Architectures Orientées Services: Haute Disponibilité, Confiance, et Sociabilité

Noura Faci

MÉMOIRE D'HABILITATION À DIRIGER DES RECHERCHES

UNIVERSITÉ CLAUDE BERNARD LYON 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

SPÉCIALITÉ : INFORMATIQUE

Présenté par **Noura Faci**

Laboratoire d'InfoRmatique en Image et Systèmes d'information

---

Architectures Orientées Services: Haute Disponibilité, Confiance, et Sociabilité

Date de soutenance : 14 décembre 2018

Devant la commission d'examen composée de :

Jamal Bentahar	Full Professor, Concordia University, Canada	Rapporteur
Khalil Drira	Directeur de recherche, LAAS-CNRS, Toulouse	Rapporteur
Daniela Grigori	Professeure des universités, Université Paris-Dauphine, Paris	Rapporteur
Djamal Benslimane	Professeur des universités, Université Claude Bernard, Lyon	Examinateur
François Charoy	Professeur des universités, Université de Lorraine, Nancy	Examinateur
Walid Gaaloul	Professeur des universités, Institut Mines Télécom, Paris	Examinateur
Franck Morvan	Professeur des universités, Université Paul Sabatier, Toulouse	Examinateur
Marta Rukoz	Professeure des universités, Université Paris Ouest, Nanterre	Examinateur

– 2018 –

Résumé Mon agenda de recherche au cours des 10 dernières années a principalement porté sur la robustesse des applications orientées-services. Cela consiste à découvrir, sélectionner, et composer des services pouvant être sujets à des défaillances au moment de l'exécution, et susceptibles de faire échouer ces applications. Pour atténuer les risques de défaillance, nous nous sommes intéressés à la pertinence de la diversité pour exécuter les applications orientées-services avec succès. Plusieurs questions de recherche ont été identifiées telles que (i) comment définir et configurer la diversité; et (ii) comment sélectionner les services pour satisfaire les besoins en tolérance aux fautes. Une phase préliminaire a examiné comment la réplication pourrait aller de pair avec la diversité, de sorte que, par exemple, plusieurs services sémantiquement équivalents (mais pas nécessairement les mêmes d'un point de vue non fonctionnel) soient regroupés ensemble. La phase suivante suivante consistait à déterminer comment améliorer la qualité de découverte de service en présence de fautes. L'idée était de s'appuyer sur les principes des réseaux sociaux pour sélectionner les services les plus appropriés, les dotant ainsi de certaines caractéristiques sociales comme qui collabore avec qui. Le travail effectué a permis de définir des réseaux sociaux de services, de les mettre en place, et d'en extraire des informations pertinentes de ces réseaux pour les besoins de la découverte. Assurer le bon usage de ces réseaux sociaux, une autre piste de recherche consistait à chercher des moyens de réglementer les actions des services (par exemple, établir et maintenir des réseaux de contacts) en utilisant la notion d'engagements. En fait, la robustesse au niveau "service" est nécessaire mais pas suffisant. Il est donc aussi important de l'inclure au niveau du processus de sélection lui-même, et donc, étendant notre processus de réflexion sur la question de robustesse. Les systèmes d'évaluation de la confiance des services sont principalement basés sur les expériences utilisateur lors de l'invocation de ces services. La question de recherche fut de comment évaluer la confiance d'un service en présence d'attaques telles que des expériences faussées. Le travail a abouti à la définition d'un modèle de crédibilité basé sur un clustering flou, un mécanisme de filtrage des utilisateurs ayant plusieurs identités et un modèle de confiance basé sur des bases de données probabilistes.

Il est bien connu que les architectures orientées-services et la gestion des processus métier vont de pair pour développer des applications d'entreprise. Nous avons jugé approprié d'explorer le tissage de principes sociaux, tels que la proximité, dans la conception et l'exécution des processus métier pour garantir la continuité de l'activité. La principale préoccupation est de comment assurer un alignement parfait des technologies Web 2.0 avec les stratégies de développement et les meilleures pratiques en entreprise. Le travail effectué a favorisé les relations sociales entre les employés d'une entreprise pour améliorer leur performance. En effet, il a été observé que les relations informelles entre les personnes existent dans les entreprises aux niveaux stratégique et opérationnel. Les questions abordées sont de (i) comment concevoir des processus métier sur la base des principes sociaux et (ii) comment assurer une exécution efficace de ces processus lors de conflits de ressources. Une phase de recherche préliminaire était de développer des réseaux dédiés basés sur les relations sociales (par exemple, la supervision et le partenariat) entre les trois composants d'un processus (tâche, personne et machine) et d'analyser la valeur ajoutée de ces réseaux pour les entreprises. Ces réseaux capturent les différents situations de colla-

boration entre tâches, entre personnes et entre machines. La phase suivante consistait à aborder la gestion des ressources dans les entreprises. L'idée était de capitaliser sur ces réseaux dédiés pour coordonner la production de ressources, leur consommation, et leur utilisation. Ce travail a permis: (i) de classer les ressources en fonction de leur nature et de leur type, (ii) identifier les conflits par catégorie de ressources et (iii) proposer des solutions pour ces conflits en utilisant les réseaux appropriés. Garantir la stabilité de ces les réseaux devient alors une nécessité. À cette fin, l'approche proposée repose sur des engagements métier et sociaux pour réguler le fonctionnement au sein de ces réseaux.

Pour les 4 prochaines années, mon agenda de recherche abordera les nouveaux défis liés à l'Internet des objets, (r)évolution du Web, tels que la combinaison à la volée des objets et les objets cognitifs. Quelques résultats préliminaires ont déjà été partagé avec la communauté scientifique.

Mots-clés composition de services, tolérance aux fautes, diversité, systèmes de confiance, robustesse, gestion des processus métier, réseaux sociaux.

Title Service-Oriented Architectures: High Availability, Trust, Sociability

Abstract My research agenda over the last 10 years has mainly revolved around the topic of robustness of service-oriented applications. This meant discovering, selecting, and composing services that could be faulty at run-time, and hence could make these applications fail. To mitigate the risks of failure, we examined diversity appropriateness for successful service-oriented applications. We raised a couple of questions that are (i) how to define and configure diversity; and (ii) how to select services so that fault tolerance requirements are met. A preliminary investigation examined how replication could work hand-in-hand with diversity so, that, for instance many replicate services (not necessarily the same from a non-functional perspective) could be grouped together. The next phase was to focus on how to improve the quality of service discovery in the presence of faults. The idea was to embrace the principles of social networks to select the most appropriate services, thus endowing them with some social characteristics like who collaborates with who. The work carried out allowed defining and setting up social networks of services and extracting relevant details from these networks for the needs of discovery. To ensure proper use of social networks, another avenue of research consisted of looking for ways that would regulate services' actions (e.g., establish and maintain networks of contacts) using the concept of commitments. In fact, robustness at the level of "service" is necessary but not sufficient. Thus, it is important to include the robustness in the selection process itself as well, and thus, extends our reflection process. We raised the question of how to evaluate a service's trust. Trust systems are mostly based on user/service experiences when requesting these services. The work resulted in the definition of a credibility model based on a fuzzy clustering, a mechanism for filtering users with multiple identities, and a trust model based on probabilistic databases.

It is well known that service-oriented architectures and business process management go hand in hand for developing enterprise applications. We deemed appropriate exploring the weaving of social principles like proximity into the design and

execution of business processes so, that, business continuity is guaranteed. The main concern is how to ensure perfect alignment of Web 2.0 technologies with business development strategies and best practices. The work carried out fostered the social relations among an enterprise's employees to perform better. Indeed, it was noted that informal relations between people exist in enterprises at strategic, management, and operational levels. We address the questions of (i) how to design business processes based on social principles and (ii) how to ensure effective execution of these processes during resource conflicts. A preliminary research phase was to develop dedicated networks based on social relations (e.g., supervision and partnership) between a process's three components (task, person, and machine) and to analyze the added value of these networks to enterprises.. These networks capture the different situations of collaboration between tasks, between persons, and between machines. The next phase was to address resource management in enterprises. The idea was to capitalize on dedicated networks to coordinate resource production, consumption, and use. The work allowed: (i) to categorize resources according to their nature and type, (ii) to identify conflicts per resource category, and (iii) to propose solutions to these conflicts by using the appropriate networks. Guaranteeing the stability of these networks is a necessity. To this end, the proposed approach relies on business and social commitments to regulate the functioning within these networks.

For the next 4-years, my research agenda will tackle new challenges related to Internet of Things, (r)evolution of the Web, such as *on-the-fly* combination of things and cognitive things. Some early findings have already been shared with the community.

Keywords Service composition, fault tolerance, diversity, trust management systems, robustness, business process management, social networks.

Laboratoire d'accueil LIRIS (UMR 5205)
43 bd du 11 novembre 1918
69622 Villeurbanne Cedex

Table des matières

Première partie

Mes travaux de recherche

Chapitre 1

Présentation générale

1.1 Contexte de recherche

Les recherches présentées dans ce manuscrit se situent à la croisée de 2 disciplines : les architectures orientées-services (en anglais, SOA) et la gestion des processus métier (en anglais, BPM). L'alliance entre ces 2 disciplines s'est montrée très bénéfique aux professionnels de l'informatique et aux entreprises [2]. SOA ne peut être utile sans une infrastructure de BPM, élément central du développement d'applications orientées-service. BPM aide à créer des modèles de processus où l'automatisation de (ou une partie de) ces processus se présente sous la forme d'invocation de services. Travaillant de pair, SOA-BPM permettent ainsi d'assembler/composer de nouvelles applications métier modélisées comme un ensemble de tâches dont certaines sont implémentées comme services.

L'étude de la robustesse de ces applications constitue le périmètre initial de mes recherches où la question principale est de comment assurer la continuité du bon fonctionnement de ces applications sujettes à la défaillance de leurs composants et propices à des attaques malveillantes. Les travaux menés s'intéressent au processus de sélection des services pour des besoins de découverte et de composition et ce, dans un contexte de fautes. Plus précisément, la problématique générale est de comment concevoir des applications orientées-service tolérantes aux fautes et superviser leur fonctionnement. Les investigations se sont orientées vers la technique de diversité. Les questions de recherche abordées sont : (i) comment définir les protocoles de diversité de services et les configurer ; et (ii) comment sélectionner les services les plus appropriés pour répondre aux besoins de tolérance aux fautes.

Avec la frénésie "sociale" (Web 2.0), le contexte de recherche a été élargi en intégrant une nouvelle discipline, l'informatique sociale (en anglais, *Social Computing*) afin de découvrir les services en se basant sur les principes des réseaux sociaux. Tout en restant dans cet esprit "social", nous nous sommes *de facto* intéressés à la confiance à accorder aux services définie en fonction des recommandations provenant d'autres consommateurs de services susceptibles d'être malveillants.

Dans ce nouveau contexte de recherche SOA-BPM-Web 2.0, de nouvelles opportuinités de recherche se sont présentées pour mettre à profit les principes des réseaux sociaux lors de la conception et de l'exécution de processus métier. Les questions de

recherche abordées sont : (i) comment concevoir des processus métier en capitalisant sur des réseaux particuliers, et (ii) comment assurer une exécution efficace de ces processus lors de conflits sur les ressources.

1.2 Démarche scientifique

Cette section présente la vision personnelle de la manière de conduire des recherches non individuelles, et sa mise en place dans les travaux menés.

1.2.1 Vision équipe-projet

Ma démarche de recherche suit une culture d'équipe-projet à court-terme (environ 4-5 années), notamment au travers d'activités de co-direction de recherche et de projets collaboratifs. La figure 1.1 montre le cycle de vie d'une équipe-projet composé des phases suivantes :

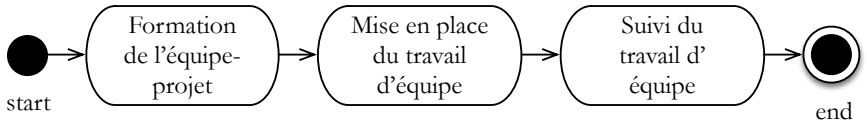


FIGURE 1.1 – Cycle de vie d'une équipe-projet

- Formation de l'équipe-projet. Des discussions sur l'objectif et attentes du projet capturent son essence même afin de se conformer à une description commune tout au long de la vie du projet. Le leadership adopté est partagé avec Mr. Benslimane (Professeur des Universités), notamment dans des co-directions de thèse, et Mr. Maamar (Full Professor), notamment dans des projets collaboratifs. La taille de l'équipe-projet varie entre 4 à 6 personnes venant de différents horizons. Des compétences en gestion de ressources humaines et du temps sont indispensables pour s'assurer du bon fonctionnement de l'équipe (e.g., maintien de l'activité et respect des délais).
- Mise en place du travail de l'équipe-projet. Afin de re-dynamiser l'équipe-projet, organiser des réunions sur les idées émergentes ou de publications s'est montré efficace pour maintenir le projet sur la bonne voie et garder les membres engagés. A l'issue des réunions, le document de travail est mis à jour. La distribution des tâches est effectuée sur la base du bon vouloir des membres acquérant ainsi plus de compétences et comprenant mieux les objectifs susceptibles d'avoir évolué et le travail restant.
- Gestion des documents de travail. Au fur et à mesure que les idées émergent, l'équipe-projet doit décider lesquelles sont importantes à conserver, celles ne rentrant pas dans le cadre du projet, et/ou celles à réserver pour une révision ultérieure. Les idées retenues sont mises au propre et développées dans un ensemble cohérent de documents de travail faisant l'objet de publications.

1.2.2 Méthodologies de recherche adoptées

Au fil de mes recherches, j'ai pû ainsi expérimenter deux manières de prospecter, l'une exploratoire et l'autre descriptive. La recherche exploratoire est menée pour étudier un problème pas encore clairement défini ou identifié dans la littérature. Elle explore un sujet de recherche avec différents niveaux de profondeur et constitue la base de recherches plus abouties. Les réflexions menées s'assimilent à un processus incrémental et intégré de l'ensemble des travaux avec comme fil directeur un travail personnel pionnier. Cette pratique s'est montrée efficace et fructueuse au vu de la cohérence générale des résultats obtenus et de leur qualité. A travers mes travaux, j'ai eu l'opportunité de participer à deux encadrements de thèse à hauteur de 50% chacun, au coté de Mr. Benslimane, Professeur des Universités dans l'équipe SOC, et aussi collaborer avec des équipes nationales et internationales.

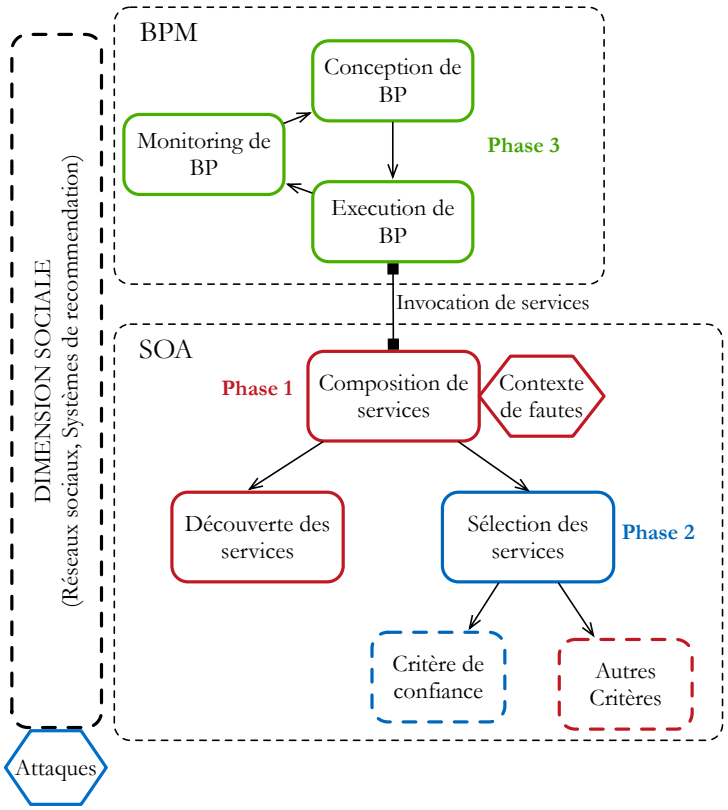


FIGURE 1.2 – Phases de recherche

La figure 1.2 retrace la chronologie des différentes phases de recherche relatives aux travaux effectués. L'objectif de la phase préliminaire (**Phase 1**) fût de rechercher comment adapter les protocoles classiques de duplication basés sur la réplication pour être appliqués dans le cas de la diversité. Un groupe de diversité représente un ensemble de services similaires capables de réaliser la même fonctionnalité d'un service abstrait caractérisé par un niveau de criticité (e.g., critique, semi-critique, et non-critique). La criticité définit la sévérité de l'impact d'éventuelles fautes du service sur le fonctionnement global de la composition. Ces recherches ont conduit

à : (i) la définition et la mise en œuvre d'une architecture multi-niveaux pour la tolérance des services aux fautes de type crash et Byzantine ; et (ii) la définition et la formalisation de protocoles pour la technique de diversité. La phase suivante portait sur l'amélioration de la qualité de la découverte de services dans un contexte de fautes. L'idée était d'adopter des principes sociaux pour sélectionner les services les plus appropriés, les dotant ainsi de caractéristiques sociales (e.g., quel service s'est substitué ou a collaboré avec tel autre). Les travaux menés ont permis de définir et de mettre en place des réseaux sociaux associés aux services, et d'en extraire des informations pertinentes à des fins de découverte.

Assurer la robustesse au niveau "service" est nécessaire mais pas suffisante. En effet, il est important de considérer aussi la robustesse au niveau du processus de sélection lui-même constituant ainsi un prolongement dans le processus de réflexion (**Phase 2**, Fig. 1.2). Il était alors question d'évaluer la confiance à attribuer aux services, et de manière générale aux ressources sur le Web (e.g., vidéos). Les systèmes de confiance sont pour la plupart basés sur les retours d'expèrience des services (ou encore des utilisateurs) après consultation de ces ressources. Les travaux ont donné lieu à la définition d'un modèle de crédibilité basé sur un clustering flou, d'un mécanisme de filtrage des utilisateurs avec des identités multiples, et d'un modèle d'évaluation de la confiance basé sur les bases de données probabilistes.

Nos résultats de recherche relatifs à la "socialisation" des services, nous ont incité à prospecter les potentialités des principes sociaux dans la conception et la mise en œuvre de processus métier à l'ère du Web. 2.0 (**Phase 3**, Fig. 1.2). La question est de comment assurer un alignement parfait des technologies Web 2.0 avec les stratégies de développement de l'entreprise. Les travaux menés ont une vision sociale de l'entreprise pour encourager les employés à être plus proactifs dans le développement des solutions répondant à leurs problématiques en sollicitant leurs réseaux de contacts. En effet, il fût constaté que les relations informelles entre les personnes arrivent à exister dans les entreprises aussi bien au niveau stratégique qu'opérationnel.

La première étape fût de développer des réseaux dédiés basés sur des relations sociales (e.g., supervision et partenariat) et d'analyser la valeur ajoutée de ces réseaux par l'entreprise. L'idée était d'étendre les relations sociales entre personnes à d'autres composants du processus métier à savoir la tâche et la machine en mettant l'accent sur le développement avec une dimension sociale. Les réseaux dédiés capturent les différentes situations de collaboration entre les composants d'un processus métier et coexistent parfaitement avec ceux plus formels où les relations entre personnes (e.g., supervision) sont déjà pré-établies. L'étape suivante consistait à répondre au problème de gestion des ressources au sein de l'entreprise. L'idée était de capitaliser sur les réseaux dédiés pour coordonner la production, la consommation, et l'utilisation des ressources. Les travaux ont permis : (i) de catégoriser les ressources selon leur nature et leur type, (ii) d'identifier les conflits par catégorie de ressources, et (iii) de proposer des solutions à ces conflits en utilisant les réseaux dédiés appropriés. L'accès aux applications Web 2.0 par les employés reste avantageux pour l'entreprise néanmoins les risques d'un mauvais usage ne sont pas à négliger. Pour cela, nous nous sommes intéressés aux restrictions à mettre en place pour contrôler l'usage de ces applications.

1.3 Socle de contributions

Les contributions majeures issues des travaux réalisés se résument à : (i) la définition d'une architecture de gestion des fautes mettant en œuvre de nouveaux protocoles basés sur la diversité ; (ii) la spécification de modèles de crédibilité et de confiance basés sur la théorie des ensembles flous et celle des probabilités, respectivement ; (iii) l'élaboration d'une technique de découverte basée sur des réseaux dédiés aux types d'interaction entre les services ; et (iv) la construction d'une méthodologie de conception et d'implémentation des processus métier selon une perspective des réseaux sociaux.

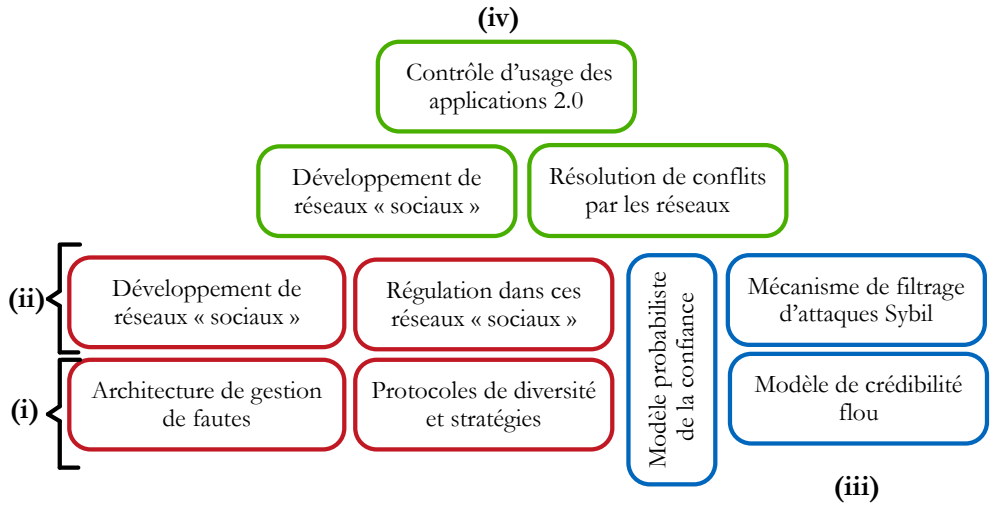


FIGURE 1.3 – Socle de contributions

- **Phase 1** (Fig. 1.2) : **Services Web hautement disponibles**

  - **C_1. Définition d'une architecture multi niveaux pour la tolérance aux fautes des services**. Une architecture *multi* niveaux pour la conception et l'exécution de services tolérants aux fautes est proposée. Les fautes considérées sont de type crash et Byzantine. Le premier niveau permet de décrire les applications à base de services en termes de services abstraits. Le deuxième niveau permet la concrétisation des services abstraits au travers des groupes de diversité. Un groupe de diversité représente un ensemble de services similaires capables de répondre à la fonctionnalité d'un même service abstrait. La duplication des services sera réalisée uniquement via les éléments d'un même groupe de diversité. Le troisième niveau correspond aux services concrets mis à la disposition des applications. L'architecture proposée permet aussi d'attacher des propriétés de criticité aux services Web abstraits pour définir la sévérité d'éventuelles fautes. Trois niveaux de criticité sont définis : critique quand la faute n'est en aucun cas acceptable, semi-critique quand des résultats

partiels sont tolérés, et non critique quand la faute est totalement acceptée. A chaque niveau de l'architecture est associé un type de composant logiciel appelé manager pour surveiller et gérer les services concrets, les groupes de diversités et la composition de services. Le comportement de chaque manager proposé est décrit sous forme d'automates à états finis.

- **C_2. Définition de protocoles de duplication adaptés à la diversité**. Deux protocoles de duplication sont définis et formalisés pour l'exécution tolérante aux fautes des services. Le protocole séquentiel permet le recouvrement de fautes via des exécutions séquentielles d'un sous-ensemble de services d'un même groupe de diversité. Le protocole parallèle réalise le recouvrement de fautes via une exécution parallèle de plusieurs éléments d'un même groupe de diversité. Un algorithme de vote est aussi proposé pour le cas du protocole parallèle. Il permet de choisir le bon résultat dans le cas de plusieurs réponses équivalentes de services d'un même groupe de diversité.
- **C_3. Formation sociale des groupes de diversité**. Il s'agit essentiellement de former un groupe de diversité et d'estimer le nombre de services similaires nécessaires pour l'exécution d'un service abstrait. Tout d'abord, une nouvelle technique de découverte basée sur les principes des réseaux sociaux est proposée. Puis, un algorithme pour le calcul du degré de redondance n exigé dans un groupe de diversité est proposé. Cet algorithme tient compte du protocole de recouvrement utilisé. Ainsi, dans le cas du protocole parallèle, il permet de déterminer le degré approprié de redondance à utiliser pour le prochain tour de vote tenant compte des fautes observées dans les précédents tours. Dans le cas du protocole séquentiel, il s'agit de réserver n services et définir leur ordonnancement.
- **C_4. Sélection de meilleurs services dans un groupe de diversité**. Une fois un groupe de diversité est configuré pour fonctionner avec n services similaires, il s'agit alors d'identifier les n meilleurs services. Des algorithmes de sélection des n meilleurs services au sein d'un groupe de diversité sont proposés. Ils sont basés sur l'historique d'exécution de services Web , les exigences des clients et les changements de contexte en terme de variation du nombre de fautes détectées (i.e., augmentation ou diminution).
- **C_5. Implémentation et expérimentation des concepts proposés**. Une mise en œuvre des protocoles de tolérance aux fautes a été réalisée. Les expérimentations montrent que la stratégie de sélection basée sur l'historique d'exécution (Log) améliore l'efficacité du groupe de diversité.

• **Phase 2** (Fig. 1.2) : **Gestion robuste de la confiance des ressources Web**

- **C_1. Modèle de crédibilité flou résilient aux évaluations biaisées**. Pour répondre à Q_1 relative à la robustesse aux attaques des évaluations biaisées/fausses, nous proposons un nouveau modèle d'évaluation de la crédibilité des utilisateurs. Le modèle proposé se base uniquement sur les évaluations fournies par les utilisateurs et n'exige aucune information

sur leurs identités ni leurs interactions, assurant une totale protection de leur vie privée. La crédibilité est évaluée selon deux critères : la fiabilité et l'expertise. La fiabilité est calculée par rapport à la proximité de l'avis majoritaire. Comme un utilisateur expert n'aura aucun intérêt à s'aligner avec la majorité, il maintient son avis indépendamment de l'avis majoritaire. Ceci se traduit, dans certains cas, comme une majorité composée de faux avis. Généralement, ces utilisateurs sont plus stricts dans leur évaluation de la ressource. Notre objectif principal est d'inclure les avis des utilisateurs présentant à la fois ces deux caractéristiques et qui sont souvent négligés dans les approches classiques. Ceci permettra de réduire l'écart pouvant exister entre les évaluations de ces utilisateurs et l'avis de la majorité actuelle. A cette fin, nous utilisons une technique de clustering flou des évaluations. Dans cette technique, l'incertitude est représentée par des frontières graduelles entre les groupes d'avis à la place de frontières nettes entre eux. Elle s'exprime par un degré d'appartenance d'une évaluation à un ou plusieurs groupes. Ainsi, les évaluations situées à la frontière du groupe majoritaire pouvant correspondre à des évaluations des utilisateurs stricts, seront incluses pour établir l'avis de la majorité. Afin de déterminer le groupe majoritaire, nous proposons trois stratégies en utilisant des valeurs qualitatives du degré d'appartenance à un cluster flou : stratégie faible, stratégie modérée et stratégie forte.

- **C_2. Modèle de filtrage des évaluations tolérant aux attaques Sybil**. Nous proposons un modèle de filtrage des évaluations afin de répondre à Q_2 concernant la robustesse de notre système contre les attaques Sybil. Le modèle combine l'évaluation de la crédibilité proposée dans C_1 avec une analyse de la structure du réseau social. Le but principal est de sélectionner parmi les évaluations collectées celles fournies par des utilisateurs crédibles ayant une seule identité. Ces utilisateurs sont plus susceptibles de fournir des évaluations cohérentes et par conséquent, dignes de confiance. Comme il s'agit d'utilisateurs réels et honnêtes, ils n'établissent des liens avec d'autres utilisateurs du réseau social seulement s'il existe un lien de confiance véritable entre eux. Nous proposons d'abord un mécanisme de distribution des capacités correspondant à un pouvoir d'évaluation que l'on souhaite accorder aux utilisateurs selon leurs crédibilités. Nous proposons ensuite un mécanise de sélection des utilisateurs basé sur des capacités distribuées et l'algorithme de flot maximal. La sélection est réalisée en partant d'un utilisateur source connu et donc fiable et puis en explorant ses liens sociaux afin de sélectionner des utilisateurs ayant des capacités maximales.

- **C_3. Modèles déterministe et probabiliste de confiance**. Nous proposons deux approches d'évaluation de la confiance d'une ressource sur le Web, une déterministe et une probabiliste pour répondre à Q_4 concernant l'agrégation des évaluations sachant que les utilisateurs n'ont pas tous le même niveau de crédibilité et certains peuvent avoir des identités virtuelles. La première approche vient consolider les évaluations des uti-

lisateurs en prenant en compte la crédibilité des utilisateurs. La prise en compte de la crédibilité de l'utilisateur permet de résoudre le problème d'incohérence des évaluations. Dans notre approche, plus les utilisateurs sont crédibles plus leurs évaluations impactent le calcul de la confiance. Cette incohérence se manifeste par une divergence des avis des utilisateurs. Troffaes a démontré que le recours à des probabilités permet de formaliser le problème de divergence d'avis [88]. Nous associons alors dans la deuxième approche la notion de crédibilité à celle de la probabilité. Les bases de données probabilistes associées à la sémantique de la théorie des mondes possibles constituent une solution intéressante au calcul de la confiance d'une ressource. Elles ont été largement utilisées pour représenter et analyser les données incertaines (extraction de données, etc.). Dans une base de données probabiliste chaque tuple a une certaine probabilité d'appartenir à la base de données. Les bases de données probabilistes offrent alors une meilleure représentation de l'incertitude liée à la crédibilité des utilisateurs et permettent aussi à travers des requêtes probabilistes un calcul incertain de la confiance d'une ressource [21].

- **C_4. Mise en œuvre et expérimentations de l'approche**. Nous proposons une implémentation de notre approche d'évaluation de la confiance des ressources sur le Web à travers le système WRTrust (Web Resource Trust). Nous avons mené plusieurs expérimentations afin d'évaluer les différents modèles proposés en termes de performance et de robustesse de WRTrust. Afin d'évaluer le critère de performance, nous avons utilisé des métriques telles que l'erreur moyenne de la valeur de confiance (RMSE), la précision et le rappel. Et afin d'évaluer le critère de robustesse, nous avons simulé des comportements malveillants tels que l'émission d'évaluations biaisées ou la création de fausses identités.

- **Phase 3** (Fig. 1.2) : **Systèmes d'entreprise selon une dimension sociale**

  - **C_1. Conception sociale des processus métier**. Nous proposons une approche de conception basée sur des relations entre les composants du processus métier, à savoir les tâches, machines, et personnes, selon deux perspectives : exécution et sociale. En vue de la démystification des relations sociales, nous associons des exigences aux tâches et des capacités aux personnes et machines. Plusieurs relations ont été identifiées telles que l'inter-échange entre tâches, la substitution entre machines, et le partenariat entre personnes. Pour capturer ces relations, trois catégories de réseaux, à savoir de configuration, de support, et sociaux, sont développés par rapport aux caractéristiques de chaque composante. Pour chaque type de relation, nous définissons des pré-conditions, conditions, et post-conditions. Une pré-condition fait référence aux raisons pour lesquelles une relation sociale entre composants est établie. Une condition indique à quel moment le réseau construit à la base de la relation sociale correspondante est utilisé pour résoudre les conflits empêchant la terminaison du processus métier. Enfin, la post-condition indique à quel moment arrêter l'utilisation du réseau.

- **C_2. Coordination sociale des processus métier**. Nous proposons une approche : (i) identifiant les conflits par type de composant (c'est-à-dire, tâche, machine et personne) dans un BP, (ii) analysant l'impact de ces conflits sur la progression de l'exécution de BP, et (iii) utilisant des réseaux de configuration et de support pour résoudre ces conflits lorsque cela est possible. Pour cela, les ressources ont été catégorisées en (i) *logique* (i.e., leur utilisation/consommation n'entraîne pas de diminution de leur niveau de fiabilité/disponibilité) et (ii) *physique* (i.e., leur utilisation/consommation conduit à une diminution de leur niveau de fiabilité/disponibilité). Cette diminution nécessite soit un remplacement des ressources ou à un réapprovisionnement en ressources. Ces ressources ont été décrites avec un ensemble de propriétés telles que *limité* (i.e., lorsque l'utilisation/consommation de la ressource est mesurable ou bien une ressource cesse d'exister à cause soit de son cycle d'utilisation/consommation soit de contraintes temporelles) et *limité mais renouvelable* (i.e., quand une utilisation/consommation de ressources atteint un certain seuil ou est soumise à des contraintes temporelles, dans les deux cas le renouvellement est possible).
- **C_3. Restrictions d'usage des actions sociales**. Nous proposons un ensemble de restrictions sur les actions sociales pour répondre aux préoccupations des entreprises face aux applications Web 2.0. Ces restrictions expliquent quoi faire (e.g., publier des notes est autorisé, mais pas participer à des sessions de discussion) et comment le faire (e.g., quel média social devrait être utilisé), quand (e.g., seulement pendant les heures de pointe), et où (e.g., seulement au bureau). Ces restrictions permettent ainsi de limiter le nombre de fois qu'une action sociale est exécutée, le contenu d'une action sociale, et le destinataire d'une action sociale. Nous avons établi une liste exhaustive d'actions sociales parmi les applications Web 2.0 les plus représentatives (e.g., Facebook, Google+). Une spécification formelle des restrictions sur les actions sociales a été proposée en utilisant un langage comme UML Object Constraint Language (OCL). Finalement, nous avons développé une approche de monitoring pour détecter les violations de restriction de sorte à garantir une conformité des usages avec les restrictions.

1.4 Organisation du mémoire

Ce manuscrit est organisé en 3 parties : (1) bilan des recherches menées, (2) prospectives de recherche, et (3) curriculum vitae long. La première partie contient 3 chapitres, un par phase de recherche. Ces chapitres sont brièvement décrits comme suit :

- Chapitre 2. Le contexte global de recherche est l'étude de la robustesse des applications orientés-service. Les travaux menés s'intéressent au processus de sélection des services pour des besoins de découverte et de composition et ce, dans un contexte de fautes. La problématique générale est comment concevoir

des applications orientées-service tolérantes aux fautes et superviser leur fonctionnement. Les investigations se sont orientées vers la technique de diversité. Les questions de recherche abordées sont : (i) comment définir les protocoles de diversité de services et les configurer ; et (ii) comment découvrir les services et sélectionner les plus appropriés pour répondre aux besoins de tolérance aux fautes.

- Chapitre 3. Assurer la robustesse au niveau "service" est nécessaire dans le Chapitre 2 mais pas suffisante. En effet, il est important de considérer aussi la robustesse au niveau du processus de sélection lui-même constituant ainsi un prolongement dans le processus de réflexion. La question est alors de comment évaluer à la confiance à attribuer aux services, et de manière générale aux ressources sur le Web. Les systèmes de confiance sont la plupart basés sur les retours d'expérience des services (ou encore des utilisateurs) après consultation de ces ressources. Les travaux ont donné lieu à la définition d'un modèle de crédibilité basé sur un clustering flou, d'un mécanisme de filtrage des utilisateurs avec des identités multiples, et d'un modèle d'évaluation de la confiance basé sur les bases de données probabilistes.
- Chapitre 4. Le contexte global de recherche est l'étude des principes sociaux dans la conception et la mise en oeuvre de processus métier. La problématique générale est de comment assurer un alignement parfait des technologies Web 2.0 avec les stratégies de développement de l'entreprise. Les travaux menés ont une vision sociale de l'entreprise pour encourager les employés à être plus proactifs dans le développement des solutions répondant à leurs problématiques en sollicitant leurs réseaux de contacts. En effet, il fût constater que les relations informelles entre les personnes arrivent à exister dans les entreprises aussi bien au niveau stratégique qu'opérationnel. Les questions de recherche abordées sont : (i) comment concevoir des processus métier en se basant sur des principes sociaux, et (ii) comment assurer une exécution efficace de ces processus lors de conflits sur les ressources.

Le deuxième partie décrit l'engagement des activités de recherche dans le cadre de directions de thèse et de collaborations internationales.

