

www.revue-eti.net

Editorial du numéro 6 p. 3
Les comités de la 6^{ème} édition de la revue eTi p. 4

Etat de l'art

Tendances des méthodes de gestion des projets informatiques p. 7
Lamia Ben Hiba, Mohammed Abdou Janati Idrissi
Ingénierie des exigences de la sécurité informatique p. 18
Meryem Kassou, Laila Kjiri

Dossier spécial: Conférence SITCAM'2011

Transformation de Fourier et moments invariants p. 31
Rachid El Ayachi, Mohamed Fakir, Belaid Bouikhalene
L'étiquetage grammatical de l'amazighe p. 44
Mohamed Outahajala, Yassine Benajiba, Paolo Rosso, Lahbib Zenkouar

Recherche & Développement

Système d'information voyageur p. 56
Youcef Belgueliel, Ramdane Maamri
Cheminement d'objets pédagogiques de type quiz p. 67
Ivan Madjarov, Zohra Bakkoury, Abdelkader Betari, Bogdan Shishedjiev

Fiche équipe

L'association GREENTIC p. 76
Aawatif Hayar
L'équipe de recherche Siweb p. 78
Ounsa Roudiès

Editorial du numéro 6

Cette 6^{ème} édition rassemble six articles et propose un dossier spécial consacré à la conférence SITACAM'2011.

La rubrique « **Etat de l'art** » présente deux articles.

Le premier article, proposé par Leila Ben Hiba et Mohamed Abdou Janati Idrissi, analyse les tendances des méthodes de gestion des projets informatiques. Entre l'approche axée-plan traditionnelle et le défi de l'agilité, il discute la pertinence des approches hybrides et détaille les critères de choix d'une méthode.

Dans le second article, Meryem Kassou et Laïla Kjiri se penchent sur l'ingénierie des exigences de la sécurité informatique et proposent une revue critique de leurs élicitation et spécification. Les auteurs établissent un rapprochement entre les visions de deux communautés, à savoir celle du Génie logiciel et celle des qualitatifs.

Le dossier spécial **SITACAM'2011** propose deux articles représentatifs des recherches contribuant au multilinguisme de l'Internet, dans le cas particulier de l'Amazigh.

La reconnaissance des caractères Amazigh (OCR) fait l'objet des travaux de Rachid El Ayachi, Mohamed Fakir et Belaid Bouikhalene qui proposent une solution combinant la transformation de Fourier, la programmation dynamique et les réseaux de neurones.

Le 2^{ème} article du dossier est une contribution de Mohamed Outahajala, Yassine Benajiba, Paolo Rosso et Lahbib Zenkour qui portent sur l'étiquetage grammatical de l'amazighe, traité en utilisant les propriétés n-grammes et un prétraitement de segmentation.

La rubrique « **Recherche et développement** » se compose également de deux articles.

Youcef Belgueliel et Ramdane Maamri décrivent un système d'information voyageur, combinant plusieurs moyens de transport (routier, aérien, etc.). Il optimise l'itinéraire en termes de coût, temps et confort, par le biais d'agents et de la composition de services.

Soumis par Ivan Madjarov, Zohra Bakkoury, Abdelkader Betari et Bogdan Shishedjiev, l'article suivant est une solution e-learning fondée sur les services Web pour l'intégration des données et ressources pédagogiques. En particulier, l'évaluation de type Quizz est implantée dans le cadre du projet XESOP. Une association de recherche, Greentic, est introduite dans la rubrique «Fiche équipe» ainsi que l'équipe de recherche Siweb, Système d'information et Web (EMI, Maroc).

Concernant notre politique d'indexation, elle s'est poursuivie avec l'ajout de EBSCO et de l'IMIST. Pour terminer, merci aux patients auteurs, aux relecteurs qui sont les garants de la qualité ainsi qu'à l'AUUF pour son soutien. Bonne lecture !

Ounsa Roudiès, Mounia Fredj
Rédactrices en chef de la 6^{ème} édition e-TI

Les comités de la 6^{ème} édition de la revue eTI

Directrices de publication

- Ounsa ROUDIES EMI, Maroc
- Dalila CHIADMI EMI, Maroc

Comité scientifique

- BELLAACHIA Abdelghani George Washington University, USA
- GIRAUDIN Jean-Pierre LIG-Sigma, France
- ROLLAND Colette Université Paris-1 Sorbonne, France

Rédactrices en chef du 6^{ème} numéro

- Ounsa ROUDIES EMI, Maroc
- Mounia FREDJ ENSIAS, Maroc

Comité de rédaction

- AHMED-NACER Mohamed USTHB, Algérie
- ALIMAZIGHI Zahia USTHB, Algérie
- AMGHAR Mustapha GIE Galileo, Maroc
- BAINA Karim ENSIAS, Maroc
- BAKKOURY Zohra EMI, Maroc
- BENHLIMA Leila EMI, Maroc
- CHIADMI Dalila EMI, Maroc
- COHEN Atika ULB, Bruxelles
- COULETTE Bernard IRIT, France
- ELEULDJ Mohcine ENSAO, Maroc
- FRONT Agnès LIG-Sigma, France
- GIRAUDIN Jean-Pierre LIG-Sigma, France
- RIEU Dominique LIG-Sigma, France
- ROLLAND Colette Paris1-Sorbonne, France
- TAMZALIT Dalila LINA, Université de Nantes, France

Comité de lecture du 6^{ème} numéro

- BAINA Karim ENSIAS, Maroc
- BAKKOURY Zohra EMI, Maroc
- BELOUADHA Fatima Zahra EMI, Maroc
- BENADADA Youssef ENSIAS, Maroc
- BENHLIMA Leila EMI, Maroc
- CHIADMI Dalila EMI, Maroc
- COHEN Atika ULB, Bruxelles
- DIOURI Ouafae EMI, Maroc,
- EBERSOLD Sophie IRIT, France
- FREDJ Mounia ENSIAS, Rabat
- GIRAUDIN Jean-Pierre LIG-Sigma, France
- MOULINE Salma FSR, UM5A, Maroc
- OULAD HAJ THAMI Rachid ENSIAS, Maroc
- RADGUI Amina FSR, UM5A, Maroc
- ROUDIES Ounsa EMI, Maroc

- TAMZALIT Dalila
- TEMGLIT Nacéra

LINA, Université de Nantes, France
USTHB, Algérie

Comité logistique du 6^{ème} numéro

- BENHLIMA Laïla, mise en page
- ELHANCHI Adnane, infographie
- MOKHTARI Abdelkrim, relecture linguistique
- RAPOSO de BARBOSA Appoline, mise en page et infographie

Partenaires



Etat de l'art

Tendances des méthodes de gestion des projets informatiques

Lamia Ben Hiba

ENSIAS, Université Mohammed V-Souissi, Madinat Al Irfane, Rabat, Maroc
benhiba.lamia@gmail.com

Mohammed Abdou Janati Idrissi

ENSIAS, Université Mohammed V-Souissi, Madinat Al Irfane, Rabat, Maroc
janati@ensias.ma

Résumé

Afin de faire face à la complexité ascendante du développement logiciel, des méthodes traditionnelles de gestion des projets informatiques ont été introduites. Axées-plan et généralement lourdes, ces méthodes ont perdu de plus en plus de partisans en faveur des nouvelles méthodes dites «agiles». Cependant, ces dernières présentent des limitations pour les projets critiques ou ceux avec une équipe de grande taille. Ainsi, ont vu le jour de nouvelles méthodes dites «hybrides», palliant les faiblesses et bénéficiant des forces des deux catégories de méthodes. Cet article présentera différentes approches traditionnelles, agiles et hybrides de gestion de projet informatique, dressera un comparatif entre les méthodes traditionnelles et agiles et enfin discutera du problème du choix de la méthode de gestion à adopter pour un projet donné.

Abstract

Traditional methods were introduced in order to face the ascending complexity of software development. Plan-oriented and generally heavy-weighted, these methods have been losing advocates in favour of new methods called Agile. These methods present certain limitations when it comes to critical projects and big teams. To benefit from both methods' strengths and to palliate their weaknesses, Hybrid methods appeared. This paper aims to present different traditional, agile and hybrid project management methods, provide a comparison between traditional and agile methods and discuss the problem of the choice of the method to adopt for a given project.

Mots-clés

gestion de projet, Pmbok, CMMI, méthodes Agile, ASD, DSDM, Scrum, XP, modèle de Boehm et Turner, modèle de Sidky

Keywords

Project management, Pmbok, CMMI, Agile method, ASD, DSDM, Scrum, XP, Boehm et Turner's model, Sidky's model

1. Introduction

L'industrie logicielle a intégré la société moderne depuis plus de 50 ans. Au début, le développement logiciel n'était que le résultat d'activités désordonnées (*cod and fix*) sans planification ni conception détaillée en amont. Cependant, la complexité croissante des logiciels a rendu difficile l'ajout de nouvelles fonctionnalités ou la réparation des *bugs* systèmes. Ainsi, fallait-il impérativement introduire des méthodes de développement logiciel. Les méthodes sont venues imposer un processus discipliné avec pour objectif de rendre le développement logiciel plus prédictible et efficace.

Les méthodes traditionnelles sont des approches axées-plan qui se basent sur un ensemble d'activités séquentielles. Un des principaux avantages de ce modèle est la réduction des risques grâce à la mise en place de vérification et de validation à la fin de chaque étape. Cependant, l'utilisation reste lourde et contraignante avec notamment une documentation conséquente et difficile à rédiger, ainsi qu'un client exclu des vérifications des différentes phases. Perçue comme rigide et limitant la créativité, l'approche traditionnelle a été remise en cause par de nouvelles méthodes dites agiles, l'objectif étant d'introduire de la flexibilité au sein des organisations. Légères, adaptatives au changement, ces méthodes ont gagné en notoriété. Pourtant, on leur reproche de ne pas être adéquates aux projets complexes ou critiques.

Partant du constat que les méthodes agiles et traditionnelles ne sont pas incompatibles, de nouvelles approches hybrides ont émergé afin d'intégrer les processus agiles dans le cycle de développement traditionnel. Ces approches sont assez récentes (début des années 2000) et elles ont donc encore besoin de perfectionnement. Cependant, la plupart d'entre elles ont contribué à déterminer la disposition des organisations à aller vers l'agilité et ainsi ont permis l'énumération de quelques facteurs influençant le choix des méthodes.

Le présent article expose un état de l'art des méthodes de gestion de projets informatiques. Nous présentons en premier lieu les principes des méthodes traditionnelles en citant quelques méthodes. La deuxième section traite des méthodes agiles, leurs principes et leurs limites en dressant une comparaison entre les principales méthodes existantes. Une comparaison entre les méthodes agiles et les traditionnelles fait l'objet de la troisième section. La quatrième section décrit les approches hybrides. Nous abordons dans la dernière section la question du choix de la méthode adéquate pour la gestion des projets informatiques.

2. Méthodes axées-plan (traditionnelles)

2.1. Principes des méthodes axées-plan

Les méthodes axées-plan sont considérées comme étant le moyen traditionnel pour le développement logiciel. Fondées sur des concepts tirés des principaux domaines de l'ingénierie, ces méthodes perçoivent le développement selon un paradigme de besoins/conception/développement couplé de processus standards et bien définis que l'organisation améliore constamment. Plusieurs cycles de vie ont été créés se fondant sur cette approche. Le plus célèbre est le cycle de vie en cascade introduit par Royce (1970), où les phases de développement sont implémentées (au moins deux fois) l'une après l'autre. Même si ce cycle de vie est considéré comme étant le modèle le plus axé-plan qui existe, Royce suggère que pour réaliser un projet nécessitant une durée d'exécution de 30 mois, on doit d'abord réaliser un projet-pilote de 10 mois, ce qui induit un aspect itératif au modèle. Ainsi, le système et le développement logiciel ne sont pas des processus linéaires. Boehm (1988) conclut que suivre le cycle de vie en cascade conduit à une approche axée-documents, ce qui force les membres de l'équipe développement à écrire des documents inutiles et difficiles à comprendre. Ainsi, Boehm présenta le cycle de vie en spirale, nouveau modèle itératif, pour pouvoir naviguer vers chacune des phases du cycle de développement et faciliter la communication et la confiance entre les membres de l'équipe. Dans les années 90, de plus en plus de méthodes itératives et/ou incrémentales ont vu le jour, mais elles maintenaient une lourde documentation et une traçabilité minutieuse à travers l'analyse des besoins, la conception et le développement.

Dans le chapitre qui suit, nous présentons deux exemples de méthodes axées-plan qui manifestent les aspects des méthodes traditionnelles. Nous commençons par le célèbre guide PMBOK (Project Management Body of Knowledge). Quoiqu'il ne représente pas une méthode au sens strict du terme, ses processus sont considérés, en pratique, comme des étapes de cycle de vie projet et ses domaines de connaissance dévoilent une linéarité commune aux méthodes axées-plan. Notre deuxième exemple est le *framework* CMMI (Capability Maturity Model Integration). Le choix de ce modèle provient du fait qu'il intègre des processus du domaine d'activité « gestion de projet » qui sont souvent implémentés d'une manière linéaire et lourde, affectant ainsi le cycle de vie projet.

2.2. Exemples des méthodes axées-plan

2.2.1. PMBOK

L'ouvrage de référence en gestion de projet traditionnelle est le PMBOK (PMI, 2008). PMBOK est un corpus de connaissances en gestion de projet qui a été internationalement reconnu (IEEE, norme ANSI) et qui décrit un ensemble de principes et de meilleures pratiques « traditionnelles » de gestion de projets applicables à la majorité des projets. PMBOK considère le projet comme étant un ensemble de processus. Un processus étant une série d'actions exécutées par des personnes et aboutissant à un résultat. Les processus de management de projet (39 en tout) s'attachent à 9 domaines de connaissances et peuvent être classés en cinq catégories : démarrage, planification, réalisation, contrôle et clôture du projet. Chaque discipline décrit les pratiques de la gestion de projet en fonction

des processus qui la composent. Ainsi, la discipline de « gestion de la qualité du projet » définira les connaissances et pratiques liées à la planification de la qualité, à l'assurance qualité et au contrôle de la qualité.

PMBOK est devenu le référentiel ultime pour la gestion des projets et ses certifications sont devenues une exigence courante dans le monde anglophone. Cependant, les partisans des nouvelles approches « agiles » reprochent au PMBOK d'être résistant au changement et de produire une documentation excessive (Boehm, 2002). Le PMI (Project Management Institute) rétorque en indiquant que le PMBOK n'est qu'un *framework* ayant pour objectif la satisfaction des clients donnant droit à l'interprétation. Ainsi, de nouveaux efforts sont-ils déployés afin d'arriver à un *mapping* des pratiques agiles dans le contexte du PMBOK, permettant ainsi de bénéficier des forces des deux approches et à pallier leurs faiblesses respectives.

2.2.2. CMMI

La famille CMMI (2006) est un ensemble de normes américaines du SEI (Software Engineering Institute) constituant un *framework* pour l'amélioration des processus de développement logiciel en particulier ceux relatifs au domaine d'activité de gestion de projet. Il est composé de pratiques clés permettant de produire un logiciel de qualité, avec une productivité accrue et dans le respect des budgets et des délais.

CMMI est un modèle qui procède d'une logique d'amélioration progressive sur 5 niveaux de maturité. Il permet d'évaluer la maturité des processus et des pratiques d'une organisation par rapport à ce qu'il faudrait faire idéalement dans les projets. Les 5 niveaux de maturité vont d'un niveau de maturité initial où les projets se déroulent de manière différente selon l'expérience des intervenants, vers la mise en place d'une organisation efficace exploitant les enseignements et accélérant l'optimisation des processus et des outils mis en place dans les précédents niveaux. Chaque niveau du modèle couvre différents domaines d'activité qui sont utilisés dans la réalisation ou le support à la réalisation de produits logiciels. Ces domaines de processus correspondent à un découpage de l'environnement de développement : Gestion des exigences, Planification de projet, *etc.* Le CMMI cadre les domaines de processus en précisant les objectifs à atteindre et les pratiques correspondantes tels que : Planifier les processus, Fournir les ressources, Assigner les responsabilités, Former les personnes *etc.*

Même si le modèle CMM/CMMI jouit d'une grande notoriété dans le milieu professionnel, on lui reproche, comme c'est le cas pour toutes les méthodes traditionnelles, d'instaurer une gestion de projet trop lourde et coûteuse. Les organisations ont cherché à alléger les processus du CMMI en essayant d'y intégrer des pratiques émanant des méthodes « légères » dites aussi agiles.

Les méthodes axées-plan focalisent donc sur une planification détaillée, une codification des processus et une rigueur d'implémentation, ce qui caractérise le paradigme classique de la gestion des projets. Dans la section suivante, nous présentons un 2^{ème} paradigme dit « agile » dont le principe est d'introduire l'agilité et la flexibilité à la gestion des projets.

3. Méthodes agiles

3.1. Principes des méthodes agiles

Les méthodes traditionnelles ont perdu un grand nombre de partisans en raison de leur rigidité et leur inadaptabilité au changement (Conboy et Fitzgerald, 2004). Les praticiens et les chercheurs ont alors eu recours à de nouvelles méthodes légères dites Agiles, avec la principale mission de faire face aux changements et donner plus de valeur au client dans l'élaboration du produit. Ces méthodes ont suscité un grand nombre de publications et de débats (Highsmith, 2001). Cependant, les recherches académiques sur le sujet restent limitées, vu que la plupart des publications sont l'œuvre de praticiens ou de consultants (Abrahamsson *et al.*, 2002).

Les méthodes agiles se sont développées séparément, jusqu'en 2001, où leurs créateurs sont arrivés à un consensus sous le label « Agile ». Les traits essentiels et communs ont alors été formalisés en un manifeste nommé « le manifeste agile » (Manifeste Agile, 2001) prônant quatre valeurs principales :

- I. priorité aux personnes et aux interactions par rapport aux procédures et outils,
- II. priorité aux applications fonctionnelles par rapport à la documentation exhaustive,
- III. priorité de la collaboration avec le client par rapport à la négociation de contrat,
- IV. priorité de l'acceptation du changement par rapport à la planification.

De ces quatre valeurs découlent douze principes généraux communs aux méthodes agiles énumérés dans le tableau 1.

1. Satisfaction du client	7. Auto-organisation
2. Coopération entre le client et les développeurs	8. Communication face à face
3. Adaptation au changement	9. Rythme soutenable
4. Livraisons fréquentes	10. Simplicité
5. Fonctionnement de l'application comme premier indicateur d'avancement du projet	11. Excellence technique (Refactorisation)
6. Construction du projet avec des personnes motivées (<i>Self-empowered team</i>)	12. Introspection

Tableau 1. Principes des méthodes agiles

3.1. Principales méthodes

Il existe une panoplie de méthodes qui incorporent les principes déjà mentionnés. Nous présentons ci-après les méthodes qui ont suscité un grand intérêt à la fois des praticiens et des chercheurs: Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), eXtreme Programming (XP), et Scrum.

3.1.1. ASD

Développée par John Highsmith (2010), cette méthode offre une approche agile et adaptative pour la gestion des projets, avec un taux de changement élevé, en se fondant sur la théorie de la complexité. La planification dans un environnement rapide et imprédictible a beaucoup de risques d'échouer. Ainsi, ASD substitue-t-il le cycle traditionnel de « planifier – concevoir – développer » par le cycle « spéculer – collaborer – apprendre »:

- V. Spéculer ie. reconnaître l'incertitude régnant sur les projets complexes et encourager l'expérimentation et l'exploration ;
- VI. Collaborer c'est-à-dire travailler conjointement (équipe de développement, clients, consultants externes et fournisseurs de logiciels) pour produire de meilleurs résultats et prendre des décisions pertinentes et à temps ;
- VII. Apprendre en testant les connaissances de l'équipe constamment, à la fin de chaque itération, pour devenir une organisation apprenante (et par extension, agile).

3.1.2. DSDM

DSDM a été créé en Angleterre par un consortium dédié (Abrahamsson P. *et al.*, 2002). La première version de la méthode a vu le jour en 1994. L'idée motrice était de fixer le *timebox* et les ressources en premier, puis d'ajuster le nombre de fonctionnalités à leur assigner. Les techniques clés de cette méthode sont :

- VIII. *timeboxing* c'est-à-dire un processus par lequel les objectifs sont réalisés à une date prédéterminée et fixe, de 2 à 6 semaines;
- IX. les règles MoSCoW (*Must, Should, Could, Won't*) classent les fonctions en (i) indispensables (sans quoi le système ne peut fonctionner), (ii) souhaitables (importantes, mais pouvant être contournées si les délais sont contraignants), (iii) possibles (facilement contournées) et (iv) éliminées (à classer lors de futurs développements);
- X. les ateliers Facilités (*facilitated workshops*) sont des séances d'échanges organisées avec les utilisateurs pour la collecte d'informations et la prise de décisions.

En 2007, DSDM Atern a vu le jour. Cette nouvelle version de DSDM apporte une définition plus claire du *framework* et un focus sur la communication (DSDM Consortium, 2007).

3.1.3. Scrum

Introduite par Schwaber et Beedle (2002), Scrum est une approche empirique qui applique la théorie de la gestion des processus industriels dans des environnements volatils au développement des systèmes. L'approche est fondée principalement sur la flexibilité, l'adaptabilité et la productivité. L'idée directrice de Scrum est que le développement des systèmes implique plusieurs variables techniques et environnementales (spécifications, délais, ressources et technologie). Ceci augmente la complexité et l'imprédictibilité, d'où le besoin de flexibilité pour remédier immédiatement à tout changement (Abrahamsson *et al.*, 2002). Les principales caractéristiques de la méthode Scrum sont :

- XI. production organisée par une série de *sprints* (itérations) d'une durée de 2 à 4 semaines,
- XII. exigences définies comme des éléments d'une liste appelée « *backlog* du produit »,
- XIII. Produit (partiel) conçu, codé et testé pendant le *sprint*,
- XIV. équipe choisissant, à partir du *backlog* de produit, les éléments qu'elle s'engage à finir,
- XV. *backlog* de sprint créé collectivement et identifiant les tâches et leur durée estimée (une à 16 heures),
- XVI. réunion quotidienne de 15 minutes organisée pour répondre aux questions: Qu'as-tu fait hier ? Que vas-tu faire aujourd'hui ? Et qu'est-ce qui ne marche pas ?
- XVII. Revue de *sprint* organisée à la fin de chaque *sprint*. L'équipe présente les réalisations ainsi qu'une démonstration des nouvelles fonctionnalités ou de l'architecture,
- XVIII. réflexion régulière sur les bonnes réalisations et les défaillances à la fin de chaque *sprint* (réunion de rétrospective).

3.1.4. eXtreme Programming (XP)

La méthode XP a été mise en œuvre par Kent Beck, Ward Cunningham et Ron Jeffries (1999). Elle est considérée comme une collection de pratiques du génie logiciel venues résoudre les problèmes posés par les longs cycles de développement. Cette méthode tire son nom de son principe de base qui est la réduction du coût des modifications, poussée à l'extrême. Pour atteindre cet objectif, XP s'appuie sur quatre valeurs: simplicité, communication, feedbacks et courage (dans le sens de « sans tabou », si un problème est détecté, il ne doit pas être caché). Ces valeurs (Beck, 1999) sont mises en œuvre au moyen de 12 « pratiques » qui guident le travail de l'équipe XP (Tableau 2).

1. Jeux de planification	7. <i>Pair programming</i> (les programmeurs travaillent en binômes)
2. Livraisons petites et fréquentes	8. Appropriation collective du code
3. Utilisation des métaphores pour l'explication de points techniques, architecturaux, fonctionnels	9. Intégration en continu
4. Simplicité de la conception	10. Rythme durable
5. Tests	11. Client sur site (Un représentant du client est présent pendant toute la durée du projet)
6. Refactorisation fréquente (réusinage fréquent du code)	12. Norme de programmation

Tableau 2. Les 12 pratiques XP

3.2. Comparaison entre les méthodes agiles

De nombreuses études comparatives ont été menées pour ordonner, analyser et comprendre le domaine dispersé des méthodes agiles. On s'intéressera principalement aux méthodes présentées dans le paragraphe précédent. Abrahamson *et al.* (2003) ont dressé une comparaison des méthodes selon quatre aspects: gestion des projets, cycle de vie, « principes abstraits versus orientation concrète » et « universellement prédéfinie versus appropriée à la situation ». Les résultats obtenus pour les méthodes ASD, DSDM, Scrum et XP sont résumés en figure 1.

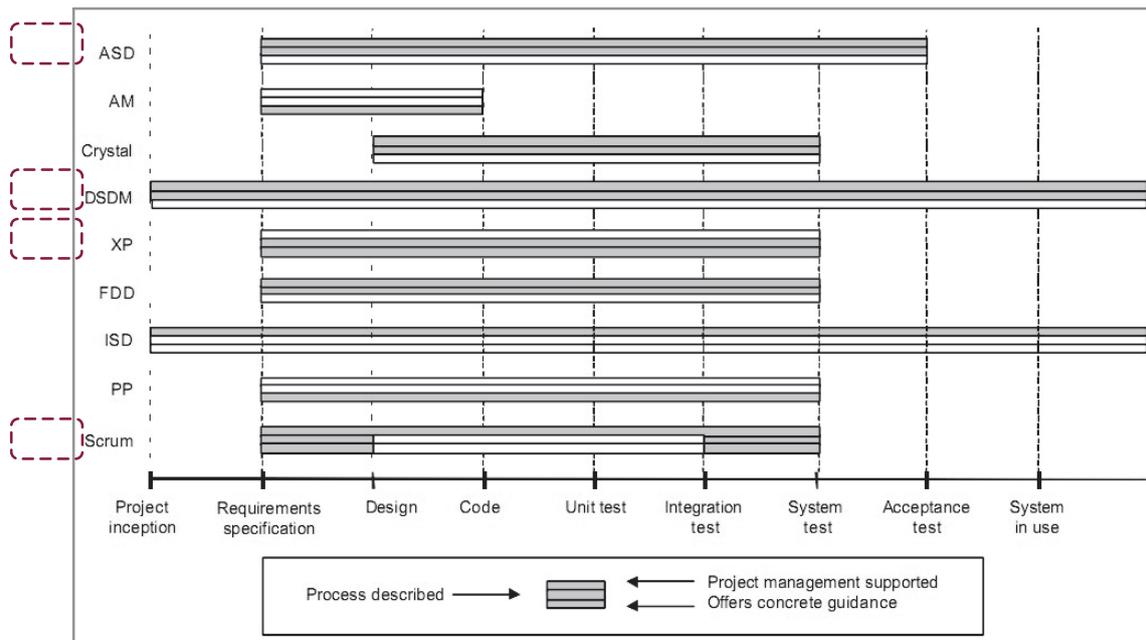


Figure 1. Comparaison des méthodes Agiles. Source: (Abrahamsson et al. 2002)

Chacune des méthodes est symbolisée par trois barres. La 1^{ère} barre indique si la méthode fournit un support pour la gestion des projets. La barre centrale indique si un processus du cycle de vie est couvert par la méthode; sa longueur indique les phases supportées par la méthode. La 3^{ème} barre indique si la méthode est fondée sur des principes abstraits (couleur blanche) ou si elle fournit une orientation concrète (couleur grise). La couleur blanche fait allusion à l'absence de l'aspect analysé tandis que le gris informe que l'aspect est couvert par la méthode.

3.2.1. La gestion des projets

Les méthodes agiles couvrent cet aspect de manière différente. Même si XP a récemment retenu de nouvelles lignes directrices sur cet aspect, il n'offre pas une vue complète de la gestion du projet. Scrum, par contre, est explicitement destiné à la gestion de projets. Schwabber et Beedle suggèrent d'utiliser d'autres méthodes pour intégrer une approche de développement fondée sur Scrum (par exemple XP). ASD stipule que l'équipe projet doit s'adapter en réponse aux changements survenus dans le projet. DSDM offre un *framework* pour améliorer la capacité de l'organisation à réagir au changement, et contribue à la gestion du projet en facilitant le travail de l'équipe de développement avec un suivi quotidien de la progression du projet.

3.2.2. Le cycle de vie

DSDM essaie de couvrir tout le cycle de développement. ASD couvre la plupart des phases sauf l'initiation du projet, les tests d'acceptation et l'utilisation du système. XP et Scrum se concentrent sur la spécification des besoins, la conception et les tests (jusqu'aux tests système).

3.2.3. Principes abstraits versus orientation concrète

Pour qu'une méthode soit utile et efficace, elle doit apporter une orientation concrète aux équipes de développement à travers des pratiques, des activités et des produits qui décrivent comment une tâche devrait être exécutée. ASD apporte plutôt des concepts et une culture. DSDM ne détaille pas de pratiques, mais il insiste sur le fait que toute organisation devrait développer ses propres pratiques. Dérivée d'un ensemble de *best practices*, XP ne préconise que l'orientation concrète. Scrum définit également un ensemble de pratiques à suivre correspondant aux phases couvertes par la méthode.

3.2.4. Universellement prédéfinie versus appropriée à la situation

Afin de pouvoir réagir au changement, les processus de développement utilisés doivent s'adapter aux situations. Une solution universellement prédéfinie propose une solution prête et non ajustable, ce qui limite visiblement la flexibilité de la méthode et donc sa capacité à réagir aux changements. Les méthodes agiles étudiées dans cet article sont toutes ajustables en fonction de la situation.

Suite à cette présentation des principales méthodes agiles, la section suivante dressera une comparaison entre les paradigmes dominants les méthodes agiles et ceux des méthodes traditionnelles de la gestion de projet.

4. Comparaison entre les méthodes agiles et traditionnelles

Les méthodes traditionnelles et les méthodes agiles se sont développées dans deux milieux diamétralement opposés. Les premières sont venues introduire de la discipline dans les grands projets gouvernementaux ayant des coûts d'échec colossaux. Les méthodes agiles, quant à elles, se sont développées dans la « bulle » du dotcom sur des petits projets Web avec des coûts d'échec frôlant le zéro. Les méthodes agiles ont introduit un nouveau paradigme pour pallier les défaillances des méthodes traditionnelles et garantir un meilleur taux de succès des projets.

Cependant, des études empiriques ont démontré que le paradigme agile n'est pas approprié à toutes les situations et que les approches traditionnelles offrent dans certains cas de meilleurs résultats. En effet, l'une des plus grandes limitations des méthodes agiles est la grande taille des équipes. Cockburn et HighSmith (2001) concluent qu'avec l'augmentation de la taille, la coordination entre les interfaces devient un problème crucial. L. Constantine (2001) et M. Fowler (2001) indiquent que la communication face-à-face prônée par les méthodes agiles devient difficile et complexe si l'équipe de développement dépasse les 20 personnes. En revanche, les méthodes traditionnelles axées-plan sont mieux appropriées aux projets à grande échelle.

Une autre limitation considérable des méthodes agiles consiste en la gestion des projets critiques. Les mécanismes de contrôle de qualité incorporés dans les méthodes agiles ne prouvent pas que les produits développés sont sans danger. Les spécifications formelles, les tests rigoureux et d'autres techniques d'analyse et d'évaluation inclus dans les méthodes traditionnelles offrent de meilleurs mécanismes, même si ils sont plus lourds, pour gérer les projets critiques (Turk *et al.*, 2002). Le tableau 3 résume les principales différences entre ces deux approches.