Chapitre 2

Services Web hautement disponibles

2.1 Introduction

Ces dernières décennies ont été marquées par le développement rapide des systèmes d'information distribués, et tout particulièrement par la démocratisation de l'accès aux données et applications via Internet [75]. Cette évolution du monde informatique a entraîné le développement de nouveaux paradigmes d'interaction entre applications comme celui des services. Ce dernier favorise les interactions entre applications distantes, autonomes et hétérogènes aussi bien au niveau des plateformes d'exécution qu'aux niveaux syntaxique et sémantique des données à manipuler. Différentes technologies implémentent le paradigme service comme les évolutions du Web, le Cloud, et plus récemment l'Internet des Objets (e.g., [76]).

Les applications orientées-services peuvent nécessiter une haute sûreté de fonctionnement, en particulier dans des domaines critiques comme la sécurité militaire et le contrôle aérien). La sûreté de fonctionnement d'une application est définie comme la caractéristique visant à pouvoir placer une confiance justifiée dans les services offerts par une application et ce en présence de fautes [10].

La sûreté de fonctionnement fait référence à 4 propriétés : la disponibilité, la fiabilité, la sécurité, et la confidentialité. Nous nous sommes particulièrement intéressés à : i) la *disponibilité* caractérisant l'aptitude d'une application à fonctionner au moment où elle est requise, et ii) la *fiabilité* exprimant l'aptitude à la continuité des services fournis par l'application.

La sûreté de fonctionnement est généralement accomplie par différentes méthodes : (1) la prévention de fautes pour éviter l'introduction de fautes aussi bien au niveau logiciel que matériel, (2) l'élimination de fautes via la maintenance corrective, (3) la prévision de fautes en évaluant le comportement de l'application en présence de fautes, et (4) la tolérance aux fautes en évitant la défaillance de l'application via des techniques de duplication telles que la réplication ([24]) et la diversité ([9]). Comme il est pratiquement impossible de prévoir et d'éviter toutes les fautes, nous nous sommes intéressés aux aspects de tolérance aux fautes des applications orientées-services comme les protocoles à mettre œuvre et stratégies de découverte et sélection

des services formant les groupes de duplication. Nous nous sommes portés sur la technique de diversité pour des raisons de coût.

La découverte de services Web est devenue avec le temps de plus en plus délicate dû à la masse de services Web offrant des fonctionnalités similaires. Longtemps, les services Web ont joué un rôle passif dans le processus de découverte. Une tendance fut de les considérer comme des éléments sociaux intégrant dans leur fonctionnement des informations sur leur environnement et leurs interactions passées (e.g., [56, 97]). *De facto*, nous nous sommes portés sur les principes des réseaux sociaux pour découvrir et sélectionner les services à inclure dans les groupes de diversité, les dotant ainsi de caractéristiques sociales (e.g., tel service s'est substitué ou a collaboré avec tel autre). Les travaux menés ont permis de définir et de mettre en place des réseaux sociaux associés aux services, d'en extraire des informations pertinentes à des fins de découverte, et d'assurer la stabilité des réseaux en termes d'engagements.

Le chapitre s'organise comme suit. La Section 2.2 introduit les hypothèses de travail et concepts de base sur lesquels repose notre approche de diversité. La Section 2.3 présente les fondements de cette approche et sa mise en oeuvre. La Section 2.4 décrit notre solution de réseaux sociaux au problème de découverte de service pour former les groupes de diversité. La Section 2.5.1 traite de la gouvernance de ces réseaux. La Section 2.6 discute du positionnement de notre travail par rapport à la littérature. Finalement, la Section 2.7 conclut ce chapitre.

2.2 Hypothèses et concepts

Il existe 4 types de défaillance/panne (e.g., [10, 19]) : *franc*, *par omission*, *temporelle* et *Byzantin*. Nous considérons 2 types en particulier : *franc* (i.e., soit le service fonctionne normalement, soit il ne fait rien) et *Byzantin* (i.e., soit le résultat délivré par un service ne permet plus d'accomplir les tâches de l'application, soit il ne respecte pas les spécifications attendues.). Les défaillances Byzantines peuvent être soit malveillantes (i.e., produites de façon intentionnelle) ou bien accidentelles (i.e., produites sans aucun objectif malicieux). Dans ce travail, nous traitons les défaillances Byzantines accidentelles. Nous considérons deux types de résultats :

- `Résultats incorrects` : Le service Web délivre des résultats ne correspondant pas à ceux attendus par l'utilisateur.
- `Résultats incomplets` : Le service Web délivre des résultats partiels à cause des ressources limitées.

2.2.1 Spécification d'une composition de services Web

Nous considérons une composition de services Web comme une agrégation de services abstraits satisfaisant les demandes des clients. Les services abstraits correspondent aux fonctionnalités des groupes de diversité. Ils seront instanciés au moment de l'exécution de la composition et ce en tenant compte des aspects de tolérance aux fautes. Soit l'exemple classique d'une planification de voyages via le site Web d'une agence de voyage. Un client veut réserver un voyage de Paris à Londres, en fournissant les dates du voyage et le numéro de la carte bancaire pour le paiement via le site.

L'agence de voyage est associée aux services abstraits ($SA_{i=1,4}$) où chaque SA_i regroupe les services Web concrets (SWs) assurant la même fonctionnalité à savoir : SA_1 (`réservation de vol`), SA_2 (`paiement`), SA_3 (`location de voiture`) et SA_4 (`réservation d'hôtel`). Au moment de l'exécution, les fonctionnalités sont réalisées via des SWs. La fonctionnalité associée à SA_1 peut être mise en œuvre par les SWs suivants : AirFranceWS, BritishAirwaysWS, LufthansaWS. En parallèle à SA_1, le client souhaite effectuer une `réservation d'hôtel` (SA_4) implémentée par les SWs : BestWesternWS et HolidayInnWS. Une fois le client a choisi son vol et son hôtel, SA_2 permet au client d'effectuer le paiement via un des SWs implémentant SA_2 : PayPalWS, GoogleCheckOutWS, et OgoneWS. Enfin, le client loue un moyen de transport entre l'aéroport et l'hôtel via SA_3 mis en œuvre par les SWs : CarRentalWS et EuropCarWS.

2.2.2 Criticité des services abstraits

La conception d'applications orientées-services tolérantes aux fautes dépend du niveau de criticité de la fonctionnalité associée aux SAs. Le niveau de criticité mesure la gravité de la panne en cas de non réalisation de la fonctionnalité du SA. Nous nous sommes basés sur les travaux de Engelmann et al. [27] pour identifier les niveaux de criticité des SAs pour les besoins de la tolérance aux fautes. Engelmann et al. définissent deux classes d'applications [27] : applications sensibles (e.g., contrôle du trafic aérien et défense militaire) et celles hautement disponibles (e.g., banques et télécommunications). Dans la première classe, les défaillances peuvent mettre en péril des vies humaines ou avoir un impact économique très élevé. Dans la seconde classe, les clients s'attendent à recevoir une réponse à leur demande, sans toutefois imposer des contraintes fortes comme le temps de réponse. Les 3 niveaux de criticité sont ainsi définis comme suit :

- SA `critique` : Un SA est dit `critique` si l'exécution du groupe de diversité implémentant la fonctionnalité associée à SA doit se terminer dans un état de `succès`.
- SA `semi-critique` : Un SA est dit `semi-critique` si l'exécution de groupe de diversité implémentant la fonctionnalité de SA se termine dans un état de `succès` ou bien rend des `résultats incomplets`.
- SA `non-critique` : Un SA est dit `non-critique` si l'exécution du groupe de diversité implémentant la fonctionnalité de SA peut se terminer dans n'importe quel état, et la défaillance du groupe peut être ignorée sans aucun impact sur la composition de services.

Le choix de la criticité est déterminé selon les besoins des utilisateurs finaux ; une fonctionnalité peut être critique pour certains et non critique pour d'autres.

2.3 Fondements et mise en œuvre de la diversité

Cette section présente un aperçu de notre approche de diversité dans le contexte des services Web ainsi que les protocoles sous-jacents d'exécution d'un groupe de diversité.

2.3.1 Description globale de l'approche

La figure 2.1 illustre la vue globale de notre approche pour rendre les applications orientées-services tolérantes aux fautes. Elle inclut tous les concepts nécessaires pour garantir le bon fonctionnement des services Web. Nous définissons trois niveaux d'abstraction : `composition de services` (CS), `groupe de diversité` (GD) et `service concret` (SC). Le niveau CS est dédié à la spécification d'une composition de services abstraits (SAs). Le niveau GD est constitué d'un certain nombre de groupes de diversité (i.e., espaces virtuels regroupant des SCs similaires) et associés chacun à service abstrait (SA). Le niveau SC fait référence à l'ensemble des SWs mis à disposition des utilisateurs et pouvant évoluer dans le temps.

Nous avons proposé une architecture pour gérer ces 3 niveaux présentée dans notre article [1]. Elles est composée de 3 types de composants : manager de CS (MCS), manager de GD (MGD) et manager de SC (MSC). Nous décrivons ci-dessous le rôle de ces différents managers :

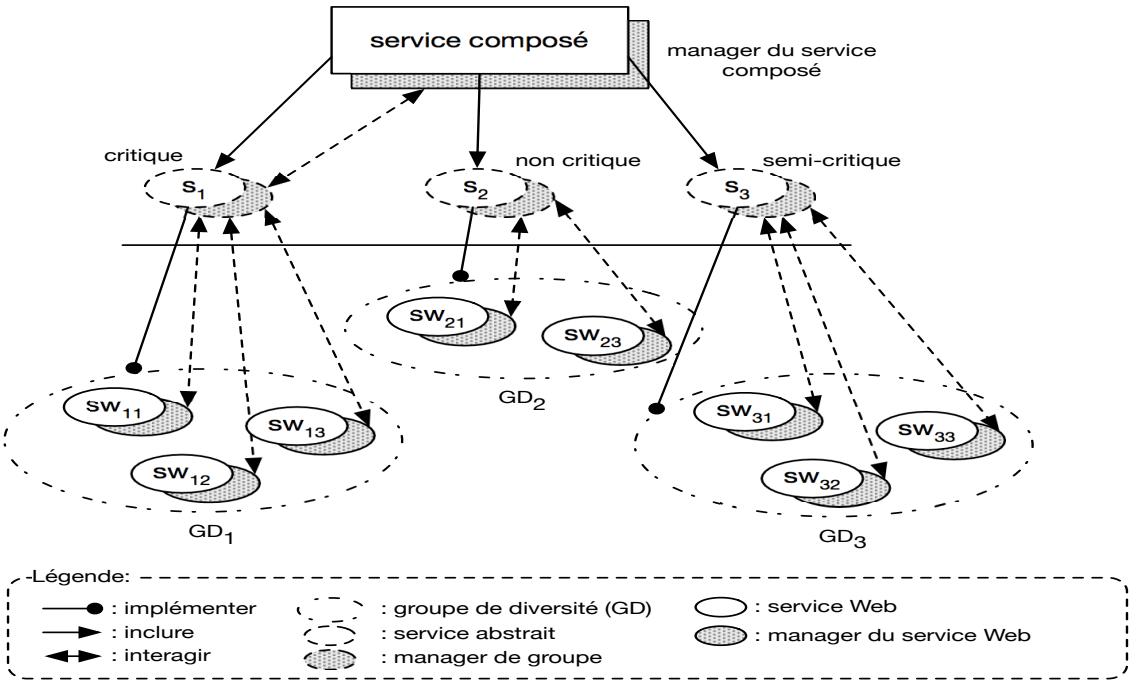


FIGURE 2.1 – **Vue Globale de l'approche de TF dans les services Web**

1. **MSC** surveille l'exécution du SW associé et signale au MGD l'échec ou la réussite du service. L'échec d'exécution peut résulter de la détection d'un crash ou d'une faute byzantine en cas de non conformité du résultat par rapport aux spécifications attendues.
2. **MGD** surveille la réalisation de la fonctionnalité associée au SA par la mise en œuvre des protocoles d'exécution séquentiel et parallèle. Il est capable de détecter des fautes byzantines dans le cas d'absence de spécifications des ré-

sultats attendus. **MGD** transmet au MCS l'état d'exécution du groupe (e.g., succès et échec) et/ou le résultat fourni par le groupe.

3. **MCS** a pour rôle la gestion de la composition. Il est doté de fonctionnalités d'invocation des GDs et d'orchestration des services en tenant compte d'éventuelles fautes détectées par les **MGDs** et de la criticité des SAs.

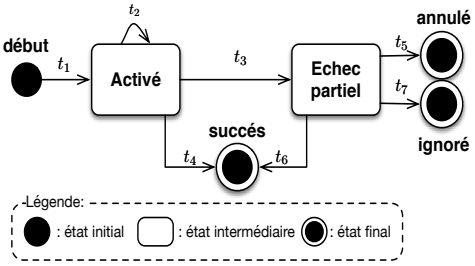
Les stratégies de recouvrement sont les points prépondérants de notre solution. Une stratégie représente un mapping approprié entre l'état observé du service Web et l'action appropriée à entreprendre. Le MGD, en charge d'observer l'exécution des SWs et leurs interactions au sein d'un groupe, rapporte au MCS l'état d'exécution du GD et ce, selon les résultats retournés par le GD ainsi que la criticité de la fonctionnalité implémentée par ce GD. Le MGD est en mesure de : i) invoquer les SWs sélectionnés selon la stratégie de recouvrement appropriée (i.e., exécution séquentielle ou parallèle), ii) signaler l'état du groupe au MCS, et iii) délivrer le résultat au MCS en charge de le transfèrer à l'utilisateur ayant invoqué l'application.

La mise en place de stratégies de recouvrement est une tâche complexe qui nécessite de prendre en compte de nombreux problèmes. D'abord, un ensemble des SWs équivalents assurant la même fonctionnalité mais implémentés différemment doit être identifié et sélectionné. Ces services sont regroupés au sein du même GD, mais seul un sous-ensemble des SWs est sélectionné pour être exécuté. Par la suite, nous discutons les problématiques relatives aux stratégies de recouvrement elles-mêmes : i) quel nombre de SWs sont à sélectionner dans le GD, ii) comment les SWs sélectionnés sont invoqués, et iii) comment le résultat approprié (i.e., correct, cohérent et/ou complet) est communiqué au client. Les 2 derniers points font référence aux protocoles d'exécution au sein du GD. Selon le même principe de la réplication, nous définissons 2 protocoles d'exécution : séquentielle (Section 2.3.2) et parallèle (Section 2.3.3). Ces protocoles ont fait l'objet d'une formalisation publiée dans notre article [28]. La formation du GD incluant la découverte de ces membres et leur sélection sera quant à elle discutée en Section 2.4.

2.3.2 Protocole d'exécution séquentielle

Dans ce protocole, la requête est envoyé à un seul SW dans le GD pour traitement. Quand une erreur est détectée (e.g., résultat non satisfaisant ou absence de résultat), un autre SW sera sélectionné et activé au sein du GD pour exécuter la requête et ce, de manière itérative. Le GD réussit (i.e., état succès) s'il arrive à un résultat correct, cohérent et/ou complet selon le niveau de criticité exigé par le client pour le SA correspondant. Le GD faillit (i.e., état échec) si l'exécution de tous les SWs au sein du GD aboutit à un échec (i.e., épuisement des SWs disponibles dans le GD). Conformément à la figure 2.1, le MGD est en charge de sélectionner un seul SW et de l'invoquer. Ce SW devient le leader du GD. Quand ce leader est en panne, le MGD vérifie s'il existe un autre service dans le même GD à sélectionner pour le désigner comme nouveau leader. Le MGD adopte le comportement décrit par un automate d'états finis (Figure 2.2).

Les états sont : initial (i.e., attente d'une invocation du SA par MSC), activé (i.e., GD en cours d'exécution), succès (i.e., retour de résultats par le GD s'avérant appropriés),



t_i	Evènement	Condition	Action
t_1	invoquer()	Null	activerS()
t_2	S(état)	SW-suivant() &(état=échec)	activerS()
t_3	S(état)	not(SW-suivant()) &(état=échec)	Null
t_4	S(état)	(état=succès)	SDG(état)
t_5	annuler()	Null	SDG(état)
t_6	S(état)	(état=succès-partiel)	SDG(état)
t_7	outrepasser()	Null	SDG(état)

FIGURE 2.2 – Modèle d'exécution du MGD en mode séquentiel

succès partiel (i.e., retour de résultats partiels), ignoré (i.e., retour de résultats s'avérant non-appropriés mais ne conduisant pas à une erreur), et échec (i.e., retour de résultats s'avérant non-appropriés mais conduisant à une erreur).

Formellement, la fonction de transition d'états ($\mathcal{T}_{seq}$) est définie comme suit (Figure 2.2) : $\mathcal{E} \times \mathcal{D} \times \mathcal{C} \times \mathcal{A} \rightarrow \mathcal{E}$ où $\mathcal{E}$ est l'ensemble des états du DG, $\mathcal{D}$ est l'ensemble des évènements déclencheur, $\mathcal{C}$ est l'ensemble des conditions à vérifier pour transiter, et $\mathcal{A}$ est l'ensemble des actions à effectuer au cours de la transition (t_i). Il est à noter que les actions du MCS deviennent des évènements déclencheur dans le MGD (e.g., invoquer(), annuler(), et outrepasser()). Les transitions relatives au comportement du MGD sont décrites dans le tableau 2.1.

Tableau 2.1 – Description des transitions en mode séquentiel

t_i	Description
t_1	invoquer() déclenche l'exécution de activerS() permettant de sélectionner un seul SW et de l'invoquer.
t_2	tant que le SW leader faillit, le MGD vérifie s'il existe un autre SW à invoquer comme leader (SW-suivant()) et exécute activerS().
t_3	quand tous les SWs du GD ont failli, le MGD change uniquement d'état.
t_4	si le SW leader du GD est parvenu au bout de son exécution, le MGD signale au MCS le succès du GD dans la mise en œuvre du SA.
t_5	suite à l'annulation du MCS (i.e., SA est critique), le MGD signale au MCS l'échec du GD dans le remise de résultats appropriés.
t_6	si le GD remet un résultat incomplet, le MGD signale au MCS le succès du GD.
t_7	si SA est non-critique (i.e., échec toléré), le MGD signale au MCS l'état ignoré du GD.

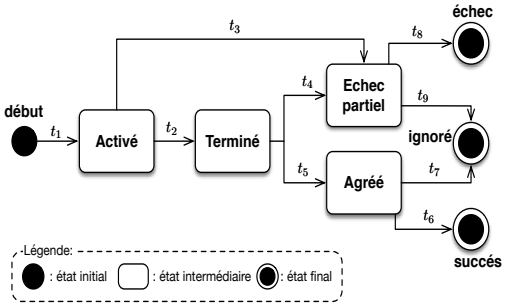
2.3.3 Protocole d'exécution parallèle

Dans ce protocole, le MGD sélectionne un sous-ensemble du GD (sGD) à invoquer et envoie la requête aux SWs (sémantiquement équivalents) pour s'exécuter de manière concurrente. Le MGD collecte les différentes réponses et en délivre une unique au MCS. Nous nous sommes inspirés de la littérature (e.g., [10], [8]) pour proposer

2 manières de constituer cette unique réponse :

- `sans vote`. Le MGD transmet la réponse du premier SW ayant terminé son exécution avec succès .
- `avec vote`. Le MGD attend de recevoir une majorité des réponses fournies par le sGD préalablement sélectionné. Ensuite, il procède à un vote majoritaire entre les réponses équivalentes. Le mécanisme sous-jacent à ce vote (ou consensus) nécessite (2×f+1) SWs pour tolérer f fautes de type Byzantine. Par la suite, nous mettons l'accent sur cette option `avec vote` à cause de sa complexité contrairement à celle `sans vote` où la solution est évidente.

De manière similaire à la Section 2.3.2, nous décrivons le comportement adopté par le MGD comme un automate d'états finis (Figure 2.3).



t_i	Evènement	Condition	Actions
t_1	invoquer()	Null	activerP()
t_2	SW-terminer()	Null	agréer()
t_3	échec-SW()	Null	SDG(état)
t_4	Null	not(agréer()	SDG(état)
t_5	Null	agréer()	sélectionner() SDG(état)
t_6	valider()	Null	SDG(état)
t_7	outrepasser()	Null	SDG(état)
t_8	annuler()	Null	SDG(état)
t_9	outrepasser()	Null	SDG(état)

FIGURE 2.3 – Modèle d'exécution du MGD en mode parallèle

Les états sont : initial (i.e., attente d'une invocation du SA par MSC), activé (i.e., sGD en cours d'exécution), terminé (i.e., fin d'exécution du sGD), agréé (i.e., consensus au sein du sGD), échec partiel (i.e., pas de consensus), succès (i.e., retour de résultat par le sGD s'avérant appropriés), ignoré (i.e., pas de retour de résultat ou bien retour de résultat par le sGD s'avérant inappropriés mais tous 2 ne conduisant pas à une erreur), et échec (i.e., pas de retour de résultat et conduisant à une erreur).

Formellement, nous definissons la fonction de transition d'états ($\mathcal{T}_{par}$) de la même manière que $\mathcal{T}_{seq}$ (Section 2.3.2) à l'exception de $\mathcal{E}$ étant l'ensemble des états du sDG (Figure 2.3). Il est à noter qu'en plus des évènements déclencheur dans $\mathcal{T}_{seq}$ (i.e., invoquer(), annuler(), et outrepasser), un évènement supplémentaire est valider(). Les transitions relatives au comportement du MGD sont décrites dans le tableau 2.2.

Dans la littérature (e.g., [89, 105]), le mécanisme de vote majoritaire consiste à considérer la valeur moyenne des réponses ou la réponse prédominante comme consensus. Cependant, ce mécanisme s'applique à des SWs ayant la même interface alors que les SWs au sein du DG offrent la même fonctionnalité mais ayant des interfaces différentes. De ce fait, nous proposons un nouveau mécanisme de vote implémenté par agréer() (Figure 2.3) en mesure d'établir un consensus parmi les SWs. Il est à noter que les réponses des SWs (R_{SW_i}) peuvent être différentes mais sémantiquement équivalentes. Nous représentons R_{SW_i} comme une instance d'un tuple de concepts (<$Concept_1$, ..., $Concept_m$>).

L'algorithme 1 décrit les différentes étapes de agréer(). L'idée est de partitionner les réponses des SWs du sDG en un ensemble de clusters ($\mathcal{C}_k$) en fonction du degré

Tableau 2.2 – Description des transitions en mode parallèle

t_i	Description
t_1	invoquer() déclenche l'exécution de activerP() permettant d'invoquer de manière concurrente les SWs du sDG.
t_2	une fois l'exécution du sGD (i.e., SWs membres) terminée (SW-terminer()), le MGD lance le mécanisme de vote (agréer()) permettant de déterminer le résultat consensuel.
t_3	dès lors que le MSW signale une panne franche de l'un des SW (échec-SW), le MDG change uniquement d'état.
t_4	dès lors qu'un consensus n'est pas pu être établi au sein du sGD (not(agréer()), le MGD le signale MCS.
t_5	dès lors qu'un consensus est établi au sein du sGD (agréer()), le MGD le signale MCS.
t_6	une fois la validation du résultat consensuel par MCS (valider()), le MGD délivre ce résultat au MCS.
t_7	une fois l'omission du résultat consensuel par le MCS (outrepasser()) et à la condition d'une non-criticité du SA associé au GD, le MGD informe MCS de son changement d'état.
t_8	une fois l'annulation par MCS (annuler()) et à la condition d'une (semi-)criticité du SA associé au GD, le MGD signale au MCS l'échec dans l'obtention d'un consensus.
t_9	une fois l'omission par MCS et à la condition d'une non-criticité du SA associé au GD, le MGD signale au MCS.

Algorithme 1 : Consensus parmi les SWs du sGD

Entrées : $\mathcal{R} = \{\mathsf{R}_{\mathsf{SW_1}}, \mathsf{R}_{\mathsf{SW_2}}, ..., \mathsf{R}_{\mathsf{SW_n}}\}$, δ, et $\mathcal{C}_1$=$\{\mathsf{R}_{\mathsf{SW_1}}\}$
Sortie : R_{cons}

1: $\mathcal{C} \leftarrow \{\mathcal{C}_1\}$; $\mathcal{R} \leftarrow \mathcal{R} - \mathcal{C}_1$; $\mathsf{R}_{cons} \leftarrow$ null ;
2: **foreach** $R_{SW_i} \in \mathcal{R}$ **do**
  $tr_1 \leftarrow faux$;
  foreach $\mathcal{C}_k \in \mathcal{C} \wedge not(tr_1)$ **do**
    $tr_2 \leftarrow vrai$;
    foreach $R_{SW_j} \in \mathcal{C}_k \wedge not(tr_2)$ **do**
      if *match*$(R_{SW_i}, R_{SW_j}) > \delta$ **then**
        $tr_2 \leftarrow faux$;
    if tr_2 **then**
      $\mathcal{C}_k \leftarrow \mathcal{C}_k \cup \{\mathsf{R}_{\mathsf{SW_i}}\}$; $tr_1 \leftarrow vrai$;
  if $not(tr_1)$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\{\mathsf{R}_{\mathsf{SW_i}}\}\}$;
  if $|\mathcal{C}_k| = \max_i |\mathcal{C}_i| > (\frac{n}{2} + 1)$ **then**
    $\mathcal{C}_{largest} \leftarrow \mathcal{C}_k$; $\mathsf{SW}_{root} \leftarrow$ centroide$(\mathcal{C}_{largest})$; $\mathsf{R}_{cons} \leftarrow \mathsf{R}_{\mathsf{SW}_{root}}$;
3: retourner R_{cons};

de matching (d). Comme le matching ne fait pas partie du contexte de recherche, nous nous sommes basés sur un algorithme existant comme celui de Paolucci et al. [66] pour définir 4 niveau de matching (i.e., exact, plugin, subsume, et fail). Ainsi, d est calculé par match() en termes de nombre de matching par niveau pour chacun des $\mathsf{Concept}_i$. Ce degré de matching entre les réponses d'un même cluster ne doit pas dépasser une certaine déviation fonctionnelle acceptable (δ). L'utilisateur final faisant appel au SA implémenté par le sDG, spécifie δ dans ses besoins et ce tenant en compte de la criticité de SA (e.g., fail toléré pour un SA non-critique). Le résultat consensuel (R_{cons}) est la réponse fournie par le centre de gravité du cluster le plus peuplé (SW_{root}).

2.4 Formation de groupes de diversité

Cette section présente 3 types d'interactions entre SWs supportant la construction de réseaux sociaux de SWs (Section 2.4.1) et la gestion de ces réseaux en termes de développement et d'exploitation à des fins de découverte de services formant les GD (Section 2.4.2).

2.4.1 Principes des réseaux sociaux pour la découverte de services

Comme précédemment mentionné en Section 2.1, les SWs devraient être en mesure de fonctionner de manière autonome pour répondre aux requêtes utilisateur. Un moyen est de garder une trace (i) des interactions passées avec des pairs, (ii) des résultats de ces interactions, (iii) des substitutions de service dans le cas de pannes, ou encore (iv) des raisons pour lesquelles ils ont été choisi au lieu d'autres pairs. Dans un de nos précédents travaux [51], cette panoplie d'interactions nous avait permis d'identifier au moins 3 types de réseaux sociaux classifiés en deux catégories : "similarité" et "différence".

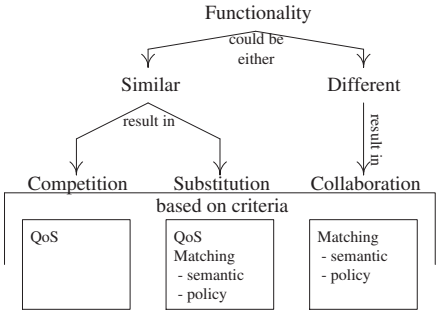


FIGURE 2.4 – Critères guidant la construction des réseaux sociaux

La Figure 2.4 montre les critères représentant les propriétés non-fonctionnelles (e.g., la QoS) et les deux types d'appariement : sémantique et politique. Les appariements mesurent les efforts à consacrer pour résoudre les conflits sémantiques et politiques. Les conflits sémantiques proviennent essentiellement des disparités conceptuelles entre les entrées des SWs et leurs sorties. Quant aux conflits de politique,

ils peuvent être dus à des contradictions entre les logiques internes des SWs. Pour illustration, seul l'appariement sémantique fût traité dans ce travail.

Nous décrivons les 3 types de réseaux comme suit :

- `Réseau de substitution`. Il contient des informations sur la similarité entre les différentes fonctionnalités de services Web. Ce réseau aide à rendre les services Web hautement disponibles en évitant le problème de pannes de SWs au moment de l'exécution. En effet, les SWs défaillants seront remplacés par d'autres similaires et opérationnels.
- `Réseau de compétition`. Il contient les mêmes informations que celui de `substitution`. En plus de la similarité entre les SWs, la sélection des SWs appropriés dépend des exigences non-fonctionnelles. Ce réseau a pour but d'aider à composer des SWs où un seul des SWs en compétition sera choisi pour répondre à la requête.
- `Réseau de collaboration`. Il contient des informations sur la différence entre les fonctionnalités de SWs. Ce réseau offre une aide pour répondre aux requêtes exigeant un assemblage de SWs. Il a aussi pour vocation d'enrichir les compositions avec des SWs supplémentaires. Dans ce type de réseau, les SWs sont disposés à collaborer les uns avec les autres.

Pour établir la nature des interactions entre SWs, notre analyse d'appariement se base sur une technique existante de matching ([61]) à partir de laquelle les degrés de similarité (DS) de complémentarité (DC) entre deux SWs (s_i et s_j) seront calculés. Noter que s_i et s_j sont complémentaires si toutes les pré-conditions de s_j correspondent aux post-conditions de s_i.

2.4.2 Gestion des réseaux sociaux

La figure 2.5 représente un aperçu de notre démarche pour gérer les réseaux de SWs décrite en 4 étapes : `clustering` des SWs, `interconnexion` des SWs, `navigation` dans les réseaux, et `ré-évaluation` des `arêtes`. Par la suite, nous considérons le DS à titre d'illustration.

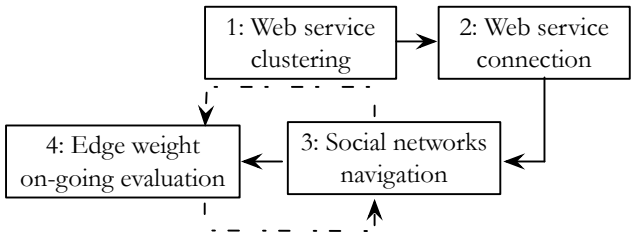
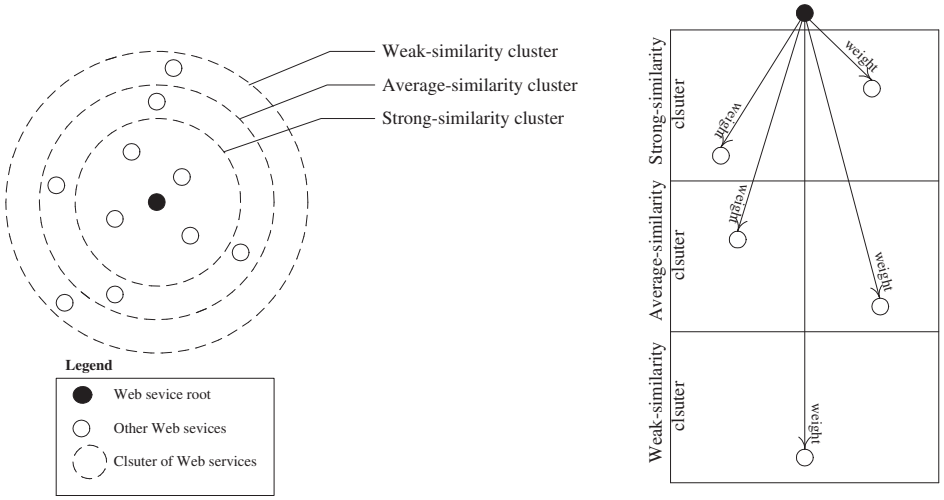


FIGURE 2.5 – Overview

`Etape 1: Clustering`. Les SWs représentés par des nœuds sont regroupés en clusters definis selon 3 classes de valeurs du DS avec un SW appelé `root` (Fig. 2.6 (a)) : `faible` (e.g., $0 < DS \leqslant 0.33$), `average` (e.g., $0.33 < DS \leqslant 0.66$), et `fort` (e.g., $0.66 < DS \leqslant 1$). Noter que nous subdivisons les valeurs du DC en 2 classes : `non` (e.g., $0 \leqslant DC \leqslant 0.49$) et `non` (e.g., $0.49 < DC \leqslant 1$). Par exemple, si un pair a un DS `fort`

avec le `root`, alors il sera placé dans le cluster à forte similarité de ce `root`. Un cluster peut déjà être rempli avec d'autres SWs. Ce processus de placement continue aussi longtemps que les SWs sont mis à disposition et acceptent d'être découverts en utilisant les réseaux sociaux.



(a) Clustering des SWs (b) Interconnexion des SWs

FIGURE 2.6 – Gestion des réseaux sociaux

`Etape 2: Interconnexion`. Les SWs sont liés par des `arêtes` selon la topologie de clusters résultat de l'étape de `clustering` conduisant ainsi à l'extension de ces réseaux sociaux (Fig. 2.6 (b)). Au démarrage (t_0), la valeur initiale du poids sur les `arêtes` (WE) entre s_i et s_j est donnée par l'équation 2.1 où s_i est le `root`. Dans un but de visualisation, la longueur des `arêtes` est proportionnelle au poids sur les `arêtes`.

$$WE_{t_0}(s_i, s_j) = DS(s_i, s_j) \tag{2.1}$$

`Etape 3: Navigation`. Comme chaque `root` constitue le point d'entrée vers son propre réseau social, la découverte de substitut(s), de collaborateur(s), ou de concurrent(s) commence par une navigation à travers différents réseaux sociaux. Selon l'activité en cours (i.e., collaboration, substitution, ou compétition), la sélection des SWs découverts dépend des facteurs suivants :

- P_c est la priorité attribuée à un cluster de SWs (Fig. 2.6 (b)). Notre intuition est de cibler en priorité les nœuds et les arêtes des clusters dits forts (i.e., ayant une priorité des plus élevées). P_c est fixe et ne change donc pas le temps.
- O_s est le coût requis par un SW s lors de sa sélection via un réseau social. Les SWs situés dans les clusters forts ont des coûts plus élevés par rapport à ceux situés dans les clusters moyens ou faibles. Ce coût est proportionnel à la la priorité du cluster et inversement proportionnel au poids de l'arête connectant le SW s au root (Equation 2.2).

$$O_{s_j} = \frac{P_c}{1 + (P_c \times WE_{t_n}(s_i, s_j))} \tag{2.2}$$

Puisque les poids des `arêtes` changent au fil du temps, le coût d'invocation d'un SW s'avère dynamique. Par conséquent, un SW est soit promu, le faisant passer à un cluster plus fort ou rétrogradé, le faisant passer à un cluster plus faible.

- E_s représente le niveau de satisfaction des expériences précédentes lors de la sélection d'un SW via un réseau social et déployé avec succès.
- L_s représente la charge actuelle d'un SW relative à ses propriétés non-fonctionnelles. Plus la charge est élevée, moins est attractif le SW.

La fonction de sélection $Select_{s_j}$ est définie comme une agrégation des facteurs susmentionnés (Equation 2.3) :

$$Select_{s_j} = \alpha_1 \times O_{s_j} + \alpha_2 \times E_{s_j} + \alpha_3 \times (1 - L_{s_j}),\ \sum_{i=1,3} \alpha_i = 1 \tag{2.3}$$

Le pair avec la plus grande valeur de $Select_s$ est considéré comme le plus appropriée pour l'activité en question (i.e., collaboration, substitution, ou compétition).

`Etape 4: ré-évaluation`. La ré-évaluation continue des `arêtes` consiste à mettre à jour les poids sur les `arêtes` lors de chaque découverte d'un pair (remplaçant, collaborateur, ou concurrent) en utilisant les réseaux sociaux. Equation 2.4 évalue les poids dans un réseau social de substitution en utilisant une fonction de prix basée sur la notion de récompense [101].

$$\begin{aligned} WE_{t+\delta t}(s_i, s_j) =& WE_t(s_i, s_j) + \\ & \alpha \times (\frac{|s_j\ selection_{t+\delta t}|}{|s_i\ failure_{t+\delta t}|} - WE_t(s_i, s_j)),\ \alpha \in [0, 1] \end{aligned} \tag{2.4}$$

où :

- δt est la période de mise à jour.
- $|s_j\ selection|$ est le nombre de fois où s_j a été substitué à s_i via le réseau social.
- $|s_i\ failure|$ est le nombre de fois où a échoué s_i.