Critères	Méthodes traditionnelles	Méthodes agiles
Approche	prédictive	adaptative
Mesure du succès	conformité au plan	valeur client
Domaine	prévisible	imprévisible/explosatoire
Retour sur investissement	à la fin du projet	au début du projet
Style de management	autocratique	décentralisé
Culture	commande, contrôle	leadership, collaboration
Environnement	stable, faible taux de changement	turbulent, taux élevé de changement
Relation client	interaction au besoin, insistance sur les exigences du contrat	un représentant du client est dévoué au projet, insistance sur la priorisation des incréments
perspective de changement	durabilité de changement	adaptabilité au changement
Focus	processus	humain
Communication	explicite	tacite
Documentation	lourde	faible
Cycles	en nombre limité	nombreux
Planification en amont	complète	minimale
Taille du projet	grande taille	petite taille
Taille de l'équipe	grande taille	petite taille/créative
Développement	conception extensive, longs incréments	conception simple, courts incréments

Tableau 2. Différences entre les méthodes agiles et traditionnelles

On peut supposer alors que les méthodes agiles et traditionnelles ne sont pas incompatibles et qu'elles pourraient être combinées en cas de besoin. Les méthodes formelles pourraient être utilisées d'une manière agile pour traiter les parties critiques du système et offrir une meilleure qualité et un taux tolérable de confiance. La section suivante présentera quelques méthodes dites hybrides qui proposent de profiter des avantages et d'esquiver les inconvénients des deux catégories de méthodes.

5. Méthodes Hybrides

Boehm et Turner (2003) affirment que les méthodes agiles, comme les méthodes traditionnelles, ont des points forts et des points faibles. Il est donc important pour les organisations d'équilibrer les deux approches pour bénéficier de leurs forces et pallier leurs faiblesses. La littérature actuelle n'offre cependant que quelques approches hybrides pour l'amélioration du développement logiciel. Dans le prochain paragraphe, nous présentons trois d'entre elles, à savoir le modèle de Boehm et Turner, le modèle de Sidky et l'intégration du CMMI et des pratiques agiles.

5.1. Le modèle de Boehm et Turner

L'un des premiers modèles à adopter les pratiques agiles est le modèle Boehm et Turner (2003). Ce modèle identifie les risques relatifs au projet et d'indique la méthode à adopter (disciplinée, agile ou une combinaison des deux). Le modèle consiste en cinq étapes clés : Analyse du risque, Comparaison du risque, Analyse de l'architecture, Ajustement du cycle de vie et Exécution et *monitoring*. Boehm et Turner ont identifié cinq facteurs d'agilité représentés par des axes d'analyse (figure 2) qui interviennent dans la sélection des méthodes agiles ou traditionnelles :

- XIX. la taille c'est-à-dire le nombre des membres de l'équipe,
- XX. la criticité du projet,
- XXI. le dynamisme qui traduit le degré de changement des spécifications,
- XXII. le personnel ou degré de compétence et d'expérience de l'équipe,
- XXIII. la culture de l'organisation.

Chaque axe représente une dimension d'agilité. Quand les données relatives au projet sont projetées sur les axes, les différents points sont reliés et les formes obtenues sont analysées. Si la forme est autour du centre, le modèle suggère l'utilisation d'une méthode agile. Une forme qui tend vers les périphéries indique l'usage d'une méthode axée-plan. Toute autre forme suggère un usage combinant les méthodes axées-plan et les pratiques agiles.

Le modèle peut certainement être utilisé comme un mécanisme pour commencer l'évaluation de l'agilité et la sélection d'un projet pilote agile. Cependant, il ne présente pas de détails quant à l'application des pratiques agiles. Le défi majeur des organisations réside dans l'adaptation des pratiques agiles à leur contexte de développement et dans l'application des pratiques les plus appropriées dans le cadre des activités de l'organisation.

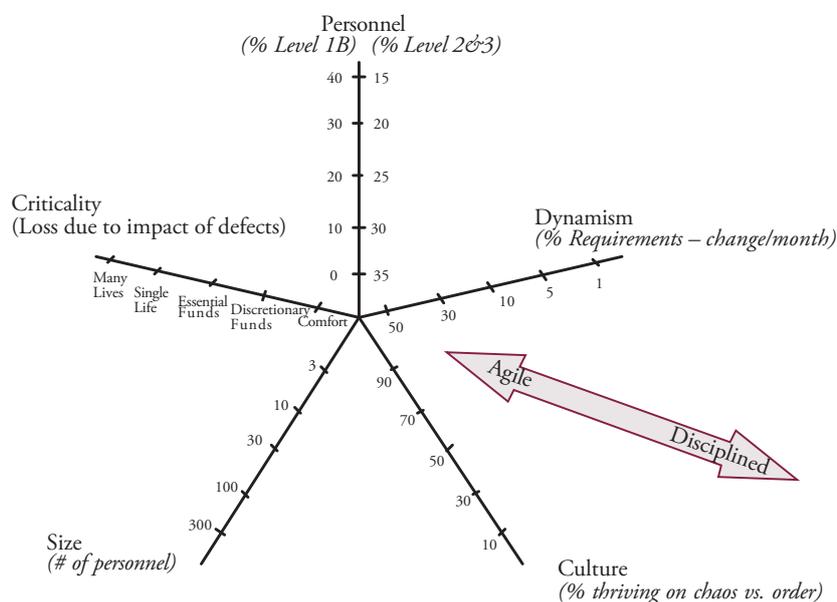


Figure 2. Les 5 facteurs d'agilité de Boehm & Turner. Source : Boehm and Turner (2003).

5.2. Le modèle de Sidky

Ahmed Sidky (2007) présente un *framework* pour l'intégration des pratiques agiles dans le processus de développement traditionnel des projets. Ce *framework* expose des niveaux pour l'évaluation de l'agilité. Comme c'est le cas pour Boehm et Turner, Sidky insiste sur la nécessité d'évaluer le besoin d'une organisation en agile en premier, avant de prendre la décision d'adopter des pratiques agiles. Ceci permet à l'organisation de n'adopter que les pratiques agiles qu'elle a la capacité de concrétiser. Sidky fournit alors un *framework* pour les projets ainsi qu'une évaluation de l'agilité de l'organisation, en quatre étapes :

- XXIV. identification des facteurs de discontinuité. Avant l'évaluation, l'organisation définit sa capacité d'adoption de l'agilité. Les facteurs de discontinuité sont les facteurs qui peuvent prévenir l'organisation de prendre la décision de se lancer dans l'agile. L'absence du soutien de la direction ou un budget insuffisant sont des facteurs de discontinuité;
- XXV. Évaluation du niveau de l'organisation par identification du plus haut niveau d'agilité que le projet pourrait atteindre;
- XXVI. Évaluation de l'inclination de l'organisation à supporter le niveau d'agilité souhaité;
- XXVII. Réconciliation en ressortir les pratiques que l'organisation veut et peut adopter.

L'agilité est évaluée sur une échelle à cinq niveaux : collaboratif, évolutif, efficace, adaptatif et environnant. Cette approche est utilisée dans le modèle IDEAL qui est un modèle d'amélioration des processus logiciels fondé sur SW-CMM. Il est approprié aux situations où les organisations ont recours à l'évaluation fondée sur CMMI. Chaque niveau d'agilité se caractérise par des pratiques agiles particulières.

Le *framework* de Sidky ouvre d'intéressants horizons pour l'adoption de l'agile, mais il reste à prouver sur le plan empirique. Il ne démontre pas pourquoi l'adoption de l'agile doit suivre l'ordre spécifié dans le *framework*. Le *framework* affirme être fondé sur CMM, cependant, il n'offre cependant pas de détails sur les points suivants :

- XXVIII. correspondance des pratiques agiles avec CMMI,
- XXIX. diminution de l'effort en évaluation pour l'adoption des pratiques agiles,
- XXX. validation de l'impact des améliorations sur le processus de développement agile (Pikkarainen, 2008).

5.3. Intégration du CMMI et des pratiques agiles

D'après le SEI, institut ayant développé le CMMI, les pratiques agiles devraient être compatibles avec CMMI, vu que ce dernier représente un cadre pour l'auto-amélioration. L'intégration des pratiques agiles avec celles des méthodes axées-plan a toujours semblé impossible aux praticiens. Ceci est principalement dû aux idées reçues sur l'intégration de l'agile et du CMMI (Glazer *et al.*, 2008). Cependant, plusieurs recherches ont essayé d'intégrer les pratiques agiles dans les processus CMMI. Les stratégies suivies pour cette intégration peuvent être classées comme suit (Alkhatib, 2004) : *front-end*, *back-end*, concurrente, *re-engineering*.

L'approche *front-end* suggère l'usage des pratiques agiles avant l'implémentation de l'approche CMMI dans le système (Boehm, 2002), contrairement à l'approche *back-end* qui stipule qu'une fois le système est développé selon l'approche CMMI, les pratiques agiles pourront être utilisées pour exercer les activités de maintenance.

La stratégie concurrente préconise une implémentation en parallèle des pratiques agiles et des processus CMMI. Un des problèmes majeurs de cette approche est l'affectation des ressources humaines aux différents modules/composants d'un même projet. Les méthodes agiles requièrent des expériences et des rôles différents de ceux des méthodes fondées sur CMM. Pour contourner ce problème, la stratégie concurrente identifie certains modules d'un projet ou des projets en entier, et puis construit en conséquence les équipes qui utiliseront l'agile ou CMMI. La stratégie de *Re-engineering* requiert beaucoup de temps et d'efforts et insiste sur la nécessité de repenser les pratiques du CMMI et des méthodes agiles. Par exemple, CMMI devrait insister sur la collaboration et sur la vitesse couplée avec la qualité; les méthodes agiles devraient générer de la documentation au-delà du code, *etc.*

Suite à cette présentation des méthodes traditionnelles, agiles et en fin hybrides, la section suivante est consacrée au choix de la méthode de gestion la plus appropriée pour un contexte de projet donné.

6. Choix d'une méthode

Les méthodes agiles promettent, contrairement aux méthodes traditionnelles, de garantir une meilleure satisfaction client, un taux inférieur de défauts, des livraisons rapides et une solution aux changements des spécifications client (Boehm and Turner, 2003). Cependant, ces méthodes ne sont appropriées qu'aux petits projets à faible complexité. Les méthodes hybrides sont alors apparues pour relever le défi d'équilibrer les deux approches, agiles et traditionnelles. R. Glass (2002) confirme qu'il n'existe pas une unique méthode pour la construction des logiciels, mais que les méthodes devraient être choisies selon les problèmes posés. Des problèmes différents requièrent des méthodes différentes. Cependant, les recherches effectuées dans ce domaine restent insuffisantes (Glass 2004). Glass suggère en effet que de nouvelles recherches sont effectuées ayant pour objectifs de :

- XXXI. établir une taxonomie des méthodes disponibles,
- XXXII. établir une taxonomie du *spectrum* des domaines des problèmes,
- XXXIII. effectuer un *matching* entre les deux taxonomies.

Glass ne nie pas les difficultés que revêtent les deux premières étapes, mais affirme que praticiens et chercheurs doivent se pencher ensemble sur la classification des méthodes et des problèmes. La même observation a été

introduite par Dyba et Dingsoyr (2008) qui conclurent qu'il y a un grand besoin en recherches pouvant déterminer les situations où les conseils des praticiens peuvent être convenablement implémentés et réduire ainsi l'écart entre la recherche et l'industrie du logiciel. En effet, les praticiens organisent fréquemment des débats pour essayer de résoudre la question : agile ou traditionnel ? La décision d'adopter une méthode de développement n'est pas facile, d'autant plus qu'en théorie, comme en pratique, le focus s'est fait sur la description des méthodes et non sur les facteurs influençant le choix de ces méthodes (Dyba et Dingsoyr, 2008). La recherche doit donc répondre au problème de l'optimalité du choix de la méthode de développement en considérant entre autres le projet, l'équipe et l'organisation.

Parmi les rares travaux qui ont pour objectif d'étudier les facteurs influençant le choix d'une méthode, citons A. Cockburn (2000) qui identifie trois facteurs : la taille de l'équipe, la criticité du système et les particularités du créateur de la méthode (partant du fait que la méthode est une prévention contre les craintes de son créateur).

Les deux premiers facteurs semblent faire unanimité chez les chercheurs. Glass (2001) quant à lui, ajoute à ces deux facteurs, le domaine d'application (scientifique ou d'ingénierie) et le degré d'innovation que nécessite le projet.

Le modèle de Boehm et Turner (2003) suggère une analyse des projets selon cinq axes : taille de l'équipe et du projet, criticité, personnel, culture de l'organisation (être à l'aise dans l'ordre ou dans le chaos) et dynamisme (pourcentage du changement des spécifications par mois).

Dans une étude plus globale menée par J. McAvoy et D. Sammon (2005), onze facteurs ont été identifiés et regroupés en quatre catégories : projet, équipe, client et organisation (tableau 3).

Catégorie	Facteurs critiques d'adoption
Projet	- Durée du projet - Acceptation du changement - Criticité du projet
Équipe	- Taille de l'équipe - Compétences des membres de l'équipe
Client	- Localisation du client - Implication du client
Organisation	- Structure du système d'organisation et de <i>reporting</i> - Processus - Exigence en documentation - Aménagement des lieux du travail

Tableau 3. Facteurs critiques de McAvoy & Sammon (2005)

Le défi que pose le choix d'une méthode est son caractère longitudinal (McAvoy et Butler, 2009). En effet, cette décision inclut de multiples autres décisions prises sur une longue période. Il faudrait donc évaluer constamment la pertinence du choix de la méthode et affiner ainsi les mesures des facteurs, au fur et à mesure de l'exécution des projets.

7. Conclusion

Il va sans dire que les méthodes agiles ont introduit un nouveau paradigme dans la gestion des projets informatiques. Venues contrer une approche axée-plan, lourde, mise en oeuvre pour gérer initialement les grands projets gouvernementaux, ces méthodes ont gagné de plus en plus de partisans au fil des années. Leur caractère léger, flexible et adaptatif aux changements ainsi que leur impact positif et indéniable sur la productivité des équipes, leur ont permis de s'imposer dans le milieu professionnel (Laanti *et al.*, 2011).

Cependant, l'approche agile présente des lacunes que seul un appariement avec les méthodes traditionnelles pourrait combler. Ainsi sont nées les méthodes hybrides qui tentent de « réconcilier » l'agile et le traditionnel, en intégrant un ensemble de pratiques agiles dans les processus de développement traditionnels. Les travaux récents de recherche se sont orientés vers le *mapping* entre les processus formels (émanant des standards tels que CMMI ou PMBOK) et les pratiques agiles. Cette combinaison s'est révélée une approche viable pour une meilleure qualité et productivité (Magdalena *et al.*, 2012). Il est pourtant important d'évaluer la disposition d'une organisation ou d'un projet à accepter l'agilité avant de pouvoir adopter ces pratiques. L'évaluation de l'amélioration des processus suite à l'adoption de l'agile est un deuxième volet qui doit être impérativement traité. Les méthodes formelles ayant parcouru un long chemin dans la voie de l'évaluation, peuvent être utilisées comme base d'un modèle d'évaluation

de l'amélioration des processus de développement. Il est pourtant primordial de garder en perspective l'aspect flexible que viennent introduire les pratiques agiles et d'alléger alors ces modèles d'amélioration. Ainsi, la mise en place d'un *framework* d'assistance et d'évaluation pour le choix et l'adoption d'une méthode (agile ou hybride), fondé sur un modèle traditionnel (CMMI ou PMBOK) semble constituer une problématique qui devrait faire l'objet de recherches à l'avenir.

Références

- Royce W. (1970). Managing the Development of Large Software Systems. *IEEE WESCON Conference*.
- Boehm B. (1988). A Spiral Model of Software Development and Enhancement. *Computer*, Vol. 21, No. 5, 61-72.
- PMI. (2008). *Guide to the project management Body of knowledge*. (4th Edition). Pennsylvania: PMI Inc.
- Boehm B. (2002). Get Ready For The Agile Methods, With Care. *IEEE Computer*, Vol. 35, No. 1, 64-69.
- CMMI Product Team. (2006) *CMMI® for Development*. (Version 1.2). Pittsburgh: Carnegie Mellon University.
- Conboy K., Fitzgerald B. (2004). Toward a conceptual framework of agile methods: A study of agility in different disciplines. *Proceedings of the ACM Workshop on Interdisciplinary Software Engineering Research, Newport Beach, CA, 2004*. 37-44.
- Highsmith J. (2001) Opening Statement. *Cutter IT Journal, The Great Methodologies Debate: Part 1*. Décembre 2001, Vol. 14, No. 12
- Abrahamsson P., Salo O., Ronkainen J., Warsta J. (2002) *Agile software development methods – Review And Analysis*. VTT Publication ESPOO. Julkaisija – Utgivare
- Manifeste Agile (2001) www.agilemanifesto.org
- Highsmith J. (2000). Retiring lifecycle dinosaurs. *Software Testing & Quality Engineering (STQE) magazine*. 2, 4 Juillet/Août, 22-28.
- DSDM Consortium (2007). DSDM Atern, www.dsdm.org/atern/
- Beck K. (1999). Embracing change with Extreme Programming. *IEEE Computer*, Issue 10 October 1999, Vol. 32, 70-77.
- Abrahamsson P., Warsta J., Siponen M.T., Ronkainen J. (2003) New directions on Agile Methods: A comparative Analysis. *Proceedings de la IEEE International conference on Software Engineering*. Portland: 2003.
- Cockburn A. and Highsmith J. (2001), Agile Software Development: The People Factor. *Computer*, Novembre 2001, 131-133.
- Constantine L. (2001), Methodological Agility. *Software Development*, Juin 2001, 67-69.
- Fowler M. (2001), *Is Design Dead ? Extreme Programming Explained*. Boston : Addison Wesley Longman.
- Turk R., France R., Rumpel B. (2002). Limitations of Agile Software Processes. *Proceedings de la 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering—XP2002, Alghero, 2002*.
- Boehm B., Turner R. (2003). Observation on Balancing discipline and agility. *Proceedings of Agile Development Conference, Salt Lake City, 2003*.
- Sidky A., (2007). *A Structured Approach to Adopting Agile Practices: The Agile Adoption Framework*. Mémoire de doctorat, Virginia Tech.
- Pikkarainen M. (2008). Towards a Framework for Improving Software Development Process Mediated with CMMI Goals and Agile Practices. *VTT Publications 695*.
- Glazer H., Dalton J., Anderson D., Konrad M., Shrum S., (2008) CMMI® or Agile: Why Not Embrace Both! <http://www.sei.cmu.edu/publications/>
- Alkhatib G I. (2004) Agile methods – CMMI-SW: linking the two extremes, *Proceedings of the Second Conference on Administrative Sciences*, Dhahran, Avril 2004, 461-472.
- Glass R.L. (2002). Searching for the Holy Grail of software engineering. *Communication the ACM*, Mai 2002, Vol. 45, No. 5.
- Glass R.L. (2004) Matching Methodology to Problem domain. *Communication of the ACM* Mai 2004, Vol. 47, No. 5
- Dyba T., Dingsoyr T. (2008) Empirical studies of agile software development: A systematic review. *Inform. Softw. Technol.*, vol. 50, no. 9-10, 833-859.
- McAvoy J., Sammon D. (2005) Agile methodology adoption decisions: An innovative approach to teaching and learning. *Journal of information and systems Education 2005 Vol. 16(4)*.
- Cockburn A. (2000). Selecting a project's methodology. *IEEE Software* Juillet/Août 2000, 64-71.
- Glass R. L. (2001). Agile Versus Traditional: Make Love, Not War! *Cutter IT Journal* Décembre 2001. V. 14, N. 12: 12-18.
- Boehm B., Turner R. (2003). *Balancing Agility and discipline: A guide for the perplexed*, Boston: Addison-Wesley.
- McAvoy J., Butler T. (2009) The role of project management in ineffective decision making within Agile Software development projects. *European Journal of Information Systems*. V. 18, N 4, Août 2009.
- Laanti M., Salo O., Abrahamsson P. (2011) Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *The Journal of Systems and Software*. V. 53. N.3: 276-290
- Magdalena A. M., Wernera C. M. L., Mendes de Araujo R. (2012) Reconciling software development models: A quasi-systematic review. *The Journal of Systems and Software*. V. 85, N. 2: 351-369

Ingénierie des exigences de la sécurité informatique

Revue de travaux de recherche de l'élicitation à la spécification

Meryem Kassou

ENSIAS, Université Mohammed V-Souissi, Madinat Al Irfane, Rabat, Maroc
meryem.kassou@gmail.com

Laila Kjiri

ENSIAS, Université Mohammed V-Souissi, Madinat Al Irfane, Rabat, Maroc
kjiri@ensias.ma

Résumé

L'expansion des réseaux d'un côté et l'importance de l'information de l'autre côté ont fortement contribué à ce que la sécurité joue un rôle important dans le développement d'applications. Cependant, les exigences de sécurité ont pendant longtemps été reléguées à la deuxième place comparativement aux exigences fonctionnelles, ce qui peut endommager les applications développées.

L'un des défis de la recherche actuelle est donc de développer des méthodes d'expression et de modélisation des exigences de sécurité qui sont expressives, simples, fiables et non coûteuses.

Cet article a pour but de passer en revue des travaux de recherche relatifs aux approches d'élicitation et de spécification des exigences de sécurité dans le contexte d'un processus de développement logiciel. Il y est question également de leur analyse, de leur classification en catégories et de la mise en évidence de leur contribution à l'ingénierie des exigences de sécurité ainsi que de leurs limites.

Abstract

Nowadays Security plays an important role in the development of applications because of network expansion on one side and of the importance of information on the other side. However, security requirements have long been relegated to second place compared to the functional requirements which can cause some damage to the applications.

One of the challenges of current research in the area of requirement engineering is to develop simple, expressive, reliable and costless methods for expressing and modelling security requirements.

The purpose of this article is to review research studies related to approaches for eliciting and specifying security requirements in the context of software development process. It aims then to analyse and classify them into categories and to highlight their contribution to security requirements engineering field and their limits.

Mots-clés

Ingénierie des exigences, Sécurité, Elicitation, Spécification formelle, Framework de modélisation

Keywords

Requirement Engineering, Security, Elicitation, Formal Specification, Modelling framework

1. Introduction

L'Ingénierie des exigences est la première étape fondamentale dans tout projet de développement d'application. Les méthodes de développement logiciel divisent les exigences logicielles en deux catégories : les exigences fonctionnelles et les exigences non fonctionnelles (Glinz, 2005). Si les exigences fonctionnelles décrivent les fonctions que l'application doit accomplir, les exigences non fonctionnelles se concentrent sur la manière dont ces fonctions sont exécutées. La sécurité est considérée comme une exigence non fonctionnelle.

Ces dernières années, l'expansion des réseaux d'une part et l'importance de l'information d'autre part ont fortement contribué à ce que les exigences de sécurité jouent un rôle clé dans le développement d'applications (Firesmith, 2003b). Cependant, les exigences de sécurité ont pendant longtemps été reléguées à la deuxième place comparativement aux exigences fonctionnelles (Matoussi et Laleau, 2008).

En effet, intégrer la sécurité au sein d'un système a toujours commencé par l'identification des besoins de sécurité après la conception de l'application. Cela pose un problème de cohérence et de cohabitation entre les exigences fonctionnelles et les exigences de sécurité. En outre, l'implémentation de l'application peut inclure des mécanismes de sécurité qui ne sont tout simplement pas nécessaires. Par ailleurs, la mise en œuvre des mécanismes de sécurité peut empêcher le fonctionnement de l'application (Giorgini, Massacci, *et al.*, 2005).

L'un des défis de la recherche actuelle est donc d'intégrer les exigences de sécurité avec le processus standard des exigences. Ainsi, les exigences de sécurité peuvent-elles être formulées et intégrées dans la conception des systèmes à un haut niveau d'abstraction en même temps que les exigences fonctionnelles. De cette manière, il devient possible de développer des applications sécurisées qui soient conçues dans le but de prévenir les violations d'une politique de sécurité (Giorgini, Massacci, *et al.*, 2005).

Plusieurs travaux sont consacrés aux méthodes d'expression des exigences de sécurité à un haut niveau d'abstraction. Nous nous intéressons dans cet article aux travaux de recherche sur les approches d'élicitation et de spécification des exigences de sécurité, dans le contexte d'un processus de développement logiciel.

Nous définirons donc, dans la première section de cet article, le processus d'ingénierie d'exigences ainsi que les particularités des exigences de sécurité. Ensuite, nous nous intéresserons à deux phases de ce processus appliqué au contexte de la sécurité, à savoir l'élicitation et la spécification des exigences de sécurité. A cet effet, nous passerons d'abord en revue, dans la deuxième section, des travaux de recherche sur les techniques d'élicitation des exigences de sécurité et expliquer pourquoi elles sont indissociables des méthodes de spécification. Puis, en section 3, nous focaliserons sur les approches de spécification des exigences de sécurité qui seront classées par catégorie. Notre travail se termine par une synthèse des approches de spécification des exigences de sécurité qui fait ressortir leur contribution à l'ingénierie des exigences de sécurité ainsi que leurs faiblesses.

2. Définitions et contexte

2.1. Exigence et processus d'ingénierie des exigences

Selon IEEE Std. 1233-1998 (1998), une exigence est une « condition ou capacité qui doit être remplie ou possédée par un système ou un de ses composants pour satisfaire à un contrat, une norme, une spécification ou tout document imposé de façon formelle ».

Une exigence formalisée doit être abstraite (elle est indépendante de la méthode de mise en œuvre), non ambiguë (elle est énoncée de manière à n'être interprétable que d'une seule manière), traçable (il est possible d'établir une relation entre la déclaration précise des besoins du client et les énoncés spécifiques de la définition du système) et vérifiable (elle doit offrir un moyen de prouver que le système satisfait à son énoncé).

Une exigence formalisée est en fait le résultat du processus d'Ingénierie des Exigences qui est la première étape importante dans le développement d'une application. Ce processus inclut plusieurs phases dont le découpage diffère selon plusieurs définitions (Nuseibeh et Easterbrook, 2000), (Kotonya et Sommerville, 1998). Cependant, typiquement un processus d'ingénierie d'exigences est un processus itératif qui est constitué des phases d'Elicitation, d'Analyse, de Spécification, de Validation et de Gestion.

L'élicitation des exigences consiste en la collecte et le développement d'exigences à partir d'une variété de sources et de parties prenantes. L'analyse se focalise sur l'examen, la compréhension des exigences élicitées et leur vérification pour la qualité en termes d'exactitude, de complétude, de clarté et de consistance. La spécification n'est autre que l'enregistrement et la documentation des exigences de sorte que celles-ci soient utilisables par les parties prenantes, et en particulier, par les développeurs qui doivent concevoir et construire le système. Enfin, la validation est la confirmation de la qualité des exigences et de leur conformité aux besoins des parties prenantes. La gestion, quant à elle, est exécutée tout au long du processus d'ingénierie des exigences. Elle inclut les activités de contrôle de l'évolution et de changement des exigences, de contrôle de version, de traçabilité des exigences et les statuts des exigences.

2.2. Particularités des exigences de sécurité

L'expression d'une exigence de sécurité se base sur l'analyse des biens et des services qui doivent être protégés et sur l'analyse des menaces de sécurité desquelles ces biens et services doivent être protégés (Firesmith, 2003a).

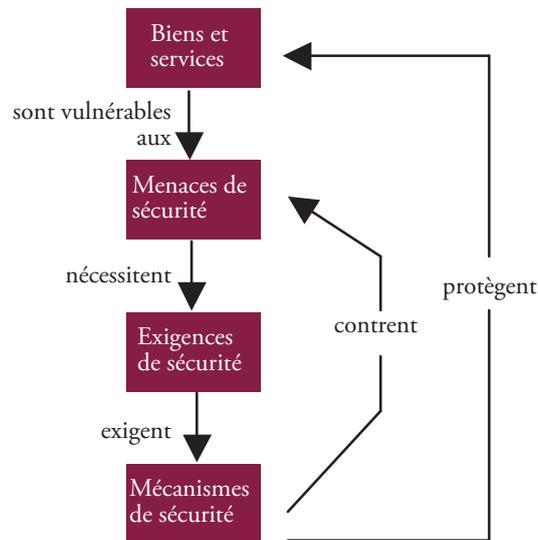


Figure 1. Menaces, exigences et mécanismes de sécurité. Source: (Firesmith, 2003a)

La Figure 1 illustre bien la relation qui existe entre biens et services, menaces de sécurité, exigences de sécurité et mécanismes de sécurité et souligne l'importance des exigences de sécurité.

L'ingénierie des exigences de sécurité a suscité chez la communauté de recherche un intérêt particulier en raison des pertes causées par des applications mal sécurisées, du fait que la sécurité n'était pas intégrée dès le début du cycle de développement ou qu'elle était mal spécifiée.

Selon Van Lamsweerde (2004), il y a trois raisons qui expliquent ce fait :

- tout d'abord, les premières phases du développement des logiciels donnent la priorité à l'élaboration des exigences fonctionnelles au détriment des exigences non fonctionnelles telles que la sécurité en vue d'obtenir un produit qui fonctionne en un court laps de temps.
- La deuxième raison est l'absence d'un mécanisme constructif et efficace pour l'élaboration d'exigences de sécurité d'une manière complète, cohérente et claire.
- La troisième raison est l'absence d'une approche précise et bien définie qui permet de produire la conception et l'implémentation des exigences de sécurité tout en assurant une prise en charge correcte de toutes les exigences et en permettant une traçabilité des exigences lors des différentes phases de développement.

Par ailleurs, selon la définition d'une exigence formalisée fournie par (IEEE Std. 1233-1998, 1998) et dans le contexte de la sécurité, certaines difficultés paraissent évidentes :

- afin d'être indépendante de la méthode de mise en œuvre, l'exigence de sécurité doit être spécifiée durant les premières étapes de développement.
- Une exigence de sécurité, rédigée dans un langage informel, peut s'avérer incohérente et ambiguë et donc difficilement vérifiable.
- Le langage de spécification de l'exigence de sécurité doit permettre une traçabilité très forte entre les phases de spécification, de conception puis d'implémentation.

L'un des défis de la recherche consacrée aux exigences de sécurité a été de développer des méthodes et des langages qui permettent de spécifier les exigences de sécurité de manière claire, expressive et non ambiguë.

Par ailleurs, l'exigence de sécurité est souvent liée à un environnement technique qui restreint le recueil des besoins à un jargon de spécialistes de sécurité et limite l'apport des autres parties prenantes en l'absence de méthodes pour guider l'opération d'identification des exigences de sécurité.

Dans la suite de cet article, nous avons jugé important de passer en revue aussi bien les approches d'élicitation des exigences de sécurité que les approches de spécification de ces exigences.

3. Elicitation des exigences de sécurité

Plusieurs techniques permettent de guider l'opération d'identification des exigences de sécurité. Nous les avons passées en revue et avons sélectionné celles qui, selon nous, sont les mieux structurées et les plus directives (nous n'avons pas mentionné par exemple les techniques basées sur les séances de brainstorming ni celles basées sur les interviews). Nous proposons de les classer selon les catégories suivantes : approches basées sur les modèles (Hadavi, Hamishagi, *et al.*, 2008), approches basées sur les scénarios (Giorgini, Massacci, *et al.*, 2005) et approches basées sur les meilleures pratiques.

3.1. Approches fondées sur les modèles

Ces approches utilisent des modèles et des patrons comme outil de connaissance ou comme support pour produire des exigences de sécurité.