Les hypothèses faites dans l'équation 2.4 sont : s_j n'échoue pas et la mise à jour du poids a bien été effectuée. En raison des changements dans les valeurs de poids affectant les longueurs des `arêtes` reliant les SW au `root` d'un réseau social (Equation 2.5), un SW peut être promu ou rétrogradé d'un cluster à un autre.

$$Length_{t+\delta t}(s_i, s_j) \quad = \quad WE_{t+\delta t}(s_i, s_j) \times n \tag{2.5}$$

Noter que les étapes 3-4 sont exécutées de manière itérative à chaque δt.

Une fois un SW inscrit dans un réseau social, il tire avantage d'être membre comme contacter d'autres pairs mais est aussi sous l'autorité d'une entité responsable du bon fonctionnement du réseau. Cependant, le SW dit social (SWS) peut exécuter des actions dont les effets pourraient "nuire" à ses pairs (e.g., révéler des détails privés), ou même ralentir le fonctionnement du réseau (e.g., diffuser des détails non pertinents). Dans la Section 2.5, nous adoptons la notion d'*engagement* pour tenir responsables les SWSs de leurs actions afin de signaler toute non-conformité à la réglementation des réseaux et en retour d'appliquer des sanctions.

2.5 Diversité et gouvernance

Cette section présente les motivations derrière l'adoption d'engagements pour réglementer l'exploitation des réseaux sociaux de SWs. Elle décrit aussi l'architecture générale supportant cette réglementation et détaille les engagements en termes de structure et de monitoring.

2.5.1 Approche fondée sur les engagements

Contrairement aux scénarios traditionnels de composition ([4, 38, 58]), nous nous appuyons sur les réseaux sociaux pour permettre aux SWSs de recommander les pairs souhaités pour collaborer dans le cas d'une composition, de recommander des pairs pouvant se substituer à eux en cas d'échec, et d'avoir connaissance des pairs en concurrence avec eux dans le cas d'une sélection [55]. La figure 2.7 représente un aperçu de notre approche comprenant deux mondes : `computation` et `operational`. D'une part, le `computation` comprend un pool de SWs et de SWSs, respectivement au niveau `service` et celui `social`. D'autre part, l'`operational` comprend un pool de requêtes utilisateur simples (e.g. conversion de devises) et complexes (e.g., planification de voyage) ainsi que différents SWS composites, respectivement au niveau `requête` et celui `composition`.

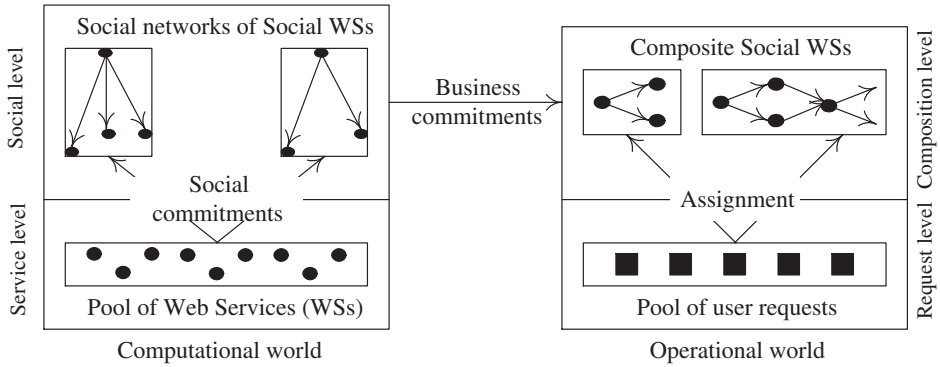


FIGURE 2.7 – Engagements pour le fonctionnement de SWs dans un réseau social

Nous identifions 2 types d'engagement :

- Engagements sociaux. Ils sont confinés aux frontières du `computation` et sont mis en place lors de l'inscription des SWs aux réseaux. Chaque réseau est géré par une entité dite autorité (sn_{auth}) dont les fonctions sont de relier les nouveaux SWs inscrits aux membres du réseau, d'évaluer les poids des arêtes et d'appliquer la réglementation au sein du réseau.
- Engagements métier. Ils relient les 2 mondes computational et operational et sont mis en place lors de la participation des SWSs à des compositions. Ces engagements permettent de garantir une participation de ces SWSs dans les compositions exécutées par des moteurs d'orchestration (comp_{orch}) et les comportements appropriés au moment de l'exécution.

Par la suite, nous identifions d'abord les responsabilités auxquelles un SWS doivent s'y rattacher après s'être inscrits sur un réseau social, puis discutons comment les engagements liés à ces responsabilités sont modélisés, gérés, et appliqués afin d'éviter les sanctions. Chaque responsabilité (Resp) est représentée avec 3 éléments : (i) soit une obligation ou permission, (ii) les actions à exécuter, et (iii) les conditions autorisant l'exécution des actions. Dans [52, 50], nous avons proposé 10 responsabilités (i.e., 5 pour chaque type d'engagement) couvrant de manière exhaustive les scénarios possibles d'interaction parmi les SWSs, sn_{auth}, et comp_{orch}. Nous en retiendrons 4 à titre d'exemple.

2.5.2 Engagements sociaux

Les engagements sociaux relèvent généralement des responsabilités contractées par un agent dit debtor, vers un autre creditor dans l'attente de voir le debtor assumer ces responsabilités [20]. Pour pallier le manque de travaux pertinents sur les réseaux sociaux de SWs, nous avons examiné la manière dont les droits des utilisateurs et leurs responsabilités sont spécifiés dans certains réseaux sociaux en ligne (e.g., Facebook et LinkedIn) dans le but de déduire des droits et responsabilités similaires pour les SWSs. Les 2 responsabilités proposées sont décrites comme suit :

- Resp_1 fait référence à l'action de collecter n'importe quel détail (d) dans un réseau social. Il faudrait indiquer l'objectif (p) de cette collecte au propriétaire de ce détail (o). Resp_1 est donc représentée comme une permission collect(d, o, valid(p)) où collect est l'action, d est une propriété non-fonctionnelle, o est le propriétaire de d (e.g., un certain SWS), p est l'objectif de la collecte de d, et valid est une fonction vérifiant la validité de p. Notons que d peut être publique (mis à la disposition de tous les SWSs du réseau), protégée (mis à la disposition de sn_{auth}), ou privée (non disponible). Nous avons étudié la spécification de P3P (acronyme de Platform for Privacy Preferences Project) pour identifier 2 objectifs : collaboration (col) relative au développement de SWs composites et substitution (sub) en rapport avec la continuité d'exécution des processus métier basés sur les SWs en cas d'échec.
- Resp_2 fait référence à la publication de détails (d) corrects sur le réseau. Resp_2 est représentée comme une `Obligation(post(d,true))` où `post` est l'action de publier et true est la véracité de d.

Une fois les responsabilités définies, nous allons nous intéresser à la modélisation des engagements associés aux responsabilités et à leur monitoring.

Modélisation. D'après le formalisme de Fonara et Colombetti [33], l'engagement associé à une responsabilité est comme suit : $\mathsf{C}_{\mathsf{Resp_i}}$(debtor, creditor, content [| condition]).

- $\mathsf{C}_{\mathsf{Resp_1}}$ ($\mathsf{sws_i}$, $\mathsf{sws_j}$, collect(d, $\mathsf{sws_j}$)|valid($\mathsf{p_d}$)) est un engagement conditionnel de $\mathsf{sws_i}$ envers $\mathsf{sws_j}$, et que si valid($\mathsf{p_d}$) est vérifié alors , collect(d, $\mathsf{sws_j}$) sera satisfait.
- $\mathsf{C}_{\mathsf{Resp_2}}$ ($\mathsf{sws_i}$, $\mathsf{sn_{auth}}$, post($\mathsf{d_{self}}$)) est un engagement de $\mathsf{sws_i}$ envers $\mathsf{sn_{auth}}$, post($\mathsf{d_{self}}$) sera satisfait.

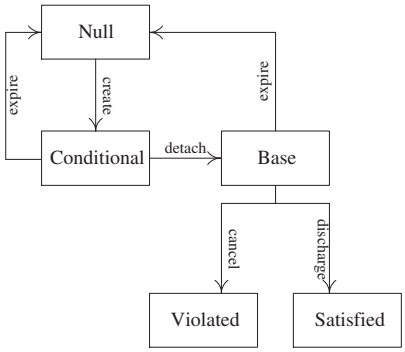


FIGURE 2.8 – Cycle de vie des engagements ([81])

Dans [81], Singh propose un cycle de vie des engagements (Fig. 2.8) au fil duquel des changements d'état se produisent suite à des interactions entre debtor (i.e., SWs et SWSs) et creditor (i.e., SWSs et sn_{auth}) et, ce sous la surveillance du creditor afin de constater une satisfaction/violation des engagements au moment de l'exécution des SWs.

Dans notre travail, les interactions debtor-creditor sont décrites par `performatif(de`: sender ; à : receiver ; content : action | statement) où performatif sont issus de la théorie des actes de langage [78] classés comme (i) exercitif incitant receiver à entreprendre une action particulière, (ii) promissif incitant receiver à s'engager dans une action future, ou (iii) expositif incitant receiver à dire la vérité. La chronologie des interactions debtor-creditor est comme suit :

1. request(de : SW_i ; à : sn_{auth} ; contenu : sign-up(reasons)) : SW_i effectue une demande d'inscription auprès de sn_{auth} accompagnée de sa motivation.
2. inform(de : sn_{auth} ; à : SW_i ; contenu : approval | refusal) : sn_{auth} informe SW_i de sa décision en se basant sur les arguments donnés par SW_i (e.g., niveau élevé de réputation/confiance) pour rejoindre le réseau.

Dans le cas approval, sn_{auth} enregistre SW_i devenant ainsi SWS_i. Des interactions supplémentaires SW_i-SW_j sont nécessaires pour les responsabilités faisant référence à des permissions (e.g., $Resp_2$) en raison de l'intérêt porté par SW_i aux détails de SW_j et en faveur du départ de SW_i du réseau :

3. request(de : SW_i ; à : SW_j ; contenu : collect(d, purpose(p)))
4. inform(de : SW_j ; à : SW_i ; contenu : approval | refusal)

Monitoring. Le suivi de la conformité de SWS_i est mise en place par sn_{auth} en exécutant create faisant passer les engagements à conditional. Nous allons passer en revue les transitions d'état reportées dans le cycle de vie en Figure 2.8 comme suit :

- conditional→ base : Si la condition est valide (i.e., valide(d,p) est vrai) ou bien il n'y a pas de condition, detach est exécutée. Selon la nature des responsabilités, les conditions sont évaluées différemment. Par exemple, $Resp_1$: SW_j vérifie que p est soit une collaboration ou une substitution.
- base→ satisfied : Si le contenu est satisfait (e.g., collect(d, SWS_j)), discharge est exécutée.

- base→ violated : Si le contenu n'est pas satisfait, cancel est exécutée.
- base→ null : Si SWS_i quitte le réseau social, expire est exécutée faisant passer tous les engagements pris à null à l'exception des engagements une fois pris sont maintenus indéfiniment.

La figure 2.8 reporte l'état violated pour indiquer que les SWSs n'ont pas honoré leur(s) engagement(s) pour diverses raisons (e.g., malveillance et pénurie temporaire de ressources de calcul). Détecter les violations est possible en utilisant le monitoring [62]. Nous distinguons 2 situations associées à violated : violation (i.e., SWSs n'exécutent pas les actions) et interdiction (i.e., SWSs exécutent les actions malgré la non-satisfaction des conditions). Nous associons également les compensations et sanctions aux creditor et debtor, respectivement.

- C_{Resp_1} : une violation se produit lorsque la collecte est effectuée sur un détail non public, et l'interdiction est établie lorsque la raison de la collecte de d n'est ni une composition, ni une substitution.
- C_{Resp_2} : une violation se produit lorsque des détails incorrects sont postés sur le réseau.

Que ce soit une violation ou une interdiction, le suivi des actions des SWSs est nécessaire et se présente comme suit.

C_{Resp_1}. Les moyens de monitoring et de compensations/sanctions sont comme suit :

- Le monitoring des violations nécessite que SW_j révèle à sn_{auth} les tentatives d'accès récurrentes à des détails non-publiques de SW_i. Si ces tentatives sont confirmées en utilisant des logs par exemple, ce sera une violation d' accès à des informations non publiques sur SW_j. Les sanctions consistent à examiner les niveaux de confiance/réputation de SW_i si il s'agit de la première fois. Sinon, faire sortir le SW_i du réseau si ses niveaux se retrouvent au dessous d'un certain seuil.
- Le contrôle des interdictions nécessite que sn_{auth} vérifie si SW_j a été bien utilisé comme composant/substitut dans une composition en cours de construction/exécution afin que SW_i puisse recueillir les détails sur SW_j. Si SW_j n'a pas été utilisé comme prévu, ce serait une interdiction de collecter des détails sur SW_j. Les compensations consistent à informer SW_j de cette situation ainsi que de lui donner plus des privilèges d'accès comme le suivi de tous les pairs demandant ses détails.

C_{Resp_2}. Les moyens de monitoring et de compensations sont comme suit :

- Le monitoring des violations nécessite que sn_{auth} vérifie la véracité des détails sur les messages de sws_i. Pour cela, sn_{auth} suit les opérations de SW_i au fil du temps. Si cette véracité n'est pas confirmée, ce sera une violation d'afficher des détails valides. Les compensations incluent la contrainte à SW_i de revoir ses détails.

2.5.3 Engagements métier

Ces engagements ont les SWSs comme debtor et les $comp_{orch}$ (Section 2.5.1) comme creditor. Rappelons que $comp_{orch}$ recherche les SWSs nécessaires à la composition, fixe les incompatibilités sémantiques entre les SWSs, lançe l'exécution des SWSs, et traite les recommandations des SWs en termes d'expansion de la composition avec de nouveaux pairs ou de remplacement en substituant le SWS défaillant avec d'autres pairs. Les responsabilités métier futures des SWSs sont conformes à leurs responsabilités sociales et définies comme suit.

- $Resp_6$. Échanger n'importe quel détail (d) dans une composition devrait indiquer l'usage prévu (u) de d par un consommateur (c). d fait référence aux données mises à la disposition de $comp_{orch}$ et aux autres membres de la composition. Nous identifions 2 possibles usages de d, à savoir alimenter les décisions prises au niveau opérationnel et/ou flux de contrôle. $Resp_6$ est représentée comme Obligation(submit(d, c, valide(u))) où submit est l'action de soumettre et valide est une fonction confirmant ou pas u.
- $Resp_8$. Recommander un SWS à $comp_{orch}$ devrait indiquer son profil (f) (i.e., propriétés fonctionnelles et non-fonctionnelles) et le rôle (f) dans une composition (i.e., collaborateur ou remplaçant). $Resp_8$ fait référence d'une certaine manière à $Resp_1$. En effet, la collecte des détails sur un SWS pour une participation possible à une composition nécessite au préalable sa permission. $Resp_8$ est représentée comme Obligation(recommend($comp_{orch}$, SWS, f, valid(r)) où recommend est l'action de recommander un service, SWS est le service recommandé à $comp_{orch}$, f et r sont ses profil et rôle, et valide est une fonction approuvant ou pas r.

Modélisation. Les engagements associés à $Resp_6$ et $Resp_8$ sont modélisés en utilisant le même formalisme que pour les engagements sociaux.

- C_{Resp_6} (SWS_i, (SWS_j, submit(d,(SWS_j) | valid(u, d)) est un engagement conditionnel de SWS_i envers SWS_j (i.e., si valide(u, d) s'avère vrai alors submit(d,SWS_j) sera satisfait).
- C_{Resp_8} (SWS_i, $comp_{orch}$, recommend($comp_{orch}$, (SWS_j, f) | valid(r,SWS_j)) est un engagement conditionnel de SWS_i envers $comp_{orch}$ (i.e., si valid(r,SWS_j) s'avère vrai alors recommend($comp_{orch}$, SWS_j, f) sera satisfait).

La chronologie des interactions debtor-creditor est comme suit :

1. request(de : SW_i [1] ; à : $comp_{orch}$; contenu : recommend(SWS_j : SW_i effectue une recommandation auprès de $comp_{orch}$ pour le rajouter à la composition.
2. inform(de : $comp_{orch}$; à : SW_i ; contenu : approval | refusal) : $comp_{orch}$ informe SW_i de sa décision relative à la recommandation.

Après approbation, $comp_{orch}$ fait de SW_j un membre de la composition et en informe les autres membres existants si nécessaire. SW_j doit alors se conformer à toutes les responsabilités de cette composition en termes d' échange des détails et de recommandation de collaborateurs et substituts.

1. SW_i est déjà membre de la composition.

Monitoring. Le suivi de la conformité de SWS_i et SWS_j est mise en place par $comp_{orch}$ en exécutant create faisant passer les engagements à conditional. Nous allons passer en revue les transitions d'état reportées dans le cycle de vie en Figure 2.8 comme suit :

- conditional→ base : Si la condition est valide (i.e., valide(u,d) est vrai) ou bien il n'y a pas de condition, detach est exécutée. Selon la nature des responsabilités, les conditions sont évaluées différemment. Par exemple, $Resp_6$: SWS_j vérifie si d est à usage contrôle ou opérationnel et $Resp_8$: $comp_{orch}$ vérifie si r est collaborateur ou substitut.
- base→ satisfied : Si le contenu est satisfait (e.g., submit(d, c, valide(u))), discharge est exécutée.
- base→ violated : Si le contenu n'est pas satisfait, cancel est exécutée.
- base→ null : Si SWS_i échoue dans une composition, alors expire est exécutée faisant passer tous les engagements pris à null.

Plusieurs scénarios associés à violated sont possibles :

- C_{Resp_6}. Une violation se manifeste lors d'une soumission de détail sans la demande explicite du consommateur. Une interdiction se produit lors d'une utilisation de d malgré la désapprobation de l'émetteur de d.
- C_{Resp_8}. Une violation se présente lors de la recommandation d'un SWS avec un profil incorrect. Une interdiction survient lors de l'ajout d'un SWS alors que son rôle n'est pas valide.

Le monitoring des engagements est à la charge de $comp_{orch}$ ainsi que des SWS_i membres et se présente comme suit.

- C_{Resp_6}. Les moyens de monitoring et de compensations/sanctions sont :
  - Le monitoring des violations nécessite de SWS_j de révéler à $comp_{orch}$ l'inadéquation des détails reçus de ce dernier. Une violation d'inonder SWS_j avec des détails inappropriés est signalée si cette situation se produit régulièrement. Les sanctions consistent à réduire le niveau de confiance des SWSs ayant envoyé ces détails à $comp_{orch}$.
  - Le monitoring des interdictions exige de SWS_i de vérifier si les détails envoyés à $comp_{orch}$ sont utilisés comme prévu (i.e., contrôle ou fonctionnalité). Une interdiction d'utiliser de manière inappropriée les détails de SWS_i est alors signalée. Les compensations consistent à informer SWS_i du déroulement des opérations et de lui donner plus de privilèges (e.g., demander la raison de l'envoi de ses détails).
- C_{Resp_8}. Les moyens de monitoring et de compensations/sanctions sont :
  - Le monitoring des violations nécessite des vérifications par $comp_{orch}$ de l'exactitude du profil de SWS_j. Si cette exactitude n'est pas confirmée, une violation de recommander de SWSs est alors reportée. Les sanctions consistent à réduire le niveau de confiance des SWS_i ayant effectué la recommandation.

- Le monitoring des interdictions exige du $\mathsf{comp}_{\mathsf{orch}}$ de vérifier si le rôle de $\mathsf{SWS}_{\mathsf{j}}$ correspond à celui prévu (i.e., collaboration ou substitution). Si cette prévision n'est pas confirmée, une interdiction d'ajouter $\mathsf{SWS}_{\mathsf{j}}$ à la composition est signalée. Les compensations consistent à informer $\mathsf{SWS}_{\mathsf{i}}$ de l'inadéquation de $\mathsf{SWS}_{\mathsf{j}}$ dans cette composition.

2.6 Positionnement

Parmi les nombreuses études visant à améliorer la sûreté de fonctionnement des services Web au niveau composant et composite, plusieurs approches se basent sur la réplication classique. Cependant, peu de travaux utilisent la diversité pour rendre les services Web composant et composite tolérants aux fautes. En règle générale, il est impossible de faire face à toute sorte de fautes dans une seule et unique solution. Par conséquent, les différentes approches sont basées sur des hypothèses de fautes à tolérer. La diversité est la meilleure solution pour éviter les fautes de mode commun pouvant persister dans des répliques identiques. Il existe deux façons d'appliquer la diversité des services, le protocole séquentiel tel qu'il est utilisé dans le *RecoveryBlocks* et le protocole parallèle tel qu'il est utilisé dans le $N - VersionProgramming$

Les stratégies de diversité ci-dessus ont été utilisées dans quelques solutions existantes. Contrairement aux mécanismes et protocoles de tolérance aux fautes basés sur la réplication qui sont bien formalisés et le contexte de leur application bien défini, ceux de la diversité restent un problème ouvert. De plus, les travaux existants manquent de détails sur la sélection des protocoles adéquats pour tolérer un type particulier de fautes, ainsi que le nombre de services Web à diverses utilisés en tenant compte de la criticité de la fonctionnalité implémentée par les services.

En effet, il n'existe aucune garantie dans une composition de services Web que l'ensemble des services la composant soit fiable. Un degré de criticité doit être attribué pour mesurer la gravité de la défaillance sur la composition. Aussi, le choix du nombre des services Web à exécuter dépend de cette mesure. D'autre part, une sélection inappropriée des services Web conduit à des résultats erronés et/ou incomplets. Une sélection des services appropriés améliore considérablement la sûreté de fonctionnement de l'ensemble de l'application. Par exemple, dans le protocole parallèle les services peuvent achever un consensus avec une valeur erronée si l'algorithme sélectionne les mauvais services similaires. Aussi, les travaux utilisant la diversité comme moyen pour tolérer les fautes, discutent rarement des stratégies de sélection.

Pour résumer, plusieurs problèmes de tolérance aux fautes dans les services composants et composites et non traités de manière satisfaisante (Section 2.3) :

1. Comment concevoir et superviser le fonctionnement d'une application à base de services qui soit tolérante aux fautes et qui utilise la notion de diversité.
2. Combien de répliques sémantiquement équivalentes par fonctionnalité, et ce en se basant sur la notion de criticité.
3. Comment configurer la diversité de services pour une meilleure sûreté de fonctionnement des applications à base de services. Plus précisément, comment identifier les meilleurs k services dans une diversité pour un protocole donné, comment les services interagissent entre eux (vote, ordonnancement).

Quoique largement étudiée, la découverte de services Web reste une tâche délicate et longue pour les utilisateurs, qui est due principalement à une offre abondante de services. Les services Web ont longtemps été conçus comme des composants passifs, réagissant uniquement à la demande des utilisateurs. Les standards existants de découverte comme UDDI permet aux entreprises de publier leurs services Web afin qu'ils puissent être identifié en utilisant une recherche par mots-clés. Alors que les UDDI prolifèrent avec leur augmentation de contenu, en les examinant tout est devenu le temps consommant et inefficace [3]. D'autres alternatives proposent des techniques de recherche d'information telles que la fréquence des termes à partir de la description des fonctionnalités d'un service Web [74]. Malheureusement, ces techniques ne sont pas révéler les relations sémantiques entre les services Web. Dans la communauté du Web sémantique, la logique de description est utilisée pour développer et déduire des mécanismes établissant la similarité entre un service Web donné et d'autres pairs [83]. Malheureusement toutes ces approches de découverte traitent les services Web comme des composants isolés n'interagissant pas entre eux, contrairement à la construction incrémentale des compositions de services.

Durant la dernière décade, une tendance est plutôt de les considérer comme des éléments sociaux qui, dans leur fonctionnement, intégreraient des informations sur leur environnement et leurs interactions passées. Cette dimension sociale découle directement de ces interactions pouvant être assimilées à certains aspects de notre vie quotidienne, par exemple la collaboration ou encore la substitution. En effet, les services Web une fois découverts peuvent se retrouver dans plusieurs situations d'interaction comme par exemple une composition ou encore un groupe de services assurant la haute-disponibilité d'autres pairs. L'idée est alors d'identifier les différents types d'interactions pouvant exister entre ces services et de représenter ces interactions passées dans des réseaux sociaux afin d'améliorer la qualité de la découverte. Le résultat de cette découverte a permis de construire des groupes de diversité efficaces pouvant s'adapter dans un environnement dynamique.

Essentiellement, notre travail met l'accent sur les interactions précédentes que les services Web ont connues afin de construire leurs réseaux sociaux (Section 2.4). Ce faisant, il devient possible d'identifier avec qui un service Web peut collaborer, qui peut se substituer à un service Web, et qui peut rivaliser avec un service Web.

Pour assurer la stabilité des réseaux sociaux, nous nous sommes basées sur la notion d'engagement. Des engagements métier et sociaux ont été formellement définis ainsi que les mécanismes de monitoring associés et les sanctions à appliquer dans le cas de violation des engagements (Section 2.5).

Singh et al. sont les premiers à préconiser l'examen des principes de l'architecture orientée services (SOA) du point de vue des engagements [80]. Les SOA traditionnelles reposent sur des abstractions de bas niveau ne permettant pas de saisir les caractéristiques intrinsèques des services métier tels que l'autonomie, la complexité, et l'adaptabilité. Au contraire, une SOA basée sur des engagements permet de soutenir la conformité "métier" sans dicter une logique opérationnelle spécifique.

En dehors des SOA, les engagements sont largement adoptés dans les systèmes multi-agents comme la vérification des interactions des agents avec le travail de El-Menshawy et. al. [26], la spécification des jeux de dialogue de persuasion avec celui de Bentahar et al. [15], l'analyse des correspondances au niveau de service (SLA)

avec celui de Paschke and Bichler [67], et enfin, mais non des moindres, la génération de protocoles corrects de contrats avec celui de Narendra [64]. El-Menshawy et. al. stipulent que les approches existantes ne parviennent pas à saisir la signification des interactions qui surviennent dans des scénarios métier réels. Au contraire, les engagements captent la signification de haut niveau des messages, ce qui rend les protocoles d'interaction flexibles et intuitifs. Bentahar et al. modélisent les jeux de dialogue de persuasion en utilisant des engagements et des arguments. Chaque jeu est spécifié par ses conditions d'entrée, sa dynamique et sa condition de sortie. Enfin, Paschke et Bichler prennent des mesures de réparation en réponse à la violation de SLA en intégrant le calcul d'événements et l'algèbre complexe d'événements/actions pour définir les règles évènement-condition-action.

2.7 Conclusion

Dans notre travail sur la tolérance aux fautes (Section 2.3), une architecture multi-niveaux de managers pour la gestion des fautes a été proposée dans le contexte de services Web. Ces managers ont le rôle de contrôler chaque niveau : composant, composite et groupe de diversité. Ils implémentent différentes techniques de détection et de recouvrement d'erreurs déclenchées quand les services Web ne satisfont plus les caractéristiques de sûreté demandées. Le comportement des managers dédiés aux différents niveaux de l'architecture proposée est modelisé par des automates à états finis. Deux nouveaux protocoles d'exécution séquentiel et parallèle du groupe de diversité ont été conçus en adaptant ceux utilisés pour la réplication. Pour chaque protocole, le degré de redondance nécessaire est calculé afin de redimensionner groupe de diversité de manière à s'accorder à l'évolution des fautes observées (e.g., nombre et type de fautes). Pour la formation de groupes de diversité, nos recherches sur la découverte de services basée sur des principes de réseaux sociaux (Section 2.4) a permis de mettre en lumière différents types d'interaction entre services : substitution, compétition, et collaboration. Une méthodologie de construction de ces réseaux a été décrite en termes d'étapes : clustering, interconnexion, navigation, et ré-évaluation. Une méthode de sélection des services a été proposée. Afin d'assurer une stabilité des réseaux sociaux, notre contribution sur les engagements (Section 2.5.1) se résume en une approche pour réguler le fonctionnement des services Web sociaux en utilisant des engagements. Deux types d'engagements ont été identifiés : social et métier. Les engagements "sociaux" font référence à des règles d'adhésion des services Web dans les réseaux sociaux, de sorte que ces services Web peuvent être appelés services Web sociaux. Les engagements "métier" font référence à la participation des services Web sociaux dans les compositions. Les engagements ont été jugés appropriés en raison des actions menées par les services Web sociaux par rapport aux services Web (réguliers) comme par exemple, l'établissement et le maintien de réseaux de contacts. La détection des violations d'engagements et des interdictions a également été examiné dans ce travail. L'objectif est de définir des récompenses et des sanctions en réponse à ces violations et interdictions, respectivement.

Chapitre 3

Gestion robuste de la confiance des ressources Web

3.1 Introduction

Avec l'apparition des applications Web 2.0 (e.g., Facebook et Amazon), les utilisateurs ont la possibilité de partager leurs avis sur les ressources (e.g., services Web) avec d'autres pairs. L'utilisateur, simple consommateur au départ, devient un acteur en produisant du contenu sur le Web (e.g., commentaires et discussions dans les forums). L'utilisateur se retrouve indécis à cause d'une part, du choix parmi une multitude de ressources et d'autre part, du risque élevé de fraudes ou d'inexactitude des avis partagés. Le critère de confiance représente alors un facteur clé dans la sélection/recommandation des ressources Web. La confiance se définit comme étant la fiabilité d'une ressource à fournir les fonctionnalités et honorer la QoS prévue. Il existe 2 types de modèles de confiance basés respectivement sur les évaluations (i.e., retours d'expérience) fournies par des pairs (e.g., [98]) et l'observation du comportement de la ressource au fil du temps (e.g., [92]). Comme les ressources Web n'ont pas toutes un comportement proprement dit, le second type de modèles est moins susceptible de convaincre les utilisateurs actuels prêtant plus d'importance aux avis d'autres pairs sur les réseaux sociaux. Pour ces raisons, nous nous sommes intéressés à celui basé sur les évaluations par les pairs. De nombreux systèmes de recommandation ont été proposés dans la littérature. La vulnérabilité de ces systèmes face aux attaques (e.g., évaluations biaisées et multiples) continue à nuire à leur efficacité et par conséquent limite leur utilisation. Rendre ces systèmes robustes s'avère alors plus important que l'amélioration de leur performance [85]. Plusieurs problèmes liés à la robustesse ont été soulevés et sont considérés encore comme des défis ouverts.

Comme les informations sur la ressource Web peuvent être incertaines, les questions de recherche abordées dans ce travail sont catégorisées selon le type d'incertitude auquel peut faire face les systèmes de recommandation. Le premier type d'incertitude résulte de l'incohérence des évaluations fournies par les utilisateurs au fil du temps. Pour y pallier, une solution est d'inclure la crédibilité des utilisateurs lors du calcul de la confiance. Cependant, les approches existantes de confiance basées sur la crédibilité supposent que les utilisateurs ont soit une bonne expertise, soit une

certaine fiabilité. Le deuxième type d'incertitude résulte de la possibilité de créer des identités virtuelles dans le système afin d'émettre de fausses évaluations dites attaques Sybil. Les approches proposées pour contrecarrer ce type d'attaques filtrent les évaluations afin de repérer les utilisateurs Sybil. Toutefois, elles se concentrent seulement sur l'identité des utilisateurs. Finalement, le troisième type d'incertitude résulte de l'incohérence des valeurs de la QoS induite par la nature dynamique de la ressource et/ou à un comportement malveillant venant du fournisseur de la ressource. Un utilisateur fait confiance à une ressource si cette dernière satisfait de manière significative un grand nombre de ses demandes. Pour adresser ces 3 limitations, nous proposons de : (1) évaluer la crédibilité en prenant compte à la fois l'expertise et la fiabilité (Section 3.3), (2) filtrer les évaluations fournies en se basant sur leur qualité et la crédibilité des utilisateurs (Section 3.4), et (3) estimer la confiance de la ressource en termes de probabilités (Section 3.5).

3.2 Fondements de la confiance

Cette section présente les types d'attaque considérés et une vue globale de notre approche pour évaluer la confiance des ressources en présence d'incertitude comme mentionné en Section 3.1.

3.2.1 Types d'attaque considérés

Généralement, la robustesse des systèmes de recommandation fait référence à leur stabilité en présence d'évaluations fausses (i.e., biaisées et/ou multiples), appelées attaques et insérées de manière intentionnelle afin d'influencer les recommandations. Influencer ces systèmes pour changer la cote d'une ressource peut être profitable à son propriétaire ou à des concurrents.

- Évaluations biaisées. Ce type d'attaque consiste à injecter des évaluations biaisées pour promouvoir ou rétrograder la ressource. Pour contrecarrer ces attaques, certaines approches (e.g., Cloud Armor [65] et RateWeb [57]) considèrent la crédibilité de l'utilisateur dans ses propos lors du calcul de la confiance. Le principe est d'établir un consensus au sein de l'ensemble des utilisateurs représentant l'avis de la majorité. Les utilisateurs proches de cet avis sont plus crédibles que ceux s'en trouvant éloignés. D'autres approches tout aussi pertinentes associent cette crédibilité au niveau d'expertise de l'utilisateur [72].
- Évaluations multiples. Ce type d'attaque dit Sybil consiste à injecter de faux profils utilisateur évaluant positivement la ressource et/ou négativement les concurrents. Pour traiter les attaques Sybil, des approches de confiance basées sur les réseaux sociaux (ex., [102] et [103]) procèdent à une étape de filtrage des évaluations avant de calculer la confiance. En effet, les utilisateurs Sybil, se dissimulant derrière des fausses identités, ont peu de chance de gagner la confiance des utilisateurs réels (non-Sybil) et par conséquent de créer un nombre limité de liens avec ces derniers.

3.2.2 Catégories d'utilisateurs

Tout utilisateur lambda dont l'avis est proche de celui de la majorité est considérée comme fiable et expert, et par conséquent, crédible. Néanmoins, un expert peut être strict dans son évaluation/jugement et par conséquent diffère de la majorité considérée comme crédible. Nous catégorisons les experts en 2 classes (i.e., stricts et non-stricts). Les stricts ont une solide expertise associée à une grande fiabilité dans une certaine communauté contrairement aux lambda. Nous nous basons sur les travaux de Schum et al. [77] pour caractériser les utilisateurs crédibles selon 3 facteurs : véracité - utilisateurs disant la vérité, objectivité - évaluations basées sur des preuves, et précision - utilisateurs estimant leurs évaluations de manière non-approximative (Tableau 3.1).

Catégorie	Véracité	Objectivité	Précision
lambda	✓	X	X
strict	✓	✓	✓
non-strict	✓	✓	X

Tableau 3.1 – Catégories des utilisateurs crédibles

Des études ont été menées en psychologie sociale (e.g., [44] et [82]) pour évaluer l'impact de la crédibilité de la source sur les croyances et les changements d'attitude. Il est démontré que les sources crédibles sont persuasives et peuvent influencer les croyances actuelles (e.g., évaluations) et les attitudes des pairs de manière significative comparée aux sources non crédibles. Par conséquent, les utilisateurs stricts, en se basant sur leur expertise, peuvent encourager les utilisateurs à revoir leurs évaluations. Pour formaliser cette situation, nous nous appuyons sur le paradigme d'apprentissage actif de Yager [99]. Dans notre travail, ce paradigme s'applique aux situations dans lesquelles les évaluations sont correctes (ou fausses) mais pas nécessairement précises, demandant un léger (ou important) raffinement par les membres appartenant à la majorité. Notre proposition consiste donc à réduire l'écart entre les évaluations des stricts et celles de la majorité actuelle de manière à parvenir à un consensus. Les stricts devraient appartenir à plusieurs groupes et influencer les croyances de ces groupes de différentes manières (e.g., fortement ou faiblement). L'appartenance à des groupes (i.e., forte et/ou faible) peut être incertaine et dépend du domaine et/ou des préférences de l'utilisateur. De fait, nous avons opté pour une technique de clustering flou des évaluations afin de traiter l'incertitude de l'appartenance d'un utilisateur à différents clusters (Section 3.3).

3.2.3 Description globale de l'approche

Pour rendre robuste la gestion de la confiance, notre idée est de considérer le facteur d'incertitude dans le processus de recommandation des ressources. L'approche proposée se base sur des évaluations quantitatives (i.e., $\in [0, 1]$) fournies par les utilisateurs et n'exige aucune information sur leurs identités ni leurs interactions, assurant une totale protection de leur vie privée. Elle comporte 3 phases (Figure 3.1) :

- Phase 1 : évaluation de la crédibilité. Deux critères caractérisent la crédibilité :

fiabilité (i.e., en terme de proximité de l'avis majoritaire) et expertise (i.e., en terme de sévérité dans les évaluations). Un expert n'a aucun intérêt à s'aligner avec l'avis majoritaire et se voit ainsi écarter de la majorité en dépit de sa fiabilité. L'idée est donc de réduire l'écart pouvant exister entre les évaluations des experts et l'avis majoritaire. Notre modèle de crédibilité se base sur la technique de clustering flou (Section 3.3.1) où les évaluations peuvent appartenir à 1-n clusters avec des degrés d'appartenance.