Les modèles d'attaque (*attack pattern*) (Barnum et Sethi, 2006) constituent un outil de connaissance qui aide à la conception, au développement et au déploiement des logiciels sécurisés. Ils découlent de la notion de « *design pattern* » appliquée dans un contexte « destructif » plutôt que « constructif ». En effet, les design patterns aident à résoudre les problèmes récurrents qui peuvent être rencontrés lors du développement d'applications. Les modèles d'attaque permettent aux développeurs d'acquérir une meilleure connaissance des attaques possibles (en se mettant à la place d'un attaquant) et des meilleures méthodes pour contrer ces attaques.

Le *System Security Engineering - Capability Maturity Model* (SSE-CMM) est un modèle qui permet d'évaluer la maturité des processus de sécurité d'un système d'information à travers un barème de cinq niveaux (Phillips, 2003). Ce modèle permet de produire quelques lignes directrices pour la production d'exigences de sécurité.

3.2. Approches fondées sur les scénarios

Ces approches sont dérivées des *use case* et permettent, à travers des cas d'utilisation dans le cadre de la sécurité, de déduire des exigences de sécurité.

Les *Misuse case* et les *Abuse Case* sont une extension conceptuelle des *use case* décrivant les actions qui ne doivent pas être exécutées, car elles pourraient engendrer des dommages aux ressources du système et aux parties prenantes. Ils présentent une solution pour l'obtention des exigences de sécurité (Sindre et Opdahl, 2001), (McDermott et Fox, 1999).

Use Case: Integrity			
Use Case Path: System Message Integrity			
Security Threat: A misuser corrupts a message that is sent from the system to a user.			
Preconditions: 1) The misuser has the means to intercept a message from the system to a user. 2) The misuser has the means to modify an intercepted message. 3) The misuser has the means to forward the modified message to the user.			
User Interactions	Misuser Interactions	System Requirements	
		System Interactions	System Actions
		The system shall send a message to a user.	The system shall ensure that modifications to the message will be obvious to the user.
	The misuser intercepts and modifies the system's message and forwards it on to the user.		
The user receives the corrupted message.			The system shall recognize that its message was corrupted.
		The system shall notify the user that its message was corrupted.	
Postconditions: The system shall have notified the user that the system's message was corrupted.			

Figure 2. Exemple de Security Case. Source : (Firesmith, 2003a)

Les *Mitigation case* sont également utilisées pour exprimer des exigences de sécurité. Les *Mitigation case* permettent de déterminer les exigences et les contre-mesures qui doivent être prises contre les scénarios offensifs des *Misuse case*. Dans ce processus, les *use case* et *Misuse case* principaux sont d'abord produits. Ensuite, chacun de ces cas est divisé en *Small case*. Pour chaque nouveau cas de *Misuse case*, un cas *Mitigation case* est présenté comme une exigence (Hadavi, Hamishagi, *et al.*, 2008).

Firesmith (2003a) considère que l'utilisation des *Misuse case* est plus efficace pour analyser les menaces que pour définir les exigences de sécurité. Il définit les *Security case* qui permettent de déterminer les exigences de sécurité en se basant sur les *Misuse case*.

Les Anti-exigences (*anti-requirements*) permettent d'exprimer les exigences relatives aux utilisateurs malveillants ou pirates. Une Anti-exigence est satisfaite lorsque la menace imposée par un attaquant met le système en une ou plusieurs catégories de risques (Hadavi, Hamishagi, *et al.*, 2008).

Pour illustrer l'approche basée sur les scénarios pour l'élicitation d'exigences de sécurité, nous avons extrait de (Firesmith, 2003a) un exemple concret de *Security use case* présenté en figure 2.

Ce scénario décrit les exigences de sécurité déduites du cas d'un utilisateur malveillant qui corrompt un message envoyé du système à l'utilisateur.

3.3. Approches fondées sur les meilleures pratiques

Ces approches se fondent sur l'expérience de plusieurs équipes pour fournir une liste d'exigences de sécurité.

CLASP, *Comprehensive, Lightweight Application Security Process*, (Viega, 2005) est une méthode structurée pour l'extraction des exigences de sécurité dès les premières étapes de développement. Elle s'est fondée sur l'expérience de plusieurs équipes de développement et sur la compilation de *best practices* pour créer un ensemble complet d'exigences de sécurité.

Les normes de sécurité, fondées sur les meilleures pratiques comme la norme ISO 17799/27001, sont d'autres ressources appropriées pour le développement des exigences de sécurité. Elles peuvent être utilisées pour capturer les exigences de sécurité en utilisant des politiques d'organisation (Su, Bolzoni, *et al.*, 2006).

Approches d'élicitation	Description	Points forts	Points faibles
Approches fondées sur les modèles	Elles utilisent des modèles et des patrons comme outil de connaissance ou comme support pour produire des exigences de sécurité	<ul style="list-style-type: none"> – Base de connaissance (réutilisation, richesse) – Exigences de sécurité détaillées 	Trop spécialisées
Approches fondées sur les scénarios	Elles sont dérivées des <i>use case</i> et permettent, à travers des cas d'utilisation dans le cadre de la sécurité, de déduire des exigences de sécurité	<ul style="list-style-type: none"> – Faciles à utiliser par les parties prenantes – Complets (élicitation, analyse, spécification) 	Exigences de sécurité de haut niveau
Approches fondées sur les meilleures pratiques	Elles se fondent sur l'expérience de plusieurs équipes pour fournir une liste d'exigences de sécurité	<ul style="list-style-type: none"> – Standards – Support facile à l'élicitation (<i>checklist</i>) 	Exigences de sécurité de haut niveau

Tableau 1. Caractéristiques des approches d'élicitation des exigences de sécurité

Nous reprenons dans le tableau 1 les approches d'élicitation des exigences de sécurité présentées dans cette section et précisons les points forts et les points faibles que nous avons relevés pour chacune d'elles.

Comme la phase d'élicitation est étroitement liée aux autres phases du processus d'ingénierie des exigences de sécurité et particulièrement à la phase de spécification, les approches utilisées pour l'élicitation des exigences de sécurité et celles utilisées pour leur spécification sont dans une grande mesure interdépendantes (Nuseibeh et Easterbrook, 2000).

4. Spécification des exigences de sécurité

4.1. Critères de classification et d'analyse

La spécification d'une exigence de sécurité utilise des langages et des outils d'expression et de modélisation des exigences de sorte que celles-ci soient utilisables par les parties prenantes, et en particulier, par les développeurs qui doivent concevoir et construire le système. Plusieurs travaux ont proposé des approches qui diffèrent les unes des autres.

Certains travaux de recherche ont utilisé les *frameworks* de modélisation des exigences (fonctionnelles et non fonctionnelles) pour le cas particulier des exigences de sécurité. D'autres travaux ont étendu ces *frameworks* avec des attributs de sécurité. Par ailleurs, certains résultats de recherche présentent des *frameworks* construits à partir de l'utilisation combinée de méthodes formelles et semi-formelles.

Dans ce qui suit, nous présentons et analysons ces différentes contributions.

4.2. Utilisation des *frameworks* de modélisation existants

Dans cette approche, la méthode de spécification utilise les *frameworks* disponibles comme i*/tropos pour modéliser certaines exigences de sécurité. Les fonctionnalités d'analyse du *Framework* sont ensuite utilisées pour la conception et la mise en œuvre.

Certains travaux ont été réalisés selon cette approche. Liu, Yu, *et al.* (2003) utilise i*/tropos pour modéliser les exigences de sécurité et de confidentialité et pour analyser les dépendances et ainsi vérifier si le système est sécurisé. Tropos est une méthodologie de développement pour construire des systèmes logiciels orientés agents. Elle utilise les concepts d'acteur, d'objectif, de plan, de ressources et de dépendance intentionnelle pour définir les obligations des acteurs (*dependees*) à d'autres acteurs (*dependers*). Tropos accorde beaucoup d'attention aux premières exigences en soulignant la nécessité de comprendre pourquoi et comment le système cible répondrait aux objectifs organisationnels.

He et Anton (2003) présentent un *framework* orienté objectif pour la modélisation des exigences de confidentialité dans le processus d'ingénierie des rôles. L'objectif de ce *framework* est de combler le fossé entre les pratiques de confidentialité des entreprises et les préférences des utilisateurs qui peuvent être en conflit. Les exigences de confidentialité sont modélisées comme des contraintes relatives aux permissions et aux rôles.

Le NFR Framework (*Non Functional Requirement Framework*) est un *framework* de modélisation d'objectifs (Mylopoulos, Chung *et al.*, 1999). Il analyse les *Soft goals* qui sont des exigences non fonctionnelles (dont la sécurité) difficiles à exprimer et sur lesquelles les parties prenantes se sont mises d'accord. Le *NFR Framework* explicite la relation entre les Exigences Non Fonctionnelles (ENF) et les décisions de conception. Cela permet de mieux comprendre l'impact de chaque décision de conception. L'intérêt principal de ce *framework* est qu'il peut être réutilisé par d'autres modèles pour manipuler les ENF (Matoussi et Laleau, 2008).

Analyse:

L'approche d'utilisation des *frameworks* disponibles a deux avantages (Giorgini, Massacci, *et al.*, 2005). D'une part, elle est peu onéreuse, car ni un nouveau langage ni un nouveau Framework pour la modélisation et l'analyse ne sont nécessaires, et d'autre part, toutes les caractéristiques du Framework de modélisation sont immédiatement utilisables. Si le Framework est équipé d'une sémantique formelle et de procédures de raisonnement formelles, ces dernières seront également héritées.

L'inconvénient majeur de cette approche est qu'elle ne répond pas à un large éventail d'exigences de sécurité. En effet, ne disposant pas de structures conceptuelles adéquates, elle ne permet pas, par exemple, de spécifier des contraintes d'autorisation ou de confidentialité. Par ailleurs, le lien entre les exigences de sécurité et les exigences fonctionnelles doit être introduit par le concepteur par des prédicats ou par des relations.

4.3. Amélioration (extension) des *frameworks* existants

Selon cette approche, les *frameworks* de modélisation sont améliorés avec une nouvelle structure conceptuelle (syntaxe et sémantique) afin de répondre à un plus large éventail d'exigences de sécurité et afin de transmettre des exigences de sécurité plus complètes et plus exactes pour l'analyse du système (Hadavi, Hamishagi *et al.*, 2008) (Giorgini, Massacci *et al.*, 2005). Par conséquent, les fonctionnalités d'analyse, de conception et de mise en œuvre du *framework* sont révisées ou améliorées pour prendre en charge des exigences de sécurité plus précises. Parmi les travaux utilisant cette approche, nous citons ci-dessous ceux qui ont étendu les *framework* UML, Tropos et KAOS, afin de modéliser des propriétés de sécurité.

UMLsec (Jurjens, 2001) est une extension d'UML pour modéliser des propriétés de sécurité comme la confidentialité

et le contrôle d'accès. UMLsec inclut toutes les analyses UML et artefacts de conception tels que les diagrammes d'activités, de déploiement, de séquences et d'états. La conception du système peut ainsi être représentée avec UMLsec. Les exigences de sécurité sur les modèles UMLsec sont ensuite vérifiées par *model-checking*.

Secure Tropos (Mouratidis, Manson *et al.*, 2003) est une extension de la méthode Tropos dont le but est de permettre de capturer de manière adéquate les exigences de sécurité. Elle a introduit l'utilisation de nouveaux concepts comme les diagrammes de sécurité, les contraintes de sécurité et les capacités de sécurité comme concepts de base.

KAOS, *Knowledge Acquisition in auTOMated Specification*, (Van Lamsweerde, 2001) est une approche semi-formelle pour l'élicitation, l'analyse et la modélisation des exigences. Le modèle d'exigences de cette approche est fondé sur la logique du premier ordre temporelle. KAOS a ensuite été étendu de sorte qu'il puisse manipuler des exigences non fonctionnelles telles que les exigences de sécurité. La méthode générale pour l'élaboration d'exigences de sécurité KAOS est fondée sur la construction progressive et sur la spécification de deux modèles concurrents (Mansour, 2009):

- un modèle intentionnel du système à construire qui couvre aussi bien le logiciel que son environnement,
- un anti-modèle dérivé du modèle qui montre comment les spécifications des éléments du modèle peuvent être menacées, pourquoi elles peuvent l'être et par qui. Les contre-mesures qui sont dérivées pour répondre aux menaces permettent de définir de nouvelles exigences de sécurité qui seront introduites dans le modèle initial.

Un autre travail (Lodderstedt, Basin *et al.*, 2002) présente un langage de modélisation fondé sur UML, SecureUML, visant à intégrer l'information de contrôle d'accès aux modèles d'applications définis avec UML. L'approche est axée sur les politiques de modélisation de contrôle d'accès et leur intégration dans un processus de développement de logiciels fondé sur les modèles. Le langage s'appuie sur une extension du modèle de contrôle d'accès RBAC (*Role Based Access Control*). RBAC est un modèle de contrôle d'accès où les utilisateurs et leurs privilèges sont découplés par rôles (Sandhu, Coyne *et al.*, 1996). SecureUML a ajouté la notion de contraintes d'autorisation pour définir les pré-conditions pour régir l'accès à une opération. (Lodderstedt, Basin *et al.*, 2002) présente l'exemple concret d'un *scheduler* sécurisé spécifié avec SecureUML que nous avons repris en figure 3. Il s'agit d'une application constituée de deux composantes: Calendrier (*calendar*) et entrée (*entry*). Un calendrier peut contenir plusieurs entrées, chacune représentant un rendez-vous avec une date de début, une date de fin et un emplacement. Chaque entrée est détenue par un utilisateur dont le nom est stocké dans l'attribut *owner*. Les expressions supplémentaires indiquées dans le schéma sont utilisés pour exprimer des informations de contrôle d'accès.

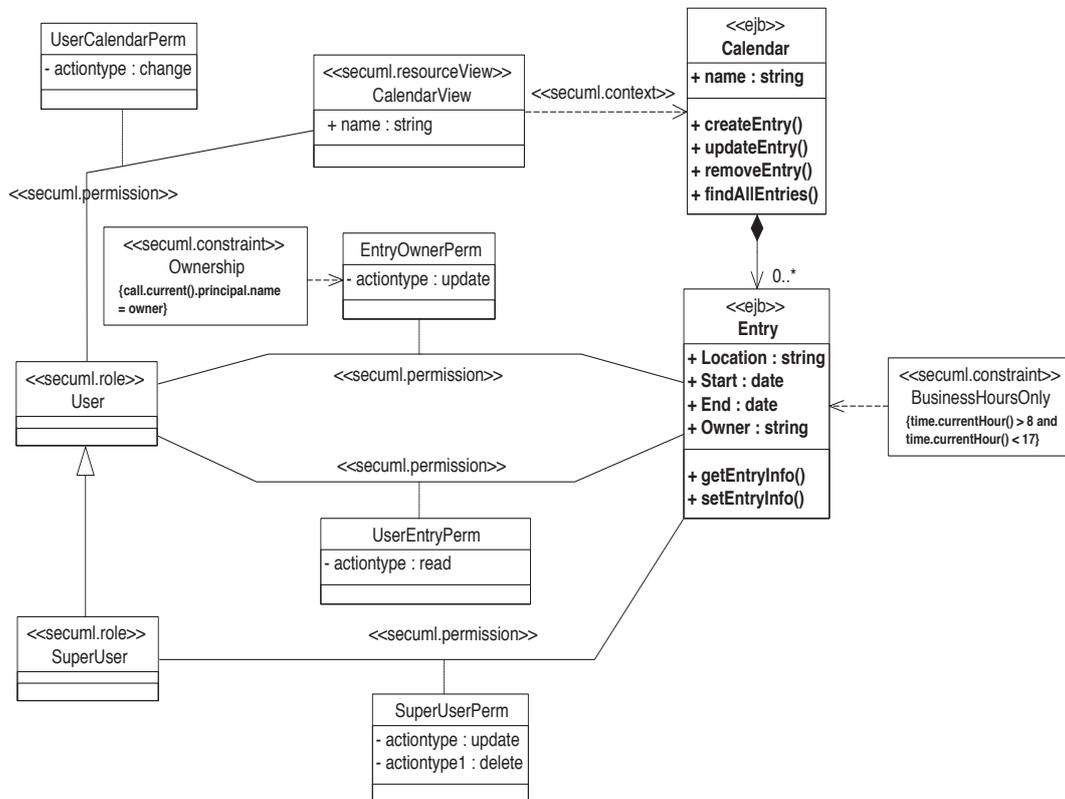


Figure 3. Exemple de scheduler sécurisé. Source : (Lodderstedt, Basin, *et al.*, 2002)

Analyse:

L'avantage de l'approche d'extension des *frameworks* de modélisation est qu'elle permet de répondre à un plus large éventail d'exigences de sécurité et de transmettre des exigences de sécurité plus complètes et plus exactes pour l'analyse du système. L'inconvénient est que les fonctionnalités d'analyse, de conception et de mise en œuvre du *framework* sont révisées ou améliorées pour prendre en charge des exigences de sécurité plus précises.

4.1. Construction de Framework de modélisation à partir de méthodes semi-formelles et formelles

Les méthodes formelles de spécification des exigences permettent de spécifier de manière rigoureuse, non ambiguë et vérifiable les exigences de sécurité. Mais un problème récurrent est leur coût élevé et leur difficulté d'utilisation. Les méthodes semi-formelles de spécification des exigences sont des méthodes fondées sur des langages en général graphiques qui disposent d'une syntaxe claire et d'une sémantique ambiguë. Elles sont donc faciles à utiliser, mais leurs faiblesses sont dues au caractère ambigu des notations semi-formelles, à leur forme limitée de spécification et d'analyse et au fait qu'elles ne disposent pas d'outils de preuve (Mansour, 2009).

Les méthodes semi-formelles et formelles peuvent être combinées de manière à se compléter. Dans ce qui suit, nous décrivons des approches qui utilisent les méthodes formelles en combinaison avec des méthodes semi-formelles, et ce dans le cadre de *frameworks* bien définis.

Le Formal Design Analysis Framework (FDAF) (Dai et Cooper, 2005) est un *framework* d'architecture orienté aspects qui permet la conception et l'analyse automatique des propriétés non fonctionnelles des architectures logicielles, en utilisant UML et un ensemble de méthodes formelles existantes. Dans le FDAF, les propriétés non fonctionnelles, y compris la sécurité, sont représentées comme des aspects. FDAF traduit une conception UML étendue semi-formelle en une notation formelle.

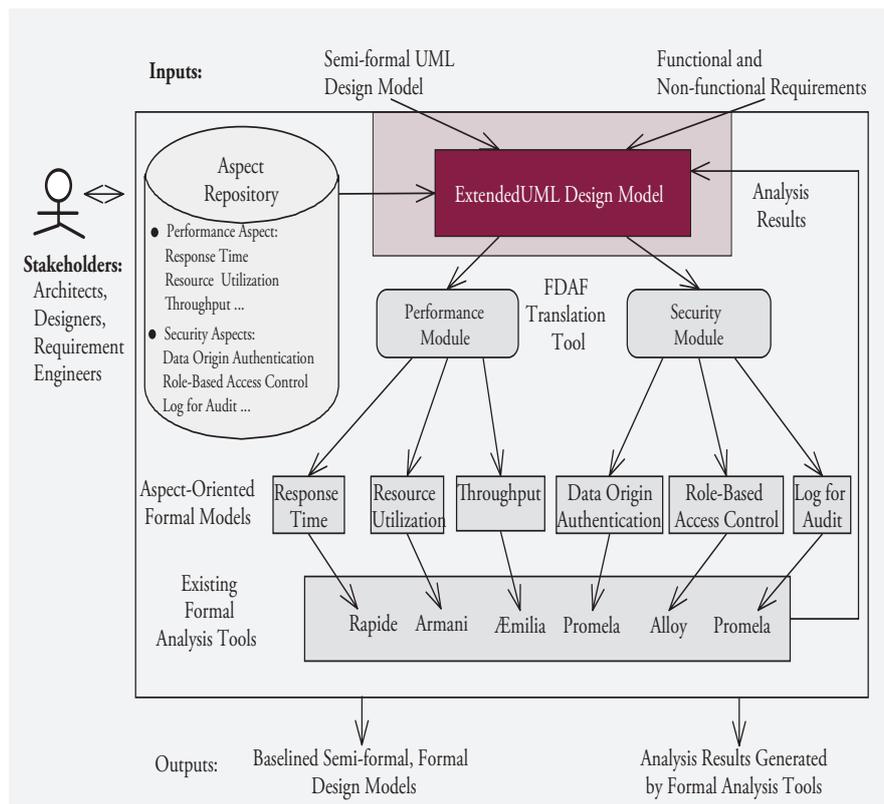


Figure 4. Formal Design Analysis Framework. Source: (Dai et Cooper, 2005)

La limite majeure de FDAF est que la variété des notations formelles utilisées pour formaliser les ENF constitue un obstacle pour étudier les différentes interactions entre les ENF. En conséquence, l'utilisateur ne peut pas détecter les conflits entre les différents types d'ENF (Matoussi et Laleau, 2008).

Une autre proposition de formalisation des Exigences Non Fonctionnelles (Botella, Burgués *et al.*, 2001) est le langage Nofun (acronyme pour Not Functional) qui décrit de manière formelle et évalue la qualité des logiciels à base de composants, en se fondant sur le *framework* des standards de qualité ISO/IEC. Un modèle de qualité est

défini par des caractéristiques générales de logiciels qui sont affinées en sous-caractéristiques puis en attributs, dans une hiérarchie à plusieurs niveaux. La sécurité fait partie des sous-caractéristiques définies par ce modèle de qualité. Les exigences de qualité en général et celles de la sécurité en particulier sont définies comme des restrictions dans le modèle de qualité.

Le langage NoFun se compose de trois parties différentes. Dans la première partie, les caractéristiques et les attributs de qualité des composants sont définis. Dans la deuxième partie, des valeurs sont affectées aux attributs de base de la qualité des composants. Troisièmement, les exigences de qualité dont font partie les exigences de la sécurité, peuvent être spécifiées pour les composants, à la fois indépendamment et dépendamment du contexte (propriétés de qualité générales ou dépendantes d'un domaine informatique particulier). Ce langage contient des mécanismes de structuration et de définition de types d'éléments qui permettent de définir des modèles de qualité de manière détaillée. Cependant, il n'offre pas de possibilité de détecter les conflits possibles entre les différentes exigences de qualité.

Mansour (2009) utilise la méthode B pour spécifier les exigences de sécurité. Ce travail propose une approche formelle qui tire les spécifications de conception d'un ensemble d'exigences de sécurité modélisées en utilisant une extension de KAOS pour la sécurité.

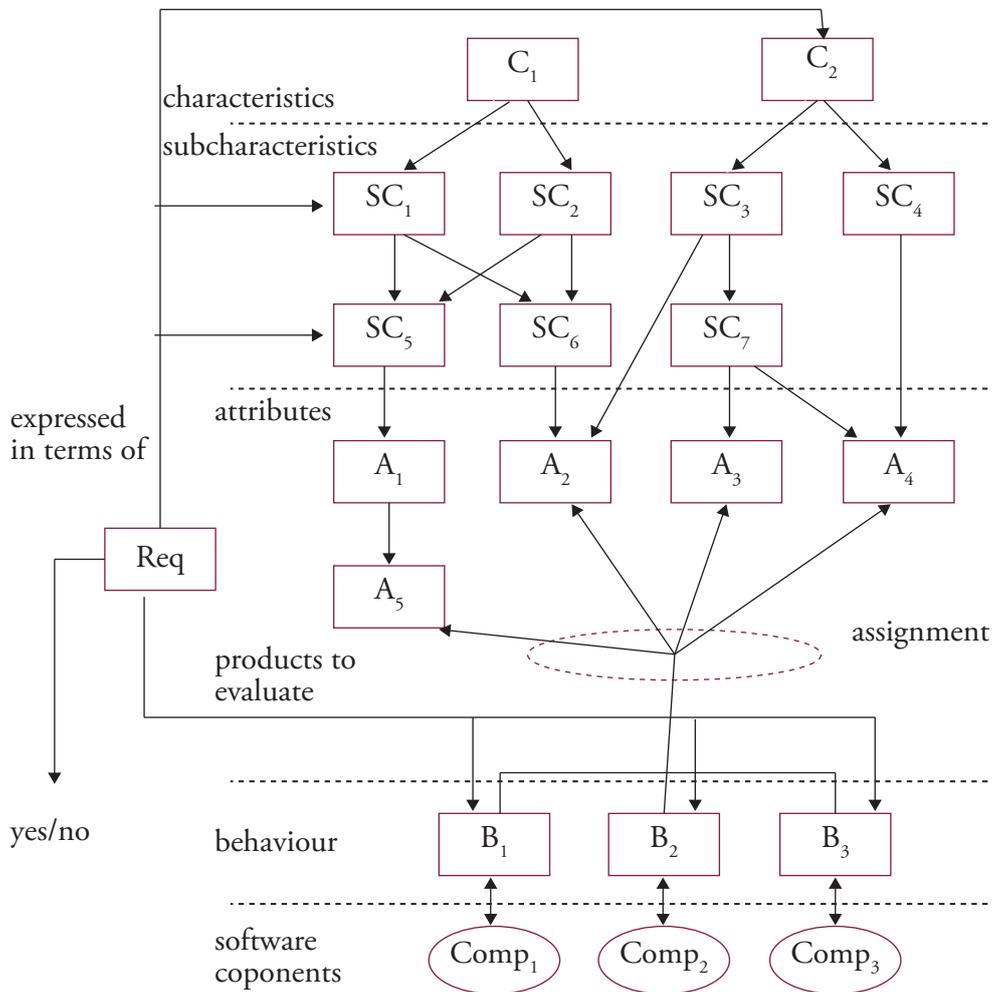


Figure 5. Modèle de qualité ISO/IEC représenté avec NoFun. Source: (Botella et al., 2001)

La première étape de l'approche consiste à transformer le modèle d'exigences de sécurité KAOS en un modèle abstrait B. La deuxième étape consiste à affiner le modèle en utilisant les mécanismes de raffinement B afin de générer des spécifications de conception et d'exécution. Ce processus de transformation et de raffinement est détaillé en Figure 6.

L'intérêt de cette approche est qu'elle permet d'utiliser le langage formel et les mécanismes de raffinement de la méthode B. Ceci permet de préserver les propriétés du système déjà prouvées dans les modèles de niveau supérieur. Par ailleurs, la méthode formelle intervient à un stade dans le processus de spécification où la frontière entre le logiciel et son environnement a déjà été établie: les exigences de sécurité sont d'abord spécifiées par KAOS qui tient compte de cette interaction (Darimont et Van Lamsweerde, 1996).

Ledang et Souqières (2002) suggèrent une démarche de formalisation des spécifications UML comme suit:

- Traduction des *use case* en B. Chaque cas d'utilisation est modélisé comme une opération B.
- Modélisation des opérations de classe. Chaque opération de classe est modélisée comme une opération B dans une machine abstraite.
- Modélisation des diagrammes d'états qui se fait en deux étapes, la création d'une opération abstraite B pour chaque événement puis la mise en œuvre (ou raffinement) de l'opération B.

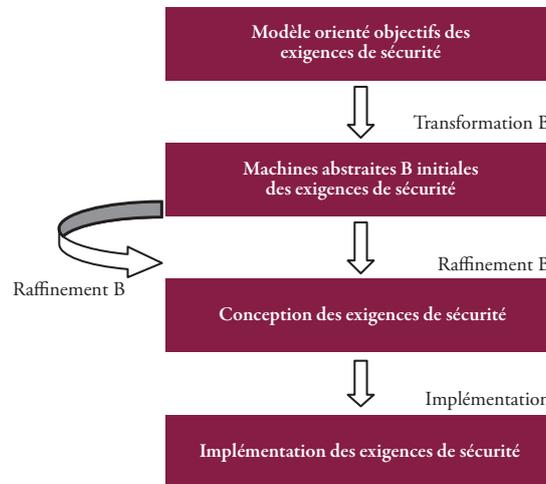


Figure 6. Dérivation formelle des spécifications de la sécurité. Source: (Mansour, 2009)

Le principal objectif pour traduire UML en B est de permettre une analyse formelle des spécifications UML à travers leurs spécifications correspondantes en B. L'autre objectif est d'utiliser les spécifications UML comme un outil pour faciliter la réalisation de spécifications en B.

Analyse:

L'utilisation combinée de méthodes semi-formelles et formelles permet de spécifier des exigences de sécurité en bénéficiant des avantages des deux méthodes: la facilité d'utilisation de la première et la rigueur de la deuxième.

Cependant, selon Van Lamsweerde (2004) et Darimont *et al.* (1996), les techniques de spécification formelles souffrent de lacunes qui expliquent pourquoi elles ne sont pas entièrement adéquates pour la phase critique d'élaboration et d'analyse des exigences. Ces lacunes sont décrites comme suit:

- La grande majorité des techniques se concentre sur la modélisation et la spécification du logiciel seul sans tenir compte de son environnement. En effet, les méthodes d'ingénierie des exigences ont besoin d'ontologies qui offrent des abstractions de haut niveau telles que *goal*, *goal refinements*, *agent*, *responsibility assignments* et ne peuvent se limiter à des abstractions de programmation comme *operation* ou *state*.
- Il n'existe pas de méthode constructive pour modéliser ou spécifier des systèmes complexes. Le problème n'est pas simplement de traduire les instructions en langage naturel dans des modèles semi-formels et/ou des spécifications formelles. L'Ingénierie des exigences, en général, prévoit que les exigences complexes soient élicitées, élaborées, structurées, interdépendantes et négociées.

Si la première lacune peut être comblée par une utilisation combinée de méthodes semi-formelles qui seront utiles lors de la phase d'élaboration des exigences de sécurité, la deuxième lacune est beaucoup plus liée à la non disponibilité d'outils d'analyse et de gestion des exigences de sécurité.

5. Synthèse

Nous avons passé en revue plusieurs approches de spécification des exigences de sécurité. Certains travaux de recherche ont utilisé les *frameworks* de modélisation des exigences (fonctionnelles et non fonctionnelles) pour le cas particulier des exigences de sécurité. Cette approche s'avère intéressante dans la mesure où elle permet d'utiliser des outils qui ont déjà fait leurs preuves et qui sont donc faciles à utiliser, car maîtrisés par les participants au processus d'élicitation, d'expression et de spécification des exigences. Cependant, ces outils présentent des limites qui sont liées à:

- leur expressivité, car ils ne possèdent pas suffisamment d'attributs pour spécifier correctement des exigences de sécurité,
- leur degré de formalisme si ces outils ne se fondent pas sur un langage formel.

D'autres recherches se sont donc consacrées à la manière de pallier ces lacunes.

- en étendant ces *frameworks* avec une expressivité adéquate liée aux exigences de sécurité,
- en utilisant des langages formels qui permettent de spécifier de manière rigoureuse, non ambiguë et vérifiable les exigences de sécurité.

Le tableau 2 résume les principales caractéristiques des approches de spécification des exigences de sécurité analysées.

Approches de spécification	Avantages	Limites
<i>Frameworks</i> de modélisation existants	Coût et facilité d'utilisation.	Expressivité limitée du langage de spécification des exigences de sécurité.
Extension des <i>frameworks</i> de modélisation	Expressivité du langage de spécification des exigences de sécurité.	Révision des outils du <i>Framework</i> pour prendre en charge des exigences de sécurité plus précises.
Combinaison de méthodes semi-formelles et formelles	Facilité d'utilisation du semi formel et rigueur du formel.	Coût des méthodes formelles et indisponibilité d'outils d'analyse et de gestion des exigences

Tableau 2. Avantages et limites des approches de spécification des exigences de sécurité

Cette analyse a permis de relever que toutes les approches de spécification des exigences de sécurité ont des avantages et des limites. La justification du choix d'une approche de spécification est donc d'abord liée à son contexte d'utilisation et à ses contraintes. Par ailleurs, elle est également liée à la méthode d'élicitation utilisée.

6. Conclusion

Les exigences de sécurité jouent un rôle important dans le développement des logiciels. L'Ingénierie des exigences de sécurité implique le développement de méthodes et d'outils qui permettent la construction de spécifications d'exigences de sécurité complètes, cohérentes et claires.

Cet article présente une revue d'approches de spécification des exigences de sécurité et une analyse de leurs apports et faiblesses. Nous avons également cité quelques approches d'élicitation des exigences de sécurité, car nous avons estimé que dans le contexte de la sécurité, elles sont parfois indissociables des approches de spécification des exigences de sécurité.

Idéalement, une approche de spécification des exigences de sécurité devrait tenir compte de tous les critères suivants : l'expressivité, la rigueur, la simplicité d'utilisation, le coût et la complétude de l'approche. En réalité, le choix de l'approche de spécification des exigences de sécurité est un compromis entre ces critères et les besoins réels de l'application.

Une des perspectives de recherche dans ce domaine serait de définir des outils intégrés qui permettraient de couvrir toutes les phases de l'Ingénierie des exigences de sécurité (élicitation, analyse, spécification, vérification et gestion) et de tenir compte des aspects organisationnels dans la définition des exigences.

Une autre perspective serait d'améliorer les méthodes d'analyse des conflits entre les exigences non fonctionnelles entre elles puis entre les exigences fonctionnelles et les exigences de sécurité.

Références

- Barnum, S., Sethi, A. (2006). *Introduction to Attack patterns*. digital. <https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/attack/585-BSI.html>
- Botella, P., Burgués, X., Franch, X., Huerta, M. (2001). Modeling Non-Functional Requirements, *Proceedings of Jornadas de Ingenieria de Requisitos Aplicada (JIRA)*, Sevilla, Spain.
- Dai, L., Cooper, K. (2005). *Formal design analysis framework: an aspect-oriented architectural framework*. PhD Dissertation, Dallas University of Texas.
- Darimont, R., Van Lamsweerde, A. (1996). Formal Refinement Patterns for Goal-Driven Requirements. *Proceedings of the 4th ACM SIGSOFT symposium on Foundations of software engineering*, San Francisco, 179-190.
- Firesmith, D.G. (2003a). Security Use Cases. *Journal of Object, Technology*. Vol. 2, No. 3, 53-64.
- Firesmith, D.G. (2003b). Engineering Security Requirements. *Journal of Object, Technology*. Vol 2, No 1, 53-64.

- Giorgini, P., Massacci, F., Zannone, N. (2005). Security and Trust Requirements Engineering. In *Foundations of Security Analysis and Design III - Tutorial Lectures, volume 3655 of Lecture Notes in Computer Science*, Springer, 237–272.
- Glinz, M. (2005). Rethinking the Notion of Non-Functional Requirements. *Proceedings of the Third World Congress for Software Quality*, Munich, Germany, Vol. II. 55-64.
- Hadavi, M.A., Hamishagi, V.S., Sangchi, H.M. (2008). Security Requirements Engineering; State of the Art and Research Challenges. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong: 19-21 March.
- He, Q., Anton, A.I. (2003). A Framework for Modeling Privacy Requirements in Role Engineering. In *Proceedings of the 9th International Workshop on Requirements Engineering- Foundation for software quality*, Klagenfurt/Velden, Austria: 16-17 Juin, 137–146.
- Software Engineering Standards Committee of IEEE Computer Society (1998), *IEEE Guide for Developing System Requirements Specifications*. Norme IEEE 1233, édition 1998.
- Jurjens, J. (2001). Towards Secure Systems Development with UMLsec, *Fundamental Approaches to Software Engineering (FASE/ETAPS)*, International Conference, Genoa.
- Kotonya, G., Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*. New York: John Wiley & Sons.
- Ledang, H., Souquières, J. (2002). Integration of UML Views using B Notations. *Proceedings of workshop on Integration and Transformation of UML Models*, Malaga, Spain.
- Liu, L., Yu, E.S.K., Mylopoulos, J. (2003). Security and Privacy Requirements Analysis within a Social Setting. In *Proceedings IEEE international conference on requirements engineering (RE'03)*, Monterey, California, 151–161.
- Lodderstedt, T., Basin, D., Doser, J. (2002). SecureUML: A UML-Based Modeling Language for Model-Driven Security. In *Proceedings of the 5th International Conference on the Unified Modeling Language*, Dresden, Germany: October 2002, 426–441.
- Mansour, R. (2009), *Formal Analysis and Design for Engineering Security (FADES)*, PhD in Computer Science and Applications, Blacksburg, Virginia: Faculty of the Virginia Polytechnic Institute and State University.
- Matoussi, A., Laleau, R. (2008). *A Survey of Non-Functional Requirements in Software Development Process*. Rapport Technique TR–LACL–2008–7. Paris: Laboratoire d'Algorithmique, Complexité et Logique (LACL), Université Paris 12.
- McDermott, J., Fox, C. (1999). Using Abuse Case Models for Security Requirements Analysis. In *Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC)*, Phoenix, Arizona.
- Mouratidis, H., Manson, G., Giorgini, P. (2003). Modelling Secure Multiagent Systems. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne-Australia*, 859-866.
- Mylopoulos, J., Chung, L., Yu, E. (1999). From Object-oriented to Goal oriented Requirements Analysis. *Communications of the ACM, Vol. 42 No. 1*, 31-37.
- Nuseibeh, B., Easterbrook, S. (2000). Requirements Engineering: A Roadmap. In *Proceedings of International Conference on Software Engineering*, ACM Press. Limerick, Ireland.
- Phillips, M. (2003), *Using a Capability Maturity Model to Derive Security Requirements*, SANS InfoSec Reading Room, GSEC Practical v1. <http://www.sans.org/rr/whitepapers/bestprac/1005.php>
- Sandhu, R.S., Coyne, E.J, Feinstein, H.L, Youman, C.E (1996). *Role-based access control models*. IEEE Computer, 29(2), 38–47.
- Sindre, G., Opdahl, A. L. (2001). Capturing Security Requirements through Misuse Cases. In *Proceedings the 14th Norwegian Informatics Conference (NIK'2001)*, Tromsø, Norway.
- Su, X., Bolzoni, D., van Eck, P.A.T. (2006). A Business Goal Driven Approach for Understanding and Specifying Information Security Requirements. In *11th International Workshop on Exploring Modelling Methods in Systems Analysis and Design (EMMSAD2006)*, Luxembourg, 465–472.
- Van Lamsweerde, A. (2001). *Building Formal Requirements Models for Reliable Software*. LNCS, Vol. 2043, Springer.
- Van Lamsweerde, A. (2004), Elaborating Security Requirements by Construction of Intentional Anti-Model. *Proceedings of the 26th International Conference on Software Engineering (ICSE)*, 148-157
- Viega, J. (2005). *The CLASP Application Security Process*. Secure Software.

Dossier spécial :
Conférence Sitacam'2011

Transformation de Fourier et moments invariants appliqués à la reconnaissance des caractères Tifinaghe

Rachid El Ayachi

Equipe de traitement de l'information et de télécommunications TIT, Facultés des Sciences et Techniques, Université Sultan Moulay Slimane, Béni Mellal, Maroc
rachidieea@yahoo.fr

Mohamed Fakir

Equipe de traitement de l'information et de télécommunications TIT, Facultés des Sciences et Techniques, Université Sultan Moulay Slimane, Béni Mellal, Maroc
fakfad@yahoo.fr

Belaid Bouikhalene

Equipe de traitement de l'information et de télécommunications TIT, Faculté polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal, Maroc
bbouikhalene@yahoo.fr

Résumé

La Reconnaissance des Caractères Optiques OCR est un outil qui vise à donner la possibilité aux ordinateurs de lire les caractères sans intervention humaine. Le problème de l'OCR est celui de faire reconnaître les caractères indépendamment de leur position, rotation et taille. Pour remédier à ce problème, on utilise les descripteurs invariants. Le système développé dans ce travail utilise la transformation de Fourier et les moments invariants pour extraire les attributs, et la programmation dynamique et les réseaux de neurones pour la classification. La reconnaissance est effectuée sur les caractères Tifinaghe imprimés.

Abstract

Optical Character Recognition OCR is a tool that aims to provide opportunities for computers to read characters without human intervention. The objective of OCR is characterization of a character by invariant descriptors in translation, rotation and scaling. In this paper, the OCR developed use invariant moments and Fourier transform in extraction phase. In the recognition phase, dynamic programming and neural network are adopted. All tests are applied on Tifinaghe printed characters.

Mots-clés

Reconnaissance des Caractères Optiques, transformation de Fourier, moment invariant, programmation dynamique, réseaux de neurones

Keywords

Requirement optical character Recognition OCR, Fourier transform, invariant moment, Dynamic programming, neuronal Network

1. Introduction

La Reconnaissance de Caractères Optiques, ROC (*Optical Character Recognition* OCR) (Fakir, 2001), (Fakir *et al.*, 2000), (El Ayachi *et al.*, 2010), (Amrouch *et al.*, 2010), (Es Saady *et al.*, 2010), (El Ayachi *et al.*, 2011b) est un sujet de recherche courant dans le domaine de la reconnaissance de forme.

Les applications de l'OCR sont nombreuses et s'appliquent à des domaines aussi variés que le guidage automatique de véhicule, la reconnaissance d'objet, la numérisation d'ouvrage, les transactions bancaires.

Dans cet article, l'objectif de l'OCR (Figure 1) est de reconnaître les caractères Tifinaghe (Blumenstein *et al.*, 2002), (Bencharef *et al.*, 2011), (Es Saady *et al.*, 2011), (El Ayachi *et al.*, 2011c) (Figure 2). Le système adopté comporte plusieurs phases : prétraitement, extraction d'attributs et reconnaissance.

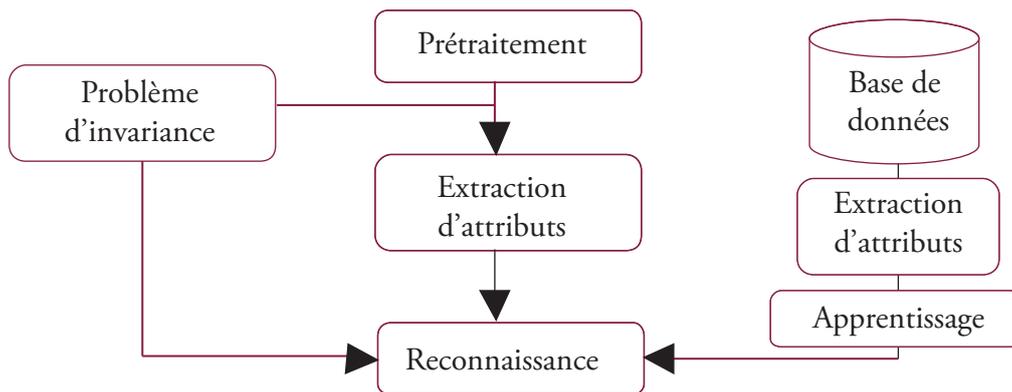


Figure 1. l'OCR

La structure de l'article est organisée de la façon suivante: la section 2 s'intéresse à la phase de prétraitement qui contient plusieurs opérations; la section 3 représente la phase d'extraction d'attributs en utilisant la transformation de Fourier et les moments invariants, puis la section 4 traite la phase de la reconnaissance à l'aide des réseaux de neurones et la programmation dynamique. La section 5 résout le problème d'invariance alors que la section 6 illustre les résultats obtenus. La section 7 donne les différentes conclusions tirées à partir de ce travail.



Figure 2. Les caractères de Tifinaghe – IRCAM

2. Pré-traitement

Après l'acquisition de l'image, le système de reconnaissance commence par la phase de prétraitement comportant les fonctions suivantes: la normalisation, la correction de l'inclinaison et la segmentation.

2.1. Normalisation

Pour éliminer les zones inutiles dans une image, la méthode d'histogramme (El Ayachi *et al.*, 2010) est utilisée. Le principe repose sur le calcul des histogrammes horizontal et vertical, ensuite, le parcours des deux histogrammes pour détecter les deux lignes L1 et L2 et les deux colonnes C1 et C2 qui permettent de déterminer les marges à supprimer (Figure 3).



Figure 3. Détermination des marges

2.2. Correction d'inclinaison

L'inclinaison (Kavallieratou *et al.*, 1999), (El Ayachi *et al.*, 2011b) (Fig.4) est un phénomène qui influence négativement sur la phase de segmentation et par conséquent sur le système de reconnaissance, il est donc indispensable d'utiliser une méthode efficace pour la correction de l'inclinaison.

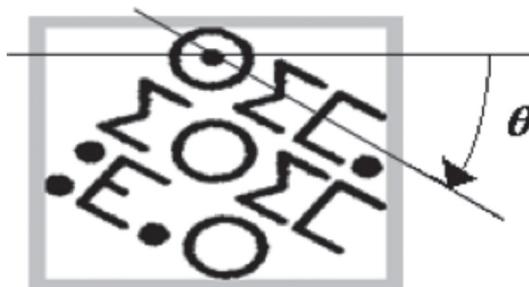


Figure 4. Angle d'inclinaison

La correction d'inclinaison passe par deux étapes: la détection de l'angle d'inclinaison et la rotation dans le sens approprié.

Pour détecter l'angle d'inclinaison, nous avons adopté la transformation de Hough (Fakir *et al.*, 2000), (El Ayachi *et al.*, 2010) qui est connue par robustesse dans ce type de problèmes.

2.3. Segmentation

La fonction de segmentation consiste premièrement à détecter les lignes dans une image contenant un texte en Tifinaghe, et secondement, à segmenter ces lignes en caractères.

Aussi, dans cette fonction de segmentation, nous avons utilisé la méthode d'histogramme (El Ayachi *et al.*, 2010): le balayage de l'histogramme horizontal de haut en bas pour déterminer les lignes (à l'aide des paramètres LH et LB), le parcours de l'histogramme vertical de gauche à droite afin de segmenter les lignes en caractères (en utilisant les paramètres CG et CD).

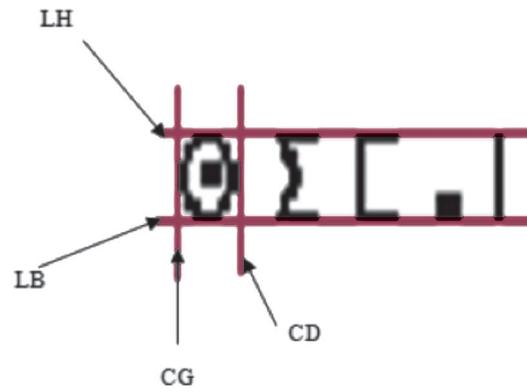


Figure 5. Détection des lignes et caractères

Cette section a traité les différentes fonctions de prétraitement à savoir : la normalisation, la correction d'inclinaison et la segmentation. La section suivante s'intéresse à la phase d'extraction d'attributs.

3. Extraction d'attributs

L'extraction d'attributs est la deuxième phase à appliquer dans un OCR, elle joue un rôle primordial dans la reconnaissance, car elle doit prendre en considération la représentation du caractère dans diverses situations telles que : la translation, la rotation et l'échelle. C'est la raison de l'utilisation des moments invariants (Fakir, 2001), (El Ayachi *et al.*, 2010), (El Ayachi *et al.*, 2011b) et de la transformation de Fourier (Ghorbel, 1993), (El Ayachi *et al.*, 2011a) dans le système traité dans ce travail.

3.1. Moments invariants

Soit f une fonction définie par : $f(x, y) = 1$ sur une région R fermée et délimitée et $f(x, y) = 0$ ailleurs. On définit le moment d'ordre (p, q) comme suit :

$$m_{pq} = \iint_R x^p y^q f(x, y) dx dy \quad , \text{ pour } p, q = 0, 1, 2, \dots \quad (1)$$

Les moments centraux peuvent être exprimés par :

$$\mu_{pq} = \iint_R (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \quad , \text{ avec } \bar{x} = \frac{m_{10}}{m_{00}} \quad , \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (2)$$

Pour une image numérique, l'équation (2) devient :

$$\mu_{pq} = \sum_{(x,y) \in R} \sum (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (3)$$

On peut facilement vérifier que les moments centraux d'ordre $p + q \leq 3$ peuvent être calculés par les formules suivantes :

$$\begin{aligned} \mu_{00} &= m_{00} \\ \mu_{10} &= 0 \\ \mu_{11} &= m_{11} - \bar{y}m_{10} \\ \mu_{20} &= m_{20} - \bar{x}m_{10} \\ \mu_{02} &= m_{02} - \bar{y}m_{01} \\ \mu_{12} &= m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2 m_{10} \\ \mu_{21} &= m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2 m_{01} \\ \mu_{30} &= m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10} \end{aligned} \quad (4)$$

$$\mu_{03} = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01}$$

Les moments centraux sont invariants par translation. Ils peuvent être normalisés, pour conserver l'invariance, par changement d'échelle et on obtient les moments centraux normalisés :

$$\alpha_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \text{ avec } \gamma = \frac{p+q}{2} + 1, \text{ pour } p+q = 2,3,\dots \quad (5)$$

Les invariants du moment suivants ont été obtenus par Hu (1962) et fréquemment utilisés comme caractéristiques pour la reconnaissance des formes.

$$\begin{aligned} \varphi_1 &= \alpha_{20} - \alpha_{02} \\ \varphi_2 &= (\alpha_{20} - \alpha_{02})^2 + 4\alpha_{11}^2 \\ \varphi_3 &= (\alpha_{30} - \alpha_{12})^2 + (3\alpha_{12} - \alpha_{03})^2 \\ \varphi_4 &= (\alpha_{30} + \alpha_{12})^2 + (\alpha_{21} + \alpha_{03})^2 \\ \varphi_5 &= (\alpha_{30} - 3\alpha_{12})(\alpha_{30} + \alpha_{12})[(\alpha_{30} + \alpha_{12})^2 - 3(\alpha_{21} + \alpha_{03})^2] \\ &+ (3\alpha_{21} - \alpha_{03})(\alpha_{21} + \alpha_{03})[3(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2] \\ \varphi_6 &= (\alpha_{20} - \alpha_{02})[(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2] \\ &+ 4\alpha_{11}(\alpha_{30} + \alpha_{12})(\alpha_{21} + \alpha_{03}) \\ \varphi_7 &= (3\alpha_{21} - \alpha_{03})(\alpha_{30} + \alpha_{12})[(\alpha_{30} + \alpha_{12})^2 - 3(\alpha_{21} + \alpha_{03})^2] \\ &+ (3\alpha_{12} - \alpha_{03})(\alpha_{21} + \alpha_{03})[3(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2] \end{aligned} \quad (6)$$

Hu a montré que ces quantités $\varphi_i, (1 \leq i \leq 7)$ sont invariantes par changement d'échelle, translation et rotation.

3.2. Transformations de Fourier et moments invariants (étude théorique)

Pour $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, sa transformée de Fourier est donnée par :

$$\widehat{f}(u, v) = \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} f(x, y) e^{-i2\pi(xu+yv)} dx dy \quad (7)$$

Cette transformation admet l'approximation discrète :

$$\widehat{f}(u, v) = \sum_x \sum_y f(x, y) e^{-i2\pi(xu+yv)} \quad (8)$$

On définit aussi le moment d'ordre (p, q) par l'équation (1) dont l'approximation discrète est donnée par l'équation :

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (9)$$

La relation entre les moments et la transformation de Fourier :

Si on calcule

$$\begin{aligned} \frac{\partial^{p+q} \widehat{f}(u, v)}{\partial u^p \partial v^q} &= \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} (-i2\pi x)^p (-i2\pi y)^q f(x, y) e^{-i2\pi(xu+yv)} dx dy \\ &= (-i2\pi)^{p+q} \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} x^p y^q f(x, y) e^{-i2\pi(xu+yv)} dx dy \end{aligned} \quad (10)$$

Pour $(u, v) = (0, 0)$

On obtient

$$\begin{aligned} \frac{\partial^{p+q} \widehat{f}(0,0)}{\partial u^p \partial v^q} &= (-i2\pi)^{p+q} \int_{-\infty-\infty}^{+\infty+\infty} x^p y^q f(x, y) dx dy \\ &= (-i2\pi)^{p+q} m_{pq} \end{aligned} \quad (11)$$

Alors,

$$m_{pq} = \frac{\partial^{p+q} \widehat{f}(0,0)}{(-i2\pi)^{p+q}} = Fe[x^p y^q f(x, y)]_{(0,0)} \quad (12)$$

D'où,

$$\begin{aligned} \frac{\partial^{p+q} \widehat{f}(u, v)}{\partial u^p \partial v^q} &= Fe[(-i2\pi)^{p+q} x^p y^q f(x, y)] \\ &= \sum_x \sum_y (-i2\pi)^{p+q} x^p y^q f(x, y) e^{-i2\pi(xu+yv)} \\ &= (-i2\pi)^{p+q} \sum_x \sum_y x^p y^q f(x, y) e^{-i2\pi(xu+yv)} \end{aligned} \quad (13)$$

On a :

$$\begin{aligned} m_{00} &= \widehat{f}(0,0) \\ m_{10} &= \frac{\partial \widehat{f}(0,0)}{\partial u} = Fe[xf(x, y)]_{(0,0)} \\ m_{01} &= \frac{\partial \widehat{f}(0,0)}{\partial v} = Fe[yf(x, y)]_{(0,0)} \\ m_{11} &= \frac{\partial^2 \widehat{f}(0,0)}{(\partial u \partial v)} = Fe[xyf(x, y)]_{(0,0)} \\ m_{02} &= \frac{\partial^2 \widehat{f}(0,0)}{(\partial v^2)} = Fe[y^2 f(x, y)]_{(0,0)} \\ m_{12} &= \frac{\partial^3 \widehat{f}(0,0)}{(\partial u \partial v^2)} = Fe[xy^2 f(x, y)]_{(0,0)} \\ m_{20} &= \frac{\partial^2 \widehat{f}(0,0)}{(\partial u^2)} = Fe[x^2 f(x, y)]_{(0,0)} \\ m_{21} &= \frac{\partial^3 \widehat{f}(0,0)}{(\partial u^2 \partial v)} = Fe[x^2 y f(x, y)]_{(0,0)} \\ m_{03} &= \frac{\partial^3 \widehat{f}(0,0)}{(\partial v^3)} = Fe[y^3 f(x, y)]_{(0,0)} \end{aligned} \quad (14)$$

$$m_{30} = \frac{\partial^3 \widehat{f}(0,0)}{(-i2\pi)^3} = Fe\left[x^3 f(x, y)\right]_{(0,0)}$$

La conclusion que nous tirons à partir de cette étude théorique est que nous pouvons utiliser la transformation de Fourier et ses dérivées pour calculer les valeurs moyennes pondérées d'une image f . Dans cette section, nous avons détaillé une étude théorique qui démontre la possibilité d'utiliser la transformation de Fourier et ses dérivées pour calculer les moments invariants d'un caractère segmenté. La prochaine section présente les méthodes de classification.

4. Reconnaissance

La reconnaissance (Al-Hajj *et al.*, 2006), (Outahajala *et al.*, 2011 ; Kessentini *et al.*, 2010), (Ali, 2008), (Ali *et al.*, 2009) est la tâche la plus délicate dans un OCR, car la réussite de ce système de reconnaissance repose sur la décision résultante. Il est donc nécessaire d'utiliser un processus de classification efficace qui a un taux de reconnaissance élevé et par conséquent un taux d'erreur diminué.

Dans cette étape, nous avons utilisé les deux méthodes suivantes : le réseau de neurones (Bouikhalene *et al.*, 2009), (El Ayachi *et al.*, 2010), (El Ayachi *et al.*, 2011b), (El Ayachi *et al.*, 2011c) et la programmation dynamique (El Ayachi *et al.*, 2010), (Chevalier *et al.*, 2003) pour la reconnaissance des caractères Tifnaghe, parce qu'elles sont connues par leur succès dans le domaine de la reconnaissance des caractères.

4.1. Réseau de neurones

La figure 6 représente l'exemple de réseau de neurones utilisé, à savoir un réseau multicouche qui contient une couche cachée.

Cet exemple de réseaux comporte :

- Une couche d'entrée à 7 cellules d'entrées (vecteur du moment invariant) $E_i = X_i$
- Une couche cachée de 3 neurones d'activations Y_j
- Une couche de sortie de 6 neurones d'activations Z_k
- 7×3 connexions entre la couche d'entrée et la couche cachée, chacune pondérée par V_{ji}
- 3×6 connexions entre la couche cachée et la couche de sortie, chacune pondérée par W_{kj}
- X_0 et Y_0 sont des scalaires

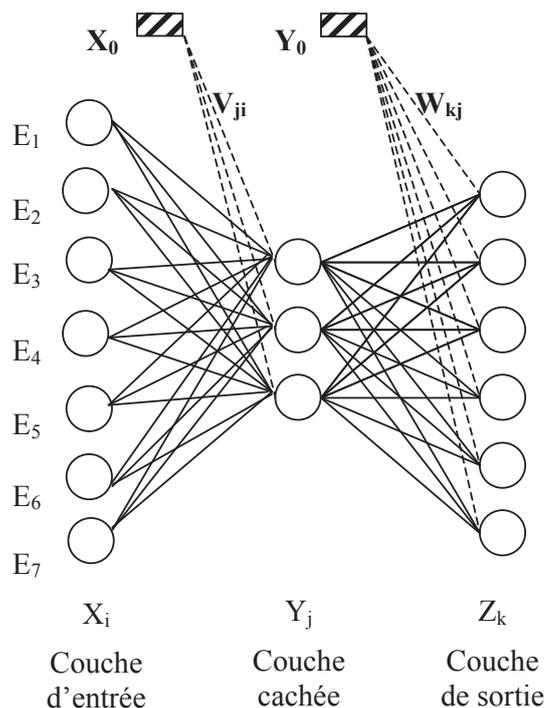


Figure 6. Réseau de neurones

Le principe de fonctionnement du réseau de neurones (Figure 6) repose sur cinq étapes.

Etape 1. Initialisation des poids des connexions.

Les poids sont pris aléatoirement.

Etape 2. Propagation des entrées.

Les entrées E_i sont présentées à la couche d'entrée: $X_i = E_i$

$$Y_j = f\left(\sum_{i=1}^7 X_i V_{ji} + X_0\right) \quad (15)$$

Ensuite de la couche cachée vers la couche de sortie, on adopte:

$$Z_k = f\left(\sum_{j=1}^3 Y_j W_{kj} + Y_0\right) \quad (16)$$

X_0 et Y_0 sont des scalaires

f est la fonction d'activation (fonction sigmoïde):

$$f(a) = \frac{1}{1 + \exp(-a)} \quad (17)$$

Etape 3. Rétro-propagation de l'erreur.

Au niveau de la couche de sortie, l'erreur entre la sortie désirée S_k et la sortie réelle Z_k est calculée par:

$$E_k = Z_k (1 - Z_k) (S_k - Z_k) \quad (18)$$

L'erreur calculée est propagée sur la couche cachée en utilisant la formule 19.

$$F_j = Y_j (1 - Y_j) \sum_{k=1}^6 W_{kj} \cdot E_k \quad (19)$$

Etape 4. Correction des poids des connexions.

On corrige les poids de connexions entre la couche d'entrée et la couche cachée par:

$$\Delta V_{ji} = \eta \cdot X_i \cdot F_j \quad \text{et} \quad \Delta Y_0 = \eta \cdot F_j \quad (20)$$

Puis, on modifie les connexions entre la couche cachée et la couche de sortie par:

$$\Delta W_{kj} = \eta \cdot Y_j \cdot E_k \quad \text{et} \quad \Delta X_0 = \eta \cdot E_k \quad (21)$$

η Un paramètre à déterminer empiriquement.

Etape 5. Boucle.

Boucler à l'étape 2 jusqu'à l'obtention d'un critère d'arrêt à définir (seuil d'erreur, nombre d'itérations).

Après l'apprentissage et l'exécution d'OCR, on utilise la distance euclidienne pour identifier les caractères de Tifinaghe:

$$d(t_k, o) = \left(\sum_{i=1}^6 (t_{ki} - o_i)^2 \right)^{1/2} \quad (22)$$

Avec, t_k la sortie désirée et o la sortie du réseau.

4.2. Programmation dynamique

La programmation dynamique est la deuxième méthode utilisée dans la reconnaissance. Elle est organisée en trois étapes.

Etape 1. Calculer la matrice d entre le vecteur du caractère segmenté V_{car} et chaque vecteur des caractères

Tifinaghe de référence V_{ref}

La matrice est donnée par:

$$d(x, y) = |V_{car}(x) - V_{ref}(y)| \quad (23)$$

Avec $x, y = 1, 2, \dots, 7$

Etape 2. Calculer le chemin optimal à partir du point (1,1) jusqu'à point (x,y) en utilisant la formule réursive suivante :

$$S(x, y) = d(x, y) + \min \begin{cases} S(x - 1, y) \\ S(x - 1, y - 1) \\ S(x, y - 1) \end{cases} \quad (24)$$

Avec $S(x,y)$ est la distance cumulée le long du chemin optimal à partir du point (1,1) jusqu'au point (x,y). $S(x,y)$ est évaluée dans l'espace [1,7]*[7,1] qui est parcouru colonne par colonne et ligne par ligne à partir du point (1,1).

Etape 3, Calculer l'indice de dissemblance :

$$D(V_{car}, V_{ref}) = \frac{S(7,7)}{7} \quad (25)$$

5. Problème d'invariance

Un OCR efficace est un système de reconnaissance qui prend en considération l'invariance du caractère en quelques situations (translation, rotation et taille). Cependant, pour les caractères Tifinaghe, cette caractéristique d'invariance pose un problème dans la phase de reconnaissance, car il y en a des caractères qui se ressemblent (⊖ et ⊕, Q et Ø, ⵏ et ⵐ). Donc, il faut ajouter un autre traitement pour régler ce problème. Ce qui est l'objectif de cette section, ou bien, dans la phase d'extraction d'attributs, il faut utiliser une autre méthode variante. Pour cela, nous allons enregistrer le caractère segmenté puis exécuter le processus pour calculer d'autres variables (SUP, INF et HORIZ) qui vont intervenir lors de la reconnaissance. Ce processus comporte deux traitements : le premier pour les caractères (Q et Ø) et (ⵏ et ⵐ) et le deuxième pour les caractères (⊖ et ⊕).

Le premier traitement (Figure 7) comprend les étapes suivantes :

- Diviser l'image en deux parties horizontalement (SUP et INF).
- Calculer le nombre de pixels noirs dans chaque partie.
- Attribuer 1 à la variable SUP et 0 à la variable INF si le nombre de pixels noirs dans la partie SUP est supérieur au nombre de pixels noirs dans la partie INF.
- Attribuer 0 à la variable SUP et 1 à la variable INF si le nombre de pixels noirs dans la partie SUP est inférieur au nombre de pixels noirs dans la partie INF.
-

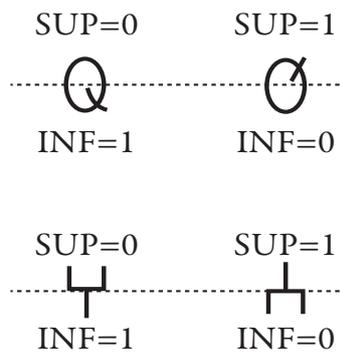


Figure 7. Premier traitement

Le tableau 1 indique les valeurs des variables SUP et INF.