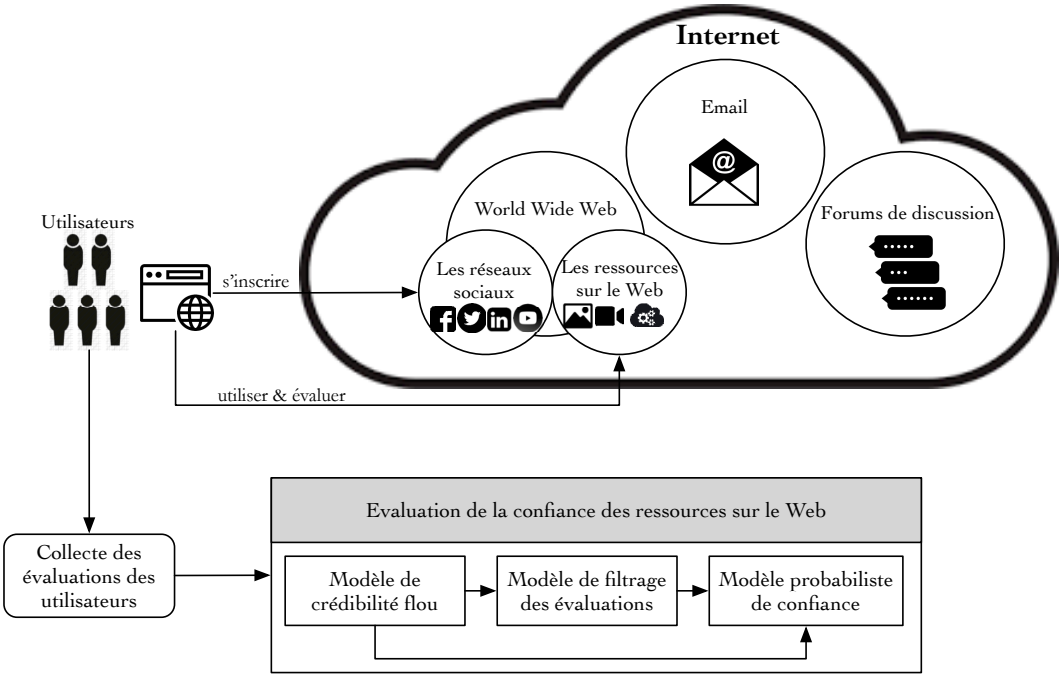


FIGURE 3.1 – Démarche d'évaluation de la confiance

- Phase 2 : filtrage des évaluations. Face aux attaques Sybil, il est nécessaire d'analyser la structure du réseau social d'utilisateurs (Section 3.4). Nous partons du principe que les utilisateurs réels et honnêtes n'établissent de liens avec d'autres pairs seulement s'il existe un lien de confiance véritable entre eux. L'idée est donc d'accorder un pouvoir d'évaluation aux utilisateurs selon leur crédibilité et de sélectionner une chaîne d'utilisateurs dont les évaluations seront incluses lors du calcul de la confiance.
- Phase 3 : calcul de la confiance. Prendre en compte la crédibilité permet de résoudre le problème d'incohérence des évaluations lors du calcul de la confiance. Cette incohérence se manifeste par une divergence des avis des utilisateurs. Troffaes a démontré que le recours à des probabilités permet de formaliser le problème de divergence d'avis [88]. Par conséquent, nous associons la notion de crédibilité à celle de probabilité. Les bases de données probabilistes associées à la sémantique de la théorie des mondes possibles constituent un modèle intéressant au calcul incertain de la confiance (Section 3.5.3).

Par la suite, nous décrivons en détail le modèle associé à chacune des phases.

3.3 Modèle flou de crédibilité

Cette section présente notre modèle flou de crédibilité en présence d'évaluations biaisées, en 3 étapes (Figure 3.2) : i) clustering des évaluations, ii) recherche du cluster majoritaire, et iii) calcul de la crédibilité.

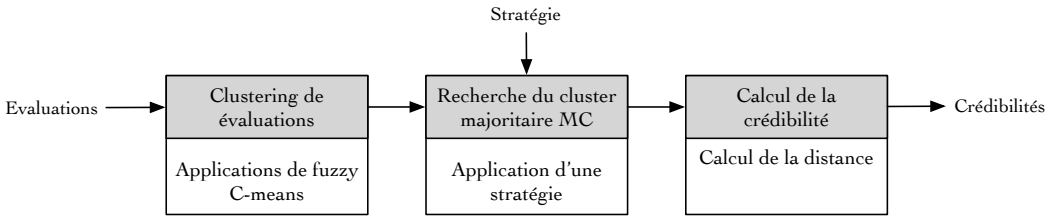


FIGURE 3.2 – Vue globale du modèle flou de crédibilité

3.3.1 Clustering des évaluations

Cette étape consiste à structurer les évaluations en classes homogènes (clusters) en terme de similarité. De manière générale, le clustering est une technique d'apprentissage non supervisé ayant pour but de déterminer des corrélations entre données. Conformément aux arguments donnés en Section 3.2.2, nous adoptons $\mathcal{C}$-means [16] comme clustering flou pour représenter les incertitudes et imprécisions liées aux évaluations. Ce type de clustering est basé sur la notion de clusters flous et permet à une donnée d'appartenir plus ou moins fortement à un cluster (C_j). Ainsi, un utilisateur strict peut appartenir au cluster majoritaire avec un certain degré d'appartenance dû à sa sévérité.

$\mathcal{C}$-means est décrit par l'algorithme 2 ayant comme paramètres :

- X_i : vecteur d'évaluations quantitatives de la ressource R fournie par un utilisateur u_i,
- m, nb : degré du flou et nombre de clusters
- epsilon : critère de terminaison pour valider la construction des clusters finaux,
- MA : matrice d'appartenance où $MA_{i,j}$ correspond au degré d'appartenance de X_i à C_j.

Le principe de cet algorithme est d'identifier les centroide(C_j) minimisant la distance euclidienne (i.e., mesure de similarité) avec X_i et les MA_{ij} associés.

3.3.2 Consensus majoritaire

Nous proposons 3 stratégies pour la recherche du cluster majoritaire. Les stratégies reposent sur les valeurs qualitatives du degré d'appartenance à un cluster flou : faible, modéré, et fort. L'idée principale est de trouver le cluster le plus peuplé en raisonnant sur les degrés d'appartenance des évaluations.

Stratégie 1 : faible. Une évaluation appartient à un cluster si son degré d'appartenance est strictement positif. Cette stratégie conserve la taille actuelle des clusters. Le cluster le plus peuplé est désigné $C_{majoritaire}$ défini par l'équation 3.1 :

Algorithme 2 : Fuzzy C-means

Entrée : $X=\{X_{i=1,n}\}$, nb, $m \in [0,1]$, epsilon
Sortie : MA, $C_{j=1,nb_{cluster}}$
1 $MA \leftarrow MA^{(0)}$; $k \leftarrow 0$
2 $centroide(C_j) \leftarrow \sum_{i=1}^{n}(MA_{ij}^{m} \times X_i) / \sum_{i=1}^{n} MA_{ij}^{m}$
repeat
  compute $MA^{(k+1)}$ such that $MA_{ij} = [\sum_{p=1}^{nb}(\frac{\|X_i - centroide(C_j)\|}{\|X_i - centroide(C_p)\|^{\frac{2}{m-1}}})]^{-1}$;
  k=k+1;
until $\left\| MA^{(k)} - MA^{(k+1)} \right\| < epsilon$;

$$C_{majoritaire}^{faible} = C_j,\ |C_j| = \max_{\forall k=1,nb}(|C_k|) \wedge\ MA_{i,k} > 0, i \in [1,n] \tag{3.1}$$

Dans cette stratégie, plus les degrés d'appartenance aux clusters sont faibles, plus le recouvrement des clusters est important. Lorsque epsilon n'est pas assez faible, les clusters flous représenteront un recouvrement total où tous les degrés d'appartenance sont non nuls. Il devient alors difficile de déterminer le cluster majoritaire. Pour remédier à ce problème, une autre stratégie dite modérée est proposée.

Stratégie 2 : modéré. Une évaluation est comptabilisée dans un cluster seulement si son degré d'appartenance est supérieur à un certain seuil gamma. $C_{majoritaire}$ est défini par l'équation 3.2 :

$$C_{majoritaire}^{modéré} = C_j,\ |C_j| = \max_{\forall k=1,nb}(|C_k|) \wedge\ MA_{i,k} \geq gamma, i \in [1,n] \tag{3.2}$$

Les stratégies faible et modéré dépendent fortement de la répartition des évaluations dans les différents clusters obtenus. Quoique $C_{majoritaire}$ est facilement identifiable, cette répartition peut porter à confusion lors d'un calcul automatique. Nous avons cherché à trouver une stratégie plus robuste basée uniquement sur les degrés d'appartenance des évaluations afin d'être sûr de sélectionner le cluster le plus peuplé.

Stratégie 3 : fort. $C_{majoritaire}$ correspond au cluster ayant le plus haut degré d'appartenance sur l'ensemble des évaluations et est défini par l'équation 3.3 :

$$C_{majoritaire}^{fort} = C_j,\ |C_j| = \max_{\forall k=1,nb}(\sum_{i \in [1,n]}(MA_{i,k})) \tag{3.3}$$

3.3.3 Calcul de la crédibilité

$C_{majoritaire}$ étant établi, la crédibilité des u_i se définit par la similarité entre X_i de dimension d et $centroide(C_{majoritaire})$. Pour des évaluations quantitatives (i.e., échelles de valeurs continues), la distance euclidienne (d_{eucl}) reste la plus utilisée pour la

mesure de similarité (Équation 3.4). La forme normalisée de d_{eucl} est préconisée pour des échelles de valeurs non-homogènes.

$$d_{eucl}(X_i, X_j) = (\sum_{k=1}^{d}(X_{i,k} - X_{j,k})^2)^{1/2},\ X_i = (X_{i,1}, ..., X_{i,d}) \tag{3.4}$$

La crédibilité de u_i (CR_i) est définie par l'équation 3.5 :

$$CR_i = 1 - d_{eucl}(X_i, centroide(C^s_{majoritaire})),\ s \in \{faible,\ modéré,\ fort\} \tag{3.5}$$

Traditionnellement, les CR_i sont utilisées dans le modèle de confiance d'une ressource res décrit par l'équation 3.6 :

$$Confiance(res) = \frac{1}{\sum_{i=1,n} CR_i} \times \sum_{i=1,n}(CR_i \times X_i) \tag{3.6}$$

Chaque futur consommateur de res (cons) fera appel à ses accointances u_i fournissant les évaluations pour calculer Confiance(res). Confiance(res) semble être une "bonne" mesure mais ne permet pas d'affirmer la fiabilité de res en raison de la connaissance limitée des u_i. Une solution serait que cons consulte d'autres pairs (u'_i) ayant au préalable établi Confiance(res, u_k) à partir d'autres évaluations fournies par leurs accointances. Par conséquent, la réputation de res (REP) est un facteur à prendre en compte lors du calcul de Confiance(res). La réputation de res est décrite par l'équation 3.7 :

$$REP(res) = \frac{1}{|\{u'_k\}|} \times \sum_{k=1,p} Confiance(res, u_k) \tag{3.7}$$

Dans [71], Ramchurn et al. différencient la confiance et la réputation ; la première est issue des interactions directes et la seconde est fournie par d'autres agents permettant d'établir la confiance. Par conséquent, nous proposons de combiner à la fois Confiance(res) et REP(res) pondérées respectivement par les scores de préférences de cons, α et β (Équation 3.8).

$$Confiance'(res) = \alpha \times Confiance(res) + \beta \times REP(res) \tag{3.8}$$

Notons que Confiance'(res) $\in$ [0,1].

3.4 Modèle de filtrage de Sybils

Cette section présente notre modèle de filtrage d'évaluations fausses et multiples (i.e., attaques Sybil), en 3 étapes (Figure 3.3) : i) élagage du réseau social d'utilisateurs (R), ii) construction du graphe de confiance (G), et iii) sélection des utilisateurs crédibles et non-Sybil. Avant de décrire ces étapes, il nous semble nécessaire de définir R et G.

Formellement, R et G sont 2 tuples <N, source, A> et <R élagué, C^N, C^A> où :

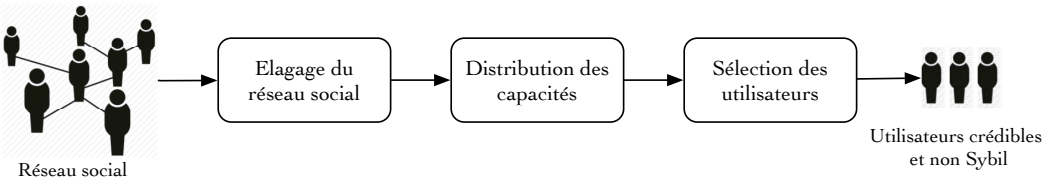


FIGURE 3.3 – Etapes pour le filtrage des évaluations

- N : ensemble des nœuds (N_i) représentant chacun un utilisateur u_i,
- source : noeud fiable servant de point de collecte de l'ensemble des évaluations fournies par N,
- A : ensemble des arcs (A_{ij}) reliant N_i à N_j.
- C^N : ensemble des capacités d'évaluation associées aux N_i,
- C^A : ensemble des capacités des A_{ij} de manière similaire à un réseau de transport.

Noter que C^{N_i} correspond à $\sum_j C^{A_{ji}}$ et revient à une capacité d'attaque si N_i est Sybil.

3.4.1 Élagage du réseau social

Notre idée est de trouver un compromis entre limiter la capacité d'attaque d'un nœud Sybil et donner au nœud honnête l'opportunité de s'exprimer. Comme C^{N_i} dépend du nombre de A_{ji}, réduire celui d'un Sybil permettrait de diminuer le taux des fausses évaluations multiples susceptibles d'être collectées par `source`. Nous adoptons la solution proposée par Tran et al. [87] de limiter le nombre de A_{ji} par nœud à un seuil prédéfini $e_{entrant}$. L'étape d'élagage décrite par l'algorithme 3 supprimant les A_{ji} excédant $e_{entrant}$ en fonction de la crédibilité des u_i. Ainsi, elle élimine les chemins redondants dans R permettant d'accélérer la collecte future des évaluations et d'éviter la propagation des évaluations d'un Sybil possédant des multiples liens avec des utilisateurs honnêtes.

3.4.2 Construction du graphe de confiance

Notre idée est d'augmenter le pouvoir d'évaluation des honnêtes et diminuer celui des utilisateurs malhonnêtes tout en respectant une capacité d'évaluation maximum (C_{max}. L'initialisation de C^A est faite de manière à permettre à un N_i honnête de transmettre une seule évaluation via ses $e_{entrant}$ A_{ji}. Nous proposons ainsi un mécanisme de distribution de capacités (ou tickets) basé sur les crédibilités CR_j (Section 3.3) et décrit par l'algorithme 4.

Chaque N_i redistribue C^{N_i} tickets via ses A_{ij} selon l'équation 3.9). Afin de conserver le flot de capacités dans G, $C^{A_{ij}}$ est arrondi $\lfloor C^{A_{ij}} \rfloor$ ou $\lceil C^{A_{ij}} \rceil$.

$$C^{A_{ij}} = \frac{C^{N_i} * CR_j}{\sum_{\forall N_k \in Fils_i} CR_k} \tag{3.9}$$

Algorithme 3 : Élagage du réseau social

Entrée : R, $e_{entrant}$
Sortie : R élagué

```
file.enfiler(source) ;
marquer(source);
while (!file.vide()) do
    i ← file.défiler() ; k =|{Aij}| ;
    if k > e_entrant then
        supprimerArcsSelonCrédibilité(k-e_entrant, i)
    supprimerArcsEntreFils(i);
    forall the Nj ∈ fils(i) ∧ nonMarqué(Nj) do
        marquer(Nj) ;
        enfiler(Nj)
```

L'algorithme 4 parcourt en largeur R élagé de manière à former un G où chaque u_i possède 1-m tickets. Le niveau de N_i est défini par rapport à sa distance de source (i.e., un nœud de niveau l possède 1-m parents de niveau l-1). N_i reçoit p tickets provenant des A_{ji}, en consomme 1 et redistribue le reste via A_{ik} aux N_k (i.e., $fils_i$).

Algorithme 4 : Distribution des capacités

Entrée : R élagué, $e_{entrant}$, C_{max}
Sortie : G

```
file.enfiler(source) ; marquer(source) ; C^source ← C_max;
while !file.vide() do
    c ← file.défiler();
    forall the Nk ∈ fils(c) ∧ nonMarqué(Nk) do
        compute C^Ack;
        if C^Ack = 0 then
            supprimer(C^Ack)
        else
            computeC^Nk ; marquer(Nk) ; enfiler(Nk) ;
```

A l'issue de cette étape, seuls les u_i possédant des tickets ont le droit de participer au processus d'évaluation.

3.4.3 Sélection des utilisateurs

Cette étape vise à sélectionner le nombre maximal d'utilisateurs crédibles et ayant une seule identité dans G. Cette sélection est formalisée comme une maximisation du flot passant par les chemins de source à un nœud puits rajouté à G le reliant aux feuilles. L'algorithme 5 est inspiré des travaux de Ford-Fulkerson [32] pour résoudre le problème de flot maximal et se base sur les fonctions suivantes :

(i) $F : A \rightarrow \mathbb{N}$ fait correspondre à A^{ij} de capacité $C^{A_{ij}}$ une valeur de flot $F(A_{ij}) \leq C^{A_{ij}}$.

(ii) $R_F : A \rightarrow \mathbb{N}$ fait référence à la capacité résiduelle telle que $R_F(A_{ij}) = C^{A_{ij}} - F(A_{ij})$.

(iii) G_F représente un état de flot dans G. G_F= <source, N, A_F, CR, C^A> où A_F= $\{A_{ij} \in A, R_F(A_{ij}) \geq 0\}$.

L'algorithme 5 retourne le flot maximal F et les u_i sélectionnés pour participer à l'évaluation. L'idée est d'atteindre un optimum global en se basant sur un choix optimal local (i.e., un chemin de source au puits contenant le nombre maximal d'utilisateurs crédibles et ayant une seule identité). À chaque itération, l'algorithme 5 commence de `source` et augmente le chemin de flot en recherchant le "meilleur" N_i selon le principe du A* [68]. Nous proposons 2 heuristiques pour la sélection du meilleur N_i : i) choisir le plus crédible, et ii) choisi le plus crédible ayant la valeur la plus basse de $h(N_i) = 1/C^{N_i}$ afin de minimiser $R_F(A_{source,\ i})$.

Algorithme 5 : Algorithme de sélection d'utilisateurs

```
Entrée : G
Sortie  : F, U={u_i}
  forall the A_ij ∈ A do
    F(A_ij) ← 0;
  U ← ∅
  repeat
    chemin = meilleurChemin(G_F, puits);
    foreach A_ij ∈ chemin do
      F(A_ij) ← F(A_ij) + 1;
      F ← F + 1 ; U ← U ∪ {N_i,N_j}
  until chemin=null;
```

3.5 Modèle probabiliste de confiance

Cette section présente notre modèle probabiliste de confiance en 2 étapes : i) modélisation des évaluations sous forme d'une base de données probabiliste (BDProb) et ii) calcul de la confiance sous forme d'interrogations à BDProb (Figure 3.4). Notre modèle de confiance se veut général en termes de modèle(s) de crédibilité utilisé(s) et paramétrable en terme de modèle d'incertitude adopté. Dans la suite, nous nous intéressons à 2 modèles d'incertitude : tuples indépendants (TID) ou blocs indépendants (BID).

3.5.1 Modélisation TID des évaluations

Il nous semble nécessaire de discuter du bon fondement des BDProbs dans le calcul de la confiance avant d'aborder la question de leur structure.

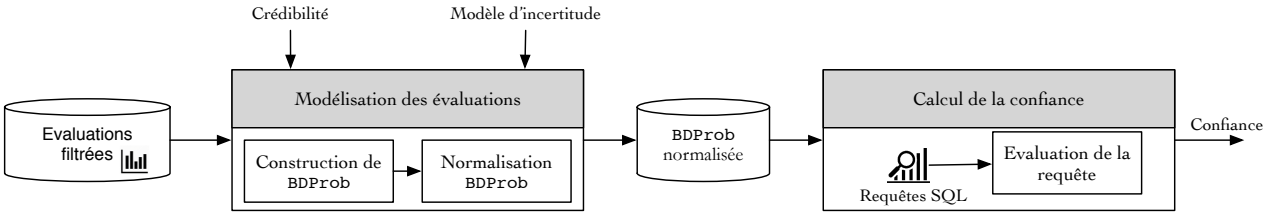


FIGURE 3.4 – Vue d'ensemble

Fondement des BDProbs. Dans [21], Dalvi démontre une corrélation entre la probabilité d'occurrence et le degré de confiance dans la validité des données (i.e., plus grande est la probabilité, plus élevé est le degré de confiance). Notre idée fût alors de faire correspondre la crédibilité des u_i considérée comme degré de confiance à la notion de probabilité. A titre illustratif, considérons u_1, u_2, et u_3 ayant une expérience avec une ressource res et la déclaration d suivante : u_i `a observé que res a satisfait ses demandes`. L'incertitude de d reflète la probabilité que d se produise réellement correspondant à CR_i. Soit 3 événements e_1, e_2, et e_3 relatifs à la déclaration par u_1, u_2, et u_3 que res ait satisfait leurs demandes. La combinaison de ces e_i lors du calcul de la confiance soulèvent les questions suivantes : compte tenu de la variation de crédibilités des u_i dans leurs évaluations de différentes res, comment agréger les évaluations et calculer la confiance de res à partir de ces 3 déclarations ? Les bases de données probabilistes constituent une solution intéressante au calcul de la confiance de res en offrant une meilleure représentation de l'incertitude sous-jacente à la crédibilité des u_i et permettant aussi une évaluation des requêtes de calcul incertain de la confiance de res [21].

Structure et sémantique de BDProb. Il s'agit de créer des tuples t_i pour stocker l'information suivante : u_i a fournit l'évaluation X_i concernant res_j. Concrètement, t_i représente l'avis de u_i sur res_j considéré comme incertain au vu de la crédibilité de u_i. t_i est alors associé une probabilité (prob(t_i)), à savoir un degré de confiance (i.e., crédibilité de u_i). Lorsque prob(t_i) est égale à 1/0, t_i est valide/invalide quelque soit la situation envisagée. BDProb a pour schéma relationnel R(res, u, eval, att_1, ..., att_n, p) où :

- res : identifiant de la ressource,
- u : identifiant de l'utilisateur,
- eval : degré de satisfaction de u pour res,
- att_1, ..., att_n : détails complémentaires (e.g., date d'émission de eval et identité du fournisseur de res),
- prob(t) : probabilité d'occurrence du tuple t correspondant à la valeur de crédibilité de u pour eval calculée par notre modèle de crédibilité flou (Section 3.3).

La sémantique de BDProb fait référence à l'ensemble de ses mondes possibles mdp_k où $mdp_k \subseteq$ BDProb est associé une probabilité (P_k) sous l'hypothèse de l'indépen-

dance des tuples (TIP). L'équation 3.10 définit P_k comme suit :

$$P_k = \prod_{t_j \in mdp_k} prob(t_j) * \prod_{t_j \notin mdp_k} (1 - prob(t_j)) \tag{3.10}$$

Normalisation de BDProb. BDProb contient des tuples rattachés à des u_i fournissant des eval sur différentes res. Un u_i peut être cohérent (i.e., toujours ou jamais crédible) ou inconsistant (i.e., variation entre crédible et non-crédible) dans leurs évaluations. En effet, certains u_i sont plus crédibles que d'autres en fournissant des évaluations correctes, et vice-versa. Si t est faux la probabilité qu'un autre t faisant référence au même u_i soit également faux se voit accroître. Ainsi, BDProb ne respecte pas le modèle des TIP pour lequel une probabilité est associée à chaque t indépendamment de l'occurrence des autres t.
Pour normaliser BDProb, nous la décomposons en 2 relations probabilistes de tuples indépendants (Figure 3.5) : PAIR et R_1. PAIR stocke tous les u_i avec leur valeur respective de crédibilité. Vu que PAIR doit être souvent mis à jour, elle est considérée comme une vue plutôt qu'une table. u_i est crédible pour res si ses évaluations sont consistantes.

PAIR

u	prob
u_1	0.1
u_3	0.88

R_1

res	u	eval	prob(t)
res_1	u_1	0.2	0.12
res_2	u_1	0.76	0.84
res_1	u_3	0.97	0.88

FIGURE 3.5 – Normalisation de BDProb

L'équation 3.11 calcule prob de PAIR (i.e., crédibilité de u_i) sur la base des évaluations fournies par le passé.

$$prob(t_i) = \prod_j CR(i, res_j) \tag{3.11}$$

3.5.2 Modélisation BID des évaluations

Dans la mesure où la crédibilité des u_i peut être évaluée selon des critères différents mais complémentaires comme l'expertise (e.g., [72]) et la fiabilité (e.g. [65]), notre idée fût de capitaliser sur m modèles de crédibilité (M_j) pour (u_i, res) associant ainsi un t à m probabilités d'occurrence.

Structure et sémantique de BDProb. Il s'agit de structurer BDProb sous forme de blocs de tuples t où chaque bloc fait référence à un modèle de crédibilité. L'hypothèse d'incertitude est relative au modèle des blocs indépendants (BID). BDProb contient des t clonés dont $prob(t_i)$ change selon le M_j. Le schéma de BDProb est R'(res, u, eval, att_1, ..., att_n, p) où mod désigne le modèle de crédibilité utilisé pour calculer p (Figure 3.6).

	res	u	eval	mod	p
t_{11}	res_1	u_1	0.2	M_1	0.07
t_{12}				M_2	0.06
t_{21}	res_1	u_2	0.76	M_1	0.44
t_{22}				M_2	0.42
t_{31}	res_1	u_3	0.97	M_1	0.43
t_{32}				M_2	0.44

FIGURE 3.6 – Représentation BID de BDProb

L'hypothèse BID réfère à 2 règles : exclusivité des tuples dans un bloc et indépendance des blocs. Nous organisons alors les tuples clonés d'un même u_i dans des blocs différents. Comme les M_j sont basés sue théorie différente, le choix de M_j est exclusif. Par conséquent, les tuples d'un même bloc correspondent à des événements disjoints ou mutuellement exclusifs validant ainsi la première règle BID. Comme les tuples appartenant à des blocs différents représentent des avis fournis par différents u, leurs tuples respectifs sont indépendants des uns des autres.

Les mondes possibles (mdp_k) de BDProb sont générés selon 2 hypothèses : M_j non-corrélés (i.e., leurs tuples respectifs peuvent être combinés ensemble) et M_j corrélés (i.e., un seul M_j utilisé à la fois). La probabilité de mdp_k est calculé par l'équation 3.10.

3.5.3 Estimation de la confiance

À notre connaissance, l'évaluation des requêtes probabilistes des BDProbs est limitée et en particulier, pour les opérateurs d'agrégation (Section 3.5.1). A titre illustratif et pour des raisons de simplicité, nous nous sommes focalisés sur le modèle TID. Par la suite, nous développons des requêtes spécifiques à BDProb pour établir la confiance de res.

Définition des requêtes. L'agrégation des eval en une valeur probabiliste peut s'exprimer à l'aide d'une requête SQL Select avg (Req). Exécuter Req sur chaque mdp_k signifie que les u_i dans mdp_k observent conjointement que res satisfait leurs demandes avec une probabilité P_k. Req est définie comme suit :

$$\textbf{Select } \textbf{avg}(eval) \textbf{ From } \text{BDProb } \textbf{where } \text{res} = \text{res}_j; \tag{3.12}$$

Exécuter Req sur $\{mdp_k\}$ aura comme résultat une valeur de confiance par mdp_k ainsi que la probabilité associée à mdp_k. Contrairement aux approches probabilistes existantes (e.g., [86] et [106]), Req peut être personnalisée selon les préférences des u_i pour estimer la confiance. Nous identifions 3 variantes de Req comme suit :

1. Req_1 calcule la confiance de res_j comme une moyenne des eval fournies par une liste prédéfinie (L) de u_i crédibles :

$$\text{Req}_1 : \textbf{Select } \textbf{avg}(eval) \textbf{ From } \text{BDProb } \textbf{where } \text{res} = \text{res}_j \textbf{ and } date \geq \text{"}2018-05-01\text{"};$$

2. Req_2 calcule la confiance de res_j comme une moyenne des eval fournies au delà d'une date donnée :

$$Req_2 : \textbf{Select avg}(eval) \textbf{ From } BDProb \textbf{ where } res = res_j \textbf{ and } u \text{ in } L;$$

3. Req_3 calcule la confiance d'un fournisseur (f_i) comme une moyenne des évaluations fournies sur toutes les res de f_i :

$$Req_3 : \textbf{Select avg}(eval) \textbf{ From } BDProb \textbf{ where } f = f_i;$$

Évaluation des requêtes. Malgré la simplicité de la sémantique des mondes possibles, la complexité d'évaluation des requêtes, même des plus simples, est exponentielle par rapport au nombre de tuples de la base de données [21]. Nous nous sommes basés sur l'approximation de avg proposée par Jayram et al. [37] pour calculer la confiance. Le problème se réduit alors à trouver une estimation efficace de l'intégrale de la fonction $h_{avg}(x)$. Cette fonction est basée sur la notion de flux de données probabiliste defini comme « une séquence de n tuples incertains $(t_i, p(t_i))$ où chaque tuple est présent dans une instance de la base avec une probabilité $p(t_i) \in (0, 1]$ (indépendamment de tous les autres tuples dans la base de données) ». Notre choix a été guidé par les critères suivants : (i) applicable pour un modèle TID, ii) bonne approximation de l'opérateur avg ; et (iii) faible complexité de l'algorithme.

Le calcul de la confiance décrit par l'algorithme 6 est effectué en 2 étapes : 1) extraire l'ensemble des évaluations correspondants à res_j, et 2) calculer l'estimation de la valeur de la confiance selon l'approximation de Jayram et al..

Algorithme 6 : Calcul de la confiance

Entrée : BDProb, res_j
Sortie : confiance
$X \leftarrow$ **Select** (*eval*) **From** BDProb **where** $res = res_j$;
$h_{avg}(x) \leftarrow \sum_i X_i.p(t_i). \prod_{j \neq i}(1 - p(t_j) + p(t_j)x)$
confiance $\leftarrow$ intégrale(0, 1, avg(x))

3.6 Positionnement

Plusieurs approches d'évaluation de la confiance ont été définies et implémentées dans divers domaines (ex. e-commerce, environnements P2P, et SOA). Les travaux basés sur la crédibilité calculent la crédibilité de l'utilisateur en utilisant des algorithmes de clustering tel que k-means (e.g., [57]), des mesures de similarité (e.g., [65, 98]) ou des modèles mathématiques (e.g., [93, 94]). Dans [57], Malik et al. proposent RateWeb, un système d'évaluation de la confiance pour les environnements orientés service ayant pour but d'optimiser la sélection et la composition des services Web. RateWeb évalue la confiance en utilisant un certain nombre de métriques prenant en compte l'aspect dynamique de ces environnements et également la possibilité de présence d'utilisateurs malveillants avec un comportement oscillant

dans le temps entre honnête et malhonnête dans leurs évaluations. Les métriques de RateWeb sont définies pour capturer la plupart des aspects de la confiance d'un point de vue social. Noor et al. [65] proposent CloudArmor, un système d'évaluation de la confiance dans le contexte du Cloud Computing. La caractéristique principale du système est la protection de la vie privée de l'utilisateur en préservant l'anonymat de leur identité et leurs interactions. Ces approches supposent que les utilisateurs possèdent une solide expertise et/ou une bonne fiabilité. Cependant, elles négligent les utilisateurs possédant à la fois l'expertise et la fiabilité. Nous appelons ces utilisateurs des stricts (ou sévères). Ces derniers n'ont aucun intérêt à s'aligner avec la majorité. Pour atteindre un consensus, nous utilisons la technique de clustering flou pour réduire l'écart entre les évaluations des utilisateurs stricts et l'avis de la majorité actuelle (Section 3.3).

Les attaques Sybil sont largement traitées dans la littérature. Certaines solutions se concentrent sur comment diminuer l'impact des attaques réalisées par des utilisateurs multi-identités (e.g., [87]). D'autres solutions préconisent la détection de ce type d'utilisateurs en analysant les profiles utilisateur. Dans [87], Tran et al. proposent SumUp conçu pour superviser les systèmes de vote/évaluation en ligne contre les attaques Sybil. SumUp exploite le réseau social des utilisateurs en affectant des capacités aux liens sociaux entre les utilisateurs, désignant la capacité de ces utilisateurs à émettre des évaluations. Cette capacité est d'autant plus élevée si les chances que l'utilisateur soit Sybil sont infimes. Par la suite, le réseau est considéré comme un graphe sur lequel est appliqué l'algorithme de ot maximal afin de collecter les évaluations des utilisateurs. SumUp définit un arc d'attaque comme un lien social que l'utilisateur Sybil a réussi à établir avec un utilisateur honnête. Ainsi, SumUp parvient à limiter le nombre de fausses évaluations émises par des utilisateurs Sybil au nombre d'arcs d'attaques. Cependant, la sélection aléatoire (e.g., la recherche en largeur à l'aveugle) des utilisateurs met en jeu le nombre et la qualité des évaluations considérées lors du calcul de la confiance d'une ressource sur le Web. A cet effet, nous préconisons la recherche heuristique pour guider la collecte des évaluations vers les utilisateurs les plus appropriées aussi bien en termes de quantité que de qualité. Cela devrait garantir une meilleure qualité de la valeur de la confiance calculée.

Dans les approches probabilistes existantes de gestion de la confiance (e.g., [86, 106]), les utilisateurs s'appuient soit sur leur propre expérience avec les ressources, soit sur les évaluations fournies par d'autres utilisateurs. Les évaluations fausses sont alors traitées par un mécanisme de filtrage approprié. Dans [86], Teacy et al. proposent TRAVOS un modèle de confiance pour les systèmes multi-agents. Dans ce modèle, un agent/pair gagne la confiance d'un autre au fil des interactions directes avec ce dernier. L'évaluation du résultat issu des interactions est binaire : réussite ou échec. Une fonction de densité modélise la probabilité d'occurrence d'une interaction réussie avec un pair. Le modèle se base sur les expériences des autres pairs pour calculer la confiance lorsque le pair n'a pas d'expérience directe. Le modèle utilise la crédibilité des agents pour filtrer les évaluations incorrectes provenant d'agents ayant des connaissances limitées ou un comportement malveillant. Dans [106], Zhou et al., proposent PowerTrust un système de confiance pour les réseaux pair-à-pair. Les noeuds évaluent les interactions avec les autres pairs et estiment la confiance localement en utilisant une technique d'apprentissage bayésien. La valeur globale de la

confiance est ensuite calculée en utilisant ces valeurs locales. Cette valeur est mise à jour périodiquement en utilisant l'algorithme de marche aléatoire (Look-ahead Random Walk). L'algorithme sélectionne les pairs avec des valeurs locales de confiance au-delà d'un seuil prédéfini pour mettre à jour la valeur globale de confiance d'un pair donné. Cet algorithme utilise une table de hachage distribuée et indexée par des valeurs locales de confiance pour classer les pairs.

Les approches probabilistes d'évaluation de la confiance (e.g., [86] et [106]) négligent l'interprétation des évaluations en présence de l'incertitude. Ceci impacte le calcul de la confiance et produit des résultats non-pertinents et inexacts. Le dilemme relatif aux évaluations fournies par les utilisateurs est le suivant : d'une part, elles peuvent réduire l'incertitude et d'autre part elles peuvent introduire d'autres types d'incertitude. En effet, chaque évaluation est vraie à un certain degré et fausse à un autre degré. Nous modélisons alors les évaluations de l'utilisateur par une base de données probabiliste, nous interprétons les évaluations en termes de mondes possibles et nous calculons la confiance d'une ressource sur le Web comme une évaluation de requêtes sur une base de données probabiliste (Section 3.5).

3.7 Conclusion

Dans le premier travail, nous avons proposé un nouveau modèle basé sur la crédibilité des utilisateurs et des données probabilistes pour le calcul de la confiance des ressources sur le Web. Dans ce modèle, nous nous sommes focalisés sur les utilisateurs stricts souvent exclus par les systèmes existants de gestion de la confiance. Nous avons discuté de l'utilisation du clustering flou pour déterminer la crédibilité d'un utilisateur. Pour le calcul de la confiance nous utilisons une approche basée sur l'évaluation de requêtes probabilistes. Nous avons développé un framework d'évaluation de la confiance fondé sur le modèle de confiance à base d'un modèle de crédibilité préalablement introduit. Finalement, nous avons mené plusieurs expérimentations montrant une amélioration significative de la qualité de la valeur de confiance. Les travaux à venir porteront sur l'incorporation de l'incertitude inhérente aux valeurs de la confiance due aux informations incomplètes/probabilistes sur les transactions avec les ressources sur le Web.