Variables	Q	Ø	ⵏ	ⵐ
SUP	0	1	0	1
INF	1	0	1	0

Tableau 1. Les variables SUP et INF

Le deuxième traitement (Figure 8) se déroule de la façon suivante :

- Tracer le contour du caractère.

- Enlever le contour externe.
- Calculer l'histogramme horizontal.
- Parcourir l'histogramme, s'il contient deux zones représentant les pixels noirs, alors attribuer 2 à la variable HORIZ, et s'il contient une seule zone, alors attribuer 1 à la variable HORIZ.
-

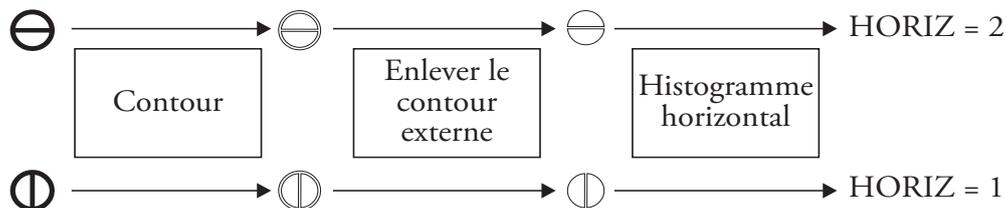


Figure 8. Deuxième traitement

Le tableau 2 indique les valeurs de la variable HORIZ.

Variable	⊖	⊕
HORIZ	2	1

Tableau 2. La variable HORIZ

Cette section a présenté des traitements adoptés pour éviter le problème de ressemblance des caractères Tifnaghe. La section suivante montre les résultats obtenus en utilisant le système de reconnaissance développé.

6. Résultats

L'image de la figure (figure 9) représente un exemple de texte en Tifnaghe.

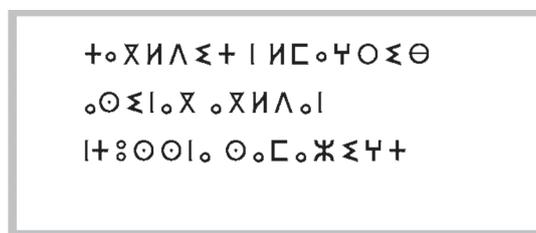


Figure 9. Exemple de texte en Tifnaghe

Après l'application de la phase de prétraitement à la figure 9, nous obtenons les figures 10 et 11.

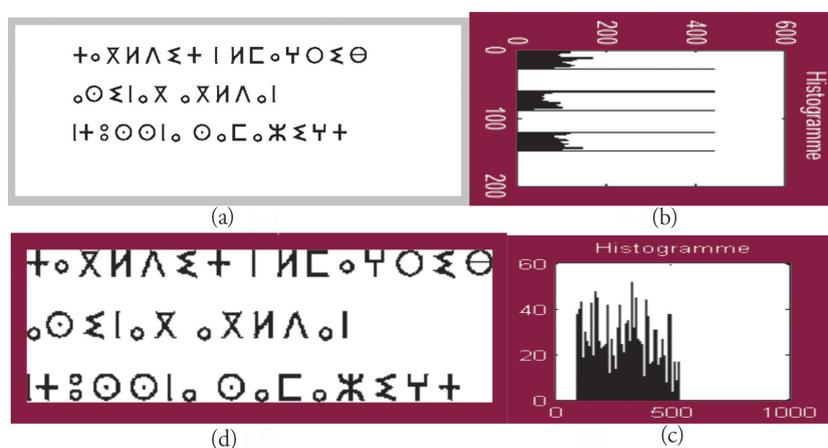


Figure 10. (a) Avant normalisation, (b) Histogramme horiz. (c) Histogramme vertical. (d) Après normalisation

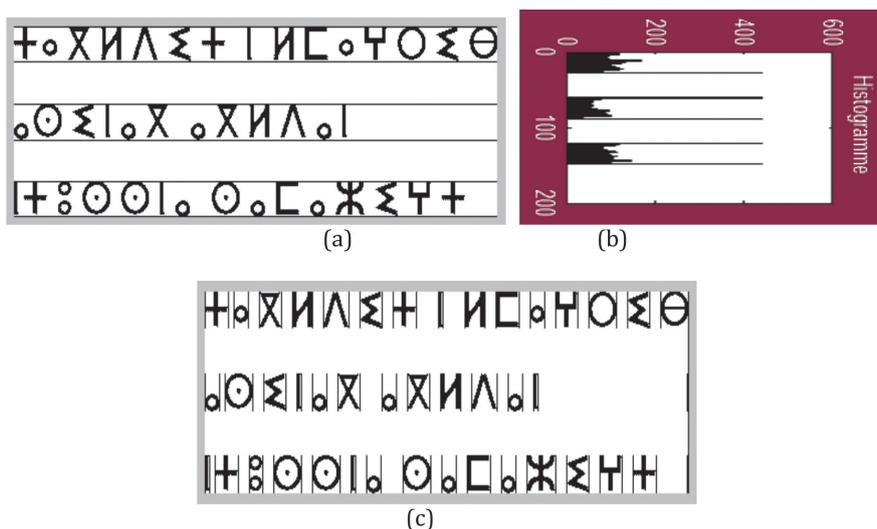


Figure 11. (a) Segmentation en lignes, (b) Histogramme horizontal, (c) Segmentation en caractères

La méthode d’histogramme rencontre un problème dans le cas de la segmentation des caractères (\times^u et κ^u), elle divise chacun de ces caractères en deux, ce qui est faux, et par conséquent, elle pose une difficulté lors de la reconnaissance. Pour remédier à ce problème, nous faisons recours à la distance d représentée dans la figure (figure 12). Tout d’abord, nous calculons la distance d , et ensuite, si $d \leq 5 \text{ pixels}$ nous annulons les colonnes 2 et 3 afin d’obtenir un seul caractère.

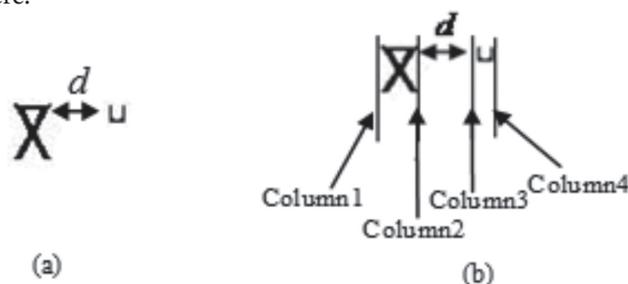


Figure 12. (a) Distance d , (b) Colonnes de segmentation

Dans le cas d’un texte incliné, la fonction de la correction d’inclinaison permet de résoudre le problème, ce qui est illustré en figure 13.

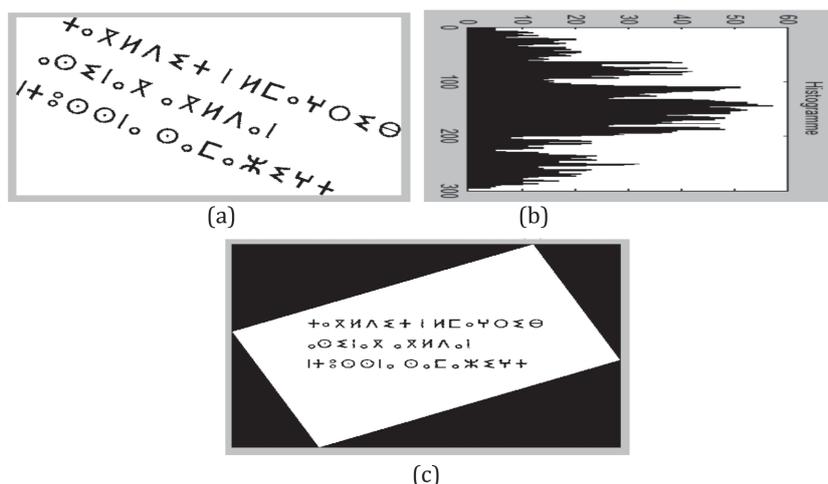


Figure 13. (a) Inclinaison, (b) Histogramme horizontal, (c) Correction d’inclinaison

La comparaison effectuée entre les deux approches appliquées dans la phase de reconnaissance (la programmation dynamique et le réseau de neurones), en se fondant sur des attributs extraits à l’aide de la transformation de Fourier et les moments invariants, est illustrée dans le tableau 3.

Transformation de Fourier et moments invariants		
	Taux de reconnaissance	Taux d'erreur
Programmation dynamique	92.76%	7.24%
Réseau de neurones	93.27%	6.73%

Tableau 3. Taux de reconnaissance et taux d'erreur

Les résultats du tableau 3 montrent que :

- Le taux de reconnaissance calculé en utilisant le réseau de neurones est supérieur au taux de reconnaissance trouvé par la programmation dynamique.
- Le taux d'erreur du réseau de neurones est inférieur au taux d'erreur de la programmation dynamique.

Dans ce système, la source des erreurs repose sur deux points :

- Problème d'invariance (\ominus et \oplus , \mathcal{Q} et \mathcal{O} , \mathcal{H} et \mathcal{V}). ie. l'existence de quelques caractères qui sont identiques en quelques situations.
- Le vecteur caractéristique extrait contient uniquement sept éléments (ce qui est insuffisant pour une reconnaissance efficace).

Pour remédier à ce problème source d'erreurs, les solutions que l'on peut proposer sont :

- Dans le cas d'invariance, la correction peut se faire à l'aide de l'utilisation de la méthode détaillée dans la partie (5) de cet article, ou bien, par l'adoption d'une méthode (phase d'extraction) qui n'a pas de problème d'invariance.
- Dans le cas du nombre d'éléments du vecteur caractéristique, on peut utiliser des méthodes qui produisent des vecteurs dont le nombre d'éléments est supérieur à sept, ou bien, regrouper plusieurs méthodes pour obtenir un seul vecteur dont le nombre d'éléments est supérieur à sept.

La base de données (Ait Ouguengay et Taalabi, 2009) utilisée est constituée de 627 images (les caractères imprimés de Tifnaghe). La configuration de l'ordinateur portable utilisée est :

Processor: Intel (R) Core (TM) 2 Duo, CPU T5870 @ 2.00 Ghz, RAM: 2.00 Go.

7. Conclusion

Dans cette recherche, nous avons traité un exemple de système de reconnaissance des caractères Tifnaghe. Ce système est composé d'un ensemble de phases : prétraitement, extraction d'attributs et reconnaissance.

La phase de prétraitement a regroupé plusieurs fonctions à appliquer sur l'image d'entrée ; la fonction de normalisation permettant l'élimination des zones indésirables afin de réduire le temps d'exécution et minimiser l'espace mémoire utilisé, la fonction de correction d'inclinaison qui facilite l'étape suivante concernant la segmentation en lignes et en caractères.

Dans l'extraction d'attributs, nous avons utilisé une méthode qui garde l'invariance du caractère en quelques situations (translation, rotation et taille), appelée moments invariants, ces moments invariants sont calculés à l'aide de la transformation de Fourier. Cette invariance a posé un problème lors de la phase de reconnaissance, ce problème est dû à l'existence des caractères de Tifnaghe identiques (s'ils ont subi une rotation par exemple). Il est résolu par l'exécution d'un traitement en parallèle avec le système de reconnaissance.

La reconnaissance est la phase la plus délicate dans un OCR, car l'efficacité de ce système se base sur la classification trouvée. C'est la raison du choix du réseau de neurones et programmation dynamique comme deux méthodes de reconnaissance grâce à leurs succès dans ce type de domaine de recherche.

Les résultats obtenus ont montré la réussite de ce système traité dans la reconnaissance des caractères Tifnaghe, c'est-à-dire le succès de l'utilisation de la transformation de Fourier dans le calcul des moments invariants.

Références

- Ait Ouguengay, Y., Taalabi, M. (2009). Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe : Phase d'apprentissage». *5e Conférence internationale sur les «Systèmes Intelligents: Théories et Applications»*, Paris: Europia, cop. 2009 (impr. au Maroc), ISBN 978-2-909285-55-3.
- Al-Hajj, R., Chafic Mokbel, M. et Likforman-Sulem, L. (2006). Reconnaissance de l'écriture arabe cursive: combinaison de classifieurs MMCs à fenêtres orientées. *Actes de CIFED'06*, Fribourg, Suisse, 271-276.
- Ali, H. (2008). A Back Propagation Neural Network Arabic Characters Classification Module Utilizing Microsoft Word. *Journal of Computer Science* 4 (9), 744-751.
- Ali, S., Al-Omari, K., Sumari, P., Al-Taweel Sadik, A. et Husain Anas, J.A. (2009). Digital Recognition using Neural Network. *Journal of Computer Science* 5 (6), 427-434.
- Amrouch, M., Rachidi, A., Elyassa, M. et Mammass, D. (2010). Handwritten Amazigh Character Recognition Based On Hidden Markov Models. *ICGST-GVIP Journal*, Vol.10, Issue 5, 11-18.
- Bencharef, O., Fakir, M., Minaoui, B. et Bouikhalene, B. (2011). Tifinagh Character Recognition Using Geodesic Distances, Decision Trees & Neural Networks. *(IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence*, 1-5.
- Blumenstein, M., C. K. Cheng et Liu, X. Y. (2002). New Preprocessing Techniques for Handwritten Word Recognition. in *Proceedings of the Second IASTED International Conference on Visualization, Imaging and Image Processing (VIIP 2002)*, ACTA Press, Calgary, 480-484.
- Bouikhalene, B., Fakir, M. et Moro, K., (2009). Reconnaissance des Caractères Tifinaghe par l'Utilisation des Réseaux de Neurones Multicouches. *SITACAM'09*, Agadir, Maroc, 12-13 Décembre.
- Chevalier, S., Geoffrois, E., et Prêteux, F. (2003). A 2D Dynamic Programming Approach for Markov Random Field-based Handwritten Character Recognition. *Proceedings IAPR International Conference on Image and Signal Processing (ICISP' 2003)*, Agadir, Maroc, 617-630.
- El Ayachi, R., Bouikhalene, B. et Fakir, M. (2011). Transformation de Fourier et Moments Invariants Appliqués à la Reconnaissance des Caractères Tifinaghe. *SITACAM'11*, Agadir, Maroc, 6-7 Mai, 41-54.
- El Ayachi, R., Fakir, M. et Bouikhalene, B. (2011). Recognition of Tifinaghe Characters Using Dynamic Programming & Neural Network. *Chapter in the book "Document Recognition and Understanding"*, ISBN 978-953-307-995-0, INTECH, 2011.
- El Ayachi, R., Fakir, M. et Bouikhalene, B. (2011). Recognition of TIFINAGHE Characters Using A multilayer Neural Network. *International Journal Of Image Processing (IJIP)*, vol. 5, Issue2, 109-118.
- El Ayachi, R., Moro, K., Fakir, M. et Bouikhalene, B. (2010). On The Recognition Of Tifinaghe Scripts. *JATIT*, vol. 20, No.2, 61-66.
- Es Saady, Y., Rachidi, A., El Yassa, M. et Mammass, D. (2011). Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character. *International Journal of Advanced Science and Technology Vol. 33*, Août, 33-50.
- Es Saady, Y., Rachidi, A., El Yassa, M. et Mammass, D. (2010). Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata. *ICGST-GVIP Journal*, Volume 10, Issue 2, Juin.
- Fakir, M. (2001). *Reconnaissance des Caractères Arabes Imprimés*. Thèse de doctorat, Faculté des Sciences, Semailia, Maroc.
- Fakir, M., Sodeyama, C. et Hassani, M. M. (2000). On the recognition of Arabic characters using Hough transform technique. *Malysian Journal of Computer Science Vol. 13, No.2*, Décembre, 39-47.
- Ghorbel, F. (1993). Application de la Transformée de Fourier Généralisée au Problème de l'invariance en Reconnaissance de Formes à Niveau de Gris. *Quatorzième Colloque GRESTI*, Juan-les Pins, 13 - 16 Septembre.
- Kavallieratou, E., Fakotakis, N. et Kokkinakis, G. (1999). Skew Angle Estimation In Document Processing Using Cohen's Class Distributions. *PRL(20)*, No. 11-13, Novembre, 1305-1311.
- Kessentini, Y., Paquet, T. et Benhamadou, A. (2010). Off-Line Handwritten Word Recognition Using Multi-Stream Hidden Markov Models. *Pattern recognition Letters (PRL)*, Vol 30, Issue 1, Janvier, 60-70.
- Outahajala, M., Benajiba, Y., Rosso, P. et Zenkouar, L. (2011). POS Tagging in Amazighe Using Support Vector Machines and Conditional Random Fields. *Natural Language Processing and Information Systems, Lecture Notes in Computer Science, Volume 6716*, 238-241.

L'étiquetage grammatical de l'amazighe en utilisant les propriétés n-grammes et un prétraitement de segmentation

Mohamed Outahajala

Laboratoire Electronique et Communication- Ecole Mohammadia d'Ingénieurs-Institut Royal de la Culture Amazighe, Maroc, outahajala@ircam.ma

Yassine Benajiba

Philips Research North America, Briarcliff Manor, USA, yassine.benajiba@philips.com

Paolo Rosso

Natural Language Engineering Lab – ELIRF, DSIC, Universidad Politécnica de Valencia, Spain, proso@dsic.upv.es

Lahbib Zenkouar

Laboratoire Electronique et Communication- Ecole Mohammadia d'Ingénieurs, Rabat, Maroc, zenkouar@emi.ac.ma

Résumé

L'objectif de cet article est de présenter le premier étiqueteur grammatical amazighe. Très peu de ressources ont été développées pour l'amazighe et nous croyons que le développement d'un outil d'étiquetage grammatical est une étape préalable au traitement automatique de textes. Afin d'atteindre cet objectif, nous avons formé deux modèles de classification de séquences en utilisant les SVMs, séparateurs à vaste marge (*Support Vector Machines*) et les CRFs, champs markoviens conditionnels (*Conditional Random Fields*) en utilisant une phase de segmentation. Nous avons utilisé la technique de 10 fois la validation croisée pour évaluer notre approche. Les résultats montrent que les performances des SVMs et des CRFs sont très comparables. Dans l'ensemble, les SVMs ont légèrement dépassé les CRFs au niveau des échantillons (92,58% contre 92,14%) et la moyenne de précision des CRFs dépasse celle des SVMs (89,48% contre 89,29%). Ces résultats sont très prometteurs étant donné que nous avons utilisé un corpus de seulement ~ 20k mots.

Abstract

The aim of this paper is to present the first amazigh POS tagger. Very few linguistic resources have been developed so far for amazigh and we believe that the development of a POS tagger tool is the first step needed for automatic text processing. In order to achieve this endeavor, we have trained two sequence classification models using Support Vector Machines (SVMs) and Conditional Random Fields (CRFs) after using a tokenization step. We have used the 10-fold technique to evaluate our approach. Results show that the performance of SVMs and CRFs are very comparable. Across the board, SVMs outperformed CRFs on the fold level (92.58% vs. 92.14%) and CRFs outperformed SVMs on the 10 folds average level (89.48% vs. 89.29%). These results are very promising considering that we have used a corpus of only ~20k tokens.

Mots-clés

étiquetage grammatical automatique, langue amazighe, TAL, apprentissage supervisé, segmentation

Keywords

automatic POS tagging, amazigh language, NLP, supervised learning, segmentation

1. Introduction

L'étiquetage grammatical¹, appelé également étiquetage morpho-syntaxique, consiste en l'annotation de chaque mot d'une phrase avec sa catégorie grammaticale. Il s'agit de la première couche au-dessus du niveau lexical et le niveau le plus bas de l'analyse syntaxique. Ainsi, le POS tagging est utilisé pour toutes les tâches du traitement automatique des langues (TAL) traitant des niveaux linguistiques supérieurs telles que l'analyse partielle, la désambiguïsation de sémantiques des mots, l'affectation des fonctions grammaticales (Cutting et al, 1992) et la reconnaissance d'entités nommées (Benajiba *et al.*, 2010a; Benajiba *et al.* 2010b). Conjointement avec l'analyse partielle, l'étiquetage grammatical est utilisé dans des tâches plus complexes (Manning et Schütze, 1999) telles que l'acquisition lexicale, l'extraction des informations, la recherche des termes d'indexation dans la récupération de l'information et les questions-réponses.

Dans la littérature, il a été montré que les approches fondées sur l'apprentissage supervisé (cf. section 2) sont les plus efficaces pour construire les étiqueteurs grammaticaux, en s'appuyant sur un corpus annoté manuellement et souvent d'autres ressources, telles que des dictionnaires et des outils de segmentation. Pour construire notre étiqueteur grammatical, nous utilisons des techniques de classification des séquences supervisées fondées sur deux approches de ces techniques, à savoir les SVMs (Support Vector Machines) et les CRFs (Conditional Random Fields). Nous nous appuyons sur un corpus de ~20k mots annotés manuellement (Outahajala *et al.*, 2011) pour former nos modèles et les caractéristiques n-grammes lexicales en vue d'une amélioration de la performance.

Le reste de l'article est organisé comme suit : à la section 2, nous présentons les travaux connexes sur les techniques d'étiquetage grammatical dans d'autres langues. Puis, dans la section 3, nous fournissons un aperçu sur la langue amazighe et le jeu d'étiquettes employé dans l'étiquetage du corpus Amazighe collecté. Dans la section 4, nous présentons les deux approches d'apprentissage supervisées utilisant les SVMs et les CRFs et qui ont été employées pour le marquage grammatical. Dans la section 5, nous décrivons les expériences et nous discutons les résultats. Enfin, dans la section 6, nous dressons quelques conclusions et nous présentons les travaux à effectuer dans le futur proche.

2. Etat de l'art

De nombreux systèmes pour l'étiquetage automatique des parties du discours ont été développés pour un large éventail de langues. Parmi ces systèmes, certains s'appuient sur les règles linguistiques et d'autres sur des techniques d'apprentissage automatique (Manning & Schütze, 1999). Les premiers POS taggers étaient principalement à base de règles. La construction de tels systèmes nécessite un travail considérable afin d'écrire manuellement les règles et de coder les connaissances linguistiques qui régissent l'ordre de leur application. Un exemple d'étiqueteur à base de règles est TAGGIT, développé par Green et Robin (Greene, Rubin, 1971) et contenant environ 3300 règles. Ce système atteint une précision de 77%. Par la suite, l'apprentissage automatique des étiqueteurs s'est avéré à la fois moins pénible et plus efficace que ceux à base de règles. Dans la littérature, de nombreuses méthodes d'apprentissage automatique ont été appliquées avec succès pour réaliser des POS taggers. Nous les citons dans ce qui suit.

- Les Modèles de Markov Cachés (HMM) (Charniak, 1993) dont les états sont des balises ou des tuples de balises. Pour un tagger bigrammes par exemple, les états de l'HMM sont les balises, les probabilités de transition sont les probabilités d'une étiquette donnant l'étiquette précédente et les probabilités d'émission sont les probabilités d'un mot sachant une étiquette donnée.
- La transformation système fondée sur la réduction du taux d'erreur (Brill, 1995) consiste en l'affectation de l'étiquette la plus fréquente d'un mot donné en utilisant un corpus de référence. Elle procède par la suite en sélectionnant la règle qui donne la plus grande erreur. Ce processus est itéré tant que les résultats d'annotation ne sont pas suffisamment proches de celles du corpus de référence.
- Les arbres de décision permettent de construire un outil d'aide à la décision (Schmid, 1999) sur la base d'un corpus de référence qui utilise ce modèle. La meilleure étiquette attribuée à un mot donné est celle qui donne la plus forte probabilité conditionnelle pour le nœud courant de cet arbre.

¹ La tradition anglophone utilise le terme part-of-speech tagging (POS tagging), c'est-à-dire étiquetage des parties du discours.

- Le modèle d'entropie maximale (Ratnaparkhi, 1996) permet la combinaison de diverses formes d'informations contextuelles sans imposer aucune hypothèse sur les données d'entraînement. Son but est de maximiser l'entropie de la distribution d'un mot à certaines contraintes contenues dans le corpus de référence.
- Les méthodes d'apprentissage automatique permettent de construire des modèles complexes (comportant de très nombreux paramètres), difficiles à faire manuellement. La qualité des modèles est souvent liée à la quantité de données utilisées dans l'apprentissage. Ainsi à partir d'exemples appris précédemment, les programmes s'appuyant sur ces méthodes affectent l'étiquette aux mots selon le contexte (Schmid, 1994; Kudo, Matsumoto, 2000; Lafferty *et al.* 2001).
- Les méthodes hybrides qui utilisent à la fois des règles à base de connaissances linguistiques codées manuellement et les méthodes d'apprentissage automatique.

Les résultats annoncés utilisant ces méthodes sont supérieurs à 95%. Bien que ces méthodes aient une bonne performance, la précision des mots inconnus est beaucoup plus faible que celle des mots connus, ce qui est problématique lorsque le corpus d'apprentissage est de petite taille. La taille du jeu d'étiquettes varie considérablement selon la langue et la finalité de la tâche d'étiquetage voulue. Leech (1997) montre que le nombre d'étiquettes varie de 32 à 270 dans les principaux corpus anglais. Dans la pratique, la plupart des analyseurs limitent le nombre d'étiquettes en ignorant certaines distinctions difficiles à désambiguïser automatiquement, ou sujettes à discussion du point de vue linguistique.

3. L'amazighe

Dans cette section, nous présenterons un bref aperçu sur la langue amazighe avant de décrire le jeu de balises adoptées dans nos expérimentations.

3.1. La langue amazighe

La langue amazighe est parlée au Maroc, en Algérie, en Libye, en Tunisie et en Égypte (oasis de Siwa). Elle est également parlée par d'autres communautés dans certaines régions du Niger et du Mali. L'amazighe est un composite de dialectes dont aucun n'a été considéré comme la norme nationale dans aucun des pays déjà susmentionnés. Avec l'émergence de la revendication identitaire, les locuteurs natifs militent pour la sauvegarde et la promotion de leur langue et de leur culture. Pour atteindre un tel objectif, certains États du Maghreb ont créé des institutions spécialisées, telles que l'Institut Royal de la Culture Amazighe (IRCAM, désormais) au Maroc et le Haut-commissariat de l'Amazighité en Algérie. Au Maroc, l'amazighe a été introduite dans les médias et dans le système éducatif. En conséquence, l'Alphabet Tifinaghe a été reconnu officiellement par le consortium Unicode le 05/07/2004, une nouvelle chaîne de télévision amazighe a été lancée le 1^{er} mars 2010, un peu plus de 3 000 écoles primaires marocaines enseignent l'Amazighe à plus de 600 000 élèves. Au niveau de l'enseignement supérieur, des filières d'études amazighes et des masters ont été créés. Le 01 Juillet 2011, les Marocains ont voté favorablement pour la nouvelle constitution du pays qui octroie le statut langue officielle à l'amazighe.

Nous remarquons une croissance notable du nombre de publications traitant de la langue et de la culture amazighes. Cependant, en TAL, la langue amazighe, comme la plupart des langues non européennes, souffre encore de la pénurie d'outils et des ressources pour son traitement automatique.

Sur le plan linguistique, la langue est caractérisée par la prolifération des dialectes en raison de facteurs historiques, géographiques et sociolinguistiques. Au Maroc, par exemple, on peut distinguer trois principaux dialectes : le tarifite dans le Nord, le tamazighte dans le centre et le tachlhitte dans le sud du pays.

En raison de sa morphologie complexe (Chafiq, 1991 ; Boukhris *et al.* 2008) ainsi que de l'utilisation des différents dialectes dans sa normalisation, la langue amazighe présente des défis intéressants à prendre en compte pour les chercheurs en Traitement Automatique des Langues (TAL). Nous en citons dans ce qui suit quelques-uns.

- L'amazighe dispose de sa propre graphie, le Tifinaghe qui s'écrit de gauche à droite.
- Le Tifinaghe ne contient pas de majuscules.
- Les noms, les noms de qualité (adjectifs), les verbes, les pronoms, les adverbes, les prépositions, les focaliseurs, les interjections, les conjonctions, les pronoms, les particules et les déterminants consistent en un seul mot se produisant entre deux blancs ou des signes de ponctuation. Toutefois, si une préposition ou un nom de parenté est suivi d'un pronom personnel, l'ensemble préposition/nom de parenté et le pronom qui suit forment une chaîne unique délimitée par des espaces ou des signes de ponctuation. Par exemple: □□ (□r) signifiant «pour, au» + □ (i) qui signifie «moi» (pronom personnel) donnent «□□□□/□□□□ (□ari/□uri)».
- Les signes de ponctuation amazighe sont semblables aux signes de ponctuation adoptés au niveau international et ont les mêmes fonctions.
- L'amazighe, à l'instar d'autres langues naturelles, peut présenter des ambiguïtés au niveau des

classes grammaticales. En effet, la même forme de surface peut appartenir à plusieurs catégories grammaticales selon le contexte dans la phrase. Par exemple, ⵜⵜⵜⵜ (illi) peut fonctionner comme un verbe à l'accompli négatif signifiant «il n'existe pas» ou comme nom de parenté signifiant «ma fille». Quelques mots tels que «ⵜ» (d) peuvent fonctionner comme préposition ou comme conjonction de coordination ou comme particule de prédication ou d'orientation.

- L'amazighe est peu dotée en ressources langagières et outils du TAL au même titre que la majorité des langues dont les recherches en TAL ont récemment commencé.

3.2. Jeu d'étiquettes

Définir le jeu de balises adéquat est une tâche essentielle dans la construction d'un POS tagger automatiquement. Il vise à définir un ensemble de balises traitables qui ne soit ni grand et nuire à la performance de l'apprentissage automatique, ni petit et n'offrir pas suffisamment d'informations pour être utilisé par les systèmes fédérateurs. Dans (Outahajala *et al.*, 2010), un ensemble contenant 13 balises (verbe, nom, adverbe, etc.) a été conçu. Pour chaque élément, les caractéristiques morphosyntaxiques et deux attributs communs ont été définis: «wd» pour mot et «lem» pour lemme, dont les valeurs dépendent de l'item lexical en question. L'ensemble des balises utilisées comprend 13 étiquettes représentant les principales parties du discours dans la langue amazighe, comme il est résumé dans le tableau 1. Ce jeu de balises est un sous-ensemble du jeu d'étiquettes présentées dans (Outahajala *et al.*, 2010). Le genre, la personne, l'aspect et d'autres informations n'ont pas été inclus dans ce jeu d'étiquettes et ont été considérés comme une piste de recherche à poursuivre à l'avenir.

Classe	Désignation
V	Verbe
N	Nom
A	Nom de qualité/Adjectif
AD	Adverbe
C	Conjonction
D	Déterminant
S	Préposition
FOC	Focalisateur
I	Interjection
P	Pronom
PR	Particule
R	Résiduel (nom étranger, nombre, date, monnaie, signe mathématique, etc.)
F	Ponctuation

Tableau 1. Jeu d'étiquettes

4. Apprentissage supervisé pour l'étiquetage grammatical

Dans cette section, nous décrivons les fondements théoriques de l'apprentissage supervisé en général, et des SVMs et des CRFs en particulier. Les bonnes performances de ces derniers ont été prouvées dans les problèmes de classification des séquences.

4.1. Apprentissage supervisé

En apprentissage supervisé, l'objectif est d'apprendre une fonction

$$h: X \rightarrow Y \quad (1)$$

où $x \in X$ sont les entrées et $y \in Y$ sont les sorties.