Dans le second travail, nous avons proposé une approche pour traiter les attaquants Sybil et les retours d'information biaisés lors du calcul de la confiance du service Web. Plus précisément, nous avons proposé un modèle de crédibilité floue de l'utilisateur et un algorithme basé sur le débit maximum pour limiter l'impact des attaquants Sybil. Notre approche vise à sélectionner le maximum non-Sybil et les utilisateurs crédibles. Nous avons également montré comment l'incertitude des évaluations associée à une rétroaction biaisée peut être modélisée comme une base de données probabiliste, et comment une confiance de service est calculée comme une évaluation de requête probabiliste. À titre de travaux futurs, nous prévoyons de poursuivre la question de la confiance en matière de service en considérant le modèle de réseau de crédit introduit dans la communauté du commerce électronique pour traiter les protocoles de confiance en l'absence d'entités de confiance centrales.

Dans le troisième travail, nous avons abordé la question de la confiance à ac-

corder aux services Web sous l'incertitude augmentant de l'absence d'évaluations cohérentes que les utilisateurs finaux fournissent avec le temps et l'incohérence de l'évaluation de la qualité de service. Dans l'approche déterministe, deux mesures de confiance sont proposées : les commentaires/évaluations des utilisateurs finaux et la réputation du service Web. La deuxième approche repose sur des bases de données probabilistes qui découlent de la théorie des probabilités couplé avec la sémantique des mondes possibles. Notre base de données probabiliste est structurée autour du modèle d'incertitude indépendant du tuple. La confiance est évaluée par en utilisant des requêtes spécifiques appliquées à la base de données probabiliste. Une mise en oeuvre des approches proposées a résulté en un framework de gestion de la confiance. Enfin, plusieurs expériences ont été menées pour évaluer l'impact du modèle de crédibilité sur confiance en la qualité et de comparer les résultats de confiance obtenus avec déterministe contre approches probabilistes. Les expériences ont démontré que la qualité de la confiance améliore en utilisant le modèle de crédibilité. Des résultats encore plus stables sont obtenus en utilisant des bases de données probabilistes. Pour un travail futur, nous explorerons possibilité d'incorporer plusieurs modèles de crédibilité dans le modèle de confiance en utilisant modèle d'incertitude indépendant du bloc pour structurer notre base de données probabiliste.

Chapitre 4

Vers des systèmes d'Entreprise 2.0 durables

4.1 Introduction

Pour réaliser leurs missions et atteindre leurs objectifs, les organisations (e.g., entreprises) conçoivent, développent et déploient des processus métier (BPs) (*aka* savoir-faire). Les BPs définissent quoi faire suite à l'occurrence d'événements internes et externes. Ces organisations sont d'autant plus désireuses de suivre le cours d'exécution des BPs pour évaluer leur efficacité et si nécessaire, réorganiser leur logique. Les défis d'aujourd'hui (e.g., politique, économique et sociétal) forcent aussi les organisations à être créatives et/ou innovantes en utilisant diverses technologies telles que le Web 2.0 (alias Web social) [12]. Par exemple, l'usage du crowdsourcing permettrait de façonner des pratiques commerciales en puisant dans l'expertise externe à l'organisation.

Dans [39], nous définissons une méthodologie de conception des BPs d'un point de vue social (Section 4.2). Elle consiste à capturer les relations entre composants du BP (i.e., tâche, personne, et machine). La tâche (i.e., unité de travail) est associée aux exigences alors que la personne comme la machine (i.e., exécuteurs de tâches) sont associés à des capacités permettant de satisfaire ces exigences. Des exemples de relations sociales sont l'échange entre les tâches, la délégation entre les personnes, la coopération entre les machines. Ces relations ont permis d'établir 3 catégories de réseaux pour extraire des détails spécifiques : i) réseau configuration de tâches (e.g., niveau de satisfaction des exigences d'une tâche sachant les capacités des exécuteurs disponibles) ; ii) réseau social de personnes (e.g., niveau de fiabilité d'une machine lors de l'attribution de tâches critiques) ; et iii) réseau support de machines (e.g., niveau d'engagement d'une personne pour aider les autres à exécuter leurs tâches). En fait, les réseaux capturent les interactions passées entre tâches (t2t), entre personnes (p2p) et entre machines (m2m).

Malgré l'assurance d'une conception fiable des BPs (e.g., [91]), il n'y a aucune garantie pour une exécution réussie du BP. Une raison est le manque de ressources car elles ne durent pas toujours éternellement et ne sont pas illimitées et/ou partageables. Dans [53], nous mettons en avant le besoin des tâches et des personnes/machines de

coordonner leur manière de consommer/utiliser les ressources disponibles. Cette coordination (i.e., allocation de ressources) permet d'éviter les conflits entre tâches, entre personnes, et entre machines et par conséquent de réduire les délais. La plupart des travaux existants sur l'allocation de ressources (e.g., [36]) n'ont pas accordé suffisamment d'attention aux dépendances entre tâches, ni entre personnes/machines en termes de production et consommation/utilisation de ressources. L'idée est alors de définir une coordination dite sociale basée sur les réseaux susmentionnés pour recommander des actions correctives en réponse à des modèles spécifiques de conflit (Section 4.3).

En dépit de l'enthousiasme engendré par les applications Web 2.0, leur pertinence sur le lieu de travail reste un sujet de questionnement pour de nombreuses entreprises. En effet, l'accès aux applications Web 2.0 par les employés reste avantageux pour l'entreprise néanmoins les risques d'un mauvais usage ne sont pas à négliger. Pour cela, nous nous sommes intéressés aux restrictions à mettre en place pour contrôler l'usage de ces applications [29]. Ces restrictions sont appliquées à la structure des actions sociales en termes de propriétés. Par souci de conformité, le suivi de ces restrictions est un must et est donc également discuté (Section 4.4).

4.2 Conception sociale de BP

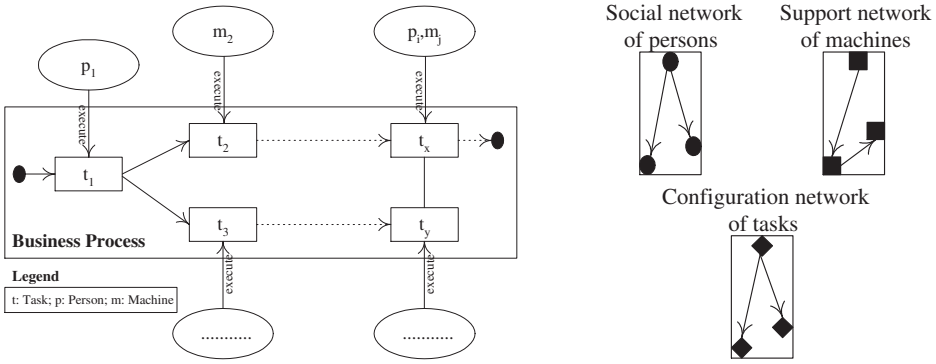
Cette section présente notre vision d'une conception sociale de BP en utilisant des réseaux d'entreprise dédiés.

4.2.1 Principes

Notre méthodologie de conception sociale favorise les liens parmi et entre composants d'un BP (i.e., tâche (t), personne (p), et machine (m)). Elle consiste en 3 étapes : i) identifier les relations sociales entre t, entre exécuteurs (i.e., p et m), et entre t et p/m (Sections 4.2.2) ; ii) développer une catégorisation de réseaux d'entreprise capturant ces relations et ce en lien avec les caractéristiques de chaque composant (Section 4.2.3) ; et iii) exploitation de ces réseaux en terme de valeur-ajoutée (Section 4.2.4). La figure 4.1 montre les éléments de conception de BPs en termes de composants et de catégories de réseaux.

Un BP simple est composé de y tâches ($t_1 \cdots t_y$). Les t_i sont généralement reliées entre elles par des dépendances de données (i.e., entrées/sorties). Elles peuvent être soit manuelles (e.g., p_1), automatisées (e.g., m_2), ou semi-automatisées (e.g., $p_i\&m_j$), et avoir n exécuteurs. Les relations entre t, entre p, et entre m sont capturées et utilisées pour construire des réseaux dédiés. Ces derniers appartiennent aux 3 catégories (i.e., configuration, support, et social), chacune spécialisée en 3 types spécifiques (Section 4.2.3).

Afin d'identifier les relations sociales, il nous semble approprié de caractériser t/p&m en termes exigences/capacités. Les exigences imposent des restrictions dans la sélection des exécuteurs (i.e., p et/ou m) (e.g., type d'exécution de t (manuelle ou automatique) et niveau d'expertise/de fiabilité nécessaire pour p/m). En plus des exigences, nous considérons t comme par défaut autonome/auto-suffisante (i.e. aucun traitement supplémentaire pour les sorties). Par exemple, la tâche de compres-



Composants du BP — Catégories de réseaux d'entreprise

FIGURE 4.1 – Eléments de conception sociale de BP en entreprise

sion/chiffrement de données nécessite une tâche de décompression/décryptage. L'attribution des t aux exécuteurs consistant en une mise en correspondance exigences-capacités ne fait pas partie de notre champ d'investigation.

4.2.2 Identification des relations

Les relations possibles entre t/p/m sont identifiées selon 2 perspectives : exécution et sociale.

- **Relations entre tâches**. D'une part, les relations possibles d'exécution (i.e., dépendances) entre t_i et t_j sont bien établies (e.g., pré-requis, parallèle, et pré-requis parallèle) [45]. D'autre part, les relations sociales sont définies comme suit :
  - collaboration(t_i,t_j) fait référence à la non-autonomie de t_i nécessitant t_j pour fournir des résultats utilisables (e.g., compresser et décompresser).
  - échange(t_i,t_j) réfère à la production de sorties similaires avec des entrées similaires et au non-chevauchement de leurs exigences. Ceci permet d'éviter une situation de blocage où t_i est échangée avec t_j ayant des exigences communes et ainsi ne pouvant pas être satisfaites.
- **Relations entre machines.** D'une part, les 2 relations possibles d'exécution entre m_i et m_j sont extraites du travail de Decker [22] : i) enable(m_i,m_j)- m_i et m_j sont assignées simultanément à une tâche commune (t) et m_i produit des sorties (internes) permettant à m_j de "plier" l'exécution de t, et ii) inhibe(m_i,m_j)- m_i et m_j sont assignées simultanément à une tâche commune (t) et m_j ne commence à exécuter sa part de tâche seulement après l'écoulement d'un certain temps de l'achèvement de celle exécutée par m_i. D'autre part, les relations sociales sont identifiées comme suit :
  - sauvegarde(m_i,m_j) se rapporte soit à des capacités similaires de m_i et m_j ou bien à l'existence d'une relation de subsomption entre leurs capacités en terme de fiabilité.
  - coopération(m_i,m_j) fait référence à sauvegarde(m_i,m_j) et à la nécessité de combiner simultanée leurs capacités respectives pour répondre aux exigences d'une tâche.

- partenariat(m_i,m_j) réfère à des capacités complémentaires de m_i et m_j et à la nécessité de combiner leurs capacités pour répondre aux exigences d'une tâche.

- **Relations entre personnes.** D'une part, les relations d'exécution entre p_i et p_j correspondent à celles définies pour les machines. D'autre part, 3 relations sociales sont identifiées comme suit :
  - substitution(p_i,p_j) se rapporte soit à des capacités similaires de p_i et p_j ou bien l'existence d'une relation de subsomption entre leurs capacités en terme d'expertise.
  - délégation(p_i,p_j) fait référence à substitution(p_i,p_j) et à un transfert de l'affectation par p_i à p_j.
  - référence(p_i,p_j) réfère à une recommandation de p_i par p_j en fonction des capacités de p_i permettant de répondre aux exigences d'une tâche.

Le tableau 4.1 résume les relations sociales entre t, entre m, et entre p, ainsi que des pré-conditions (i.e., quand établir la relation), conditions (i.e., quand utiliser la relation) et post-conditions (i.e., quand se défaire de la relation). Les relations avec/sans un astérisque font référence à une exploitation au moment de la conception/exécution du BP.

Tableau 4.1 – Récapitulatif des relations sociales

Relations between	Social Relation types	Pre-Conditions	Conditions	Post-Conditions
Tasks	Collaboration*	Task not self-contained	Output processing required	Task becomes self-contained
	Interchange	Produce similar outputs in receipt of similar inputs	Lack of appropriate executor	Produce different outputs in receipt of similar inputs
Machines	Backup	Have similar capacities	Machine failure	Changes in capacities
	Cooperation	Engage in backup	combined similar capacities required	Release from backup
	Partnership*	Capacities complementing each other	Combined separate capacities required	Changes in capacities
Persons	Substitution	Have similar capacities	Person unavailability	Changes in capacities
	Delegation	Engage in substitution	Capacities available elsewhere	Release from substitution
	Referral	Be aware of other's capacities	Other's capacities required	Changes in capacities

4.2.3 Catégorisation des réseaux

Les réseaux sont constitués de noeuds et d'arêtes représentant, respectivement, les composants du BP et les relations entre eux. A titre illustratif, seuls les réseaux de tâches et de machines sont formalisés.

Tâche-Perspective exécution. Une tâche (t_i) est une unité de travail concrète agissant sur l'environnement et caractérisée par des exigences (Section 4.2.1). t_i est définie comme un 5-tuple :

$$< \mathcal{E}\text{-Req}_{t_i}, \mathcal{E}\text{-PreCond}_{t_i}, \mathcal{E}\text{-E/S}_{t_i}, \mathcal{E}\text{-Cond}_{t_i}, \mathcal{E}\text{-PostCond}_{t_i} >$$

où $\mathcal{E}$-$\texttt{Req}_{t_i}$ sont les exigences de t_i, $\mathcal{E}$-$\texttt{PreCond}_{t_i}$ est un ensemble d'évènements autorisant une possible affectation de t_i à un exécuteur (i.e., $\mathcal{E}$-$\texttt{Req}_{t_i}$ sont satisfaits), $\mathcal{E}$-$\texttt{E/S}_{t_i}$ représente les données dont t_i a besoin pour s'exécuter et les résultats retournés par t_i après exécution, $\mathcal{E}$-$\texttt{Cond}_{t_i}$ est un ensemble de conditions à satisfaire avant de commencer l'exécution de t_i, et $\mathcal{E}$-$\texttt{PostCond}_{t_i}$ est un ensemble d'évènements à vérifier avant de détacher t_i de l'exécuteur et confirmer une exécution réussie de t_i. En cas d'échec, t_i est détachée de l'exécuteur seulement après la prise de mesures correctives.

Tâche-Perspective sociale. Une tâche (t_i) est une unité de travail abstraite s'engageant avec une autre tâche (t_j) dans une relation sociale. t_i est définie comme un 4-tuple :

$$< \mathcal{S}\text{-Relation}_{(t_i,t_j)}, \mathcal{S}\text{-PreCond}_{(t_i,t_j)}, \mathcal{S}\text{-Cond}_{(t_i,t_j)}, \mathcal{S}\text{-PostCond}_{(t_i,t_j)} >$$

où $\mathcal{S}$-$\texttt{Relation}_{(t_i,t_j)}$ est une relation sociale entre t_i et t_j, $\mathcal{S}$-$\texttt{PreCond}_{(t_i,t_j)}$ est un ensemble d'évènements à vérifier pour connecter t_i à t_j (i.e., soit ($\mathcal{E}$-$\texttt{Output}_{t_i}$ et $\mathcal{E}$-$\texttt{S}_{t_j}$ est équivalente) ou ($\mathcal{E}$-$\texttt{S}_{t_i}$ nécessite un traitement supplémentaire de t_j car t_i n'est pas autonome)), $\mathcal{S}$-$\texttt{Cond}_{(t_i,t_j)}$ est un ensemble de conditions à satisfaire avant de déclencher l'utilisation de $\mathcal{S}$-$\texttt{Relation}_{(t_i,t_j)}$, et $\mathcal{S}$-$\texttt{PostCond}_{t_i,t_j}$ est un ensemble d'évènements à vérifier avant de déconnecter t_i de t_j (i.e., soit (($\mathcal{E}$-$\texttt{E}_{t_i}$ et $\mathcal{E}$-$\texttt{E}_{t_j}$ sont équivalentes) et ($\mathcal{E}$-$\texttt{S}_{t_i}$ et $\mathcal{E}$-$\texttt{S}_{t_j}$ ne sont pas équivalentes)) ou ($\mathcal{E}$-$\texttt{S}_{t_i}$ ne nécessite pas de traitement supplémentaire par t_j car t_i est non-autonome)).

Relation-Perspective exécution. Une relation ($r_{(t_i,t_j)}$ établit une dépendance concrète entre t_i et t_j mise en œuvre au moment de l'exécution. $r_{(t_i,t_j)}$ est définie comme un triplet :

$$< t_i, t_j, \mathcal{E}\text{-Type} >, \quad \mathcal{E}\text{-Type} \in \{\text{prerequis}, \text{parallel}, \text{prerequis-parallel}\}$$

où prérequis (i.e., $\mathcal{E}$-$\texttt{PostCond}_{t_i}$ est valide et $\mathcal{E}$-$\texttt{PostCond}_{t_i} \subseteq \mathcal{E}$-$\texttt{PreCond}_{t_j}$), parallèle prérequis (i.e., $\mathcal{E}$-$\texttt{PostCond}_{t_i}$ est valide, $\mathcal{E}$-$\texttt{PostCond}_{t_i} \subseteq \mathcal{E}$-$\texttt{PreCond}_{t_j}$, et $\mathcal{E}$-$\texttt{S}_{t_i} \cap \mathcal{E}$-$\texttt{E}_{t_j} \in \varnothing$), et parallèle (i.e., $\mathcal{E}$-$\texttt{PreCond}_{t_i}$ et $\mathcal{E}$-$\texttt{PreCond}_{t_j}$ sont valides, $\mathcal{E}$-$\texttt{Pre/PostCond}_{t_i} \cap \mathcal{E}$-$\texttt{Post/PreCond}_{t_j} = \varnothing$, $\mathcal{E}$-$\texttt{S}_{t_i} \cap \mathcal{E}$-$\texttt{E}_{t_j} \neq \varnothing$, et $\mathcal{E}$-$\texttt{S}_{t_j} \cap \mathcal{E}$-$\texttt{E}_{t_i} \neq \varnothing$).

Relation-Perspective sociale. Une relation ($r_{(t_i,t_j)}$) correspondant à $\mathcal{S}$-$\texttt{Relation}_{(t_i,t_j)}$ établit un lien entre t_i et t_j indépendamment des relations d'exécution pouvant exister entre elles. $r_{(t_i,t_j)}$ est définie comme un triplet :

$$< t_i, t_j, \mathcal{S}\text{-Type} >, \quad \mathcal{S}\text{-Type} \in \{\text{collaboration}, \text{échange}\}$$

Evaluation du poids de $\mathsf{r}_{(\mathsf{t_i},\mathsf{t_j})}$. Dans [39], nous proposons des formules de poids (w) par type de relation selon les perspectives exécution et sociale. A titre illustratif, seul 1 type donné est considérée par perspective :

Relation prérequis. La formule 4.1 évalue le poids d'une arête indiquant le niveau d'intersection entre $\mathcal{E}$-$\mathtt{PostCond}_{\mathsf{t_i}}$ et $\mathcal{E}$-$\mathtt{PreCond}_{\mathsf{t_j}}$ à satisfaire.

$$w^{\mathcal{E}}_{PreReq(\mathsf{t_i},\mathsf{t_j})} = \frac{|\mathcal{E}\text{-}Post\text{-}Condition_{\mathsf{t_i}} \cap \mathcal{E}\text{-}Pre\text{-}Condition_{\mathsf{t_j}}|}{|\mathcal{E}\text{-}Pre\text{-}Condition_{\mathsf{t_j}}|} \tag{4.1}$$

Relation d'échange. La formule 4.2 évalue le poids d'une arête en termes d'échanges antérieurs.

$$w^{\mathcal{S}}_{interchange_{(\mathsf{t_i},\mathsf{t_j})}} = \frac{|interchangeSuc_{(\mathsf{t_i},\mathsf{t_j})}|}{|interchange_{(\mathsf{t_i},\mathsf{t_j})}| * |failure_{\mathsf{t_i}}|} \tag{4.2}$$

où $|interchangeSuc_{int_{(\mathsf{t_i},\mathsf{t_j})}}|$ est le nombre de fois où $\mathsf{t_i}$ a été échangée avec $\mathsf{t_j}$ avec succès (i.e., exécuteur trouvé pour $\mathsf{t_j}$), $|interchange_{(\mathsf{t_i},\mathsf{t_j})}|$ est le nombre de fois où $\mathsf{t_i}$ a été échangée avec $\mathsf{t_j}$, et $|failure_{\mathsf{t_i}}|$ est le nombre de fois où $\mathsf{t_i}$ n'a pas été exécuté faute d'exécuteurs.

Machine-Perspective exécution. Une machine ($\mathsf{m_i}$) est une unité concrète de traitement recevant des tâches selon ses capacités (Section 4.2.1). $\mathsf{m_i}$ est définie comme un 5-tuple :

$$< \mathcal{E}\text{-}\mathsf{Cap}_{\mathsf{m_i}}, \mathcal{E}\text{-}\mathsf{PreCond}_{\mathsf{m_i}}, \mathcal{E}\text{-}\mathsf{E/S}_{\mathsf{m_i}}, \mathcal{E}\text{-}\mathsf{Cond}_{\mathsf{m_i}}, \mathcal{E}\text{-}\mathsf{PostCond}_{\mathsf{m_i}} >$$

où $\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_i}}$ représente l'ensemble des capacités de $\mathsf{m_i}$, $\mathcal{E}$-$\mathtt{PreCond}_{\mathsf{m_i}}$ est l'ensemble d'évènements à vérifier pour accepter la programmation des tâches à exécuter par $\mathsf{m_i}$ (i.e., $\mathsf{m_i}$ satisfait aux exigences), $\mathcal{E}$-$\mathtt{E}_{\mathsf{m_i}}$ et $\mathcal{E}$-$\mathtt{S}_{\mathsf{m_i}}$ font référence aux tâches ordonnancées par $\mathsf{m_i}$ et aux résultats produits par $\mathsf{m_i}$, $\mathcal{E}$-$\mathtt{Cond}_{\mathsf{m_i}}$ est un ensemble d'évènements à vérifier avant de commencer l'exécution des tâches par $\mathsf{m_i}$, et $\mathcal{E}$-$\mathtt{PostCond}_{\mathsf{m_i}}$ est un ensemble d'évènements à vérifier avant de pouvoir détacher une tâche de $\mathsf{m_i}$ tels que confirmer le statut d'exécution de la tâche et mettre à jour $\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_i}}$.

Machine-Perspective sociale. Une machine ($\mathsf{m_i}$) est une unité abstraite de traitement s'engageant avec une autre machine ($\mathsf{m_j}$) dans une relation sociale. $\mathsf{m_i}$ est définie comme un 4-tuple :

$$< \mathcal{S}\text{-}\mathsf{Relation}_{\mathsf{m_i}}, \mathcal{S}\text{-}\mathsf{PreCond}_{\mathsf{m_i}}, \mathcal{S}\text{-}\mathsf{Cond}_{\mathsf{m_i}}, \mathcal{S}\text{-}\mathsf{PostCond}_{\mathsf{m_i}} >$$

où $\mathcal{S}$-$\mathtt{Relation}_{(\mathsf{m_i},\mathsf{m_j})}$ est une relation sociale entre $\mathsf{m_i}$ et $\mathsf{m_j}$, $\mathcal{S}$-$\mathtt{PreCond}_{\mathsf{m_i}}$ est un ensemble d'évènements à vérifier pour connecter $\mathsf{m_i}$ à $\mathsf{m_j}$ ensemble (i.e. soit $\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_i}}$ et $\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_j}}$ sont équivalents [1] ou bien ($\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_i}}$ et $\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_j}}$ sont requises simultanément)), $\mathcal{S}$-$\mathtt{Cond}_{(\mathsf{m_i},\mathsf{m_j}}$ est un ensemble de conditions à satisfaire pour déclencher l'utilisation de $\mathcal{S}$-$\mathtt{Relation}_{(\mathsf{m_i},\mathsf{m_j})}$ (i.e., échec de $\mathsf{m_i}$ ou insuffisance des $\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_i}}$), et $\mathcal{S}$-$\mathtt{PostCond}_{\mathsf{m_i}}$ est un ensemble d'évènements à vérifier

1. $\mathsf{m_i}$et$\mathsf{m_j}$ font le même travail.

pour déconnecter $\mathsf{m_i}$ de $\mathsf{m_j}$ (e.g., changement de $\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_i}}$ ou $\mathcal{E}$-$\mathtt{Cap}_{\mathsf{m_j}}$).

Relation-Perspective exécution. Une relation ($\mathsf{r}_{(\mathsf{m_i},\mathsf{m_j})}$ établit un lien entre $\mathsf{m_i}$ et $\mathsf{m_j}$ mis en œuvre au moment de l'exécution (Section 4.2.2). $\mathsf{r}_{(\mathsf{m_i},\mathsf{m_j})}$ est définie comme un triplet :

$$< \mathsf{m_i}, \mathsf{m_j}, \mathcal{E}\text{-Type} >, \quad \mathcal{E}\text{-Type} \in \{\mathsf{enables}, \mathsf{inhibit}\}$$

Relation-Perspective sociale. Une relation ($\mathsf{r}_{(\mathsf{m_i},\mathsf{m_j})}$ établit un lien entre deux machines $\mathsf{m_i}$ et $\mathsf{m_j}$ correspondant à $\mathcal{S}$-$\mathtt{Relation}_{(\mathsf{m_i},\mathsf{m_j})}$. $\mathsf{r}_{(\mathsf{m_i},\mathsf{m_j})}$ est définie comme un triplet :

$$< \mathsf{m_i},\ \mathsf{m_j},\ \mathcal{S}\text{-Type} >, \quad \mathcal{S}\text{-Type} \in \{\mathsf{sauvegarde}, \mathsf{cooperation}, \mathsf{partenariat}\}$$

Evaluation du poids de $\mathsf{r}_{(\mathsf{t_i},\mathsf{t_j})}$. Nous proposons des formules de poids (w) par type de relation selon les perspectives exécution et sociale. A titre illustratif, seul 1 type donné est considérée par perspective :

Relation d'activation. La formule 4.3 évalue le poids d'une arête connectant $\mathsf{m_i}$ à $\mathsf{m_j}$.

$$w^{\mathcal{E}}_{Enables(\mathsf{m_i},\mathsf{m_j})} = \frac{\sum_{t\in(\mathcal{T}_{\mathsf{m_i}}\cap\mathcal{T}_{\mathsf{m_j}})} |ReqOutput_{t,(\mathsf{m_i},\mathsf{m_j})}|}{\sum_{t'\in\mathcal{T}_{\mathsf{m_j}}} |ReqOutput_{t',\mathsf{m_j}}|} \tag{4.3}$$

où $\mathcal{T}_m$ est l'ensemble des tâches affectées à m, $|ReqOutput_{t,(\mathsf{m_i},\mathsf{m_j})}|$ est le nombre de fois où $\mathsf{m_i}$ produit des résultats en interne en faveur de $\mathsf{m_j}$ pour une affectation à la même tâche t, et $|ReqOutput_{t',\mathsf{m_j}}|$ est le nombre de fois où $\mathsf{m_j}$ a nécessité des sorties d'autres machines pour l'affectation de $\mathsf{t'}$.

Relation de coopération. La formule 4.4 évalue le poids d'une arête connectant $\mathsf{m_i}$ à $\mathsf{m_j}$.

$$w^{\mathcal{S}}_{cooperation_{(\mathsf{m_i},\mathsf{m_j})}} = \frac{|supportSuc_{\mathcal{T}_{\mathsf{m_i}},(\mathsf{m_i},\mathsf{m_j})}|}{|support_{\mathcal{T}_{\mathsf{m_i}},(\mathsf{m_i},\mathsf{m_j})}| * |addCapacities_{\mathcal{T}_{\mathsf{m_i}},\mathsf{m_i}}|} \tag{4.4}$$

où $|supportSuc_{\mathcal{T}_{\mathsf{m_i}},(\mathsf{m_i},\mathsf{m_j})}|$ représente le nombre de tâches dans $\mathcal{T}_{\mathsf{m_i}}$ s'étant exécutées avec succès et ce avec le support de $\mathsf{m_j}$, $|support_{\mathcal{T}_{\mathsf{m_i}},(\mathsf{m_i},\mathsf{m_j})}|$ est le nombre total de tâches dans $\mathcal{T}_{\mathsf{m_i}}$ affectées à $\mathsf{m_i}$ et s'étant exécutées avec le support de $\mathsf{m_j}$, et $|addCapacities_{\mathcal{T}_{\mathsf{m_i}},\mathsf{m_i}}|$ est le nombre de tâches dans $\mathcal{T}_{\mathsf{m_i}}$ pour lesquelles $\mathsf{m_i}$ a nécessiter des capacités supplémentaires d'autres machines pour répondre aux exigences de ces tâches.

4.2.4 Valeur ajoutée des réseaux

Cette section analyse à quel moment les réseaux peuvent être utilisés et quels détails pertinents fournissent-ils aux concepteurs de BP (e.g., niveaux de criticité des tâches (T), de fiabilité des machines (M) et de centralité des personnes (P)).

T-Réseau d'échange. Il est utilisé lorsque les exigences d'une tâche ne peuvent pas être satisfaites au moment de l'exécution (e.g., manque d'exécuteurs possibles). Ainsi, une autre tâche similaire avec des exigences différentes est proposée à travers ce réseau. Ces exigences sont supposées cohérentes (e.g., niveau de sécurité plus élevé ou équivalent) avec celles ne pouvant pas être satisfaites. Selon la formule 4.2, comparer le poids d'une arête avec un certain seuil ($\mathsf{s}_{\mathsf{exchange}}$) permet de dire si les exigences d'une tâche sont faciles à satisfaire par rapport aux capacités des exécuteurs existants. Ce réseau aide ainsi le concepteur à examiner les exigences et/ou capacités et ainsi établir le niveau de criticité de la tâche.
T-Réseau de collaboration. Il est utilisé lorsqu'une tâche ne s'avère pas autonome et nécessite un traitement supplémentaire de ses sorties par une autre tâche recommandée par ce réseau. Comparer le poids d'une arête avec un certain seuil ($\mathsf{s}_{\mathsf{col}}$) permet de dire si les tâches sont faiblement ou fortement couplées en terme de fréquence de collaboration. Cela pourrait aider à résoudre les problèmes d'appariement sémantique entre ces tâches.
M-Réseau de sauvegarde. Il est utilisé pour recommander une machine pouvant terminer une tâche initialement affectée à une machine n'étant plus disponible (e.g., respect impératif de la durée maximale d'exécution). Comparer le poids d'une arête avec un certain seuil ($\mathsf{s}_{\mathsf{backup}}$) permet d'indiquer le niveau de fiabilité d'une machine et sa facilité de remplacement par rapport aux capacités des machines disponibles. Cela pourrait aider à éviter une machine pour la prochaine sélection et améliorer la maintenance.
M-Réseau de coopération. Il est utilisé lorsque les capacités d'une machine s'avèrent insuffisantes pour répondre aux besoins d'une tâche (e.g., haute priorité d'une tâche nécessitant une rapide prise en charge). Selon la formula 4.4, comparer le poids d'une arête avec un certain seuil ($\mathsf{s}_{\mathsf{coop}}$) permet de déterminer le niveau de préparation d'une machine en termes de capacités à recevoir des tâches à exécuter.
M-Réseau de partenariat. Il est utilisé lorsque les capacités des machines prises séparément s'avèrent insuffisantes pour répondre aux besoins d'une tâche nécessitant la combinaison de leurs capacités (e.g., nature du BP comportant des tâches parallèles). Comparer le poids d'une arête avec un certain seuil ($\mathsf{s}_{\mathsf{part}}$) permet de dire si les machines sont des partenaires faibles ou forts. Cela pourrait aider à sélectionner les machines en concurrence.
P-Réseau de substitution. Il est utilisé lorsqu'une personne exécutant une tâche s'avère indisponible pour une autre tâche censée se terminer (e.g., congé de maladie). Ainsi, une autre personne est proposée à travers ce réseau pour accomplir cette tâche. Le poids d'une arête permet d'établir le niveau d'engagement d'une personne pour aider les autres à accomplir leurs tâches et *vice versa*.
P-Réseau de délégation. Il est utilisé lorsque les capacités d'une personne s'avèrent insuffisantes pour répondre aux exigences d'une tâche (e.g., retards dans l'exécution d'autres tâches). Ainsi, une autre personne est proposée à travers ce réseau pour accomplir cette tâche. Le poids d'une arête permet d'établir le niveau d'engagement d'une personne à soutenir les autres à accomplir leurs tâches et *vice versa*.
P-Réseau de référence. Il est utilisé lorsqu'une personne suggère à d'autres de compléter une tâche initialement affectée en raison de ses capacités à répondre aux exigences de cette tâche. Comparer le poids d'une arête avec un certain seuil ($\mathsf{s}_{\mathsf{referral}}$)

permet d'établir le niveau d'adéquation des recommandations par le réseau.

4.3 Coordination sociale de BP

Cette section présente notre approche basée sur les réseaux d'entreprise (Section 4.2) pour la résolution de conflits sur les ressources lors de l'exécution du BP.

4.3.1 Fondements

Notre approche de coordination sociale consiste en 4 étapes (Figure 4.2). L'étape 1 catégorise les ressources requises pour l'exécution des BP/tâches (Section 4.3.2). L'étape 2 définit la manière dont t/m/p d'un BP "se lient" aux ressources au moment de l'exécution (Section 4.3.3). L'étape 3 catégorise les conflits sur les ressources survenant entre t, entre t, et entre p (Section 4.3.4). Enfin, l'étape 4 analyse l'adéquation de certains réseaux de t/m/p pour résoudre certains conflits (Section 4.3.5).

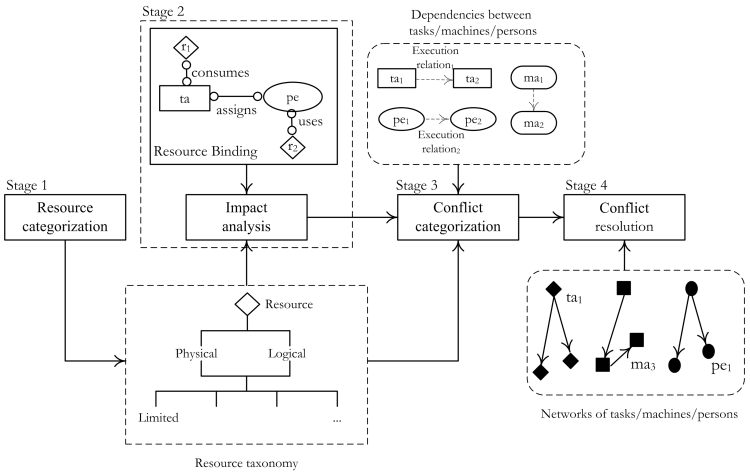


FIGURE 4.2 – Aperçu global d'une coordination sociale

Par la suite, nous adoptons les notations suivantes :

- T : ensemble fini de tâches ($t_i \in T$),
- M : ensemble fini de machines ($m_j \in M$),
- P : ensemble fini de personnes ($p_k \in P$),
- R : un ensemble de ressources ($r_q \in R$) disponibles pour T, M, et P.

Au moment de l'exécution, t_i consomme des r_q, alors que m_j et p_k utilisent des r_q pour réaliser les t_i du BP.

4.3.2 Catégorisation des ressources

Afin de proprement identifier les conflits sur r_q, il nous semble nécessaire de catégoriser r_q en : i) logique (i.e., pas de diminution du niveau de fiabilité et/ou de disponibilité des r_q après l'utilisation/consommation), et ii) physique, (i.e., diminution du niveau de fiabilité et/ou de disponibilité des r_q après l'utilisation/consommation). Cette diminution peut nécessiter un réapprovisionnement en r_q/remplacement des r_q [2]. Les ressources sont également décrites par des propriétés comme suit :

- limité (l) : consommation/utilisation de r_q est quantifiable ou bien cessation d'existence de r_q conformément à son cycle d'utilisation/consommation ou dû des contraintes temporelles,
- limité mais renouvelable (lr) : après un certain seuil de consommation/utilisation de r_q ou r_q sujette à des contraintes temporelles, un renouvellement est possible, et
- non-partageable (ns) : consommation/utilisation simultanée de r_q nécessitant un ordonnancement.

Par défaut, une ressource est définie comme illimitée (ul) et/ou partageable (s). La table 4.2 donne des exemples de ressources par catégorie et propriété.