Les objets d'entrée appelés instances ou exemples peuvent être de tout type, selon la tâche d'apprentissage particulière. En TAL, une entrée peut être un classement des documents, des chaînes de mots à étiqueter avec une séquence d'étiquettes (ce qui est notre cas).

Selon la nature de l'espace Y de sortie, les tâches d'apprentissage peuvent être classées en plusieurs types :

- Classification binaire avec $Y = \{-1, +1\}$
- Multi classification avec $Y = \{1, \dots, K\}$ (ensemble fini de balises)
- Régression avec $Y = \mathbb{R}$
- Prédiction structurée dans le cas où les sorties sont complexes. Par exemple, dans une tâche d'étiquetage des séquences telles que le POS-tagging, $Y = \{1 \dots, K\}$ signifie que la sortie est une séquence d'étiquettes de longueur égale à la longueur de la chaîne d'entrée.

4.2. Les séparateurs à vaste marge

Les SVM ou séparateurs à vaste marge ont été introduits par Vapnik (1995). Ils sont connus pour leur performance et leur bonne généralisation. Ils ont été utilisés pour des problèmes de reconnaissance différents et ont donné de bons résultats en pratique.

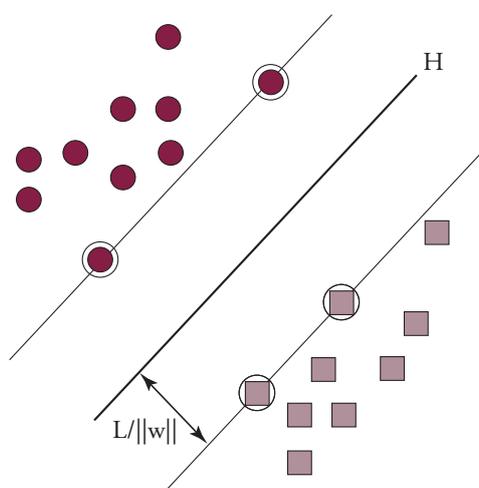


Figure.1. Séparation des régions par un hyperplan

L'objectif des SVMs étant de trouver un hyperplan optimal, on les appelle « séparateurs à vaste marge ». La marge est la distance entre la frontière de séparation et les échantillons les plus proches (Figure.1) qui sont appelés vecteurs supports. Dans les SVMs, la frontière de séparation est choisie de sorte à maximiser la marge. Le problème est de trouver cette frontière séparatrice optimale, à partir d'un ensemble d'apprentissages. Ceci est fait en formulant le problème comme un problème d'optimisation.

En TAL, les SVMs sont appliquées à la catégorisation de textes (Kudu, Matsomoto, 2000), l'analyse partielle (Diab *et al.*, 2007), la reconnaissance d'entités nommées (Benajiba *et al.*, 2010), etc. Plusieurs étiqueteurs grammaticaux ont été réalisés sur la base des SVMs. C'est le cas de l'arabe (Diab *et al.*, 2004; Diab *et al.*, 2007), du bengali (Ekbal, Bandyopadhyay, 2008), etc. Les SVMs atteignent des performances élevées sans apprentissage même en utilisant plusieurs caractéristiques. Ils réagissent également bien avec les données éparses et bruitées.

En ce qui concerne la tâche de l'étiquetage grammatical de l'amazighe, le processus d'apprentissage a été conduit en utilisant YamCha², un outil fondé sur les SVMs. Pour la classification, nous avons utilisé TinySVM³, un outil public pour la reconnaissance des motifs.

4.3. Les champs markoviens conditionnels

Les CRFs ou champs markoviens conditionnels sont des modèles probabilistes discriminants introduits par (Lafferty *et al.*, 2001) pour l'annotation séquentielle. Il s'agit de graphes non orientés. Etant donnée une séquence d'observation, le modèle conditionnel indique les probabilités de séquences d'étiquettes possibles. En plus des avantages des MEMMs, les CRFs peuvent être imaginés comme un modèle à états finis avec des probabilités de

² <http://chasen.org/~taku/software/yamcha/>

³ <http://chasen.org/~taku/software/TinySVM/>

transition non normalisée. Ils sont appliqués dans de nombreuses tâches du TAL, telles que la reconnaissance d'entités nommées (Benajiba *et al.*, 2010), l'analyse syntaxique partielle (Sha, Pereira, 2003) et l'extraction d'information à partir de tables (Pinto *et al.*, 2003). En relation avec l'étiquetage grammatical, les CRFs ont été utilisés pour de nombreuses langues, telles que l'Amharique (Adafre, 2005) et le Tamoul (Lakshmana, Geetha, 2009). Les CRFs sont définis par des champs aléatoires X et Y décrivant respectivement chaque unité de l'observation et son annotation, et par un graphe $G = (V, E)$ dont V est l'ensemble des nœuds et E l'ensemble des arcs, avec $V = X \cup Y$. Deux variables sont reliées dans le graphe si elles dépendent l'une de l'autre. Chaque étiquette dépend des étiquettes précédentes et suivantes.

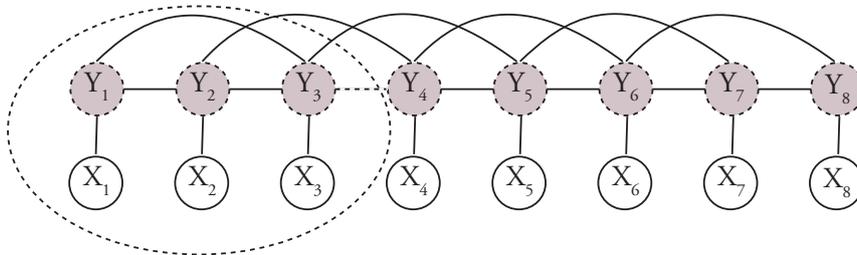


Figure.2. Exemple d'un graphe des CRFs

Dans un CRF, étant données une observation x et une annotation y on a la relation suivante (Lafferty et al, 2001) :

$$p(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \exp \left(\sum_k \lambda_k f_k(y_c, x, c) \right) \quad (2)$$

où

- C est l'ensemble des cliques -sous-graphes complètement connectés- de G
- y_c est l'ensemble des valeurs prises par les variables de Y sur la clique c
- $Z(x)$ est un coefficient de normalisation défini de telle sorte que la somme sur y de toutes les probabilités $p(y|x)$ pour une donnée x fixée soit égale à 1
- f_k sont des fonctions caractéristiques définies à l'intérieur de chaque clique c et sont à valeurs réelles. La valeur de ces fonctions peut dépendre des étiquettes présentes dans une certaine clique c ainsi que de la valeur de l'observation x
- λ_k sont des poids à valeurs réelles. Ils permettent de fixer l'importance à chaque fonction f_k . Leurs valeurs sont fixées lors de la phase d'apprentissage en cherchant à maximiser la log-vraisemblance sur des exemples déjà annotés formant le corpus de référence.

Nous avons utilisé l'outil CRF++⁴, une implémentation open source des CRFs pour la segmentation et l'étiquetage des données, en utilisant le même ensemble de données que celui utilisé avec Yamcha.

5. Expérimentations et analyse des erreurs

5.1. Corpus

Notre corpus se compose de textes extraits d'une variété de sources telles que la version amazighe du site Web de l'IRCAM⁵, le périodique «Inghmisen n usinag» (bulletin d'information de IRCAM) et des manuels scolaires. Les textes ont été annotés en utilisant l'outil AnCoraPipe (Bertran et al., 2008). La vitesse d'annotation de ce corpus est de 80 et 120 mots par heure. Afin de calculer l'accord entre les annotateurs, des textes ont été choisis aléatoirement et révisés par des annotateurs différents. Le résultat de cette expérience a donné un accord de 94,98%. Les remarques et les corrections communes ont été généralisées à l'ensemble du corpus dans la deuxième validation par un annotateur autre que celui qui a fait l'annotation. Le format d'entrée pour Yamcha et CRF++ est le même (voir Figure 3 et Figure 4 ci-dessous). La première colonne de l'entrée est formée des mots de notre corpus, la dernière colonne est réservée à la classe grammaticale, et le cas échéant les caractéristiques lexicales n-grammes sont mises dans les colonnes entre les deux (voir Figure 4). i.e., les première et dernière colonnes.

⁴ <http://crfpp.sourceforge.net/>

⁵ <http://www.ircam.ma/>

Nous décrivons ci-dessous un exemple dans lequel nous n'utilisons pas de segmentation du format d'entrée pour la phrase « *ar as ttHyyaln i tmGra ann sg usggwas lli izrin* ». [En Français : Ils se préparaient au mariage depuis l'année dernière]:

ar	PR
as	S_P
ttHyyaln	V
i	S
tmGra	N
ann	D
sg	S
usggwas	N
lli	P
izrin	V
.	F

Figure 3. Un extrait à partir du corpus d'apprentissage

ar	a	-	-	-	r	-	-	-	PR
as	a	-	-	-	s	-	-	-	S_P
ttHyyaln	t	tt	ttH	ttHy	n	ln	aln	yaln	V
i	-	-	-	-	-	-	-	-	S
tmGra	t	tm	tmG	tmGr	a	ra	Gra	mGra	N
ann	a	an	-	-	n	nn	-	-	D
sg	s	-	-	-	g	-	-	-	S
usggwas	u	us	usg	usgg	s	as	was	gwas	N
lli	l	ll	-	-	i	li	-	-	P
izrin	i	iz	izr	izri	n	in	rin	zrin	V
.	-	-	-	-	-	-	-	-	F

Figure 4. Un extrait à partir du corpus d'apprentissage en utilisant les propriétés lexicales

Dans cet article, nous explorons deux groupes d'expériences, basés sur les SVMs et les CRFs, avec ou sans propriétés lexicales (voir paragraphe 5.2). Dans la première expérimentation, nous ne segmentons pas les mots contenant plusieurs morphèmes lexicaux, et nous utilisons «S_P» et «N_P» pour désigner respectivement les prépositions et les noms de parenté lorsqu'ils sont suivis des pronoms personnels. Dans la deuxième expérimentation, nous segmentons les prépositions et les noms de parenté quand ils sont utilisés avec les pronoms personnels. Toutefois, ceci est un cas problématique, car la fonction inverse n'est pas déterministe. En effet, si par exemple l'union des deux morphèmes «dg» et «s» peut donner soit «digs» ou «dags», signifiant [en lui/elle]. Ainsi, une fois nous avons subdivisé le mot composé en ses morphèmes constituants, il n'est plus possible de calculer la forme originale après l'étiquetage grammatical en ne se basant que sur la sortie du POS tagger. Une solution à cette question est de prendre la forme la plus utilisée dans le corpus en se basant sur la fréquence des mots, ainsi on prendra pour ce cas la forme «digs». Dans toutes nos expériences, nous avons utilisé deux ensembles de balises celui présenté ci-dessus (voir le tableau 1) et ce même ensemble plus les deux balises «S_P» et «N_P» correspondant aux prépositions et aux noms de parenté lorsqu'ils sont employés avec les pronoms.

5.2. Caractéristiques pour l'étiquetage

Dans cet article, nous explorons deux ensembles de caractéristiques basées sur le texte réel et faciles à extraire. Dans le premier sous-ensemble d'expérimentations, nous utilisons les mots qui entourent le mot à étiqueter ainsi que leurs étiquettes dans une fenêtre de +/- 2 mots (Figure 3). Dans le deuxième sous-ensemble (illustré dans la Figure 4), nous ajoutons aux premières caractéristiques les propriétés lexicales n-grammes du mot à étiqueter et des mots entourant ce mot, avec la même fenêtre de +/- 2. Les propriétés n-grammes se composent des i premiers et i derniers n-grammes caractères, avec i variant de 1 à 4.

5.3. Les expérimentations

Dans nos premières expériences sur l'étiquetage grammatical (Outahajala *et al.* 2011b), nous avons montré que la courbe d'apprentissage croît avec l'augmentation de la taille du corpus d'entraînement.

Partie#	SVMs	SVMs (avec les propriétés lexicales)	CRFs	CRFs (avec les propriétés lexicales)
0	81,01	86,86	83,19	86,95
1	76,02	83,86	80,7	84,98
2	85,64	91,66	87	90,86
3	82,56	88,34	86,45	88,58
4	83,55	88,24	85,8	88,87
5	83,28	89,99	86,24	90,48
6	76,59	85,38	79,98	85,38
7	79,07	86,6	81,79	87,96
8	87,35	91,38	88,88	91,14
9	84,64	90,41	86,79	91,35
Moyenne	81,97	88,27	84,68	88,66

Tableau 2. Résultats de la validation croisée en 10 des parties

Nous avons mené et évalué nos expériences en utilisant une validation croisée en 10 parties, i.e. l'entraînement avec 90% du corpus de référence et l'utilisation de 10% pour le test, en répétant l'expérience dix fois et en prenant à chaque fois une tranche différente du corpus.

Partie#	SVMs	SVMs (avec les propriétés lexicales)	CRFs	CRFs (avec les propriétés lexicales)
0	82,85	87,94	84,46	87,31
1	78,27	85,06	81,55	85,9
2	87,59	92,58	87,9	91,42
3	83,95	89,62	87,39	89,22
4	85,06	89,02	86,93	89,26
5	86,08	91,38	87,6	91,62
6	79,27	86,42	82,9	87,18
7	81,34	86,96	83,69	88,96
8	88,54	92,47	89,32	91,79
9	86,45	91,49	88,65	92,14
Moyenne	83,93	89,29	86,01	89,48

Tableau 3. Résultats de la validation croisée en 10 parties après une phase de segmentation

Vu l'impact positif de la segmentation, comme prétraitement, à l'instar de ce qui se fait pour d'autres langues comme l'arabe (Diab *et al.*, 2004), nous avons subdivisé les prépositions et les noms de parenté lorsqu'ils sont utilisés conjointement avec les pronoms personnels, et nous avons obtenu de meilleurs résultats, comme il est indiqué dans le tableau 4. Par la suite, vu l'amélioration des résultats obtenus en utilisant cette phase de segmentation comme prétraitement à l'étiquetage, nous avons décidé de réaliser un étiqueteur, en s'appuyant sur les méthodes

d'apprentissage automatique. Pour ce faire, nous avons entraîné deux modèles à base de séquences d'étiquettes en utilisant cinq étiquettes: {B-WORD, I-WORD, B-SUFF, I-SUFF, O}. Le corpus utilisé pour l'entraînement de ce segmenteur a été construit de manière semi-automatique. Sur la base d'expérimentations, nous avons constaté que les performances des SVMs et des CRFs sont très comparables. Le modèle à base des SVMs a légèrement dépassé celui basé sur les CRFs (99.95% contre 99,89%).

5.4. Expérimentations et discussion des résultats des étiqueteurs

Pour une meilleure compréhension du comportement du système, nous avons examiné la matrice de confusion pour l'expérience qui a donné la plus grande précision. L'analyse de la matrice de confusion présente toutes les étiquettes erronées comme le montrent les tableaux 4 et 5.

	N	A	V	P	D	S	C	AD	PR	FOC	F	I	R
N	93,1	0,3	1,8	0,6	3,9	0	0	0	0	0	0,3	0	0
A	18,2	63,6	18,2	0	0	0	0	0	0	0	0	0	0
V	5,4	0,3	93	0	0	0,7	0	0	0,7	0	0	0	0
P	0,7	0	0,7	91	5,5	0,7	0,7	0	0,7	0	0	0	0
D	3,3	0	1,1	9,9	84,6	0	0	1,1	0	0	0	0	0
S	0,5	0	1	0,5	0	94	2,1	0,5	1,6	0	0	0	0
C	0	0	0	2,1	2,1	2,1	83,3	4,2	4,2	2,1	0	0	0
AD	23,2	0	7,1	1,8	1,8	3,6	1,8	60,7	0	0	0	0	0
PR	0	0	0	0	1,9	0,6	0	0,6	96,8	0	0	0	0
FOC	0	0	0	0	40	0	0	0	0	60	0	0	0
F	0	0	0	0,2	0	0	0	0	0	0	99,8	0	0
I	36,4	0	0	0	0	0	0	0	18,2	0	0	45,4	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0

Tableau 4. La matrice de confusion en pourcentage en utilisant les SVMs avec les caractéristiques lexicales

En analysant les erreurs les plus fréquentes dans les deux matrices de confusion, nous avons constaté que les adjectifs sont souvent étiquetés comme substantifs. Ceci est dû au fait que les adjectifs peuvent être utilisés comme des noms. En optant pour une non-distinction entre les noms et les adjectifs, nous obtenons une amélioration de 0,73 et un meilleur score de 90,02% en utilisant la validation croisée en 10 parties avec les SVMs. La même expérience avec les CRFs permet une amélioration de 0,77 et un meilleur score de 90,25%.

	N	A	V	P	D	S	C	AD	PR	FOC	F	I	R
N	94,6	2,4	2,1	0,2	0,3	0,1	0,2	0	0	0	0	0	0,1
A	12,6	82,3	4,6	0	0	0	0	0	0	0,6	0	0	0
V	2,6	1,5	93,3	0	0,4	1,5	0	0,4	0,4	0	0	0	0
P	3,7	0	0	75	13,9	0,9	0,9	0,9	3,7	0,9	0	0	0
D	2,4	0	0	4,8	82,5	0	0	2,4	7,9	0	0	0	0
S	0	0	0,2	0,3	0	99	0,5	0	0	0	0	0	0
C	1,7	0	0,6	0	0,6	2,9	91,4	0	2,9	0	0	0	0
AD	23,8	0	0	9,5	0	9,5	14,3	42,9	0	0	0	0	0
PR	0	0	1,1	1,1	2,3	1,1	9,2	1,1	83,9	0	0	0	0
FOC	0	0	0	0	0	0	0	0	50	50	0	0	0

F	0	0	0	0	0	0	0	0	0	0	100	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0
R	3,1	0	1,6	1,6	0	4,7	0	0	4,7	0	6,3	0	78,1

Tableau 5. La matrice de confusion en pourcentage en utilisant les CRFs avec les caractéristiques lexicales.

Le taux d'erreur des pronoms est également élevé en raison du chevauchement important entre eux et les déterminants. Les verbes constituent une autre source commune d'erreurs. Comme le montre le Tableau 5, le POS tagger basé sur les CRFs a étiqueté 4,1% des verbes comme des noms et des adjectifs et de 1,6% comme des prépositions, alors que le POS-tagger basé sur les SVMs a étiqueté 5,7% des verbes comme des noms et des adjectifs. Pour les autres classes, le POS-tagger basé sur les SVMs a de meilleurs résultats dans l'étiquetage des pronoms, des déterminants, des adverbes, des focalisateurs et des particules.

Certaines particules ont un niveau d'ambiguïté important, par exemple la particule \square (d) qui a plusieurs balises possibles, en fonction du contexte. Par exemple, le mot «d» pourrait être :

- Une conjonction de coordination: «tamaziGt d tiknulujiyin timaynutin» [l'amazighe et les nouvelles technologies],
- Une préposition: «iman d ubrid» [il a pris le chemin],
- Une particule de prédication: «d argaz» [il est un homme]
- Une particule d'orientation: «asi d tikint tamjahdit» [prend le grand bol].

En analysant les corpus d'entraînement et de test, nous avons observé que les mots inconnus sont importants. Les mots mal classés du corpus de test sont invisibles dans le corpus d'entraînement. Aussi, certaines erreurs existent-elles encore dans notre corpus manuellement annoté.

6. Conclusions et travaux futurs

Dans cet article, nous avons essayé de décrire les caractéristiques morphosyntaxiques de la langue amazighe. Nous avons abordé la conception de deux jeux d'étiquettes et deux étiqueteurs grammaticaux basés sur les SVMs et les CRFs. La précision obtenue a atteint 92,58%. Pour valider nos résultats, nous avons utilisé un petit corpus annoté manuellement d'environ ~20k mots, les caractéristiques lexicales et une phase de prétraitement de segmentation et la technique de 10 fois la validation croisée. Le tagueur grammatical utilisant les CRFs a atteint 89,48% alors que celui basé sur les SVMs a atteint une précision de 89,29%.

Comme perspective, nous visons l'amélioration des performances de l'étiqueteur en élargissant le jeu d'étiquettes en utilisant plus de données étiquetées et obtenues avec les techniques d'apprentissage semi-supervisé et l'apprentissage actif.

Remerciements: nous tenons à remercier tous les chercheurs linguistes de l'IRCAM de leur aide précieuse. Les travaux du troisième auteur ont été financés par le projet de recherche EU FP7 Marie Curie PEOPLE-IRSES 269180 WiQ-Ei, MICINN TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 (Plan I+D+i), VLC/CAMPUS Microcluster on Multimodal Interaction in Intelligent Systems.

Références

- Adafre, S. F. (2005), Part of Speech tagging for Amharic using Conditional Random Fields. *In Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pp. 47-54.
- Benajiba Y., Diab M., Rosso P. (2010a), Arabic Named Entity Recognition: A Feature-Driven Study. In: IEEE Transactions on Audio, Speech and Language Processing, vol. 15, num. 5. *Special Issue on Processing Morphologically Rich Languages*, pp. 926-934. DOI: 10.1109/TASL.2009.2019927.
- Benajiba Y., Zitouni I., Diab M., Rosso P. (2010b), Arabic Named Entity Recognition: Using Features Extracted from Noisy Data. *In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL-2010*, Uppsala, Sweden, July 11-16, pp. 281-285.
- Boukhris, F. Boumalk, A. El moujahid, E., Souifi, H. (2008). *La nouvelle grammaire de l'amazighe*. Publications de l'IRCAM.
- Bertran, M., Borrega, O., Recasens, M., Soriano, B. (2008), AnCoraPipe A tool for multilevel annotation. *Procesamiento del lenguaje Natural*, n° 41. Madrid, Spain.

- Brants, T. (2000), TnT - A Statistical Part-of-Speech Tagger. In *Proceedings. Of the 6th Applied Natural Language Processing Conference*. Seattle, USA.
- Brill, E. (1995), Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*. vol 21, pp. 543—565.
- Chafiq, M. (1991), *ةيغيزامأل ي ف اسرد نوعرأو ةعبرأ*. éd. Arabo-africaines.
- Charniak, E. (1993), *Statistical Language Learning*. MIT Press, Cambridge
- Cutting, D., Kupiec, J., Jan Pedersen, J. Sibun, P. (1992), Practical Part-of-Speech Tagger. Xerox Palo Alto Research Center.
- Diab, M., Hacioglu, K., Jurafsky, D. (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. *Proceedings of Human Language Technology-North American Association for Computational Linguistics (HLT-NAACL)*.
- Diab, M., Hacioglu, K., Jurafsky, D. (2007), Arabic Computational Morphology: Knowledge-based and Empirical Methods, chapter 9. Springer.
- Ekbal, A.; Bandyopadhyay, S. (2008), Part of Speech Tagging in Bengali Using Support Vector Machine. In *Information Technology, ICIT '08*, pp. 106-111.
- Greene, B.B., and Rubin, G.M. (1971), Automatic Grammatical Tagging of English. Department of Linguistics, Brown University, Providence, R.I.
- Kudo, T., Yuji Matsumoto, Y. Use of Support Vector Learning for Chunk Identification. *Proceeding of CoNLL-2000 and LLL-2000*.
- Lafferty, J. McCallum, A. Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML-01*, pp. 282-289. 2001.
- Lakshmana Pandian S., Geetha T. V. (2009), CRF Models for Tamil Part of Speech Tagging and Chunking. In: *Proc. ICCPOL '09*. Springer-Verlag Berlin, Heidelberg
- Manning, C., Schütze, H. (1999), *Foundations of Statistical Natural Language Processing*. The MIT Press.
- Leech, G. (1997). Introduction to corpus annotation. In R. Garside, G., Leech, & A. McEnery (Eds.), *Corpus annotation: Linguistic information from computer text corpora*. pp. 1-18. London: Longman.
- Outahajala M., Zenkour L., Rosso P., Martí A. (2010), Tagging Amazighe with AncoraPipe. In: *Proc. Workshop on LR & HLT for Semitic Languages, 7th International Conference on Language Resources and Evaluation, LREC-2010*, Malta, May 17-23, pp. 52-56.
- Outahajala M., Zenkour L., Rosso P. (2011a), Building an annotated corpus for Amazighe. *Will appear In Proc. of 4th International Conference on Amazigh and ICT*. Rabat, Morocco.
- Outahajala M., Benajiba Y., Rosso P., Zenkour L. (2011b), POS tagging in Amazighe using Support Vector Machines and Conditional Random Fields. In. *Proc. of 16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011, LNCS(6716)*, Springer-Verlag, pp. 238-241.
- Pinto, D., McCallum, A., Wei, X., Croft. W. B. (2003), Table extraction using conditional random fields. In *SIGIR '03: Proceedings of the 26th annual international*, pp. 235-242, New York, USA.
- Ratnaparkhi, A. (1996), A Maximum Entropy Model for Part-Of-Speech Tagging. In *proc. of EMNLP*, Philadelphia, USA.
- Schmid, H. (1999), Improvements in Part-of-Speech Tagging with an Application to German. In *Proc. of the ACL SIGDAT-Workshop*. Academic Publishers, Dordrecht, 13-26.
- Sha, F. and Pereira F. (2003), Shallow Parsing with Conditional Random Fields. In *Proc. of Human Language Technology*.
- Schmid, H. (1994). Part-of-speech tagging with neural networks. In *Proc. of international conference on Computational Linguistics*, Kyoto, Japan.
- Vapnik, Valdimir N. (1995). *The Nature of Statistical Learning Theory*. Springer Verlag, New York, USA.
- Zenkour L. (2008), Normes des technologies de l'information pour l'ancrage de l'écriture Amazighe. *revue Etudes et Documents Berbères n°27*, pp. 159-172.

Recherche et développement

Systeme d'information voyageur à base d'agents pour la recherche d'itinéraires multimodaux

Youcef Belgueliel

Département d'informatique, Université Mohamed Boudiaf, BP 1583 RP M'sila, 28000 Algérie,
Bel_you28@yahoo.fr

Ramdane Maamri

Laboratoire Lire, département d'informatique, Université Mentouri, Route Ain Elbey, Constantine,
25000 Algérie, Rmaamri@yahoo.fr

Résumé

Avec le développement des technologies de l'information et de communication, les chercheurs et les industriels s'intéressent à l'amélioration de la qualité de service offert aux usagers du transport en commun. Il s'agit essentiellement d'améliorer la qualité de l'information de transport en termes de facilité d'accès, de disponibilité et d'efficacité, pour aider les clients à se déplacer moyennant un système d'information voyageur. Dans cette optique, nous nous appuyons sur la théorie multi-agents. Notre travail qui vise à concevoir un système d'information voyageur d'aide au déplacement multimodal, a pour but d'utiliser une approche de composition des services Web pour calculer l'itinéraire le plus court en termes de coût, de temps de parcours et de confort, en vue d'offrir une assistance les voyageurs.

Abstract

With the development of information and communications technology, researchers and industrialists are interested to improve the quality of service provided to public transportation users. It is essentially a question of improving the quality of transport information in terms of access facility, availability and efficiency in order to help customers to move to an information system traveler. In this context, the aim of our project, founded on multi-agents paradigm, is to design a Traveler Information System supporting the multimodal movement, and more specifically to calculate the shortest route in terms of cost, of time (weather) of crosses (goes through) and of comfort, to assist the travellers by using a new approach of Web services composition.

Mots-clés

systeme d'information, information multimodale, systeme multi-agents, transport multimodal, aide au déplacement

Keywords

information systems, multimodal information, multi-agent, multimodal transport

1. Introduction

Un client du transport multimodal souhaite avoir à disposition une information sur l'offre de déplacements et sur les itinéraires respectant ses critères. La recherche de cette information et la composition d'itinéraire peuvent se faire par le client lui-même. Cela nécessite une forte connaissance sur l'offre de transport, de différents sites d'information des opérateurs des réseaux de transport ainsi qu'une disponibilité de la part du client et une maîtrise

de l'outil informatique. Pour éviter cette tâche fastidieuse de recherche, de mémorisation des différentes éventualités, de combinaison de différents moyens de transport et pour l'aider dans cette démarche de prise de décision, nous proposons un système qui aide les passagers à formuler leurs requêtes de recherche d'itinéraire et à planifier leur déplacement. Il s'agit de fournir toute information utile et pertinente sur les différents modes de déplacement afin d'une part, d'améliorer le confort et l'efficacité des trajets à un niveau individuel, et d'autre part, de favoriser l'usage multimodal et raisonné des différents modes de transport à l'échelle collective (Uster, 2004).

Dans la suite de cet article, nous présentons dans la section 2 les problèmes de déplacement des voyageurs (problématique). La section 3 présente la configuration du réseau de transport, puis la section 4 expose l'architecture du système proposé. La section 5 explique les différents algorithmes utilisés dans notre système et la normalisation des critères de déplacement. Nous terminons par une conclusion.

2. Problématique

La planification d'un déplacement par un client de réseaux de transports nécessite la consultation de plusieurs systèmes d'information d'aide au déplacement gérés par des opérateurs de transports en commun. Les systèmes d'information d'aide au déplacement pour le transport en commun existant sont généralement des systèmes monomodaux et offrent une information sur un seul moyen de transport. Il existe des systèmes d'information multimodaux qui offrent une information portant sur plusieurs modes de transport gérés par un même et seul opérateur. Ces systèmes restent limités géographiquement. Plusieurs exemples existent. Nous citons parmi la catégorie de systèmes monomodaux, Phebus de la Société de transport de Versailles et Solea pour le transport à Mulhouse et ses banlieues. Parmi les systèmes multimodaux se limitant à un même opérateur, nous citons le site Transpole, opérateur de la métropole lilloise, lePilote pour Marseille et ses banlieues, ou RATP pour la région de l'Île-de-France.

Pour fédérer les projets et pour capitaliser les recherches et les expériences en termes d'information multimodale, le PREDIT, Programme de Recherche et de l'Innovation dans les Transports terrestres¹, a créé le PREDIM², une plate-forme de recherche et d'expérimentation pour le développement de l'information multimodale.

Parmi les projets qui concernent l'information multimodale, nous citons :

- Le projet INFOPOLIS qui vise à améliorer la qualité de l'information produite par les systèmes existants,
- Le projet ISCOM qui est un projet de développement de techniques multimédias pour diversifier les techniques de diffusion de l'information multimodale,
- le projet TRIDENT³ qui vise à assurer l'interopérabilité entre les systèmes monomodaux ou régionaux, par les techniques d'intégration et d'échange d'information en informatique, telles que XML ou les Web services,
- le système allemand DELFI qui donne des solutions multimodales sans assurer le suivi des déplacements en cas de perturbation (Zidi, Hammadi *et al.*, 2004).

Afin d'éviter cette surcharge de consultation, notre travail vise à concevoir un Système d'Information Voyageur Multimodale (SIVM) pour l'aide au déplacement multimodal. Il s'agit d'automatiser cette démarche de recherche et de composition d'itinéraires, pour fournir une information multimodale via un système intégrateur, en s'appuyant sur la théorie des systèmes multi-agents (SMA) pour l'intégration et la médiation des systèmes d'information d'aide au déplacement de différents opérateurs de transport.

La majorité des opérateurs de transport disposent d'un Système d'Information d'Aide au Déplacement, qui est généralement composé d'une base de données décrivant le réseau de transport de l'opérateur et un algorithme de calcul d'itinéraires (ACI). La base de données décrite à travers des informations statiques telles que toutes les lignes, les arrêts, les itinéraires et les horaires de changement, l'ensemble du réseau et des services offerts à l'utilisateur. ACI utilise la base de données locale pour rechercher et trouver les itinéraires qui s'adaptent au mieux à la requête de l'utilisateur.