Tableau 4.2 – Catégories de ressources et exemples

Resource		Tasks	Examples of resources
Category	Property		
Logical	Unlimited (ul)	Put together medical team for surgery	Patient medical record (read mode)
	Limited (l)	Prepare on-call medical shifts	Doctors' weekly schedules (valid for one week only)
	Limited but renewable (lr)	Prepare interns' access rights to labs	File of interns (internship possible extension)
	Shareable (s)	Prepare patient for surgery	Patient lab results
	Non-shareable (ns)	Report on surgery outcome	Patient medical record (update mode)
Physical	Unlimited (ul)	Not applicable	Not applicable
	Limited (l)	Prepare patient for surgery	Anesthetic injection
	Limited but renewable (lr)	Carry out surgery on patient	Surgical staple cartridge
	Shareable (s)	Carry out surgery on patient	Oxygen tank
	Non-shareable (ns)	Check patient's vitals	Blood pressure tensiometer

Le cycle de consommation/utilisation (cc/uc) d'une ressource r_q est représenté comme un diagramme de transition d'état d'une ressource (Figure 4.3) indépendamment de la catégorie et tenant compte des propriétés (Tableau 4.2). cc/uc ont pour objectif d'imposer des contraintes de consommation/utilisation sur les ressources. Par la suite, la discussion sera sur cc. D'une part, les états (s_i) sont : not made available (r_q n'est ni créée ni produite), made available (créée ou produite), not consumed (en attente d'être liée à une tâche), locked (réservée à une tâche en préparation

2. Le remplacement peut être le résultat d'une dégradation.

de sa consommation), unlocked (libérée par une tâche après consommation), consumed (liée à une tâche en cours d'exécution), withdrawn (abandon de $\mathsf{r_q}$ après l'avoir détachée de toutes les tâches), et done (mise à jour selon la propriété de $\mathsf{r_q}$). D'autre part, les transitions d'état de $\mathsf{r_q}$ ($\mathsf{trans_j}$) sont : start, waiting to be bound, consumption approval, consumption update, lock, release, consumption reject, consumption completion, renewable approval, et no longer useful.

A titre d'exemple, nous listons 2 séquences d'états (s)-transitions (s) représentant le cycle de consommation de $\mathsf{r_q}$, chacune pour une propriété donnée et notée $r(cc_{\mathsf{propriété}}) = s_i \overset{trans_i}{\longrightarrow} s_{i+1} \overset{trans_{i+1}}{\longrightarrow} s_{i+2} \dots s_{j-1} \overset{trans_{j-1}}{\longrightarrow} s_j$. Une liste plus complète est proposée dans [54].

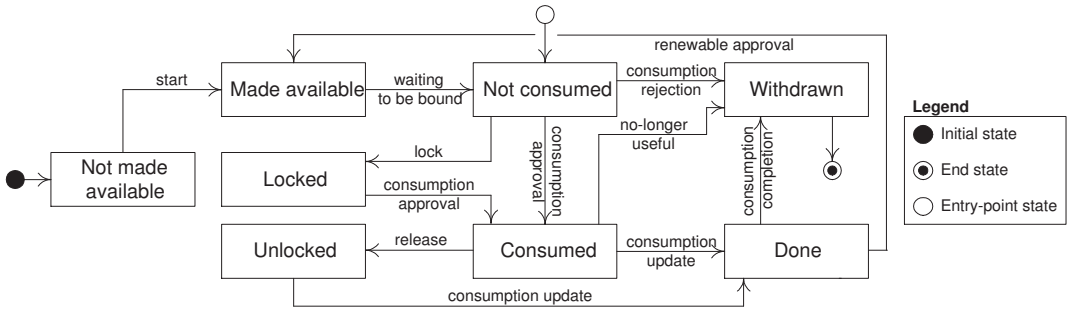


FIGURE 4.3 – Daigramme d'états d'une ressource

- $r(cc_{\mathsf{l}})$: not made available $\overset{start}{\longrightarrow}$ made available $\overset{waiting\ to\ be\ bound}{\longrightarrow}$ not consumed $\overset{consumption\ approval}{\longrightarrow}$ consumed $\overset{consumption\ update}{\longrightarrow}$ done $\overset{consumption\ completion}{\longrightarrow}$ withdrawn. La transition de done à withdrawn puis à end-of-state protège $\mathsf{r_q}$ (e.g., liste de patients présentant des maladies contagieuses) de toute nouvelle tentative de consommation par des tâches (e.g., soumettre une liste de patients aux autorités sanitaires après détection d'une maladie) après un cycle de consommation. A done, les paramètres de $\mathsf{r_q}$ (e.g., niveau de précision) sont mis à jour et $\mathsf{r_q}$ est détachée (ou pas) des tâches.
- $r(cc_{\mathsf{lr}})$: not made available $\overset{start}{\longrightarrow}$ made available $\overset{waiting\ to\ be\ bound}{\longrightarrow}$ not consumed $\overset{consumption\ approval}{\longrightarrow}$ consumed $\overset{consumption\ update}{\longrightarrow}$ done $\overset{renewable\ approval}{\longrightarrow}$ made available. La transition de done à made available permet de régénérer $\mathsf{r_q}$ (e.g., fichier d'internes) pour un autre cycle de consommation (e.g., extension de stage).

4.3.3 Binding de ressources

Le binding des $\mathsf{t}/\mathsf{m}/\mathsf{p}$ aux ressources $\mathsf{r_q}$ et son impact sur $\mathsf{r_q}$.

- $\mathsf{consume}(\mathsf{t_i}, r_i)$: l'exécution de $\mathsf{t_i}$ (e.g., effectuer une opération sur le patient) nécessite de consommer r_i (e.g., cartouche d'agrafes chirurgicales comme ressource physique). Indépendamment du succès ou échec de cette exécution, son impact sur r_i diffère comme suit :

- Logique : (ul : pas d'impact), (l : withdrawn dans cc_l), (lr : pas d'impact), (s : no impact), et (ns : no impact).
- Physique : (ul : non applicable), (l : niveaux de disponibilité et/ou de fiabilité diminuent selon l'état done dans cc_l), (lr : niveaux de disponibilité et/ou de fiabilité diminuent avec la possibilité d'augmenter le niveau de disponibilité selon les états done et made available), et (s, ns : niveaux de disponibilité et/ou de fiabilité diminuent avec la dégradation de r_i).

- use($p/m_j, r_j$, consume(t_i, r_i)) : l'exécution de t_i (e.g., effectuer une opération sur le patient) par p/m_j (e.g., chirurgien) nécessite l'utilisation de r_j par r_j (e.g., agrafeuse chirurgicale). Cette exécution conduit également à consommer r_i. L'impact de cette exécution sur r_j est similaire à celui impact de la consommation d'une ressource par une tâche comme décrit ci-dessus.

Conformément au cycle de consommation (Figure 4.3), l'exécution de t par des p et/ou m peut produire d'autres ressources disponibles à d'autres p/m. De même, la consommation de ressources par t peut produire d'autres ressources disponibles pour d'autres t (mais pas pour p/m).

4.3.4 Catégorisation des conflits

Les conflits sur les ressources r_q sont identifiés à la base des propriétés de r_q et ne font pas référence aux futurs exécuteurs. Il est important de distinguer le moment où une ressource est produite et le moment où une ressource est consommée, de sorte que les contraintes temporelles soient prises en compte (seules les propriétés de ressources générant des conflits sont discutées ci-dessous). Dans [54], une liste exhaustive des conflits entre t/p/m a été proposée. A titre d'exemple, nous considérons un seul type de conflits entre t ($\mathcal{T}$-Conflict$_1$) et adoptons la notation suivante :

1. $t_i/m_i/p_i \rightarrow r_i$ signifie qu'une ressource est disponible pour $t_i/m_i/p_i \nrightarrow r_i$;
2. $r_{i,j}$ signifie qu'une ressource produite par une certaine tâche t_i est transférée à une autre tâche t_j ;
3. $r_{i,\{j,k,\cdots\}}$ généralise $r_{i,j}$ mais cette fois la ressource est partagée entre plusieurs tâches.

$\mathcal{T}$-Conflict$_1$ apparaît quand une relation prerequisite entre t_i et t_j existe, consume(t_i, r_i) $\rightarrow$produit ($t_i, r_{i,j}$), et t_j a besoin de $r_{i,j}$ (i.e., $t_j \nrightarrow r_j$, non r_j est disponible pour t_j). Formellement, ((consume(t_i, r_i) $\rightarrow$produire ($t_i, r_{i,j}$))$\wedge$(consume($t_j, r_{i,j}$) $\rightarrow$ produit (t_j, r))). Les conflits possibles sur $r_{i,j}$ (et éventuellement $r_{i,k,\cdots\},j}$ et $r_{i,\{j,k,\cdots\}}$) sont comme suit :

Ressources logiques.

— l : 2 cas résultent de la relation prerequisite entre $t_{\{k,\cdots\}}$ (e.g., remplir les documents nécessaires) et t_j (e.g., diriger le patient vers le département approprié) sachant que la même relation existe entre t_i (e.g., vérifier les signes vitaux du patient) et t_j.

a) $r_{i,j}$ (e.g., rapport sur les niveaux vitaux) cesse d'exister (e.g., l'échantillon de sang n'est plus valide) avant que l'exécution de t_j ne commence ;

$\mathsf{t_j}$ attend $t_{\{k,\cdots\}}$ pour produire $r_{\{k,\cdots\},j}$ (e.g., approbation de la compagnie d'assurance) ; (au moins un) $t_{\{k,\cdots\}}$ est toujours en cours de réalisation (e.g., retard dans la réception de l'approbation de la compagnie d'assurance).

b) Un seul cycle de consommation de $r_{i,j}$ est autorisé (par type de propriété) mais il s'avère que plusieurs cycles de consommation sont nécessaires pour terminer l'exécution de $\mathsf{t_j}$ et la consommation de $r_{\{k,\cdots\},j}$ produites par $t_{\{k,\cdots\}}$.

- lr : le renouvellement de $r_{i,j}$ (e.g., fichier d'internes) n'est pas possible pour $\mathsf{t_j}$ (e.g., étendre les droits d'accès des stagiaires aux laboratoires) en raison de contraintes.
- ns :$r_{i,\{j,k,\cdots\}}$ (e.g., dossier médical du patient) doit être publié par $t(\{k,\cdots\}$ (e.g., inclure le rapport de chirurgie dans le dossier médical du patient) de sorte que $\mathsf{t_j}$ (e.g., préparer des visites de suivi) continue. Cela arrive quand une relation prerequisite entre t_i (e.g., effectuer une chirurgie) et $t_{\{j,k,\cdots\}}$.

Ressources physiques.

- l : niveaux de disponibilité/fiabilité de $r_{i,j}$ (e.g., injection anesthésique) ne sont pas suffisants/appropriés (e.g., dosage limité) pour l'exécution de $\mathsf{t_j}$;
- lr : le renouvellement de $r_{i,j}$ n'est pas possible pour $\mathsf{t_j}$ en raison de contraintes tel que le budget.
- s : $r_{i,\{j,k,\cdots\}}$ (e.g., réservoir d'oxygène) partagées entre $\mathsf{t_j}$ (e.g., effectuer une opération sur le patient) et $t_{\{k,\cdots\}}$ (e.g., prodiguer des soins post-opératoire) ne permet pas à $\mathsf{t_j}$ de terminer à temps en raison d'une diminution des niveaux de disponibilité/fiabilité de $r_{i,\{j,k,\cdots\}}$.

— ns : similaire à ns dans le cas de ressources logiques.

4.3.5 Stratégies de résolution de conflict

L'exécution d'une tâche peut être dirigée par des propriétés transactionnelles quand une tâche faillit ou réussit [46]. La figure 4.4 montre ces propriétés, à savoir, *pivot* (i.e., si terminée avec succès, ses effets d'exécution restent inchangés pour toujours et ne peuvent pas être annulés), *retriable* (i.e., sûre de se terminer avec succès après plusieurs activations finies), et *compensable* (i.e., ses effets d'exécution peuvent être sémantiquement défaits).

Le rôle des réseaux d'échange et de couplage dans la résolution de $\mathcal{T}$- Conflit$_1$ est discuté comme suit. A titre d'exemple, le cas a) de ressources logique avec la propriété limited est analysé.

$r_{i,j}$ cesse d'exister avant que l'exécution de $\mathsf{t_j}$ ne commence ; $\mathsf{t_j}$ attend que $t_{\{k,\cdots\}}$ produisent $r_{\{k,\cdots\},j}$; et au moins un des $\mathsf{t_k}$ est toujours en cours d'exécution ou a échoué.

Les états actuels des tâches et des ressources selon leurs cycles de vie et de consommation respectifs : state($\mathsf{t_i}$) sont : done ; state($r_{i,j}$) : withdrawn ; state ($\mathsf{t_j}$) : not-activated ; et state($\mathsf{t_k}$) : soit activated ou soit failed. L'analyse pour la propriété $\mathcal{T}$- Conflict$_1$/ limited est donnée dans la Table 4.3. L'objectif est de re-produire $r_{i,j}$

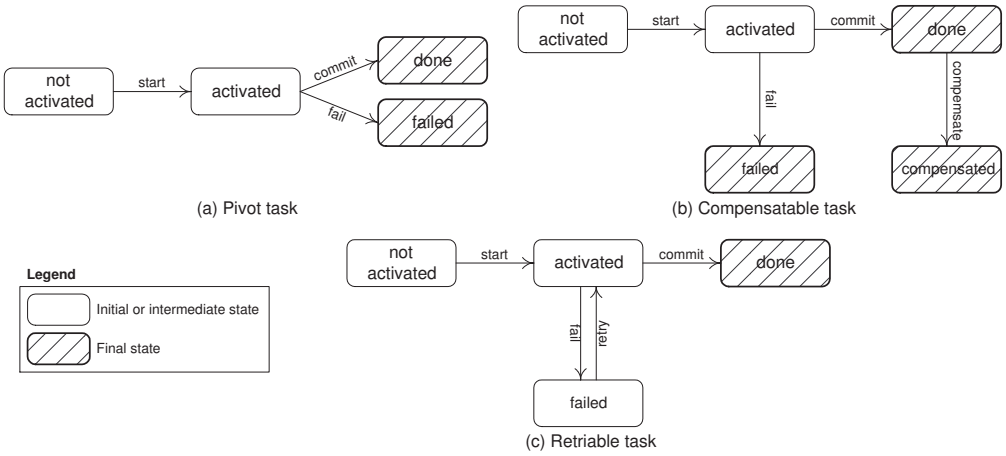


FIGURE 4.4 – Cycle de vie d'une tache par propriété transactionnelle [46]

(ou de produire $r_{i',j}$ avec $t_{i'}$ étant obtenu via le réseau d'échange de $\mathsf{t_i}$). A cause de l'échec de $\mathsf{t_k}$, $r_{k',j}$ est produite par $t_{k'}$ recommandé par le réseau d'échange de t_k.

Tableau 4.3 – Actions possibles de coordination pour $\mathcal{T}$-Conflict$_1$/limited/case a

Transactional properties		Coordination actions	Networks involved
$\mathsf{t_i}$	$\mathsf{t_k}$		
Null	Null	- re-perform $\mathsf{t_i}$ to re-produce $r_{i,j}$ - re-perform $\mathsf{t_k}$ to produce $r_{k,j}$	N/A
	Pivot	Deadlock	N/A
	Compensatable	Deadlock	N/A
	Retriable	- re-perform $\mathsf{t_i}$ to re-produce $r_{i,j}$ - replace $\mathsf{t_k}$ with $t_{k'}$ then perform$t_{k'}$ to produce $r_{k',j}$	Interchange($\mathsf{t_k}, t_{k'}$)
Compensatable	Null	- compensate $\mathsf{t_i}$; either re-perform $\mathsf{t_i}$ to re-produce $r_{i,j}$ or replace $\mathsf{t_i}$ with $t_{i'}$ then perform $t_{i'}$ to produce $r_{i',j}$ - either re-perform $\mathsf{t_k}$ to produce $r_{k,j}$ or replace $\mathsf{t_k}$with$t_{k'}$ then perform $t_{k'}$ to produce $r_{k',j}$	Interchange($\mathsf{t_i}, t_{i'}$) Interchange($\mathsf{t_k}, t_{k'}$)
	Pivot	Deadlock	N/A
	Compensatable	- compensate $\mathsf{t_i}$; either re-perform $\mathsf{t_i}$ to re-produce $r_{i,j}$ or replace $\mathsf{t_i}$ with $t_{i'}$ then perform$t_{i'}$to produce$r_{i',j}$ - replace $\mathsf{t_k}$ with $t_{k'}$ then perform $t_{k'}$to produce$r_{k',j}$	Interchange($\mathsf{t_i}, t_{i'}$) Interchange($\mathsf{t_k}, t_{k'}$)
	Retriable	- compensate $\mathsf{t_i}$; either re-perform $\mathsf{t_i}$ to re-produce $r_{i,j}$ or replace $\mathsf{t_i}$ with $t_{i'}$ then perform $t_{i'}$ to produce $r_{i',j}$ - re-perform $\mathsf{t_k}$ to produce $r_{k,j}$	Interchange($\mathsf{t_i}, t_{i'}$)

4.4 Restrictions d'usage des applications Web 2.0

Cette section présente notre approche de contrôle d'usage des applications Web 2.0 en entreprise.

4.4.1 Définition d'une action sociale

Les applications Web 2.0 permettent aux utilisateurs d'effectuer diverses actions telles que l'inscription à des comptes, la mise à jour de profils, la diffusion de messages à des amis et la rédaction de commentaires. Cependant, toutes ces actions ne sont pas qualifiées de «sociales». Nous considérons une action comme sociale quand il y a un motif clair et/ou l'intention d'un utilisateur d'atteindre des pairs (inconnus) (par exemple, demander de l'amitié) ou d'engager des pairs (inconnus) dans une production collaborative ou coopérative footnote Collaborative implique une implication active de l'utilisateur alors que coopérative implique une implication passive de l'utilisateur. du contenu (par exemple, co-auteur ou taguer un rapport technique).

Pour identifier des actions sociales représentatives à travers la variété des applications Web 2.0, nous avons consulté Facebook ™, Twitter ™, Wikipedia ™, Google services ™, entre autres les services offerts, les communautés ciblées, et à quelle(s) fin(s).

Conformément à notre définition de l'action sociale, nous regroupons les actions sociales en trois catégories représentatives (Table 4.3, [49]) : communication, sharing et enrichissement Il convient de mentionner qu'une action sociale peut appartenir à plus d'une catégorie à la fois. Pour des raisons de simplicité, ceci n'est pas considéré dans ce travail.

4.4.2 Propriétés des actions sociales

Puisque les actions sociales visent à soutenir la communication entre des parties (non connues) (individus et/ou entreprises), nous avons examiné comment d'autres communautés de recherche, par exemple, l'intelligence artificielle distribuée (DAI), examinent la communication. La communauté DAI met FIPA-ACL [3] En avant pour prendre en charge la communication entre les agents logiciels qui coopèrent pour résoudre les problèmes complexes [31]. Un message FIPA-ACL se compose de quatre propriétés : *performative*, *expéditeur*, *destinataire* et *contenu*. *Performative* désigne un acte communicatif spécifique à partir d'une liste de vingt-deux actes autorisés (par exemple, inform, request et ack) ; *sender* indique l'agent qui exécute le *performative* ; *receiver* indique le (s) destinataire (s) de l'agent ; et enfin, *content* dénote l'objet véhiculé par le *performative* entre *sender* et *receiver*. Par analogie avec FIPA-ACL, nous considérons trois propriétés (l'une s'applique uniquement à certaines actions) qui serviront de base à la définition de la structure des actions sociales et, également, à la surveillance de leur exécution pour assurer le respect des restrictions.

- `Stakeholders` : la propriété se réfère à ceux qui participent à une action sociale en termes de qui l'initie et qui réagit à elle. Cette propriété est obligatoire pour les actions sociales (par exemple, chat) qui nécessitent une présence "continue"

3. Fondation pour les agents physiques intelligents - Langage de communication des agents.

de toutes les parties prenantes lors de l'exécution de ces actions. En ce qui concerne FIPA-ACL, `stakeholders` correspond à *sender* et *receiver*.

- `Content` : la propriété fait référence à l'objet mis à disposition pour et/ou par les parties prenantes d'une action sociale. Cela pourrait être du texte, image, audio, etc. En ce qui concerne FIPA-ACL, `content` correspond à *content*.
- `Tool` : la propriété fait référence à une application Web 2.0 (par exemple, Facebook ™et Google Talk ™) qui rend une action sociale disponible pour exécution par le (s) acteur (s). En ce qui concerne FIPA-ACL, `tool` n'est pas pris en compte. Nous avons jugé nécessaire cette propriété afin de garantir que les restrictions sur les actions sociales prennent en compte les caractéristiques intrinsèques de chaque application Web 2.0 . Par exemple, Facebook ™est utilisé pour poster du texte, partager des photos, jouer à des jeux, etc. tandis que Twitter ™est utilisé pour poster de courts textes et partager des vidéos, seulement.

Table 4.4 présente les propriétés des actions sociales, une par catégorie, selon l'étude de cas de l'agence de voyages. Des restrictions possibles sont également incluses dans le tableau à des fins d'illustration. Pour développer des restrictions, nous avons établi une analogie avec les termes juridiques de l'utilisation des logiciels comme les droits de propriété intellectuelle (par exemple, les brevets et les droits d'auteur) et les questions de sécurité (par exemple, sensibilité et véracité). promouvoir les produits des concurrents). Nous appliquons des restrictions aux propriétés d'une action sociale et/ou à l'action sociale elle-même afin que les restrictions soient définies de manière exhaustive. Cela permet d'avoir un meilleur réglage des restrictions, que ce soit au niveau de l'action, au niveau de la propriété ou aux deux niveaux.

4.4.3 Definition des restrictions

Lors de la définition des restrictions, nous avons développé un diagramme de classe UML pour représenter tous les concepts nécessaires associés à une action sociale et ensuite les relations entre ces concepts. Ce diagramme contient des informations sur toute action sociale (classe abstraite `social`) ainsi que sur ses trois propriétés principales (`stakeholders`, `content` et `tool` classes). La classe abstraite `Social` est spécialisée dans les classes abstraites `communication`, `sharing` et `enrichissement` qui sont à leur tour spécialisées dans des classes d'actions sociales spécifiques comme `chat`, `co-author` et `recommend`. Ces derniers contiennent des méthodes (e.g., `openSession()` et `sendMessage()`) qui prennent en charge l'exécution d'actions sociales. Il convient de noter que malgré le contenu riche du diagramme de classes UML proposé, les restrictions intrinsèques à certains domaines d'application/études de cas ne sont pas gérées. À cette fin, nous suggérons d'utiliser Object-Constraint Language (OCL).

Dans la section 4.4.2, nous mentionnons que les restrictions s'appliquent à la fois à l'action elle-même et aux propriétés de l'action. Les restrictions sur les actions dépendent des états (par exemple, `activé` et `suspendu`) qui définissent le cycle de vie d'une action. Fig. 4.5 est un exemple de cycle de vie pour une action sociale, par exemple chat. Il se compose de cinq états qui sont non-activés, activés (décomposés en trois sous-états : contact identifié, communication établie, et contenu échangé) ,

Tableau 4.4 – Illustration des propriétés des actions sociales et des restrictions

	Properties	**Examples**	**Restrictions over**	
			Properties	**Action itself**
Communication category : chat	Stakeholders	Customers and travel consultants	Customer's identity should be known (anonymity not allowed) to the consultant at start-up time Customer should not be a group of persons	No chat during certain time slots and/on days No more than m new chat sessions per day
	Content	Tips for customers on long-distance trips	No more than n characters per chat message Attachments are not allowed during a chat session	No more than$n$concurrent active chat sessions at a time No more than p minutes of activity per chat session
	Tool	Google Talk	Google Talk 2.0	
Sharing category : co-author	Stakeholders	Senior and junior trainers	All co-authors' identities should be known	No more than$n$concurrent active co-authoring sessions
	Content	Business activity reports	Financial statements (e.g., sales and revenues) cannot be modified on-line	No more than$m$edits per co-authoring session
	Tool	Wikipedia	Browser 2.0	No more than$p$co-authoring sessions per day
Enrichment category : recommend	Stakeholders	N/A	N/A	No more than n characters per recommendation
	Content	Ratings on training courses	No competitors' products and/or services are to be recommended No recommendations on marketing strategies are shared without prior approval	No more than n recommendations per day
	Tool	Facebook^{TM}and Google+^{TM}	Google+ 1.2^{TM}	

suspendu, terminé et abandonné. L'état aborted est extrêmement important pour détecter les violations de restriction (appel à un contrôle continu des actions sociales selon la section 4.4.4) et donc des actions correctives sont prises comme l'arrêt de l'action et/ou l'initiateur de l'action . Les transitions entre états sont étiquetées avec des méthodes de classe. Dans la figure 4.5, ces transitions incluent openSession(), inviteContact(), sendMessage(), suspendSession(), resumeSession() , et closeSession()

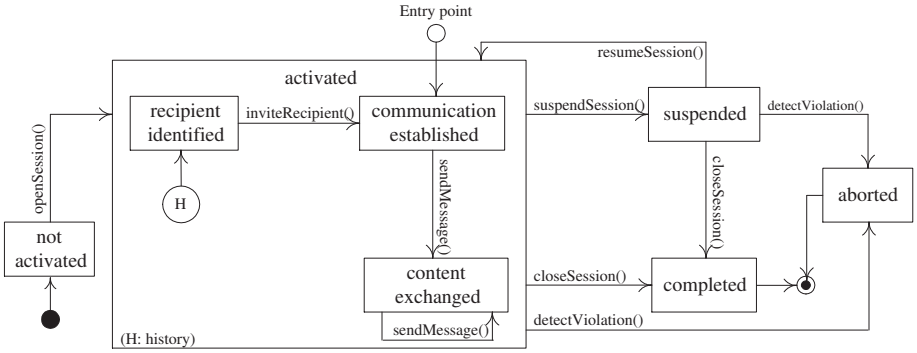


FIGURE 4.5 – Representation du cycle de vie du chat

En utilisant le cycle de vie d'une action sociale, nous définissons des suites d'états (s) et des transitions ($trans$) dénotées comme$social\ action_{s_j}$ avec s_j comme un état final dans ce cycle de vie : $social\ action_{s_j} = s_i \overset{trans_i}{\longrightarrow} s_{i+1} \overset{trans_{i+1}}{\longrightarrow} s_{i+2} \dots s_{j-1} \overset{trans_{j-1}}{\longrightarrow} s_j$ (Les indices dans les noms d'état et de transition sont donnés uniquement à des fins de notation). Nous listons ci-dessous des exemples de séquences associées aux états finaux completed et aborted pour le cycle de vie de `chat`.

- $chat_{completed}$: not activated $\overset{openSession()}{\longrightarrow}$ `activated/contact identified` $\overset{inviteContact()}{\longrightarrow}$ `activated/communication established` $\overset{sendMessage()}{\longrightarrow}$ `activated/content exchanged` $\overset{closeSession()}{\longrightarrow}$ completed. L'initiateur commence une nouvelle conversation avec un certain destinataire connu. Le chat reste actif jusqu'à ce que la transition de contenu échangé à terminé soit satisfaite (par exemple, l'initiateur ferme le chat).

- $chat_{aborted}$: not activated $\overset{openSession()}{\longrightarrow}$ `activated/contact identified` $\overset{inviteContact()}{\longrightarrow}$ `activated/communication established` $\overset{sendMessage()}{\longrightarrow}$ `activated/content exchanged` $\overset{detectViolation()}{\longrightarrow}$ aborted. La transition de activated/content exchanged vers aborted **empêche** l'initiateur d'envoyer un message en dehors des intervalles de temps autorisés, par exemple.

Comme indiqué précédemment, nous utilisons OCL pour spécifier les restrictions (les lecteurs sont référés à [35] pour plus de détails sur OCL). Une définition OCL commence par une spécification de contexte qui identifie l'objet de classe (instance) qui sera soumis à des contraintes. Une contrainte appelée invariant (`inv`) définit une condition (c'est-à-dire une expression booléenne basée sur des opérateurs logiques et relationnels) qui doit toujours être vraie pendant l'instanciation de classe.

Dans la même contrainte, `self` indique l'objet que la condition cible. Nous considérons les opérations pertinentes de l'OCL suivantes : *(i)* `forAll` vérifie si une collection d'objets satisfait une certaine condition ; *(ii)* `oclIsKindOf` vérifie si un objet est une instance d'une certaine classe ; *(iii)* `includes` vérifie si un objet appartient à une certaine collection d'objets ; *(iv)* `oclInState` renvoie true si l'objet est dans un certain état ; *(v)* `selects` renvoie le sous-ensemble d'objets dans une certaine collection d'objets qui satisfont une certaine condition ; *(vi)* `taille ()` renvoie le nombre d'objets contenus dans une certaine collection d'objets ; et *(vii)* `isEmpty` renvoie true si une collection d'objets est vide.

Nous utilisons aussi deux opérateurs unaires OCL : *(i)* →indique qu'une certaine opération est appliquée sur un ensemble d'objets ; et *(ii)* `def` définit des méthodes spécifiques à usage interne telles que la navigation dans le diagramme de classes et la sélection d'objets. L'OCL limite la portée (c'est-à-dire la condition (`pre`)) et l'effet (c'est-à-dire la post-condition (`post`)) des méthodes de classe sans se référer à l'implémentation de la méthode. En fait, OCL utilise des invariants pour les objets et des conditions pré et post pour les méthodes de classe. Dans la suite, nous associons des contraintes avec des noms tels que $R_i$et suggérons des exemples de restrictions.

Certaines restrictions sur les propriétés de chat sont listées ci-dessous :

1. Stakeholders : e.g., l'identité du participant doit être connue (anonymat non autorisé) à l'employé au moment du démarrage

   $$\begin{array}{ll} \texttt{context} & Chat \\ \texttt{inv}\ R_1\texttt{:} & \texttt{self}.stakeholders \rightarrow forAll(s : Stakeholder \mid s.Name <> null) \end{array}$$

   où `self`.stakeholders contient l'ensemble des instances Stakeholder représentant les participants qui participent à une session de discussion.|sépare l'objet de la condition.

2. Content : pas plus de$n$caractères par message de discussion

   $$\begin{array}{ll} \texttt{context} & Chat \\ \texttt{inv}\ R_3\texttt{:} & \texttt{self}.messages() \rightarrow forAll(m : Message \mid m.Size < n) \end{array}$$

   où `messages()` est défini comme suit :

   $$\begin{array}{ll} \texttt{context} & Chat\ \texttt{def:}messages() : Set(Message) = \\ & self.content \rightarrow select(m : Content \mid m \rightarrow oclIsKindOf(Message)) \end{array}$$

   où `self.content` contient l'ensemble des instances `Content` représentant les messages échangés pendant une session de discussion.

3. Tool : e.g., l'outil de chat approuvé par l'entreprise devrait être utilisé

   $$\begin{array}{ll} \texttt{context} & Chat \\ \texttt{inv}\ R_5\texttt{:} & \texttt{self}.tool \rightarrow oclIsKindOf(Internal) \end{array}$$

Certaines restrictions sur chat lui-même sont basées sur son cycle de vie et sont listées ci-dessous :

$$\begin{array}{ll} \texttt{context} & \mathit{Stakeholder} \\ \texttt{inv R}_7\texttt{:} & \mathit{self.activatedChats}() \rightarrow \mathit{size}() \ <= \ n \end{array}$$

où `activatedChats()` renvoie l'ensemble des sessions de conversation actives et défini comme suit :

$$\begin{array}{ll} \texttt{context} & \mathit{Stakeholder}\ \texttt{def:}\mathit{activatedChats}() : \mathit{Set}(\mathit{Chat}) = \\ & \mathit{self.initiatedChats}() \rightarrow \mathit{select}(c : \mathit{Chat} \mid c.\mathit{oclInState}(\mathit{activated})) \end{array}$$

où `initiatedChats()` renvoie l'ensemble des sessions de discussion initiées par un acteur et définies comme suit :ws :

$$\begin{array}{ll} \texttt{context} & \mathit{Stakeholder}\ \texttt{def:}\mathit{initiatedChats}() : \mathit{Set}(\mathit{Chat}) = \\ & \mathit{self.actions} \rightarrow \mathit{select}(a : \mathit{Action} \mid a \rightarrow \mathit{oclIsKindOf}(\mathit{Chat})) \end{array}$$

où `self.actions` contient l'ensemble des instances `Action` que l'acteur a initiées.

4.4.4 Monitoring des restrictions

Pour assurer le respect des restrictions définies précédemment, une surveillance est jugée nécessaire. L'objectif est de détecter les violations de restriction, puis de mettre en œuvre les sanctions nécessaires (par exemple, avertir les utilisateurs et désactiver les actions sociales) en réponse à ces violations.

Notre approche de monitoring est basée sur les événements liés aux restrictions, par exemple, `self.TimeSlot -> includes (currentTime)` où `currentTime` est l'événement. Pour développer cette approche, nous ajoutons une nouvelle méthode (`detectViolation ()`) à la classe `Action`. `detectViolation ()` " observe " les événements liés à la même restriction et vérifie si cette restriction est violée. Une attention particulière est accordée aux états avant l'état `aborted` dans le cycle de vie d'une action sociale. Cependant, ces états antérieurs peuvent déclencher une sorte d'indéterminisme dû au grand nombre possible d'états successeurs qui sont connectés à ces états antérieurs et sont différents de l'état `aborted`. Pour éviter cet indéterminisme, nous avons utilisé la condition `pre` dans la définition OCL de `detectViolation ()`. En outre, nous avons spécifié la condition `post` dans cette définition OCL pour nous assurer que `aborted` est l'état suivant lorsque `detectViolation()` détecte une violation.

A titre d'illustration, nous utilisons le cycle de vie du `chat` qui inclut plusieurs séquences ($seq(chat^{j=1,n}_{aborted})$) qui mènent à aborted :

- $seq(chat^{1}_{aborted})$: not activated $\overset{openSession()}{\longrightarrow}$ `activated/contact identified` $\overset{detectViolation()}{\longrightarrow}$ aborted.

  La transition $detectViolation()$ devrait empêcher l'initiateur d'un chat de converser avec des participants inconnus selon R_1. Nous définissons la contrainte OCL sur cette transition comme suit :

$$\begin{array}{ll} \texttt{context} & \mathit{Chat} :: \mathit{detectViolation}() \\ \texttt{pre :} & \mathit{oclInState}(\mathit{activated} :: \mathit{contact\ identified}) \\ \texttt{post :} & \mathit{not}(R_1)\ \texttt{implies}\ \mathit{oclInState}(\mathit{aborted}) \end{array}$$

où l'opérateur `implique` permet de formaliser l'instruction suivante : siR_1est violé, alors l'état de la session `chat` passera à `aborted`.

- $seq(chat^2_{aborted})$: not activated $\overset{openSession()}{\longrightarrow}$ `activated` $\overset{detectViolation()}{\longrightarrow}$ aborted.
  $detectViolation()$: La transition devrait empêcher l'initiateur d'un chat d'ouvrir$n$sessions concurrentes à la fois selon R_7. Nous définissons la contrainte OCL sur cette transition comme suit :

  `context` $Chat :: detectViolation()$
  `pre :` $oclInState(activated)$
  `post :` $not(R_7)$ `implies` $oclInState(aborted)$

4.5 Positionnement

Cette section présente le positionnement de nos travaux sur le cycle de vie d'un BP selon une dimension sociale par rapport à l'état de l'art.

Plusieurs initiatives de recherche combinent les processus métier et les applications collaboratives au sens social (e.g., micro-blogging) comme les travaux de Brambilla et al. [18] et ceux de Koschmider et al. [42]) décris ci-après. Dans [18], Brambilla et al. mettent en avant une notation spécifique pour concevoir des processus métier sociaux. Ils stipulent que les réseaux sociaux deviennent cruciaux dans la stratégie globale des organisations sans compromettre les pratiques commerciales des solutions BPM conventionnelles. Malgré ces avantages, il y a un manque de notations appropriées pour refléter les aspects sociaux dans les modèles de processus métier. La notation des auteurs comprend un ensemble de nouveaux types d'événements et de tâches, tels que diffuser et publier de l'information, et inviter à participer à une activité de collaboration. Les auteurs mentionnent également que la socialisation dans le contexte des BP nécessite une catégorisation des acteurs, une visibilité de l'état des processus et un niveau de participation sociale. Dans [42], Koschmider et al. montrent comment les réseaux sociaux aident à renforcer la confiance entre les acteurs des processus métier. Un premier réseau permet de fournir une vue organisationnelle des processus métier en suggérant une distance moyenne entre les acteurs ayant participé à des processus métier existants et ceux participant à l'élaboration de processus opérationnels. Un second réseau montre les relations entre les concepteurs utilisant un système de recommandation pour construire le modèle du processus métier. Dans la littérature, mixer le logiciel social avec la conception des BP est abordé pour montrer le potentiel de ces relations sociales lors de la conception des processus métier. Nous avons identifié deux autres perspectives non encore examinées. Une serait d'identifier les relations sociales appropriées au contexte particulier des entreprises et une autre serait d'ajuster les processus métier en tirant avantage de ces relations sociales. Notre solution de "mixage" rentre dans le cadre de ces deux perspectives (Section 4.2).

La gestion des conflits de ressources a fait l'objet de nombreuses études dans le cadre de l'exécution des processus métier. Par exemple, Han et al. [100] classent les situations de conflit de ressources en deux grandes catégories. La première catégorie

se concentre sur les situations liées à l'organisation où les changements dans les structures et les ressources liées à l'organisation (e.g., changements de personnel) peuvent avoir un impact direct sur l'exécution du processus métier. La deuxième catégorie se concentre sur les situations dans lesquelles les données non-utilisées par le processus peuvent être modifiées indépendamment par d'autres processus. Un autre travail est celui de Azevedo et al. [11] où une analyse ontologique de l'architecture d'entreprise est présentée. Cette analyse met particulièrement l'accent sur la capacité de ressources. Le but principal est d'identifier les problèmes sémantiques dans l'architecture proposée et de suggérer des recommandations pour d'éventuelles améliorations. En résumé, les techniques existantes de gestion des conflits de ressources passées en revue se concentrent principalement sur la disponibilité des ressources et l'aspect temporel. A notre connaissance, aucun d'entre eux ne considère les aspects sociaux dans la gestion des conflits survenant lors de l'exécution du processus métier. Notre solution se base sur ces aspects pour suivre les interactions entre les tâches, entre les personnes, et entre les machines et, par conséquent, pour résoudre ces conflits (Section 4.3). Par exemple, une tâche est remplacée par une autre tâche similaire en raison du manque d'exécuteurs, ou encore une personne évite de faire équipe avec une autre personne en raison d'expériences passées infructueuses. Le rôle des applications Web 2.0 dans la vie d'aujourd'hui, en général, et sur le lieu de travail, en particulier, suscite un vif débat dans notre société. Les avantages sont, par exemple, de contacter plus de personnes et d'exploiter les données/connaissances issues des interactions sociales. Par contre, il est possible de distraire les employés et, ainsi, favoriser les risques liés à la sécurité [79]. En plus de sécuriser les passerelles Web pour répondre à ces violations, nous estimons important de «contrôler» les actions pouvant mener à des violations. Dans la communauté R&D (e.g., [14, 17, 23, 70, 47] et [95]), l'accent est principalement mis sur la sécurité et la confidentialité des données des applications Web 2.0. Par exemple, le travail de Bhatti et al. [17] évalue de manière empirique le rôle des contrôles d'accès dans les entreprises utilisant des applications sociales. Les auteurs associent des contrôles à des réglementations définissant comment les données peuvent être consultées et partagées via ces applications. Par exemple, " *... toute divulgation par inadvertance de ces données à des tiers entraînera une violation de la confidentialité importante, et donc des contrôles d'accès doivent être mises en place pour garantir que les messages soient toujours envoyés en toute sécurité par des utilisateurs autorisés.*" [17]. Malheureusement, le contrôle d'accès de Bhatti et al. ne tient pas compte des actions sociales que les employés exécutent en termes de fréquence, de temps, de validité, etc. Les résultats de ces actions sont des sources potentielles de violations sur les données. A notre connaissance, il y a un manque d'approches assistant les utilisateurs dans leur manière d'utiliser efficacement les applications Web 2.0 et/ou suggèrant des actions préventives plutôt que correctives. Des cas comme la diffusion de photos sur le Web et l'affichage de renseignements personnels montrent les graves dommages que ces actes ont sur la vie privée des gens [73]. Notre solution consiste à "contrôler" les actions avant de les exécuter (Section 4.4).

4.6 Conclusion

Dans le cadre de notre recherche sur l'entreprise sociale (e.g., [30]), un premier travail fût la conception sociale des processus métier. Cette conception se base sur la définition de relations (ou interactions) "sociales" entre les composants du processus métier, à savoir les tâches, les personnes, et les machines. Ces interactions révèlent, par exemple, quelle personne est principalement sollicitée pour un partenariat spécifique avec d'autres personnes, quelle machine fonctionne mieux avec d'autres, et quelle tâche est "facile" à remplacer par d'autres tâches. Une méthodologie de construction de ces réseaux a été développée. Un second travail fût de gérer les conflits de ressources lors de l'exécution des processus métier. Plusieurs étapes ont été definies : catégoriser les ressources nécessaires à l'exécution du processus (logiques *versus* physiques), catégoriser les conflits sur les ressources entre les tâches, entre les machines et entre les personnes, et enfin analyser le rôle de certains réseaux de tâches, de personnes et de machines dans la résolution de ces conflits. Les ressources sont caractérisées par des propriétés (e.g., limitée, illimitée, et renouvelable) ayant un impact sur les niveaux d'utilisation et de consommation et ce par catégorie de ressources. Nos actions de coordination aux conflits s'appuient sur ces réseaux. Pour garantir la cohérence et la faisabilité de ces actions, nous avons associé des propriétés transactionnelles aux tâches, et de manière analogique, des propriétés d'activité aux personnes et des propriétés opérationnelles aux machines. Le troisième travail présente une approche pour assurer le bon usage des actions sociales sur le lieu de travail. Malgré l'enthousiasme suscité par les applications et les technologies WebN 2.0, de nombreuses entreprises les considèrent encore comme des sources de distraction et de violation de la sécurité. De nombreux cas d'abus sont signalés et justifiés par l'absence de lignes directrices et de campagnes de sensibilisation. Notre solution consiste en un ensemble de restrictions permettant de limiter, par exemple, le contenu d'une action sociale et le destinataire d'une action sociale, et le nombre de fois qu'une action sociale est exécutée. Des exemples d'actions sociales incluent poster, partager, commenter et co-auteur. Les restrictions ont été illustrées dans le contexte d'une société de voyages et démontrées à travers une application chat personnalisée déployée sur Google + Hangouts^{TM}.

Deuxième partie

Prospectives de recherche

5.1 Introduction

Financer ses activités de recherche est une préoccupation majeure. Les agences et programmes de financement sont divers et multiples que ce soit à l'échelle nationale (e.g., ANR et ANRT) et/ou internationale (e.g., ERC). Plusieurs articles (e.g., [43] et [48]) discutent des clés de réussite pour financer des projets ou contrats de recherche comme : (i) choisir un défi scientifique majeur, de préférence reconnu par la communauté scientifique et définir de sous-défis dérivés de celui-ci, de plus petite taille et pouvant être plus facilement mis en œuvre et évalués, et (ii) affiner le plan de recherche, les méthodes d'évaluation, le consortium, et le budget de manière à garantir une consistance entre ces différents éléments. Il est très important d'évaluer les compétences nécessaires et suffisantes pour accomplir le travail de recherche et d'estimer le budget adéquat. S'imprégner de la philosophie des agences de financement est aussi primordial car cela permet de mieux comprendre leurs attentes auxquelles les objectifs du projet devront se conformer. L'Union Européenne, par exemple, finance des projets pour faire avancer la science. Par conséquent, il devient impératif de s'orienter vers des problèmes scientifiques dont les enjeux font partie des priorités de l'Union Européenne.

Nul doute que l'un des domaines les plus en vogue est l'Internet des objets (Internet of Things (IoT)) faisant disparaître les frontières entre réalité et fiction. Selon Gartner [4], 6,4 milliards d'objets connectés étaient utilisés en 2016, en hausse de 3% par rapport à 2015, et atteindront 20,8 milliards en 2020. Pour soutenir cette croissance, l'IoT devrait faire face à différents obstacles aux niveaux : (i) infrastructure (e.g., diversité des technologies de développement et normes de communication [5]), et (ii) méthodologie (e.g., absence d'une discipline de génie logicielle orientée-IoT [104] et nature passive des objets [63, 25]). Cela conduit au problème de confiance des utilisateurs en l'IoT en raison de l'intrusion/invasion dans leur vie privée. La littérature fait référence à plusieurs initiatives pour rendre les objets réactifs et proactifs à l'environnement. Les objets peuvent, par exemple, contacter des pairs, traduisant une attitude collaborative, former des communautés dynamiques si nécessaire, être responsables de leurs actions comme les objets sémantiques [40], Internet des objets sociaux [6], et l'Internet des agents [69].

La carte de recherche proposée comporte 2 volets, chacun se rapportant à l'un des deux niveaux susmentionnés. Les architectures micro-services constituent le premier volet dans lequel seront traités les problèmes liés à la découverte de micro-services 5G incontournables pour un IoT *fiable* (Section 5.2). Le second volet traitera l'intégration de IoT-BPM pour un IoT *performant* et *durable* (Section 5.3).

4. www.gartner.com/newsroom/id/3165317.

5.2 Architectures micro-services

Les microservices sont des composants logiciels distribués modulaires qui peuvent être déployés dans le nuage. Les applications de micro-services sont souvent plus flexibles et plus légères, car elles peuvent fonctionner sous la forme d'une collection de composants logiciels plus petits partageant un système d'exploitation (OS) pouvant héberger d'autres applications utilisant le même système. C'est ce qu'on appelle la virtualisation de système d'exploitation, dans laquelle le système d'exploitation est divisé pour exécuter plusieurs applications de microservices à la fois. L'approche est similaire à la façon dont la virtualisation des serveurs divise un serveur en compartiments pour différents utilisateurs utilisant des machines virtuelles (VM). Cependant, dans certains cas, les microservices sont plus efficaces que les machines virtuelles car ils peuvent partager un système d'exploitation, alors que chaque machine virtuelle nécessite son propre système d'exploitation. Cela signifie que les microservices peuvent lancer des applications tout en consommant moins de ressources.

5.2.1 Micro-services pour la virtualisation des fonctions réseau

La virtualisation des fonctions réseau (en anglais, NFV) est une technologie émergente qui vise à réduire les coûts et à apporter de l'agilité en découplant les fonctions réseau du matériel sous-jacent. NFV est un moyen de créer des applications et des services qui peuvent être déployés avec un logiciel pour fonctionner sur n'importe quelle plate-forme matérielle standard, plutôt que de s'appuyer sur des solutions d'infrastructure propriétaires.

Les objectifs de la NFV et des microservices sont très alignés. Avant NFV, les applications et services réseau étaient souvent déployés à l'aide de matériel et de logiciels spécialisés et exclusifs qui ne pouvaient fonctionner que dans des installations spécifiques. NFV permet de virtualiser le logiciel et les services, et encore de les exécuter dans un modèle de Cloud, afin qu'ils puissent être déployés dans n'importe quel environnement avec une infrastructure standardisée, souvent appelée NFVI. Les micro-services sont également conçus pour le déploiement dans le Cloud à l'aide de matériels et de systèmes d'exploitation standard, ce qui permet d'installer des applications distribuées sur une infrastructure cloud tout en conservant une flexibilité maximale.

Les micro-services peuvent être utilisés pour construire des services NFV. La meilleure façon de penser aux micro-services est de simplifier les grands systèmes logiciels complexes en les décomposant en sous-composants et en les distribuant sur de nombreux serveurs informatiques ou dans le Cloud. Cela permet aux applications d'être gérées et coordonnées sur une grande infrastructure virtualisée.

Le nouveau travail de thèse, proposé en collaboration avec l'équipe SARA du LAAS, vise à élaborer une approche de NFV prenant en charge l'ensemble du cycle de vie des VNF (Virtualized Network Functions). Ce dernier consistera à : (i) développer et construire les VNF, (ii) les déployer et les configurer, et (iii) les exécuter et les gérer. Comme pour la première phase, des mécanismes appropriés de description, de publication et de découverte basés sur la sémantique sont prévus. Pour la deuxième phase, une architecture globale pour un approvisionnement économique et

agile des VNF sera conçue, mise en œuvre et évaluée. Enfin, pour la dernière phase, un algorithme innovant sera conçu pour optimiser la gestion et le fonctionnement des VNF en cours d'exécution. Des paramètres peuvent être pris en compte tels que l'évolution élastique, le placement optimal dans le réseau d'hébergement, la latence, le coût global, etc. Des études de cas diverses servant d'illustration seront examinées tout au long de la thèse, tels que les cas d'utilisation appartenant aux réseaux de diffusion de contenu, à l'Internet des objets et aux réseaux de télécommunications 4G/5G.

5.2.2 Micro-services pour la fusion de SI hétérogènes

Ce projet de thèse CIFRE, récemment accepté, se situe dans le domaine des systèmes d'information et leur urbanisation. Elle vise à apporter des solutions scientifiques et opérationnelles à la problématique d'évolution de systèmes d'information dans un contexte de fusion d'entreprises. La problématique étudiée inclura la migration d'applications monolithiques existantes bâties sur des ERP propriétaires et hétérogènes vers des applications bâties sur le concept de micro-services pour mieux supporter les futures évolutions comme l'intégration d'*objets connectés* dans le système d'information (incluant différents processus métier), et en faciliter la maintenance. Il sera alors question de proposer une méthode d'identification automatique des micro-services à partir d'applications existantes conçues de façon autonome et donc présentant des hétérogénéités structurelles et sémantiques. L'approche préconisée est basée sur la définition d'une ontologie commune à différentes entreprises exerçant des activités similaires, de proposer un langage de spécification de la sémantique des applications en utilisant les concepts de cette ontologie, et d'identifier ensuite automatiquement les micro-services par différentes techniques de clusterisation (classique k-mean, flou c-mean, hiérarchique HAC, etc.) des spécifications similaires. Cette identification des micro-services doit aussi être accompagnée d'une méthode de génération automatique des schémas de bases de données des micro-services.

Cette thèse traitera ensuite de la problématique de la migration et de l'intégration des données des différents systèmes existants vers les différentes bases de données associées aux micro-services identifiés. En effet, les micro-services disposeront de leurs propres bases de données. Cette opération de migration inclura aussi une approche d'intégration puisque les différents systèmes existants possèdent des données communes. Cette intégration des données doit bien entendu résoudre (a) les problèmes hétérogénéité structurelle et sémantique des données, (b) la prise en compte des différences de règles métiers des systèmes existants avec le nouveau système, et (c) la prise en compte du contexte de chaque système existant (dans lequel des informations implicites donc non représentées sont utilisées). En effet, cette notion du contexte est très importante et doit pouvoir aider à expliciter les informations implicites des systèmes d'information existants.

5.3 Intégration sociale IoT-BPM

Les approches existantes pour intégrer l'IoT dans les processus métier [59, 60] étendent généralement les langages de workflows traditionnels avec des constructions (par exemple, les rôles des objets connectés) pour définir la logique métier des processus métier prenant en compte les objets connectés. Cependant, cette façon de définir la logique métier restreint les opérations des objets connectés et les empêche donc de s'engager dans une collaboration ad-hoc/opportuniste avec d'autres pairs. Nous présentons à la suite deux perspectives complémentaires à l'intégration IoT-BPM.

5.3.1 Processus d'objets connectés

Comme nouveau moyen d'intégrer l'IoT dans les BPs, la technique de Storytelling permettrait d'identifier les choses en fonction de leurs capacités, de les prendre en charge pour prendre en charge de nouvelles fonctionnalités, de faciliter la (dé-)connexion des objets connectés grâce à des relations prédéfinies et enfin inciter ou pénaliser les objets connectés en réponse à leur participation constructive ou destructrice aux processus métier. Le résultat de cette intégration est le processus d'objets (en anglais, Process of Things (PoT)). Le PoT serait une nouvelle façon de capitaliser sur les opportunités IoT. Notre objectif est de veiller à ce que les objets ne fonctionnent plus comme des silos mais contribuent collectivement à offrir des services à valeur ajoutée aux utilisateurs finaux.

Cela pourrait être possible en identifiant les relations entre les objets connectés à partir lesquelles ces objets vont développer des *réseaux* de contacts. Un objet utilise ces réseaux pour ajouter d'autres objets candidats pour une éventuelle participation dans les processus, pour éviter les conflits avec les objets avant de les inclure au processus, et pour consolider sa collaboration avec d'autres objets connectés. Atzori et al. développent un Internet social des objets connectés bâti sur ces relations [7] et soulignent l'importance de *" exploiter les relations sociales entre les objets connectés, et non plus parmi leurs propriétaires "* [6] Selon Khan et al., ces objets constitueront une collaboration sociale IoT environnement [41].

La mise en place d'un PoT nécessitera d'analyser les objets selon deux perspectives : *capability* (l'objet de ce projet) et *compatibility*. D'une part, la capacité prescrit les fonctions d'un objet connecté lorsqu'il devient fonctionnel et donc prêt à agir en collaboration avec d'autres pairs pendant la mise en place d'un PoT. Les capacités comprennent la détection, le stockage, le traitement et la diffusion avec l'option de les combiner (par exemple, la détection et le traitement). Nous nous référons à ces capacités sous la forme *individuelle* et suggérons également des capacités de *groupe* (par exemple, persuasion et négociation) qui nécessitent l'implication simultanée de plusieurs éléments pour atteindre ces capacités. Dans ce projet, nous commencerons par nous focaliser sur les capacités individuelles. D'autre part, la compatibilité indique la préoccupation d'un objet connecté avec la participation d'autres objets connectés dans le même PoT. Cela pourrait être établi en fonction des préférences des objets à travailler avec d'autres afin que les risques de conflits soient atténués et/ou évités. Par conséquent, faire participer les objets dans le même processus nécessiterait de

les identifier en fonction de leurs capacités, d'assurer leur regroupement collaboratif sans générer de conflits, et d'assurer leur bonne connexion.

5.3.2 Vers des objets cognitifs

Selon un livre d'IBM 2015 [34], l'Internet des objets (IoT) doit être aussi doté de plus d'intelligence afin que de meilleurs résultats puissent être obtenus. Cette intelligence pourrait prendre forme grâce à l'informatique cognitive. Dans une perspective similaire, Wu et al. argumentent que " ... sans une capacité cognitive globale, IoT est comme un stegosaurus maladroit (i.e., des muscles mais pas de cerveau" [96]. L'IoT cognitif est un terme utilisé par Wu et al. pour décrire la future génération des objets connectés. En ligne avec cette tendance cognitive, un autre rapport indique clairement que l'Internet des objets dépend fortement de l'intelligence des objets connectés pour exploiter les opportunités de l'IoT comme de meilleurs services pour connecter les objets entre eux [90]. En parallèle, les organisations s'appuient sur les processus métier (BP) pour remplir leurs missions et atteindre leurs objectifs. Les processus métier sont au centre de toutes les initiatives que les organisations entreprennent. En effet, un processus (*aka* know-how) "... n'est rien de plus que le fruit d'une leçon apprise dans le passé, transformé en norme par un groupe d'experts et établi en tant que obligatoire pour ceux devant effectuer le travail de manière efficace ". Il est évident que la nature passive des objets connectés (confinés au rôle de fournisseurs de données) est un obstacle à l'évolution de l'IoT. Un nouveau travail de thèse, récemment initié, fait référence à la conception d'objets connectés cognitifs. Dans ce projet, nous explorons le mixage de l'informatique cognitive avec IoT donnant naissance aux objets cognitifs, dans le contexte particulier des BPs. L'injection de capacités cognitives dans l'IoT aboutirait à des objets cognitifs (en anglais, Cognitive Things (CTs)) avec lesquels le BP devrait interagir (c'est-à-dire, ne pas agir seulement mais aussi les diriger) selon leurs besoins et exigences ainsi que l'environnement (en termes de contraintes, par exemple) de ces CTs. Notre objectif est de donner aux objets des capacités de raisonnement, d'apprentissage et d'adaptation de sorte que le processus métier ferait intervenir dans son modèle de processus. Bien que certains pourraient être sceptiques sur l'autonomisation des objets connectés, de plus en plus de chercheurs soutiennent qu'au vu des avancées technologiques et une démocratisation des objets connectés (i.e., coûts abordables voire faibles) la connectivité du "tout" devient omniprésente" [84]. À la suite de ces progrès, les objets connectés du quotidien sont en mesure de se connecter et d'être programmables dynamiquement. Nous préconisons des interactions hommes-objets connectés initiés à la volée et une adaptation des capacités des objets connectés en à ces interactions.

5.4 Conclusion

Cette partie prospective a pour objectif de faire ressortir un engagement dans une activité de direction de recherche (i.e., thèses de doctorat) et celle de la mise en place d'une équipe "projet", notamment des projets de recherche, respectivement. Deux principales directions de recherche sont proposées : architectures micro-services IoT et intégration sociale IoT-BPM. La première direction s'intéresse aux probléma-

tiques niveau infrastructure IoT comme la question d'optimisation dans la migration des données IoT et celle de découverte et d'ordonnancement dans la composition des VNFs. La seconde direction permet d'étudier de la synergie IoT-BPM sous ses différentes formes comme la cognition des objets connectés. Certaines des initiatives de recherche décrites se dans le cadre de co-supervision de thèses alors d'autres feront l'objet de projets de recherche à l'échelle nationale comme un ANR et à l'échelle internationale comme le H2020.

BIBLIOGRAPHIE

[1] H. Abdeldjelil, N. Faci, Z. Maamar, and D. Benslimane. A diversity-based approach for managing faults in web services. In *AINA*, pages 81–88, 2012.

[2] S. Adam and J. Doerr. How to better align bpm & soa - ideas on improving the transition between process design and deployment. In *Workshop on Business Process Modeling, Development, and Support*, 2008.

[3] E. Al-Masri and Q. H. Mahmoud. WSCE : A Crawler Engine for Large-Scale Discovery of Web Services. In *IEEE International Conference on Web Services (ICWS'2007)*, Salt Lake City, Utah, USA.

[4] M. Alrifai, D. Skoutas, and T. Risse. Selecting Skyline Services for QoS-based Web Service Composition. In *the 19th International World Wide Web Conference (WWW'2010)*, Raleigh, North Carolina, USA, 2010.

[5] D. Andročec, B. Tomaš, and T. Kišasondi. Interoperability and Lightweight Security for Simple IoT Devices. In *Information Systems Security Conference (ISS'2017) held in conjunction with the 40^{th} Jubilee International Convention on Information and Communication Technology, Electronics, and Microelectronics (MIPRO'2017)*, Opatija, Croatia, May 2017.

[6] L. Atzori, A. Iera, and G. Morabito. SIoT : Giving a Social Structure to the Internet of Things. *IEEE Communications Letters*, 15(11), November 2011.

[7] L. Atzori, A. Iera, G. Morabito, and M. Nitti. The Social Internet of Things (SIoT) - When Social Networks Meet the Internet of Things : Concept, Architecture and Network Characterization. *Computer Networks*, 56(16), 2012.

[8] A. Avižienis, J-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1), 2004.

[9] A. Avizienis. *Software Fault Tolerance*, volume 2, chapter The Methodology of N-Version Programming, pages 22–45. John Wiley & Sons, 1995.

[10] A. Avizienis, J-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1) :11–33, 2004.

[11] C.L.B. Azevedo, M.E. Iacob, J.P.A. Almeida, M. van Sunderen, L.P. Pires, and G. Guizzardi. Modeling Resources and Capabilities in Enterprise Architecture : A Well-Founded Ontology-based Proposal for ArchiMate. *Information Systems*, 54, 2015.

[12] Y. Badr and Z. Maamar. Can Enterprises Capitalize on Their Social Networks ? *Cutter IT Journal*, 22(10), October 2009.

[13] T. Baker, E. Ugljanin, N. Faci, M. Sellami, Z. Maamar, and E. Kajan. Everything as a resource : Foundations and illustration through internet-of-things. *Computers in Industry*, 94 :62–74, 2018.

[14] A. Beach, M. Gartrell, and R. Han. Solutions to Security and Privacy Issues in Mobile Social Networking. In *International Conference on Computational Science and Engineering (CSE'2009)*, Vancouver, BC, Canada, 2009.

[15] J. Bentahar, B. Moulin, and B. Chaib-draa. Specifying and Implementing a Persuasion Dialogue Game Using Commitments and Arguments. In *First International Workshop on Argumentation in Multi-Agent Systems (ArgMAS'2004)*, New York, NY, USA, 2005.

[16] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, 1981.

[17] R. Bhatti, C. Gaspard, and C. Nita-Rotaru. Access Control in Social Enterprise Applications : An Empirical Evaluation. In *IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCS'2013 Workshops)*, Philadelphia, PA, USA, 2013.

[18] M. Brambilla, P. Fraternali, and C. Vaca. A Notation for supporting Social Business Process Modeling. In *Fourth Workshop on Business Process Management and Social Software (BPMS2'2011) held in conjunction with The Seventh International Conference on Business Process Management (BPM'2011)*, Lucerne, Switzerland, 2011.

[19] S. Bruning, S. Weissleder, and M. Malek. A fault taxonomy for service-oriented architecture. In *10th IEEE High Assurance Systems Engineering Symposium*, pages 367–368, Washington, DC, USA, 2007. IEEE Computer Society.

[20] C. Castelfranchi. Commitments : From Individual Intentions to Groups and Organizations. In 1^{st} *International Conference on Multi-Agent Systems (ICMAS'1995)*, San Francisoc, CA, USA, 1995.

[21] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal*, 16(4), 2007.

[22] K. Decker and V.R. Lesser. Generalizing the Partial Global Planning Algorithm. *International Journal Cooperative Information Systems*, 1(2), 1992.

[23] B. Dinerman. Social Networking and Security Risks, 2011. GFI White Paper, http ://www.gfi.com/whitepapers/Social_Networking_and_Security_Risks.pdf.

[24] S. Dustdar and L. Juszczyk. Dynamic replication and synchronization of web services for high availability in mobile ad-hoc networks. *Service Oriented Computing and Applications*, 1(1), 2007.

[25] DZone. The Internet of Things, Application, Protocls, and Best Practices. Technical report, DZone, https ://dzone.com/guides/iot-applications-protocols-and-best-practices, 2017 (visited in May 2017).

[26] M. El-Menshawy, J. Bentahar, and R. Dssouli. Verifiable Semantic Model for Agent Interactions using Social Commitments. In *Second Workshop on LAnguages, methodologies and Development tools for multi-agent systemS (LADS'2009)*, Torino, Italy, 2009.

[27] C. Engelmann, S.L. Scott, C. Leangsuksun, and X. He. Symmetric Active/Active High Availability for High-Performance Computing System Services : Accomplishments and Limitations. In *International Symposium on Cluster Computing and the Grid*, Lyon, France, 2008.

[28] N. Faci, H. Abdeldjelil, Z. Maamar, and D. Benslimane. Using diversity to design and deploy fault tolerant web services. In *WETICE*, pages 73–78, 2011.

[29] N. Faci, Z. Maamar, V. A. Burégio, E. Ugljanin, and D. Benslimane. Web 2.0 applications in the workplace : How to ensure their proper use ? *Computers in Industry*, 88 :1–11, 2017.

[30] N. Faci, Z. Maamar, E. Kajan, and D. Benslimane. Research Roadmap for the Enterprise 2.0 : Issues & Solutions. *Scientific Publications of the State University Of Novi Pazar Journal, Series A : Applied Mathematics, Informatics & Mechanics*, 2(2), 2014.

[31] ACL FIPA. FIPA ACL Message Structure Specification. *Foundation for Intelligent Physical Agents*, 2002.

[32] L. R. Ford and D. R. Fulkerson. A simple algorithm for finding maximal network flows and an application to the hitchcock problem. *Canadian journal of Mathematics*, 1957.

[33] N. Fornara and M. Colombetti. Operational Specification of a Commitment-based Agent Communication Language. In 1^{st} *International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS'2002)*, Bologna, Italy, 2002.

[34] H. Green. The Internet of Things in the Cognitive Era : Realizing the Future and Full Potential of Connected Devices. www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WWW12366USEN, December 2015.

[35] Object Management Group. *Object Constraint Language, Version 2.3.1*, 2012.

[36] R.-H. Hwang, C.-N. Lee, Y.-R. Chen, and D.-J. Zhang-Jian. Cost Optimization of Elasticity Cloud Resource Subscription Policy. *IEEE Transactions on Services Computing*, 7(4), 2014.

[37] T. S. Jayram, S. Kale, and E. Vee. Efficient aggregation algorithms for probabilistic data. In *Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, USA, 2007.

[38] I. Jureta, S. Faulkner, Y. Achbany, and M. Saerens. Dynamic Web Service Composition within a Service-Oriented Architecture. In *IEEE International Conference on Web Services (ICWS'2007)*, Salt Lake City, Utah, USA, 2007.

[39] E. Kajan, N. Faci, Z. Maamar, A. Loo, A. Pljaskovic, and Q. Z. Sheng. The Network-based Business Process. *IEEE Internet Computing*, 18(2), 2014.

[40] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan. Smart Semantic Middleware for the Internet of Things. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO'2008)*, Funchal, Madeira, Portugal, 2008.

[41] W.Z. Khan, M.Y. Aalsalem, M.K. Kha, and Q.-A. Arshad. When Social Objects Collaborate : Concepts, Processing Elements, Attacks and Challenges. *Computers & Electrical Engineering*, 58 :397–411, 2017.

[42] A. Koschmider, M. Song, and H.A. Reijers. Social Software for Modeling Business Processes. *Journal of Information Technology*, 25(3), 2010.

[43] You-Na Lee, John P. Walsh, and Jian Wang. Creativity in scientific teams : Unpacking novelty and impact. *Research Policy*, 44(3) :684–697, 2015.

[44] W.A. Lesko. *Readings in Social Psychology : General, Classic and Contemporary Selections*. Boston : Allyn & Bacon, 1997.

[45] B. Limthanmaphon and Y. Zhang. Web Service Composition with Case-Based Reasoning. Adelaide, Australia, 2003.

[46] M. Little. Transactions and Web Services. *Communications of the ACM*, 46(10), 2003.

[47] N. Luis G., B.-V. Tina, A. Nirav, K. Anup K., S. Jaime S., and S. Munindar P. Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems. *The Knowledge Engineering Review*, (1), March 2016.

[48] T. Luukkonen. Conservatism and risk-taking in peer review : Emerging erc practices. *Research evaluation*, 21(1) :48–60, 2012.

[49] Z. Maamar, V. Burégio, N. Faci, D. Benslimane, and Q. Z. Sheng. "Controlling" Web 2.0 Applications in the Workplace. In *International IEEE Enterprise Distributed Object Computing Conference (EDOC'2015)*, Adelaide, Australia, 2015.

[50] Z. Maamar, N. Faci, K. Boukadi, Q. Z. Sheng, and L. Yao. Commitments to regulate social web services operation. *IEEE Trans. Services Computing*, 7(2), 2014.

[51] Z. Maamar, N. Faci, L. Krug Wives, Y. Badr, P. Bispo Santos, and J. Palazzo M. de Oliveira. Using Social Networks to Web Services Discovery. *IEEE Internet Computing*, 15(4), July/August 2011.

[52] Z. Maamar, N. Faci, M. Luck, and S. Hachimi. Specifying and implementing social web services operation using commitments. In *ACM Symposium on Applied Computing SAC*, 2012.

[53] Z. Maamar, N. Faci, S. Sakr, M. Boukhebouze, and A. Barnawi. Network-based social coordination of business processes. *Information Systems*, 58, 2016.

[54] Z. Maamar, N. Faci, S. Sakr, M. Boukhebouze, and A. Barnawi. Network-based Social Coordination of Business Processes. *Information Systems*, 58, 2016.

[55] Z. Maamar, H. Hacid, and M. N. Hunhs. Why Web Services Need Social Networks. *IEEE Internet Computing*, 15(2), March/April 2011.

[56] A. Maaradji, H. Hacid, J. Daigremont, and N. Crespi. Towards a Social Network Based Approach for Services Composition. In *Proceedings of the 2010 IEEE International Conference on Communications (ICC'2010)*, Cap Town, South Africa, 2010.

[57] Z. Malik and A. Bouguettaya. Rateweb : reputation assessment for trust establishment among web services. *Very Large Data Bases (VLDB) Journal*, 18(4), 2009.

[58] B. Medjahed and Y. Atif. Context-based Matching for Web Service Composition. *Distributed and Parallel Databases, Springer*, 21(1), January 2007.

[59] G. Meroni. Integrating the Internet of Things with Business Process Management : A Process-aware Framework for Smart Objects. In *CAiSE'2015 Doctoral Consortium at the 27th International Conference on Advanced Information Systems Engineering (CAiSE'2015)*, pages 56–64, Stockholm, Sweden, 2015. Springer Link.

[60] S. Meyer, A. Ruppen, and C. Magerkurth. Internet of Things-Aware Process Modeling : Integrating IoT Devices as Business Process Resources. In *International Conference on Advanced Information Systems Engineering (CAiSE'2013)*, pages 84–98, Valencia, Spain, 2013. Springer Link.

[61] L. Min, S. Weiming, H. Qi, and Y. Junwei. A Weighted Ontology-based Semantic Similarity Algorithm for Web Services. *Expert Systems with Applications*, 36(10), December 2009.

[62] S. Modgil, N. Faci, F. Rech Meneguzzi, N. Oren, S. Miles, and M. Luck. A Framework for Monitoring Agent- based Normative Systems. In *8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'2009)*, Budapest, Hungary, 2009.

[63] A. M. Mzahm, M. S. Ahmad, and A. Y. C. Tang. Agents of Things (AoT) : An Intelligent Operational Concept of the Internet of Things (IoT). In *International Conference on Intellient Systems Design and Applications (ISDA'2013)*, Bangi, Malaysia, 2013.

[64] N. C. Narendra. Generating Correct Protocols from Contracts : A Commitment-based Approach. In *2008 IEEE Congress on Services - Part I (SERVICES I 2008)*, Honolulu, Hawaii, USA, 2008.

[65] T.H. Noor, Q.Z. Sheng, A.H.H. Ngu, A. Alfazi, and J. Law. Cloud armor : A platform for credibility-based trust management of cloud services. In *The ACM Conference on Information and Knowledge Management (CIKM)*, 2013.

[66] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara. Semantic matching of web services capabilities. In *International Semantic Web Conference*, pages 333–347, 2002.

[67] A. Paschke and M. Bichler. Knowledge Representation Concepts for Automated SLA Management. *Decision Support Systems*, 46(1), 2008.

[68] J. Pearl. Heuristics : intelligent search strategies for computer problem solving. 1984.

[69] P. Pico-Valencia and J. A. Holgado-Terriza. Semantic Agent Contracts for Internet of Agents. In *IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW'2016)*, Omaha, NE, USA, 2016.

[70] M. Pradeep K., A. Nirav, and S. Munindar P. Engineering Privacy in Social Applications. *IEEE Internet Computing*, 20(2), March/April 2016.

[71] S.D. Ramchurn, D. Huynh, and N.R. Jennings. Trust in multi-agent systems. *Knowledge Engineering Review*, 19(1), 2004.

[72] T. Riggs and R. Wilensky. An algorithm for automated rating of reviewers. In *1st ACM/IEEE-CS joint conference on Digital libraries*, pages 381–387. ACM, 2001.

[73] P. Roth. Data Protection Meets Web 2.0 : Two Ships Passing in the Night. *UNSW Law Journal*, 33(2), 2010.

[74] A. Sajjanhar, J. Hou, and Y. Zhang. Algorithm for web services matching. *Lecture notes in computer science*, 3007 :665–670, 2004.

[75] D.S. Sawicki and W.J. Craig. The democratization of data : Bridging the gap for community groups. *Journal of the American Planning Association*, 62(4) :512–523, 1996.

[76] H. et al. Schaffers. Smart cities and the future internet : Towards cooperation frameworks for open innovation. In *The Future Internet*, pages 431–446, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[77] D.A. Schum and J.R. Morris. Assessing the competence and credibility of human sources of intelligence evidence : contributions from law and probability. *Law, Probability and Risk*, 6(1), 2007.

[78] J. Searle. *Speech Acts : An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[79] D. Sherry. Web 2.0 Security Threats and How to Defend Against Them, 2010 (visited September 2015).

[80] M. N. Singh, A. K. Chopra, and N. Desai. Commitment-Based Service-Oriented Architecture. *Computer*, 42(11), November 2009.

[81] M. P. Singh. An Ontology for Commitments in Multiagent Systems : Toward a Unification of Normative Concepts. *Artificial Intelligence and Law*, 7(1), 1999.

[82] B. Sternthal, L.W. Phillips, and R. Dholakia. The persuasive effect of source credibility : A situational analysis. *The Public Opinion Quarterly*, 42(3), 1978.

[83] A. Sudhir and S. Rudi. Automatic Matchmaking of Web Services. In *IEEE International Conference on Web Services (ICWS'2006)*, Washington, DC, USA, 2006.

[84] A. Taivalsaari and T. Mikkonen. A Roadmap to the Programmable World : Software Challenges in the IoT Era. *IEEE Software*, 34(1), 2017.