Pour produire l'information multimodale et afin de la composer à partir de plusieurs systèmes d'information d'aide au déplacement de différents opérateurs, il est nécessaire de procéder à une intégration de données (Figure 1). La stratégie d'intégration adoptée peut être :

- une intégration de données, localisées dans les différentes bases de données des différents opérateurs afin de constituer une base de données globale et centralisée.
- une intégration d'applications par la production et la composition de l'information multimodale à partir des résultats de plusieurs systèmes d'information d'aide au déplacement (SIADs) existants, interrogés automatiquement.

1 www.predit.prd.fr

2 www.predim.org

3 www.erfco.com/activiti/projects/trident/home.hm

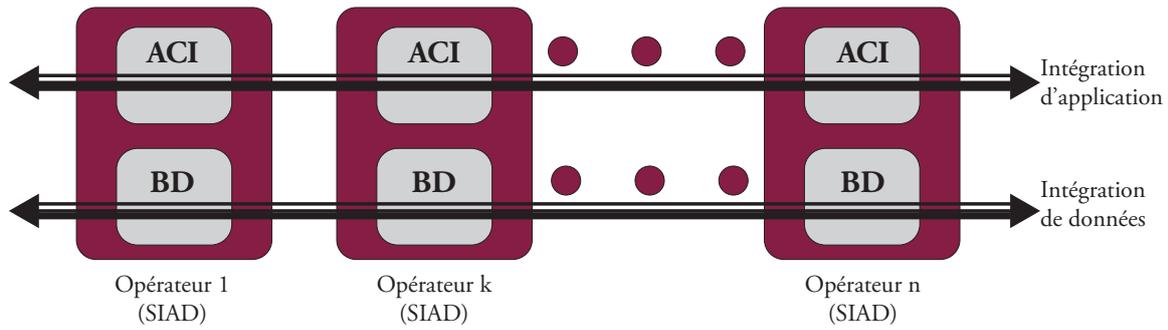


Figure 1. Stratégies d'intégration

Le SIVM devient alors l'intermédiaire entre les différentes sources d'information hétérogènes et distribuées d'une part et les clients d'autre part. Ce système doit être capable à la fois de trouver la bonne source d'information pour l'interroger selon les différentes requêtes des utilisateurs, et de regrouper les informations de manière cohérente pour répondre aux requêtes.

3. Configuration d'un réseau de transport multimodal

Le réseau de transport multimodal sur une vaste zone géographique ne peut être que la juxtaposition de plusieurs sous-réseaux régionaux (Figure 2) liés par des stations communes (pôles d'échanges).

Un réseau régional peut également réunir plusieurs réseaux départementaux et un réseau départemental, à son tour, peut réunir plusieurs réseaux de communes. Au niveau des départements et des communes, les liaisons des réseaux se fait par un ensemble de lignes de bus. Dans les agglomérations urbaines, la gestion du réseau est généralement associée à un opérateur qui exploite plusieurs modes de transport sur cette zone.

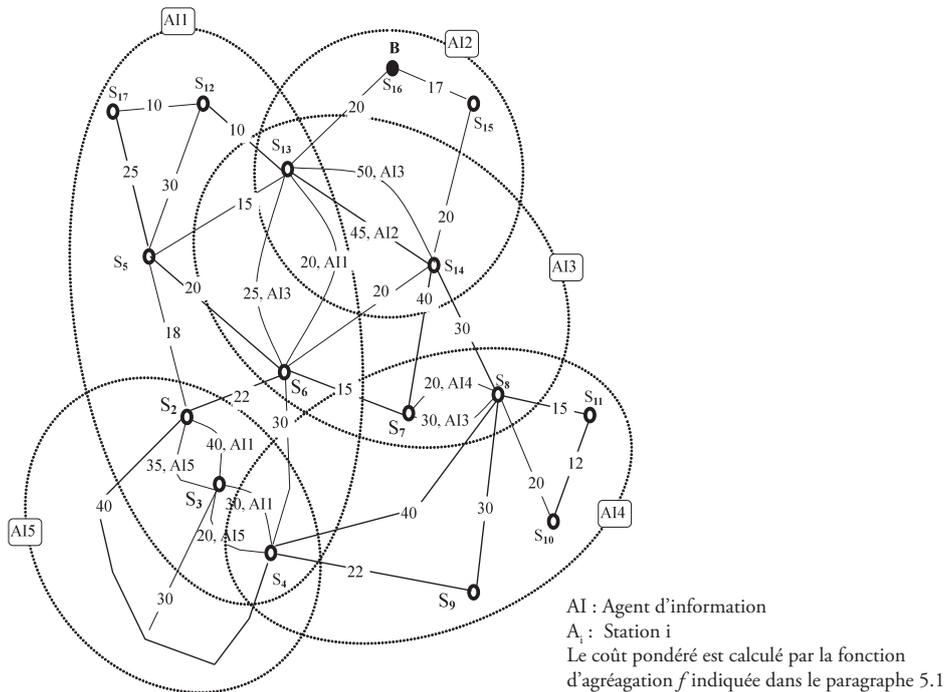


Figure 2. Réseaux de transport multimodal et sous-réseaux

4. Architecture du système proposé

Afin de planifier des déplacements, nous proposons un système d'information voyageur multimodal SIVM (Figure 3) qui a pour rôle d'assister l'utilisateur de transport en commun dans l'organisation de ses déplacements en respectant ses critères (coût, temps de parcours, confort).

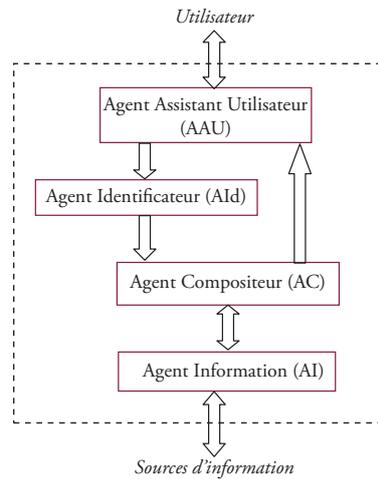


Figure 3. Architecture multi-agents

Afin de concevoir ce système d'information multimodal, nous avons proposé une organisation multi-agents. Les systèmes multi-agents sont des ensembles d'entités autonomes nommées agents qui ont un comportement collectif afin d'atteindre une fonction désirée (Colline, 2003). Pour concevoir le SIVM, nous commençons par associer à chaque système d'information d'aide au déplacement (SIAD) un opérateur de transport, et par conséquent, à chaque réseau de transport, un agent d'information **AI** (Figure 9). L'AI est un extracteur d'information à partir du SIAD. Pour ce faire, l'AI dispose en interne de l'adresse réseau (Internet ou autre) de ce SIAD, et des protocoles de communication qui lui permettront de lui envoyer des requêtes et d'en recevoir des réponses.

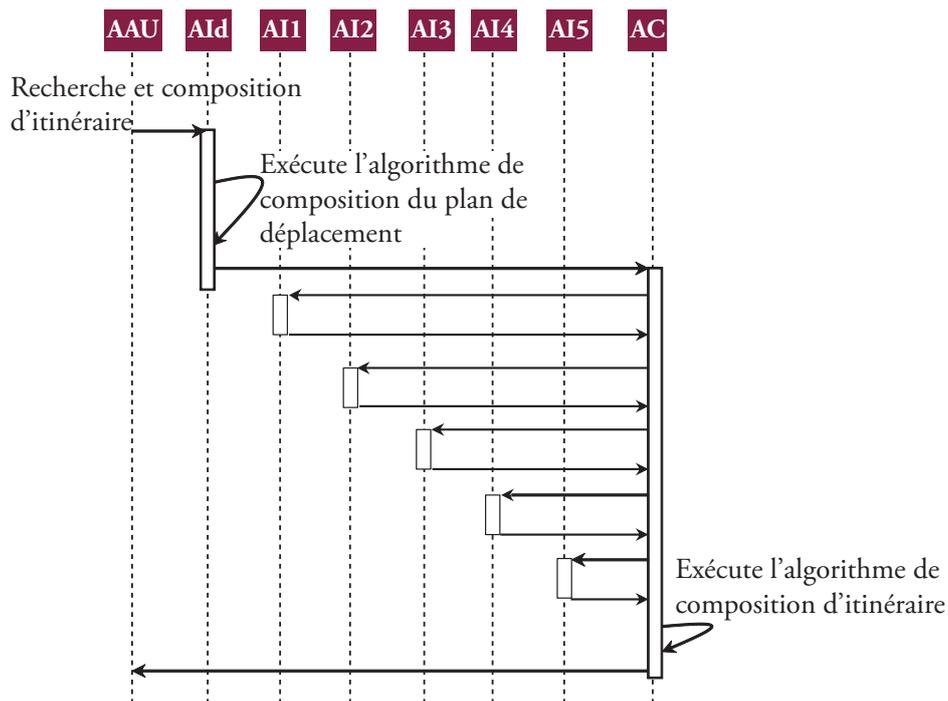


Figure 4. Diagramme de séquences d'interaction entre agents

Un AI, comme le SIAD auquel il est associé, est capable de répondre à une requête d'itinéraire, du type aller de A vers B, si A et B sont des stations gérées par le même agent.

La composition et la recherche d'itinéraire seront affectées à un agent que nous appelons dans ce contexte, Agent Compositeur **AC** (Figure 8). Un agent AC est capable de composer une offre globale d'itinéraire en interrogeant différents AI en récupérant différentes offres locales.

L'AC ne dispose pas d'une vue globale de tous les AI du SIVM. En effet, pour répondre à une requête donnée, l'AC n'est pas capable de déterminer un domaine de recherche, d'où l'intérêt d'un Agent Identificateur **AIId** (Figure 7). Ce dernier est capable de délimiter les domaines de recherche relatifs aux différentes requêtes. Ceci revient à leur envoyer l'ensemble des AI sélectionnés qui définissent ce domaine et l'ensemble des stations d'intersection.

Par ailleurs, nous associons ensuite à chaque utilisateur un Agent Assistant Utilisateur **AAU** (Figure 6) qui permettra à ce dernier de formuler sa requête et de recevoir les résultats.

Les interactions entre les agents pour rechercher et pour composer un itinéraire global sur un environnement distribué sont illustrées par le diagramme UML de séquences (Figure 4).

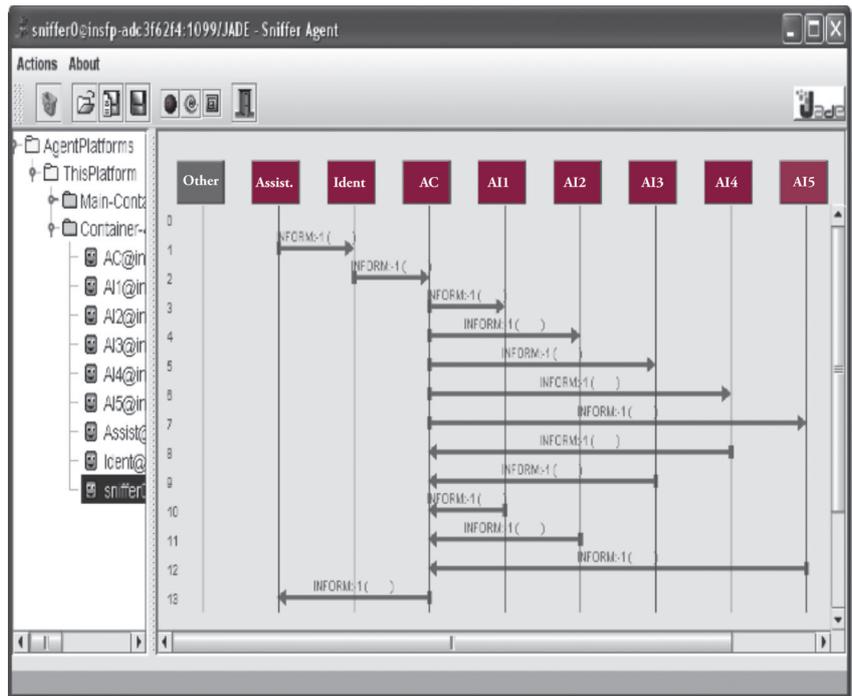


Figure 5. Messages échangés entre agents (Agent Sniffer)

Pour le développement du SIVM et la simulation des résultats de nos approches de recherche et de composition d'itinéraires multimodaux, nous avons choisi la plate-forme JADE (Java Agent DEvelopment Framework). JADE est une plate-forme multi-agents qui offre la possibilité d'intégration des services Web (Greenwood, 2005), la communication entre les agents du SIVM capté par l'agent «sniffer» de la plate-forme JADE est donnée par la figure suivante (Figure 5).

Suite à cette description du SIVM, nous abordons dans la suite la description du comportement propre à chaque agent. Cette description est illustrée par des diagrammes UML d'activités qui permettent de retracer leurs différents scénarios de fonctionnement.

4.1. Agent Assistant Utilisateur (AAU)

L'AAU permet à l'utilisateur du système de formuler sa requête et de préciser ses critères de recherche. En effet, quand il reçoit une requête de l'utilisateur, il la transmet à l'agent identificateur (AId), et attend la réponse. S'il s'agit d'un refus, il demande à l'utilisateur de reformuler sa requête, sinon il lui présente les résultats.

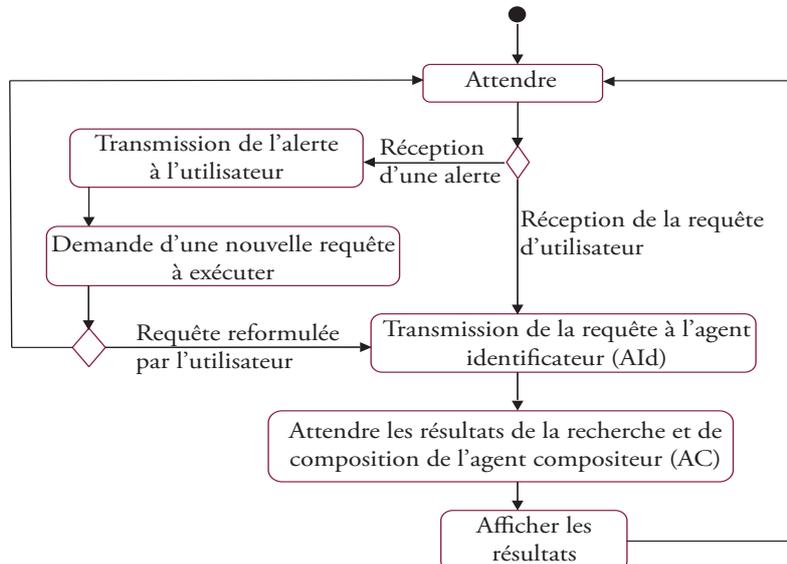


Figure 6. Comportement de l'agent AAU

4.2. Agent Identificateur (AId)

Cet agent décompose la requête reçue auprès de l'AAU pour identifier la station de départ et la station d'arrivée, puis applique l'algorithme de composition du plan de déplacement (présenté plus loin), pour trouver l'ensemble des agents qui doivent intervenir pour réaliser ce déplacement. C'est le graphe d'adjacence. L'Aid indique les stations communes existantes (pôles d'échanges) entre ces agents.

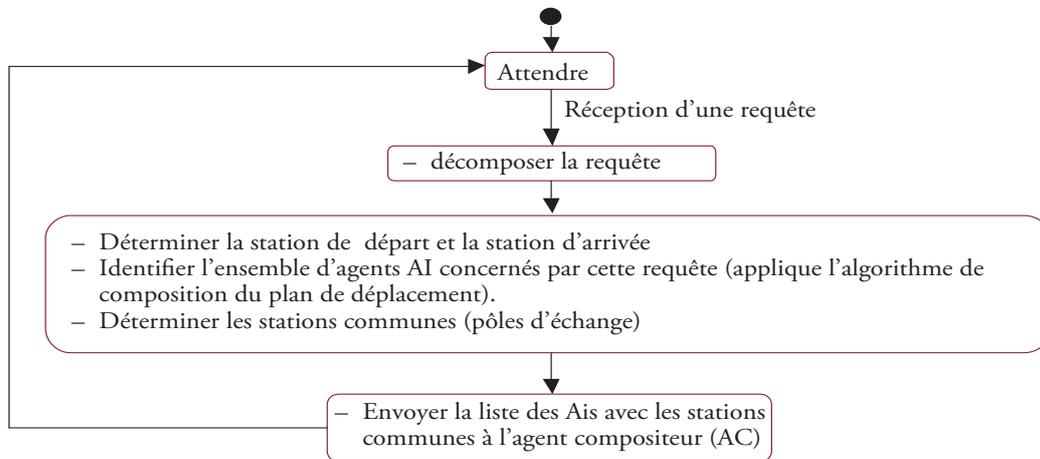


Figure 7. Comportement de l'agent AId

4.3. Agent Compositeur (AC)

A partir des résultats reçus par les AIs, l'agent AC compose le graphe de déplacement possible de la requête, puis lance l'algorithme qui calcule le plus court chemin, afin d'obtenir la (les) solution(s) possible(s), et la (les) transmettre à l'agent (AAU).

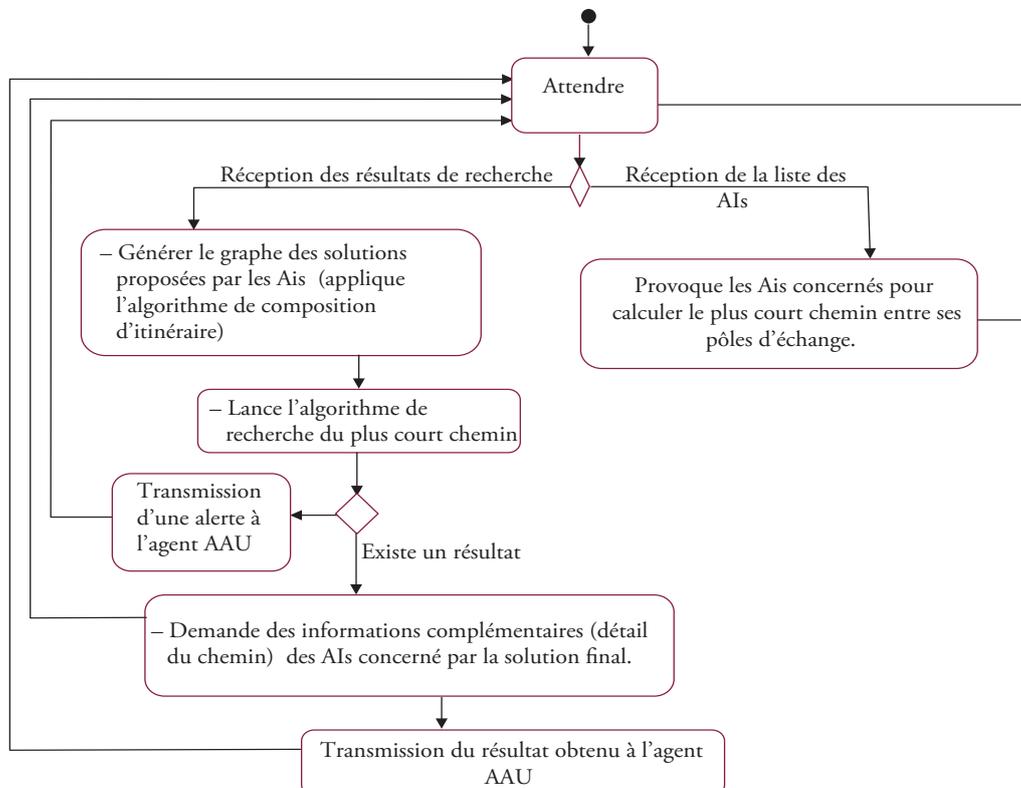


Figure 8. Comportement de l'agent AC

4.4. Agent Information (AI)

On associe à chaque système d'information d'aide au déplacement (SIAD) un agent qui se charge de l'extraction d'information à partir de SIAD associé.

Pour ce faire, l'AI doit disposer en interne de l'adresse réseau (Internet ou autre) de ce SIAD, et des protocoles de communication qui lui permettront de lui envoyer des requêtes et d'en recevoir des réponses. Chaque agent est responsable de son réseau de transport. En effet, l'AI récupère en premier les données auprès du SIAD correspondant et calcule ensuite le plus court chemin entre les pôles d'échanges dans sa zone après la normalisation des critères par l'application de la méthode d'agrégation (présentée plus loin). Le chemin partiel résultat est envoyé à AC. Ce chemin peut être le résultat de la requête, si le lieu de départ A et le lieu d'arrivée B sont des stations gérées par le même agent.

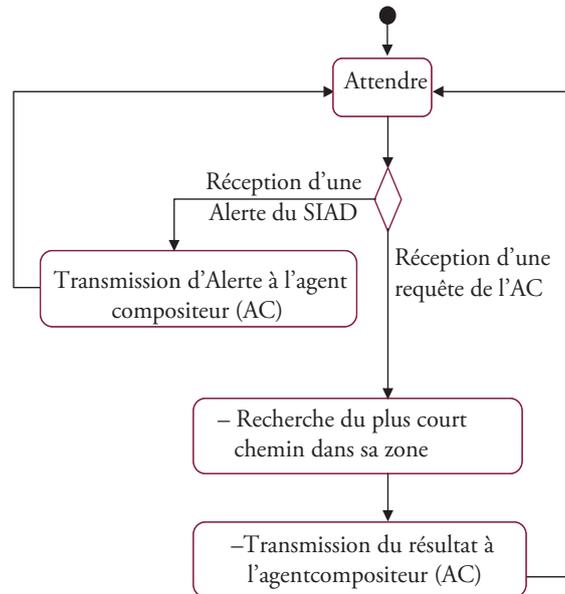


Figure 9. Comportement de l'agent AI

5. Optimisation multicritères

Les problèmes de cheminement sont parmi les plus anciens de la théorie des graphes. Le problème le plus typique de cette catégorie est celui du plus court chemin. En effet, il est rencontré dans de nombreuses applications. L'algorithme le plus utilisé pour la recherche du plus court chemin est sans doute celui de Dijkstra (Meskine et Gendre 2001) (Lacomme, Prins *et al.*, 2003). Dans la littérature, l'algorithme de Dijkstra s'applique aux graphes pondérés avec une seule valeur non négative pour chaque arc. Dans ce sens, nous effectuons une normalisation des critères de déplacements présentés dans la section suivante.

5.1. Normalisation des critères

Notre objectif est d'aider l'utilisateur de transport en commun à optimiser ses déplacements selon plusieurs critères. Dans notre système, nous optimisons trois critères : le coût, le temps de parcours et le confort. Les utilisateurs de transport en commun cherchent souvent à les optimiser ensemble.

L'optimisation de ces critères est parfois difficile, parce que l'optimisation de l'un entraîne la dégradation de l'autre. Dans ce sens, nous utilisons une fonction d'agrégation f (Ben Khaled, Kamoun *et al.*, 2005) qui calcule le coût de chaque arc du graphe du réseau. L'objectif est de minimiser le coût total du chemin qui représente la somme des coûts des arcs traversés. Les coefficients de pénalité sont calculés suivant le choix de l'utilisateur : il a le choix de favoriser un critère par rapport aux deux autres, ou bien de leur donner des importances égales.

$$f = \alpha * C_1 + \beta * C_2 + \delta * C_3$$

C_1 , C_2 et C_3 représentent les critères d'optimisation tels que le coût, le temps de parcours et le confort. C_1 est calculé par la formule suivante :

$$C_1 = \chi * \sum_{i=0}^z C_3^i * C_1^{ik}$$

Où

i représente les zones

C_3^i représente le coefficient de confort correspondant à la zone i

C_1^{ik} représente le prix de titre de déplacement dans la zone i en utilisant le mode k

$\chi \leq 1$ représente le coefficient utilisé par les modes de transport pour le calcul du prix interzones

$$C_2 = \sum_{i=0}^p t_i^{m_i} + \sum_{j=0}^n T_{Att}^j$$

$t_i^{m_i}$ est le temps de parcours de l'arc i par le mode de transport m_i

T_{Att}^j est le temps d'arrêt dans la station j

$$C_3^i = i * \mu$$

Où i appartient à l'ensemble $\{1, 2, 3\}$ et μ dépend des exploitants du transport en commun.

Cette variable caractérise les classes existantes dans le transport en commun.

α , β et δ sont des coefficients de pénalité pour ces critères, leur somme est égale à 1. La fonction f ramène le problème d'optimisation multicritère à un problème d'optimisation monocritère.

5.2. Procédure de composition d'itinéraire

Aller d'une station A à une station B dans un RTE (Réseau de Transport Etendu) nécessite une coopération de plusieurs AIs. Notre système répond à la question «quels sont les AIs qui devraient coopérer pour réaliser le déplacement de A à B?» en utilisant l'algorithme de composition du plan de déplacement. Il répond aussi à la question «comment faire coopérer les AIs?» en utilisant l'algorithme de composition d'itinéraires.

5.2.1. Algorithme de composition du plan de déplacement

C'est l'algorithme utilisé par l'AId. Pour construire le plan de déplacement initial à partir d'une requête «aller de A vers B», l'algorithme détermine les agents qui peuvent intervenir ensemble pour réaliser ce déplacement. Cet algorithme est composé des étapes suivantes.

1^{ère} étape: l'agent AId détermine au départ quel est le AI_A qui correspond à la station de départ A et quel est le AI_B correspondant à la station d'arrivée B, puis teste si A et B ne sont pas gérés par le même AI. L'algorithme se poursuit par l'étape 2.

2^{ème} étape: l'agent AId utilise le graphe d'adjacence (Figure 10) pour sélectionner les k plus petits groupes de collaboration capables de répondre. Ces groupes contiennent le minimum d'agents pour minimiser le temps global de calcul de l'offre. Pour déterminer ces k groupes d'agents l'AId applique l'algorithme classique de k plus courts chemins (Van Der Zijpp et Fiorenzo Catalano, 2005) qui s'applique sur les graphes évalués et positifs entre le sommet de départ (AI_A) et le sommet d'arrivée (AI_B). Dans notre cas, il s'agit de minimiser le nombre de sommets à impliquer dans ces k chemins.

Cet algorithme de k plus courts chemins, est fondé sur une heuristique de pénalité. Il applique l'algorithme Dijkstra pour calculer le plus court chemin, ensuite il modifie le graphe en imposant une augmentation des poids des arcs du plus court chemin. Il applique ensuite Dijkstra pour trouver le 2^{ème} plus court chemin, etc.

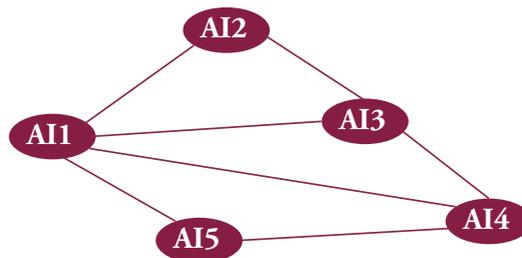


Figure 10. Graphe d'adjacence

Dans notre exemple, AI_5 est le sommet de départ et AI_2 , le sommet d'arrivée.

Si $K=3$, nous obtenons des collaborations possibles définies par les ensembles $\{AI_5, AI_1, AI_2\}$, $\{AI_5, AI_1, AI_3, AI_2\}$, $\{AI_5, AI_4, AI_3, AI_2\}$

	AI ₁	AI ₂	AI ₃	AI ₄	AI ₅
AI ₁		{s ₁₃ }	{s ₆ , s ₁₃ }	{s ₄ }	{s ₂ , s ₃ , s ₄ }
AI ₂	{s ₁₃ }		{s ₁₃ , s ₁₄ }	∅	∅
AI ₃	{s ₆ , s ₁₃ }	{s ₁₃ , s ₁₄ }		{s ₇ , s ₈ }	∅
AI ₄	{s ₄ }	∅	{s ₇ , s ₈ }		{s ₄ }
AI ₅	{s ₂ , s ₃ , s ₄ }	∅	∅	{s ₄ }	

Tableau 1. Matrice d'adjacence

Le graphe d'adjacence est représenté dans l'agent AId par la matrice d'adjacence symétrique (Tableau 1) indiquant aussi les stations communes (pôles d'échange) correspondant à l'intersection des différents réseaux de transport impliqués.

5.2.2. Algorithme de Composition d'Itinéraires

Cet algorithme de composition appliqué par l'agent compositeur AC se fonde principalement sur l'algorithme de Dijkstra distribué (Wang et Kaempke, 2004) pour calculer le plus court chemin de manière distribuée. Cet algorithme se compose de deux étapes.

1^{ère} étape: construire le graphe d'intersection étendu Gint (A; B) (Figure 11). Il s'agit du graphe des itinéraires possibles qui relient les stations A et B moyennant les stations communes, gérées par plusieurs AIs. Pour construire le graphe Gint (A; B), chaque AI intervient et relie les stations communes qui le concerne. Chaque agent fournit à l'AC l'information qui concerne sa zone. En d'autres termes, étant donné (1) les sommets d'intersection, (2) les plus courts chemins entre ces stations (3), les sommets A et B, l'agent compositeur AC construit le Gint (A; B) moyennant un groupe d'agents (plan de déplacement initial) proposé par l'agent AId.

2^{ème} étape: une fois le graphe Gint (A, B) construit, l'AC applique l'algorithme de calcul du plus court chemin. Une fois ce chemin choisi (Figure 12), chaque AI, dont l'arc a été retenu est appelé à le détailler. L'AI décrit alors l'itinéraire qu'il propose, en termes de modes de transport à utiliser.