[85] M. Tavakolifard and K. C. Almeroth. A taxonomy to express open challenges in trust and reputation systems. *Journal of Communications*, 7(7) :538–551, 2012.

[86] W. T. Teacy, J. Patel, N. R. Jennings, and M. Luck. Travos : Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2), 2006.

[87] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *6th USENIX Symposium on Networked Systems Design and Implementation*, Berkeley, CA, USA, 2009.

[88] M.C.M. Troffaes. Generalizing the conjunction rule for aggregating conflicting expert opinions. *International Journal of Intelligent Systems*, 21(3), 2006.

[89] W-T. Tsai, Y. Chen, D. Zhang, and H. Huang. Voting multi-dimensional data with deviations for web services under group testing. In *International Conference on Distributed Computing Systems Workshops*, 2005.

[90] E. Ugljanin, Z. Maamar, M. Sellami, and N. Faci. Process of Things : Ensuring a Successful Connection Between Things. *Cutter Business Technology Journal*, 12(29), 2016.

[91] W.M.P. van der Aalst, M. Dumas, F. Gottschalk, A.H.M. ter Hofstede, M. Rosa, and J. Mendling. Preserving Correctness during Business Process Model Configuration. *Formal Aspects of Computing*, 22(3-4), 2010.

[92] Y. Wang and M.P. Singh. Formal trust model for multiagent systems. In *International Joint Conference on Artifical Intelligence*, Hyderabad, India, 2007.

[93] J. Weng, C. Miao, and A. Goh. Protecting online rating systems from unfair ratings. *Trust, Privacy, and Security in Digital Business Lecture Notes in Computer Science*, 3592, 2005.

[94] A. Whitby, A. Josang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Workshop on Trust in Agent Societies hold in the Autonomous Agents and Multi Agent Systems Conference*, 2004.

[95] J. Williams. Social Networking Applications in Health Care : Threats to the Privacy and Security of Health Information. In *Workshop on Software Engineering in Health Care*, Cape Town, South Africa, 2010.

[96] Q. Wu, G. Ding, Y. Xu, S. Feg, Z. Du, J. Wang, and K. Long. Cognitive Internet of Things : A New Paradigm Beyond Connection. *IEEE Internet of Things Journal*, 1(2), April 2014.

[97] X. Xie, B. Du, and Z. Zhang. Semantic Service Composition based on Social Network. In *17th International World Wide Web Conference (WWW'2008)*, Beijing, China, 2008.

[98] L. Xiong and L. Liu. Peertrust : Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7), 2004.

[99] R. R. Yager. Participatory learning : A paradigm for building better digital and human agents. *Law, Probability and Risk*, 3(1), 2004.

[100] H. Yanbo, S. Amit, and B. Christoph. A Taxonomy of Adaptive Workflow Management. In *Conference on Computer Supported Cooperative Work (CSCW'1998)*, Seattle, WA, USA, 1998.

[101] B. Yu, Cuihong Li, M.P. Singh, and K. Sycara. A dynamic pricing mechanism for p2p referral systems. In *AAMAS'04 : International Conference on Agents and Multi-Agent Systems*, New York, USA, 2004.

[102] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit : A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2008.

[103] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard : Defending against sybil attacks via social networks. *SIGCOMM Comput. Commun. Rev.*, 2006.

[104] F. Zambonelli. Key Abstractions for IoT-Oriented Software Engineering. *IEEE Software*, 34(1), January-February 2017.

[105] Z. Zheng and M. R. Lyu. An adaptive qos-aware fault tolerance strategy for web services. *Empirical Software Engineering*, 15(4) :323–345, 2010.

[106] R. Zhou and K. 2007 Hwang. Powertrust : A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel and Distributed Systems*, 18(4), 2007.

Troisième partie

Curriculum Vitae Long

Cursus universitaire et professionnel

Titres et diplômes

- Habilitation à Diriger des Recherches de l'Université Claude Bernard Lyon 1, Spécialité Informatique. 2018
- Doctorat de l'Université Reims-Champagne-Ardenne, Spécialité Informatique. 2007

Déroulement de carrière

- 2004-2006. Attachée Temporaire d'Enseignement et de Recherche à l'Université Reims-Champagne-Ardenne, Département Informatique.
- 2007-2008. Chercheure post-doctoral à King's College London (Royaume Uni), Equipe « Agents and Intelligent Systems ».
- Depuis 2008. Maître de conférences, département Informatique (IUT Lyon 1), Université Lyon 1.

Activités de recherche

I. Mes publications en chiffres (dblp, SJR, Core)

Tableau 5.5 – Tableau récapitulatif de la production scientifique

Type de production scientifique	Nombres
Articles de journaux internationaux	**19 dont 11 (Q_1) & 4 (Q_2)**
Articles de conférences internationales	**32 dont 9 (A) & 17 (B)**
Chapitres d'ouvrage	**3 ([32], [33], [39])**
Lettres	**3 ([15], [16], [3])**
Articles de revues nationales avec comité de lecture	**1 ([23])**
Articles de conférences nationales avec comité de lecture	**3 ([27], [43], [54])**

Tableau 5.6 – Liste des journaux internationaux sélectifs

Rang	Nom du journal	Références
Q_1	IEEE Transactions on Services Computing	[29]
	IEEE Internet Computing	[28], [40]
	IEEE Access	[10]
	Computers in Industry Elsevier	[1], [11]
	Information Systems Elsevier	[18]
	Artificial Intelligence and Law	[22]
	Engineering Applications of AI	[35]
	Simulation Modelling Practice and Theory	[41]
	ACM SIGSOFT Software Engineering Notes	[51]
Q_2	Concurrency and Computation : Practice and Experience	[2]
	Journal of Systems and Software Elsevier	[17]
	Service Oriented Computing and Applications Springer	[12], [21]

Tableau 5.7 – Liste des conférences internationales sélectives

Rang	Nom de la conférence	Références
A	BPM	[4]
	ICSOC	[31]
	EDOC	[19], [26]
	WISE	[36]
	AAMAS	[46], [47], [54]
	CIA	[48]
B	ICSOFT	[7], [8], [9]
	ADBIS	[24]
	WIMS	[44]
	ICEBE	[34]
	SAC	[37]
	RCIS	[5]
	AINA	[6], [13], [38]
	WETICE	[20], [30], [42]
	EUMAS	[49]
	CEEMAS	[52]
	SELMAS	[53]

II. Encadrement doctoral et scientifique

1) Encadrement de thèses soutenues : 2

a) Nom, Prénom : Abdeldjelil, Hanane
Titre : Adaptation dynamique de compositions de services Web dans un contexte de fautes
Directeurs : N. Faci (50%), D. Benslimane (PR, 50%)
Date début : 02/11/2009 **Date fin** : 20/11/2013
Type de financement : Bourse de thèse du gouvernement algérien
Publications : 2 conférences internationales (WETICE'11 et AINA'12), et 1 conférence nationale NOTERE'11.

b) Nom, Prénom : Saoud, Zohra
Titre : Approche robuste pour l'évaluation de la confiance des ressources sur le Web
Directeurs : N. Faci (50%), D. Benslimane (PR, 50%)
Date début : 04/10/2012 **Date fin** : 14/12/2016
Type de financement : Allocation de recherche
Publications : 1 journal Systems and Software Elsevier, 1 revue nationale Ingénierie des Systèmes d'Information, 3 conférences internationales (ADBIS'15, NETYS'15, et WETICE'14), et 1 conférence nationale NOTERE'15.

2) Encadrement de thèses en cours : 3

a) Nom, Prénom : Nouar, Nour El Houda
Titre : Virtualized Network Functions, Semantic Description, Publication and Discovery in Content Delivery Networks
Directeurs : S. Yangui (MCF, 50%), N. Faci (20%), S. Tazi (MCF HDR, 20%), K. Drira (DR, 10%)
Date début : 15/10/2017 **Date fin prévue** : 15/09/2020
Type de financement : Bourse CNRS

b) Nom, Prénom : Zouaghi, Imen
Titre : Enabling robust deep learning in Internet of Things
Directeurs : N. Faci (50%), A. Hadj Kacem (PR, 50%)
Date début : 15/01/2018 **Date fin prévue** : 28/12/2020
Type de financement : Bourse gouvernement tunisien

c) Nom, Prénom : Daoud, Mohamed Taoufik
Titre : Approche d'identification automatique des micro-services et de migration de données dans un contexte de fusion de systèmes d'information hétérogènes
Directeurs : N. Faci (50%), D. Benslimane (PR, 50%)
Date début : 19/09/2018 **Date fin prévue** : 18/09/2021
Type de financement : Bourse Cifre

3) Encadrement de stage Master Recherche : 4

a) Nom, Prénom : Romdhani, Senda
Titre : Approche pour l'évaluation du trust dans le contexte de la mutation d'objets connectés
Encadrants : N. Faci (40%), C. Ghedira (PR, 30%), N. Bennani (MCF, 30%)
Date début : 01/02/2018 **Date fin** : 15/07/2018
Type de financement : Projet inter-équipe LIRIS

b) Nom, Prénom : Bellaaj, Ameni
Titre : Modélisation de processus métier reconfigurables
Encadrants : N. Faci (100%)
Date début : 01/03/2017 **Date fin** : 30/06/2017
Type de financement : Bourse gouvernement tunisien
Publication : 1 journal Concurrency and Computation : Practice and Experience

c) Nom, Prénom : Khedher, Ibtissem
Titre : Mécanisme de filtrage d'attaques multi-identités dans l'évaluation de la confiance
Encadrants : N. Faci (100%)
Date début : 01/04/2015 **Date fin** : 30/06/2015
Type de financement : Bourse gouvernement tunisien

d) Nom, Prénom : Hachimi, Salahddine
Titre : Impact Analysis of Web Services Substitution on Configurable Compositions
Encadrants : N. Faci (100%)
Date début : 01/03/2011 **Date fin** : 30/06/2011
Type de financement : Equipe SOC
Publication : 1 chapitre de livre et 1 conférence internationale (SAC'12)

4) Encadrement de séjours doctoraux : 1

a) Nom, Prénom : Masmoudi, Abir (3 Année de thèse à l'Université de Sfax, Tunisie)
Titre : Classification sémantique des messages échangés sur les réseaux sociaux
Encadrants : N. Faci (50%), M. Barhamgi (MCF, 50%)
Date début : 01/03/2017 **Date fin** : 15/07/2017
Type de financement : Projet EU RiskTrack
Publication : 1 conférence internationale (AINA'18)

5) Supervision de stage postdoctoral : 1

a) Nom, Prénom : Saoud, Zohra
Titre : Evaluation des risques de radicalisation en ligne
Superviseurs : N. Faci (50%), M. Barhamgi (MCF, 50%)
Date début : 20/10/2017 **Date fin** : 30/06/2018
Type de financement : Projet EU RiskTrack
Publication : 1 conférence internationale (AINA'18)

III. Principales activités d'enseignement

Tableau 5.8 – Tableau récapitulatif

Discipline	Année	Niveau	Heures	Établissement
Bases de Données Architectures logicielles Ergonomie	2018-2019	DUT	220	Université Lyon 1 (IUT et FST)
	2017-2018	DUT	96	
	2016-2017	DUT	200	
	2014-2015	DUT/LP	250	
	2013-2014	DUT/LP	241	
		M2	19	
	2012-2013	DUT/LP	200	
		M2	19	
	2011-2012	DUT/LP	259	
		M2	29	
	2010-2011	DUT	230	
		M2	19	
	2009-2010	DUT	212	
		M2	19	
	2008-2009	DUT	150	

Positionnement : Une partie des enseignements en M2 ont porté sur les fondements de la tolérance aux fautes dans les services Web en lien avec l'un des travaux de thèse. Pour ceux en DUT, les architectures logicielles (principes, techniques, et outils) font *de facto* référence aux architectures orientées-service de pairs avec la modélisation de processus métier.

IV. Responsabilités

• **Scientifiques**

[1] Contrat industriel

- Titre : Approche d'identification automatique des micro-services et de migration de données dans un contexte de fusion de systèmes d'information hétérogènes
- Type : ANRT (Cifre)
- Dates : 09/18-09/21
- Partenaires : LIRIS et Groupe EMERAUDE (Entreprise SILAC)
- Responsables : N. Faci (LIRIS), D. Benslimane (LIRIS), et T. Barthelet (SILAC)

[2] Participation à un projet européen H2020

a) Informations sur le projet

- Acronyme et Titre : RiskTrack -Tracking tool based on social media for risk assessment on radicalisation
- Lien : http://www.risk-track.eu
- Dates : 10/16–09/18
- Partenaires : Universidad Autónoma de Madrid (Espagne), Université Lyon1 (France), Parc Sanitari Sant Joan (Espagne), et Cyprus Neuroscience and Technology Institute (Chypre)
- Budget : 462 722.78 € (budget Lyon 1 : 129 000 €)

b) **Responsabilité de 2 lots de travail** intitulés « Knowledge Representation in RiskTrack » et « Software Prototype ». Productions : 2 livrables (04/17 et 12/17) et 1 version α du prototype de l'interface RiskTrack en 10/17.

c) Publications : 1 article de journal IEEE Access ([10]) et 1 article de conférence internationale de rang B AINA 2018 ([6])

d) Autres implications (ex., recrutement et supervision d'un chercheur post-doctoral, relecture de livrables des autres lots de travail, participation à la traduction en français du site Web de RiskTrack)

[3] Co-responsable scientifique d'un projet Moyen-Orient

a) Informations sur le projet

- Titre : Towards Web 2.0 Applications Use-Guidelines for UAE Enterprises
- Dates : 09/12–08/14
- Partenaires : Zayed University (UAE) et Université Lyon 1
- Budget : 17 000 €

b) Publications : 1 article de journal Computers in Industry Elsevier ([11]), 1 article de conférence EDOC 2015 ([26]), et 1 papier Démo ([20]).

[4] Coordinateur scientifique d'un projet CNRS PEPS 3S

a) Informations sur le projet

- Acronyme et Titre : SecSKY - Sécurisation du Ciel
- Dates : 06/17–12/17
- Partenaires : LIRIS Lyon et LAMSADE Paris Dauphine
- Budget : 10 000 €

b) Publications : 1 conférence internationale de rang B AINA 2018 ([6]) et 1 soumission à IEEE Transactions on Knowledge and Data Engineering (TKDE) en Décembre 2017.

[5] Co-coordinateur scientifique d'un projet inter-équipe LIRIS

a) Informations sur le projet

- Acronyme et Titre : TuMuLT - Trust in Mutable Things
- Dates : 11/17–09/18
- Equipes LIRIS : SOC et DRIM
- Budget : 5 000 €

b) Contribution : le projet a démarré en Novembre 2017

[6] Participation au projet CNRS PEPS FaSciDo

a) Informations sur le projet

- Titre : Protection de la vie privée dans les systèmes intelligents cyber physiques
- Dates : 09/14–08/15
- Equipes : SOC (LIRIS, Lyon) et Data Science (LAMSADE, Université Paris-Dauphine)
- Budget : 20 000 €

b) Contribution : Méthode de préservation de données dans un contexte service

• **Administration de la science**

a) Responsabilités recherche

- Responsabilité Master Recherche du parcours Technologie de l'Information et du Web (TIWe), département Informatique, Faculté des Sciences et Technologies (Octobre 2011 à Septembre 2015). Durant ce mandat, participation à la maquette de formation recherche 2016-2021.

b) Responsabilités pédagogiques

- Responsable du centre d'enseignement "Outils et Méthodes de Génie Logiciel" (2009 - 2013 puis 2014 - 2015), département Informatique, IUT Lyon 1.

c) Participation à des commissions locales

- Membre de comités de sélection (2017, 2011), Spécialité Informatique, Université Lyon 1
- Membre du comité consultatif (depuis 2011), Spécialité Informatique, Université Lyon 1
- Membre du comité de pilotage des licences professionnelles Devops et SID (depuis 2011), IUT Lyon 1

d) Participation à la vie de l'équipe

- Co-responsable de la maintenance du wiki et de la plateforme logicielle de l'équipe SOC (depuis 2016 - . . .).

V. Diffusion des travaux

• **Rapporteur de thèses de doctorat à l'étranger**

a) **MEHDI** Mohamad. **Titre** : Trust and Reputation Management in Web Services : A Probabilistic Approach. Concordia University, Canada, 2015. https://www.concordia.ca/encs/computer-science-software-engineering/news/theses-defenses.html

• **Invitations**

a) Journées de travail INS2I (Objets Communicants : Algorithmes, Architectures et Applications). **Titre** : Services de sécurité pour la protection des objets connectés et des données collectées. Siège CNRS, 2017. http://www.cnrs.fr/ins2i/spip.php?article2563

b) Séminaire invité à l'Université Paris-Dauphine. **Titre** : Network-based Social Coordination of Business Processes. Pôle Sciences des données, 2016. https://www.lamsade.dauphine.fr/spip.php?article487

c) Séminaire invité à l'Université de Namur. **Titre** : Fault Tolerant Web Services. Belgique, 2011. https://www.unamur.be/services/sevrex/omalius/2011/libre-cours-81/lc81-p11

• **Instance d'évaluation à l'étranger**

a) Rapporteur technique externe de projets R&D depuis 2016 pour ICT Fund (UAE) équivalent de l'ANRT en France. https://www.tra.gov.ae/ictfund

• **Animation**

Organisation de workshops, journées thématiques, et groupes de travail

a) Co-Chair de la session Démonstration de ICSOC'19 (17th International Conference on Service-Oriented Computing. Toulouse, France). http://www.icsoc.org/

b) Co-Chair du workshop AICTSS (Advanced ICT Technologies for Secure Societies) conjointement avec la conférence internationale DEXA 2017 (28th International Conference on Database and Expert Systems Applications), Lyon, France. http://www.dexa.org/dexa2017

c) Co-Chair du workshop CCAC (Cloud and Context-Aware Computing) conjointement avec la conférence EUSPN 2017 (8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks), Lund, Suède. http://cs-conferences.acadiau.ca/euspn-17

Relecteur dans des journaux internationaux et nationaux

- IEEE Transactions on Services Computing : 2018, 2017, 2016, 2015, 2014
- IEEE Internet of Things Journal : 2018
- Distributed and Parallel Databases, Springer : 2018
- Sustainable Computing Journal, Elsevier : 2018
- Transactions on Large-Scale Data- and Knowledge-Centered Systems, Springer : 2018
- IEEE Access : 2017, 2016
- Future Generation Computer Systems, Elsevier : 2018, 2017
- Concurrency and Computation : Practice and Experience, Wiley : 2017
- Revue en Intelligence Artificielle, Hermès Science Publications : 2014
- Technique et Science Informatiques, Hermès Science Publications : 2014
- **etc.**

Comités de programme de conférences internationales et nationales

- CoopIS'18 : The 26th International Conference on Cooperative Information Systems. Valletta, Malta. 2018-10-26 - 2018-10-28.
- ICSOC'18 : The 16th International Conference on Service-Oriented Computing. Zhejiang, China. 2018-11-12 – 2018-11-15.
- WISE'18 : The 19th International Conference on Web Information Systems Engineering. Dubai, UAE. 2018-11-12 – 2018-11-15.
- DEXA'18 : The 29th International Conference on Database and Expert Systems Applications. Regensburg, Germany. 2018-09-03 – 2018-09-06.

- WETICE'18 : The 27th International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises. Paris, France. 2018-06-27 - 2018-06-29.
- AICCSA'18 : The 15th ACS/IEEE International Conference on Computer Systems and Applications. Quaba, Jordan. 2018-10-28 - 2018-11-01.
- ICSOC'17 : The 15th International Conference on Service-Oriented Computing. Malaga, Spain. 2017-11-13 – 2017-11-16.
- DEXA'17 : The 28th International Conference on Database and Expert Systems Applications. Lyon, France. 2017-08-28 – 2017-08-31.
- WETICE'17 : The 26th International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises. Poznan, Poland. 2017-06-21 – 2017-06-23.
- AICCSA'17 : The 14th ACS/IEEE International Conference on Computer Systems and Applications. Hammamet, Tunisia. 2017-10-30 - 2017-11-03.
- ICSOC'16 : The 14th International Conference on Service-Oriented Computing. Banff, Canada. 2016-10-10 – 2016-10-13.
- WETICE'15 : The 24th International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises. Larnaca, Cyprus, 2015-06-15 – 2015-06-17.
- CAL'15 : 9ème conférence francophone sur les architectures logicielles. Hamamet, Tunisie. 2015-05-13 -2015-05-15.
- WETICE'14 : The 23th International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises. Parma, Italy, 2014-06-23 – 2014-06-25.
- CAL'14 : 8ème conférence francophone sur les architectures logicielles. Paris, France. 2014-06-10 – 2014-06-11.
- **etc.**

Prix et distinction

a) Microsoft Azure Research Award. **Titre de la proposition** : Everything-as-a-Resource for the design and development of an IoT-driven, Cloud-based Healthcare System. Janvier 2017.

NB : En lien avec cette proposition, 1 article publié dans Computers in Industry Elsevier (94) 2018 ([13]).

Séjours hors Lyon

a) LAMSADE - Université Paris Dauphine Date début : 01/09/15 Date fin : 30/03/16 Titre : Confiance et intégration des données dans les workflows scientifiques.

b) Adaptive security and privacy group - Open University (Royaume Uni). Date début : 01/04/16 Date fin : 30/06/16 Titre : Confiance dans l'Internet des Objets.

c) LAAS - Université de Toulouse. Date début : 15/03/18 Date fin : 15/06/18 Titre : Confiance dans un Internet d'Objets Mutants.

- **Diffusion logicielle**

a) Plateforme de démonstration OASIC. Co-responsable de cette plateforme au sein de l'équipe SOC pour la diffusion des travaux de ses doctorants dont ceux de Z. Saoud. https://projet.liris.cnrs.fr/soc/doku.php?id=platform

b) Ontologie de Radicalisation en ligne (2017). Mise à la disposition de la communauté scientifique de recherche. Notre ontologie regroupe un ensemble de connaissances issues des domaines tels que la psycho-criminologie, la sociologie et la linguistique. Auteurs : M. Barhamgi et N. Faci. http://liris.cnrs.fr/radicalisation

Liste des publications

[1] Z. Maamar, T. Baker, **N. Faci**, M. Sellami and E. Ugljanin. Everything as a Resource : Foundations and Illustration through Internet-of-Things. *Computers in Industry*, 2018. (JCR IF : 1.658, Q_1)

[2] W. Benallal, M. Barhamgi, D. Benslimane, Z. Maamar, **N. Faci**, and A. Bellaaj. A Knowledge-based approach to manage configurable business processes. *Concurrency and Computation : Practice and Experience*, 2018. (Q_2)

[3] Z. Maamar, T. Baker, S. Kallel, M. Sellami, E. Ugljanin, and **N. Faci**. Cloud versus Edge : Who Serves the Internet-of-Things Better ? In *Internet Technology Letters*, 2018.

[4] Z. Maamar, M. Sellami, **N. Faci**, E. Ugljanin and Q. Z. Sheng. Integration of Internet of Things into Business Processes. In *International Conference on Business Process Management (Forum), Sidney, Australia*, LNBIP (329), 2018. (A)

[5] Z. Maamar, **N. Faci**, K. Boukadi, E. Ugljanin, M. Sellami, T. Baker, and R. Angarita. How to Agentify the Internet-of-Things ? In *International Conference on Research Challenges in Information Science (RCIS), Nantes, France*, 2018. (B)

[6] A. Masmoudi, M. Barhamgi, **N. Faci**, Z. Saoud, D. Benslimane, and D. Camacho. An Ontology-based Approach for Mining Radicalization Indicators from Online Messages. In *Advanced Information Networking and Applications (AINA), Krakow, Poland*, 2018. (B)

[7] **N. Faci**, Z. Maamar, T. Baker, E. Ugljanin, and M. Sellami. In Situ Mutation for Active Things in the IoT Context. In *International Conference on Software Technologies (ICSOFT), Porto, Portugal*, 2018. (B)

[8] Z. Maamar, **N. Faci**, M. Sellami, E. Ugljanin, and E. Kajan. Everything-as-a-Thing for Abstracting the Internet-of-Things. In *International Conference on Software Technologies (ICSOFT), Porto, Portugal*, 2018. (B)

[9] Z. Maamar, **N. Faci**, M. Sellami, E. Ugljanin, and E. Kajan. Cognitive Computing Meets The Internet of Things. In *International Conference on Software Technologies (ICSOFT), Porto, Portugal*, 2018. (Short Paper, B)

[10] R. Lara-Cabrera, A. Gonzalez-Pardo, K. Benouaret, **N. Faci**, D. Benslimane, and D. Camacho. Measuring the Radicalization Risk in Social Networks. *IEEE Access*, 2017. (JCR IF : 1.27, Q_1)

[11] **N. Faci**, Z. Maamar, V. Buregio, E. Ugljanin, and D. Benslimane. Web 2.0 applications in the workplace - how to ensure their proper use ? *Computers in Industry*, 2017. (JCR IF : 1.658, Q_1)

[12] Z. Maamar, **N. Faci**, M. Sellami, K. Boukadi, F. Yahya, S. Sakr, and A. Barnawi. On business process monitoring using cross-flow coordination. *Service Oriented Computing and Applications*, 2017. (SJR IF : 0.618, Q_2)

[13] Z. Maamar, M. Sellami, **N. Faci** and S. Lefebvre. Detecting and Tackling Run-Time Obstacles in Social Business Processes. In *Advanced Information Networking and Applications (AINA), Taipei, Taiwan*, 2017. (B)

[14] T. Baker, A. Hussien, M. Hanneghan, and **N. Faci**. Game-based learning in real life business context using didactic mash-up concepts. In *Annual International Conference of Education, Research and Innovation (ICERI), Seville, Spain*, 2017.

[15] E. Ugljanin, Z. Maamar, M. Sellami, and **N. Faci**. Process of Things : Ensuring a Successful Connection Between Things. In *Cutter Business Technology Journal*, 2017.

[16] Z. Maamar, **N. Faci**, S. Kallel, M. Sellami, and E. Ugljanin. Software Agents meet Internet of Things. In *Internet Technology Letters*, 2017.

[17] Z. Saoud, **N. Faci**, Z. Maamar, and D. Benslimane. A fuzzy-based credibility model to assess web services trust under uncertainty. *Journal of Systems and Software*, 122, pp. 496-506, 2016. (JCR IF : 1.352, Q_1)

[18] Z. Maamar, **N. Faci**, S. Sakr, M. Boukhebouze, and A. Barnawi. Network-based social coordination of business processes. *Information Systems*, Elsevier, 58, pp. 56-74, 2016. (JCR IF : 1.456, Q_1)

[19] E. Ugljanin, **N. Faci**, V. Burégio, and Z. Maamar. How to restrict web 2.0 applications use in the workplace ? Example of Google+ Hangouts. In *Enterprise Distributed Object Computing (EDOC, Demo session), Vienna, Austria*, pp. 1-4, 2016. (B)

[20] E. Ugljanin, **N. Faci**, M. Sellami, and Z. Maamar. Tracking users' actions over social media : Application to Facebook. In *Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE, Demo session), Paris, France*, pp. 255-256, 2016. (B)

[21] **N. Faci**, M. Petrocchi, G. Costantino, F. Martinelli, and Z. Maamar. A quality model for social networks populated with web services. *Service Oriented Computing and Applications*, 9(2), pp. 139-155, 2015. (SJR IF : 0.618, Q_2)

[22] S. Modgil, N. Oren, **N. Faci**, F. Meneguzzi, S. Miles, and M. Luck. Monitoring compliance with E-contracts and norms. *Artificial Intelligence and Law*, 23(2), pp 161-196, 2015.

[23] Z. Saoud, **N. Faci**, Z. Maamar, and D. Benslimane. Un modèle de crédibilité basé sur le clustering flou pour une évaluation probabiliste de la confiance des ressources sur le web. *Ingénierie des Systèmes d'Information*, 20(6), pp. 79-98, 2015.

[24] Z. Saoud, **N. Faci**, Z. Maamar, and D. Benslimane. Sybil tolerance and probabilistic databases to compute web services trust. In *Advances in Databases and Information Systems (ADBIS), Poitiers, France*, pp. 458-471, 2015. (B)

[25] Z. Saoud, **N. Faci**, Z. Maamar, and D. Benslimane. Web services trust assessment based on probabilistic databases. In *Networked Systems (NetSys), Agadir, Morocco*, pp. 397-410, 2015.

[26] Z. Maamar, V. Burégio, **N. Faci**, D. Benslimane, and Q.Z. Sheng. "Controlling" Web 2.0 applications in the workplace. In *Enterprise Distributed Object Computing (EDOC), Adelaide, Australia*, pp. 191-200, 2015. (A)

[27] Z. Saoud, **N. Faci**, Z. Maamar, and D. Benslimane. Impact of sybil attacks on web services trust assessment. In *Nouvelles Technologies de la Répartition (NOTERE), Paris, France*, 2015.

[28] E. Kajan, **N. Faci**, Z. Maamar, A. Loo, A. Pljaskovic, and Q.Z. Sheng. The network-based business process. *IEEE Internet Computing*, 18(2), pp. 63-69, 2014. (JCR IF : 1.713, Q_1)

[29] Z. Maamar, **N. Faci**, K. Boukadi, Q.Z. Sheng, and L. Yao. Commitments to regulate social web services operation. *IEEE Trans. Services Computing*, 7(2), pp. 154-167, 2014. (JCR IF : 3.049, Q_1)

[30] Z. Saoud, **N. Faci**, Z. Maamar, and D. Benslimane. A fuzzy clustering-based credibility model for trust assessment in a service-oriented architecture. In *Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE, AROSA track), Parma, Italy*, pp. 56-61, 2014. (B)

[31] Z. Maamar, S. Sakr, **N. Faci**, M. Boukhebouze, and A. Barnawi. SUPER : social-based business process management framework. In *International Conference on Service-Oriented Computing (ICSOC, Demo session), Paris, France*, pp. 413-417, 2014. (A)

[32] Z. Maamar, Y. Badr, **N. Faci**, and M. Sheng. Realizing an Ecosystem of Social Web Services : Concepts, Issues, and Existing Initiatives. In *Advanced Web Services, Springer*, 2014. (Book chapter)

[33] Z. Maamar, **N. Faci**, E. Kajan and E. Ugljanin. Social Web Services Management. In *Demand-Driven Web Services : Theory, Technologies, and Applications*, 2014. (Book chapter)

[34] Z. Maamar, **N. Faci**, S. Kouadri Mostéfaoui, and E. Kajan. Network-based conflict resolution in business processes. In *e-Business Engineering (ICEBE), Coventry, United Kingdom*, pp. 132-137, 2013. (B)

[35] F. Meneguzzi, S. Modgil, N. Oren, S. Miles, M. Luck, and **N. Faci**. Applying electronic contracting to the aerospace aftercare domain. *Engineering Applications of AI*, 25(7), pp. 1471-1487, 2012. (JCR IF : 2.027, Q_1)

[36] Z. Maamar, **N. Faci**, Q.Z. Sheng, and L. Yao. Towards a user-centric social approach to web services composition, execution, and monitoring. In *Web Information Systems Engineering (WISE), Paphos, Cyprus*, pp. 72-86, 2012. (A)

[37] Z. Maamar, **N. Faci**, M. Luck, and S. Hachimi. Specifying and implementing social web services operation using commitments. In *Symposium on Applied Computing (SAC), Trento, Italy*, pp. 1955-1960, 2012. (B)

[38] H. Abdeldjelil, **N. Faci**, Z. Maamar, and D. Benslimane. A diversity-based approach for managing faults in web services. In *Advanced Information Networking and Applications (AINA), Fukuoka, Japan*, pp. 81-88, 2012. (B)

[39] S. Hachimi, **N. Faci**, and Z. Maamar. Impact analysis of web services substitution on configurable compositions. In *Distributed Computing Innovations for Business, Engineering and Science book, IGI Global*, 2012. (Book chapter)

[40] Z. Maamar, **N. Faci**, L. Krug Wives, Y. Badr, P. B. dos Santos, and J. Palazzo Moreira de Oliveira. Using social networks for web services discovery. *IEEE Internet Computing*, 15(4), pp. 48-54, 2011. (JCR IF : 1.713, Q_1)

[41] Z. Maamar, L. Krug Wives, Y. Badr, S. Elnaffar, K. Boukadi, and **N. Faci**. Linkedws : A novel web services discovery model based on the metaphor of "social networks". *Simulation Modelling Practice and Theory*, 19(1), pp. 121-132, 2011. (JCR IF : 1.38, Q_1)

[42] **N. Faci**, H. Abdeldjelil, Z. Maamar, and D. Benslimane. Using diversity to design and deploy fault tolerant web services. In *Enabling Technologies : Infrastructures for Collaborative Enterprises (WETICE, AROSA track), Paris, France*, pp. 73-78, 2011. (B)

[43] **N. Faci**, Z. Maamar, H. Abdeldjelil, and D. Benslimane. Vers un framework intégrant les principes des réseaux sociaux dans la découverte de services web. In *Nouvelles Technologies de la Répartition (NOTERE), Paris, France*, 2011.

[44] Z. Maamar, **N. Faci**, Y. Badr, L.K. Wives, P.B. dos Santos, D. Benslimane, and J. Palazzo Moreira de Oliveira. Towards a framework for weaving social networks principles into web services discovery, In *International Conference on Web Intelligence, Mining and Semantics (WIMS), Sogndal, Norway*, 2011. (B)

[45] Z. Guessoum, J-P. Briot, **N. Faci**, and O. Marin. Towards reliable multi-agent systems : An adaptive replication mechanism. *Multiagent and Grid Systems*, 6(1), pp. 1-24, 2010.

[46] S. Modgil, **N. Faci**, N. Oren, F. Meneguzzi, S. Miles, and M. Luck. A framework for monitoring agent-based normative systems. In *Autonomous Agents and Multiagent Systems (AAMAS), Budapest, Hungary*, pp. 153-160, 2009. (A*)

[47] F. Meneguzzi, S. Modgil, N. Oren, S. Miles, M. Luck, **N. Faci**, C. Holt, and M. Smith. Monitoring and Explanation of Contract Execution : A Case Study in the Aerospace Domain. In *Autonomous Agents and Multiagent Systems (AAMAS), Industry and Applications Track, Budapest, Hungary*, 2009. (A*)

[48] **N. Faci**, S. Modgil, N. Oren, F. Meneguzzi, S. Miles, and M. Luck. Towards a Monitoring Framework for Agent-Based Contract Systems. In *Cooperative Information Agents (CIA), Prague, Czech Republic*, LNAI 5180, pp. 292-305, 2008. (A)

[49] **N. Faci**, Z. Guessoum, and O. Marin. DimaX : A Fault-Tolerant Multi-Agent Platform. In *European Workshop on Multi-Agent Systems (EUMAS), Lisbon, Portugal*, 2006. (B)

[50] J-P. Briot, Z. Guessoum, S. Aknine, A.L. Almeida, J. Malenfant, O. Marin, P. Sens, **N. Faci**, M.A. de C. Gatti, and C.J. Pereira de Lucena. Experience and prospects for various control strategies for self-replicating multi-agent systems. In *International Workshop on Self-adaptation and self-managing systems (SEAMS) hold in conjunction with International Conference on Software Engineering (ICSE), Shanghai, China*, 2006.

[51] Z Guessoum, **N. Faci**, and J-P. Briot. Adaptive replication of large-scale multi-agent systems : towards a fault-tolerant multi-agent platform. In *ACM SIGSOFT Software Engineering Notes*, 30(4), 2005.

[52] Z Guessoum and **N. Faci**. Towards Reliable Large-Scale Multi-agent Systems. In *Central and Eastern European conference on Multi-Agent Systems (CEEMAS), Budapest, Hungary*, LNCS 3690, pp 430-439, 2005. (B)

[53] Z Guessoum, **N. Faci**, and J-P. Briot. Adaptive Replication of Large-Scale Multi-agent Systems - Towards a Fault-Tolerant Multi-agent Platform. In *Software Engineering for Multi-Agent Systems, Research Issues and Practical Applications (SELMAS), Saint Louis, USA*, LNCS 3914, pp 238-253, 2005.

[54] Z Guessoum, M. Ziane, and **N. Faci**. Monitoring and Organizational-Level Adaptation of Multi-Agent Systems. In *Autonomous Agents and Multiagent Systems (AAMAS), New York, USA*, pp 514-521, 2004. (A*)

[55] Z Guessoum, J-P. Briot, **N. Faci**, and O. Marin. Un mécanisme de réplication adaptative pour des SMA tolérants aux pannes. In *Journées francophones sur les systèmes multi-agents (JFSMA), Paris, France*, pp 135-148, 2004.

[56] Z Guessoum, J-P. Briot, and **N. Faci**. Towards Fault-Tolerant Massively Multiagent Systems. In *Workshop on Massively Multi-Agent Systems (MMAS), Kyoto, Japan*, LNCS 3446, pp 55-69, 2004.

Quatrième partie

Annexes