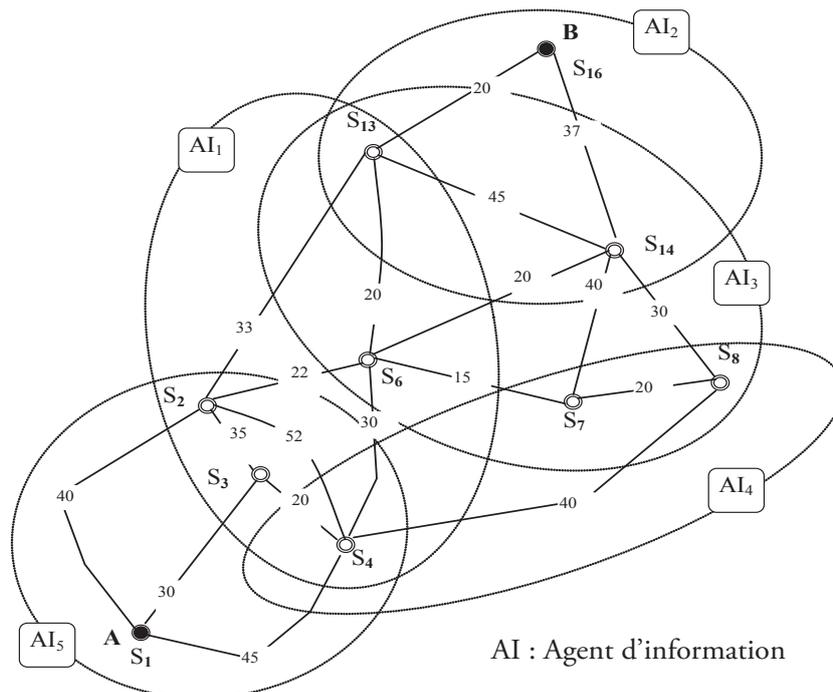


Figure 11. Graphe d'intersection Gint(A; B)

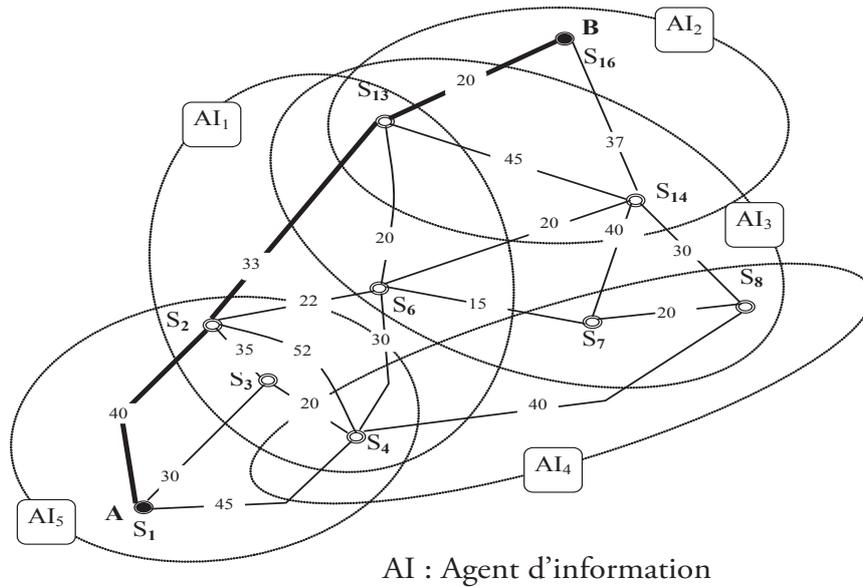


Figure 12. Plus court chemin sur le $Gint(A;B)$

5.2.3. Scénario global de composition d'itinéraire

Le scénario global de composition d'itinéraire est illustré par le diagramme UML d'activités (Figure 13):

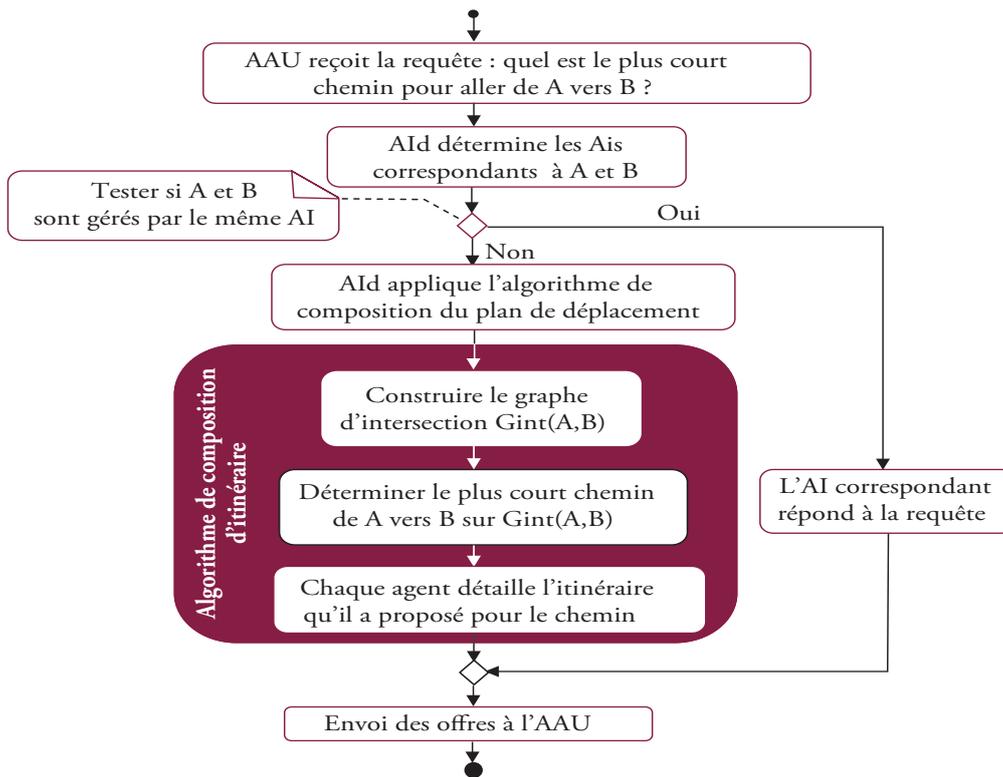


Figure 13. Scénario global de composition d'itinéraire

6. Conclusion

Dans cet article, nous avons présenté un système d'information voyageur multimodal (SIVM) fondé sur l'approche multi-agents. L'objectif de ce système est d'aider les voyageurs à organiser ou à planifier leurs déplacements. Ce système rend l'architecture plus flexible et ouverte vers d'autres systèmes. Il suffirait de modifier les compétences de quelques agents pour adapter le système à un nouveau besoin d'utilisateur, à un nouveau protocole de communication avec un fournisseur de données, ou à un nouvel algorithme de calcul d'itinéraire. Le système reste ainsi ouvert, pour inclure de nouveaux agents d'information. L'organisation multi-agents présentée assure la mise à jour de l'information. En effet, si chaque AI assure la mise à jour des données du SIAD auquel il est associé, l'information multimodale composée à partir des différents AIs serait automatiquement mise à jour.

Références

- Ben Khaled, I., Kamoun, M.A., Zidi, K., Hammad, S. (2005). Vers un système d'information voyageur multimodal (SIM) à base de système multi agent (SMA). REE. Janvier.
- Collège Systèmes Multi-Agents de l'AFIA, Groupe de Travail « Collectif, Interaction, Emergence » (COLLINE): Glossaire 2003; Document disponible à l'adresse suivante: <http://www.irit.fr/COLLINE/DOCUMENTS/GlossaireColline.rtf>
- Greenwood, D. (2005). JADE Web Service Integration Gateway (WSIG). Whitestein Technologies. Jade Tutorial. AAMAS.
- Lacomme, P., Prins, C., Sevaux, M.(2003). Algorithmes de graphes. Edition EYROLLES chapitre6 175-214.
- Meskine, A., Gendre, P.(2001). Algorithmes et calculs d'optimisation d'itinéraires pour l'information multimodale; implémentation d'un prototype pour les transports collectifs avec horaires. Novembre.
- Uster, G.(2004). pour une mobilité raisonnée, mobilités.net , collection 'Questions numériques' pp 320-324.
- Van Der Zijpp, N.J., Fiorenzo Catalano, S. (2005). Path enumeration by finding the constrained K-shortest paths, Transportation Research Part B, Vol 39, pp 545-563.
- Wang, K., Kaempke, K. (2004). Shortest route computation in distributed systems, Computers and operation research, Vol 31, pp 1621-1633.
- Zidi, K., Hammadi, S., Borne, P.(2004). Méthode évolutionniste pour l'aide au déplacement dans le transport multimodal perturbé. 5ème Conférence francophone de modélisation et de simulation (MOSM'04): « Modélisation et simulation pour l'analyse et l'optimisation des systèmes industriels et logistiques ». Du 1er au 3 septembre, Nantes (France).

Cheminement d'objets pédagogiques de type quiz

Une approche orientée services Web

Ivan Madjarov

Aix-Marseille Université, LISIS UMR 6168, Avenue Escadrille Normandie-Niemen, 13397 Marseille Cedex 20, France, ivan.madjarov@lisis.org

Zohra Bakkoury

Université Mohammed V – Agdal, École Mohammadia d'Ingénieurs, BP 765, Agdal, Rabat, Maroc, bakkoury@emi.ac.ma

Abdelkader Betari

Aix-Marseille Université, LIF UMR 6166, 163 Avenue de Luminy case 920, 13288 Marseille Cedex 9, France, abdelkader.betari@univmed.fr

Bogdan Shishedjiev

Université Technique de Sofia, 8 bd Kliment Ohridski, Sofia-1000, Bulgarie
bogi@tu-sofia.bg

Résumé

Cet article propose une structure XML sémantique conforme aux standards e-learning en vigueur pour la création, le stockage et la diffusion de quiz. Nous présentons la construction d'un parcours pour l'apprenant et un environnement auteur intégré dans une architecture orientée services. Nous apportons une solution basée sur les services Web pour l'intégration des données pédagogiques et des applications qui les animent. Nous discutons de la structuration des données et des moyens d'évaluation des connaissances dans le cadre d'un système de gestion de contenu pédagogique. Nous présentons ensuite nos travaux de recherche en cours et une partie des résultats du projet XESOP. Nous présentons enfin notre concept pour la réalisation et l'implémentation de l'éditeur de questionnaires basé sur les technologies des services Web.

Abstract

This paper proposes a standards-compliant XML semantic structure for e-learning quizzes. We present an authoring environment integrated into SOA architecture, and the construction of a learner's path. We provide a Web service-based solution for data and educational applications integration. We discuss about data structuring and knowledge assessment tools as part of a Learning Course Management System.

Mots clés

Ingénierie pédagogique, questionnaire, services Web, télé-enseignement, LCMS

Keywords

e-Learning, LCMS, quiz, Web services, authoring tools

1. Introduction

Le Web 2.0 est une plate-forme interactive et collaborative offrant les outils pour la communication, les interactions multiple et plurielle, les échanges et le partage de contenus et d'applications hétérogènes entre internautes favorisant ainsi la construction des réseaux sociaux. Nos activités de recherche concernent à la complexité des systèmes qui visent à exploiter les ressources du Web en utilisant l'Internet comme infrastructure logicielle. L'objectif est de concevoir et de développer des solutions pour faciliter l'accès et l'exploitation des données et des applications en se fondant sur les technologies des services Web.

L'utilisation des questionnaires (QCM) dans l'enseignement en présentiel est une pratique courante. Dans l'enseignement à distance, les questionnaires en ligne existent depuis les débuts de cette forme d'apprentissage. L'utilisation de ces moyens pédagogiques débouche sur la problématique de la pérennité des questionnaires et la possibilité de les intégrer dans un environnement de gestion de contenus pédagogiques (LMS pour *Learning Management System*). Les moyens de création, basés sur des techniques diverses et variées et disponibles dans l'espace des logiciels libres ou commerciaux, ne favorisent pas la réutilisation et l'intégration de ces quiz. Quant aux moyens de recherche et d'extraction de données, la situation est plus complexe à cause des différences de sémantique et de format de stockage. L'application d'un quiz dans un apprentissage en ligne demande aussi la sauvegarde des résultats et leur exploitation par l'enseignant. Toutes ces étapes sont technologiquement liées dans une plate-forme *e-learning*. Pour leur réalisation, l'intérêt est de se conformer à un standard (IMS QTI, 2005) et pour leur application à un système *e-learning* qui respecte les normes d'interopérabilité, en particulier SCORM (2004) et LOM (IEEE, 2002). Étant donné que la spécification IMS QTI est basée sur un schéma XML, on peut facilement séparer le contexte (exercices interactifs, questions et réponses) de la forme (représentation graphique, affichage, impression) pour une réutilisation à partir d'une base de données dans un format natif.

Un test ou exercice interactif IMS-QTI est représenté alors physiquement par un fichier XML. Il est donc possible de le réaliser à l'aide d'un simple éditeur de texte ou d'un éditeur XML. Toutefois, cette activité nécessite une bonne connaissance des éléments particuliers de la spécification. L'auteur risque, dans ce cas, de perdre beaucoup de temps pour un résultat qui reste incertain à la création des questions dans toutes leurs formes (question simple, texte à trou, *drag & drop*, listes déroulantes, etc.). Il est important de mentionner ici l'existence de nombreux logiciels permettant la génération de fichiers XML conformes à la spécification IMS QTI. L'interprétation de ces fichiers par la suite n'est en aucun cas automatique. La représentation graphique et interactive d'un quiz demande certaines compétences dans le domaine des technologies de transformation d'une structure XML dans un format reconnu par les navigateurs ou les *players*. La maîtrise de certains logiciels de conception graphique tels que Flash, Authorware, Gimp ou autres est demandée et représente en réalité une forte contrainte. À la sortie d'un quiz, l'auteur doit prendre soin des résultats; les enregistrer localement ou les communiquer à une autre plate-forme. L'intégration de toutes ces étapes demande une approche plus complexe et réfléchie (Joosten *et al.*, 2005).

Dans cet article, nous proposons un environnement auteur intégré et basé sur les technologies SOA (*Service Oriented Architecture*) pour la création, le stockage et la diffusion de quiz issus de la spécification IMS QTI. Nous discutons de la structuration des données définie dans le standard IMS-QTI et des moyens d'évaluation des connaissances dans le cadre d'un système de gestion de contenu pédagogique (LCMS pour *Learning Content Management System*). Nous présentons ensuite nos travaux de recherche en cours et une partie des résultats du projet XESOP (Madjarov et Boucelma, 2011). En conclusion, nous nous interrogeons sur le suivi et les futurs travaux à entreprendre.

2. Structuration des données

L'IMS est proposé par un groupe de travail (IMS, 2011) composé de membres issus de différents domaines: enseignement, entreprise et organisations gouvernementales. Le groupe a pour objectif de définir des spécifications pour l'interopérabilité des systèmes de l'enseignement et de promouvoir leur implantation dans le domaine des applications et des services basés Web. Les principes de base pour ces spécifications sont: l'interopérabilité, l'accessibilité, la réutilisation, la pérennité, l'indépendance et la portabilité. Certaines de ces spécifications traitent du contenu pédagogique:

- La spécification «IMS-Meta Data Specification» définit des ressources pédagogiques à travers une structure d'éléments et métadonnées se basant sur la syntaxe et la sémantique des objets pédagogiques définis dans la spécification LOM (IEEE, 2002).
- La spécification «IMS-Content Packaging Specification» définit la structure, l'organisation et l'échange d'un ensemble d'objets pédagogiques regroupés en package.
- La spécification «IMS-Question & Test Interoperability Specification» décrit la conception et la mise en œuvre des éléments pédagogiques servant à l'évaluation des apprenants. L'objet pédagogique de base est la question, suivie de méthodes de prise en charge de la réponse.

Par ces trois spécifications, l'IMS définit l'organisation et la structure d'un système de gestion d'enseignements. Les ressources sont organisées selon le modèle «Content Framework» qui, de son côté, définit deux sous-modèles:

«Content Packaging» pour le regroupement des ressources physiques décrites à la base des métadonnées et «Content Management» qui organise les données de gestion d'un système pédagogique. Les deux parties sont dotées de la description des interfaces vers les principaux utilisateurs selon leur position dans le processus d'interaction avec le système: les auteurs, les administrateurs et les apprenants (Figure 1).

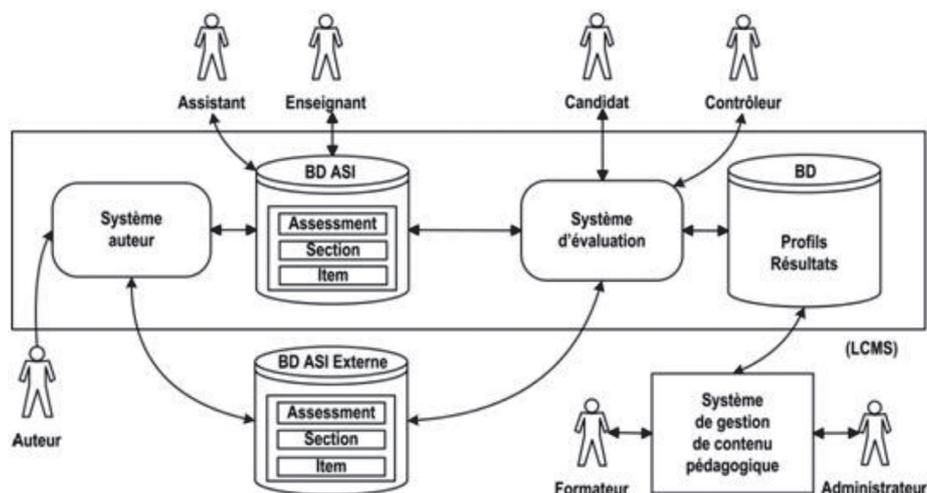


Figure 1. Interactions des acteurs dans un système d'évaluation continu

La partie «package» (Anderson et McKell, 2001) est autonome. Elle est destinée à l'auteur pour la construction d'une ressource pédagogique réutilisable en regroupant l'ensemble des objets pédagogiques. Cette partie peut être prise en charge par des logiciels et des systèmes auteur qui respectent les standards en vigueur. De cette manière le «package» est incorporable dans un LCMS. Il est constitué d'une partie «manifest» en format XML qui contient la description de l'organisation du contenu pédagogique. La deuxième partie contient la description des objets pédagogiques. Physiquement, ils peuvent faire partie d'une base de données XML native (Madjarov, 2010) ou d'une arborescence de fichiers sur des sites distribués. Un objet pédagogique (IEEE, 2002) est défini comme «toute entité, sur un support numérique ou non, pouvant être utilisé pour l'apprentissage, l'enseignement ou la formation». Le concept des objets pédagogiques (LO pour *learning object*) est fondamental dans les systèmes *e-learning* en tant qu'entités basées sur le standard LOM (IEEE, 2002). C'est un modèle de métadonnées reconnu pour décrire la nature, la syntaxe et la sémantique d'un objet pédagogique. Il peut être utilisé tel quel ou en construction avec d'autres afin de construire les entités d'enseignement telles que les cours. Les méta informations du «Content Management» assurent la possibilité de concevoir des cours de type documents virtuels personnalisables.

Dans nos travaux sur le projet XESOP, nous avons étendu l'arborescence de base avec des composants qui mettent en évidence des éléments pédagogiques pertinents, tels que: tableaux, exercices, équations mathématiques, graphisme vectoriel, questionnaires (Madjarov et Boucelma, 2006), (Madjarov et Betari, 2008), (Madjarov et Boucelma, 2011).

3. La spécification IMS QTI et l'évaluation des connaissances

Le déroulement d'une formation en présentiel ou à distance est normalement finalisé par une évaluation. Cette phase finale dans le domaine du *e-learning* représente un enjeu majeur. D'une part, l'évaluation des connaissances peut être vue comme une contribution à la réussite d'un enseignement, et d'autre part, comme un objet pédagogique particulier qui s'ajoute à la construction des savoirs et des compétences des apprenants.

La spécification de l'IMS-QTI permet de représenter la structure de données d'une question (*item*), d'un test (*assessment*) et des résultats correspondants (*results report*). Cette spécification est extensible, adaptable et fondée sur le métalangage XML pour la conception des applications spécifiques (Figure 2). Elle n'influe pas sur les concepts pédagogiques ni sur les choix techniques de programmation ou d'interface graphique (Joosten ten Brinke *et al.*, 2005). Cette partie est laissée à la libre appréciation des concepteurs d'outils logiciels et des producteurs de contenus (Michel et Rouissi, 2003). Pour une question (*item*), il faut préciser le barème, le rendu visuel et les retours à l'apprenant. Il est parfois utile de regrouper des séries de questions (*items*) dans un test (*assessment*) ce qui se fait en utilisant des sections qui elles-mêmes peuvent contenir d'autres sections. Cette structure (Figure 1) correspond au modèle ASI (*Assessment Section Item*) (Global Learning Consortium, 2011).

Le scénario d'utilisation place l'apprenant (*participant*) devant un exercice de type test. La section représente un container pour le regroupement d'autres sections qui contiennent un ou plusieurs éléments de base (*item*). Cet élément représente un bloc contenant la question (question) avec tous ses propriétés et attributs. Une question finalement peut être vue comme un *learning object* indissociable et intégré au niveau du «Content Packaging».

La version 2.1 d'IMS-QTI offre douze questions types (IMS, 2005): Question simple Vrai/Faux (*True/False*) (Figure 2); Question à choix multiples (*Multiple choice*); Question à réponses multiples (*Multiple-response*); Question à choix multiples avec réponse sur une zone sensible d'une image (*Image hotspot*); Question à réponses multiples avec plusieurs zones sensibles d'une image; Question à choix multiples avec réponse en fonction du mouvement du curseur (*Drag-and-drop*); Question avec un tri de texte ou images (*Ordered-response*); Question à réponse graphique; Question avec réponse textuelle (*FIB-string*); Question avec une réponse numérique (*FIB-numeric*); Question où la réponse demande un glissement d'objets graphiques; combinaisons des types existants. Le standard propose également des facilités pour la définition du niveau de difficulté des questions et intègre des algorithmes de calcul des scores. Dans la version 2.1 de l'IMS-QTI, nous pouvons constater l'implémentation du XHTML 2.0 et du MathML.

```

<questestinterop>
  <item ident="IMS_ex_01">
    <presentation label="ex_01">
      <material>
        <mattext>Paris est la capitale de la France.</mattext>
      </material>
      <response_lid ident="F01" rcardinality="Single" rtiming="No">
        <render_choice>
          <response_label ident="T">
            <material><mattext> Vrai </mattext></material>
          </response_label>
          <response_label ident="F">
            <material><mattext> Faux </mattext></material>
          </response_label>
        </render_choice>
      </response_lid>
    </presentation>
  </item>
</questestinterop>

```

Paris est la capitale de la France

Vrai

Faux

Figure 2. Exemple de code XML pour une question de type vrai/faux.

La partie gauche de la Figure 2 présente le code XML pour décrire une question simple. Comme nous pouvons le constater, il est complètement démesuré par rapport à la brièveté et la simplicité de la présentation visuelle de la question (la partie droite). La complexité du code IMS-QTI demande une démarche de caractérisation des ressources d'évaluation plus complexe et générique qui dépasse largement les outils logiciels de génération automatique de balises des «simples» éditeurs XML actuellement proposés (Michel et Rouissi, 2003). Le développement d'un outil générique et plus évolué s'impose donc. Cet outil doit dépasser le stade de la gestion de simples objets comme les formulaires à compléter et fournir une aide réelle comme les systèmes auteur le font. En effet, la richesse et la complexité des spécifications nécessitent une intelligence logicielle capable à la fois de générer les balises suffisantes et d'accompagner l'auteur dans sa tâche de conception de ressources pédagogiques (Giacomini, 2005). Cet accompagnement doit fournir aussi les moyens de réutilisation de ces ressources pédagogiques et leur intégration à partir d'autres ressources hétérogènes. Il s'agit de concevoir un modèle et un système auteur qui s'intègre dans les systèmes de gestion de contenus pédagogiques (Figure 3). Le modèle d'intégration des données et des applications dans notre travail est basé sur les technologies des services Web (Madjarov et Boucelma, 2006). Ainsi, une architecture orientée services (SOA) est-elle proposée en Figure 4.

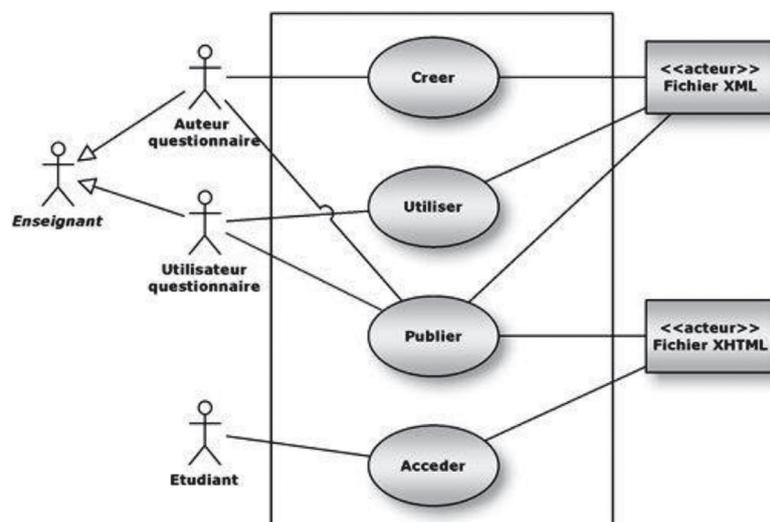


Figure 3. Les cas d'utilisation pour les types d'acteurs

Le standard IMS-QTI permet dans une certaine mesure de prendre en compte la personnalisation des acteurs selon leurs fonctions et l'utilisation des ressources. IMS en effet a prévu neuf types d'intervenants : administrateur, auteur, enseignant, formateur, étudiant, *etc.* Ces informations sont librement renseignées dans la balise <item metadata>. Les acteurs et leurs interactions sont présentés en Figure 3. L'auteur crée un questionnaire en format XML et le publie. Un autre, nommé utilisateur de questionnaire peut réutiliser le fichier XML déjà créé pour le publier et inclure le lien URL dans son cours. On voit que ces deux acteurs présentent des similitudes dans leurs relations. On peut l'exprimer en créant un acteur générique, voire virtuel, qui modélise les aspects communs aux différents acteurs. Dans notre modèle, l'acteur Enseignant est la présentation virtuelle des acteurs Auteur et Utilisateur. Le dernier et le plus important acteur est l'Étudiant. Celui-ci accède au fichier contenant le code-question pour évaluer ses connaissances.

Le choix des questions est également personnalisé selon le profil pédagogique, éducationnel, réactionnel du participant, en renseignant les attributs correspondants. Le flux de questions peut être filtré selon la difficulté de la question et le niveau éducationnel.

La séquence des questions est gérée à travers la spécification IMS-SS (2003). Le langage de la spécification est utilisé pour décrire un cheminement (*sequence*) à travers un ensemble d'activités pédagogiques, c'est-à-dire l'ordre dans lequel les activités pédagogiques seront présentées à l'apprenant et les conditions pour lesquelles les ressources sont sélectionnées, délivrées ou outrepassées durant l'apprentissage. Cette spécification «simple» (en réalité assez complexe) envisage un nombre limité de stratégies de cheminement d'activités pédagogiques. Ces activités regroupent les cours, les tests, les exercices et elles sont organisées de manière hiérarchique correspondant à l'arborescence de leurs dépendances fonctionnelles et/ou pédagogiques.

L'interactivité dans un test en ligne est un besoin pédagogique d'une importance capitale. Par certaines balises de l'IMS-QTI servent à gérer une interactivité minimale. En effet, <itemfeedback> permet de renseigner les possibles réactions du système. Les balises <hint> et <duration> proposent des aides au participant en fonction de ses réponses ou selon le temps de sa réaction.

En conclusion, une utilisation effective des standards IMS-QTI demande une démarche plus réfléchie dans la direction de la conception d'un ensemble (suite ou famille) d'outils avec une intelligence logicielle capable, à la fois, d'organiser la structuration et la personnalisation des questionnaires, la gestion des séquences de questions et d'assurer une interactivité plus élaborée avec l'apprenant.

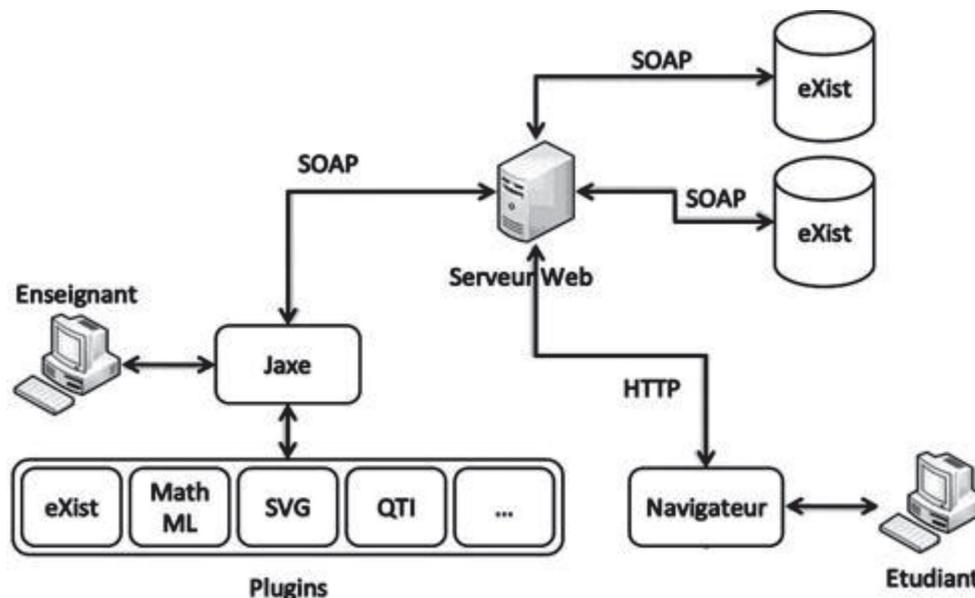


Figure 4. Architecture SOA du système XESOP

4. Architecture du système

L'objectif principal de notre projet est la réalisation d'un enseignement ouvert, collaboratif et à distance par l'implantation d'une plate-forme *e-learning* fondée sur des logiciels libres en s'appuyant sur les technologies de l'Internet et des standards en ingénierie pédagogique.

Notre plate-forme «Xesop» assure la création d'un contenu d'enseignement et la construction d'un parcours pour l'apprenant, au moyen d'une structure sémantique conforme aux standards SCORM (2004) et IMS (*Global Learning Consortium, 2011*). Les éléments essentiels du projet consistent d'une part, en des supports sémantiques pour la création, la présentation et le stockage des objets pédagogiques basés sur des technologies XML et d'autre

part, en des moyens pour la gestion du parcours de l'étudiant (Madjarov *et al.*, 2004) et l'exécution d'exercices à distance fondés sur les Services Web (Madjarov et Boucelma, 2006).

Notre conception peut être présentée fonctionnellement par le scénario suivant : l'auteur des cours crée ses objets pédagogiques à l'aide d'outils sémantiques à savoir textes, équations mathématiques, graphiques vectoriels, questionnaires, exercices, etc. (Figure 4). Le stockage de ces objets est possible dans un format XML conforme à une structure de métadonnées préalablement définie par un *XML Schema* (W3C, 2004) et en accord avec les standards *e-learning* en vigueur. Cette grammaire garantit, par la suite, la conformité des documents pédagogiques enregistrés dans une base de données *XML native* (eXist), et sert à la «bonne formation» des sorties générées à la demande dans des formats standardisés de l'espace Web (XML, HTML, XHTML MP, PDF, JPEG, etc.).

La réutilisation de ces documents dans un espace collaboratif entre auteurs est assurée par le format natif (XML) de leur création et leur accessibilité par l'appel d'un service Web. Ainsi, l'auteur d'un cours peut-il facilement intégrer des composants propres à des composants disponibles dans la base des objets pédagogiques initialement appartenant à un autre cours et créés par un autre auteur dans un environnement différent. L'intégration des données par le moyen des services Web apporte un potentiel reconnu. Ainsi, l'intégration de données pédagogiques avec des sources externes est-elle facilement réalisable à l'aide des services Web à travers des *wrappers* appropriés.

L'apprenant de son côté, après s'être identifié dans le système pour une formation, choisit et poursuit un cours à travers des textes, questionnaires et exercices. S'il a besoin d'informations complémentaires, il peut adresser une requête au système de gestion des services Web (SGSW). Ce système joue le rôle de médiateur entre les parties: (1) auteur, (2) formation et (3) bases de données d'objets pédagogiques (BDOP). Le service dédié va rechercher les informations dans les bases de données connues et accessibles (Interactions: SOAP - Serveur UDDI - WSDL - Service Web - Poste client). Le résultat retourné est présenté côté apprenant dans un format adapté aux navigateurs Web. Dans ce cas, notre objectif est atteint par une intégration de données avec transformation de formats à partir de sources hétérogènes via les technologies des services Web.

4.1. Le concept de l'éditeur IMS-QTI

La réalisation de cet éditeur QTI répond à un besoin existant et poursuit l'idée évolutive de notre famille d'éditeurs sémantiques *open source* qui font partie intégrante du système Xesop (Madjarov *et al.*, 2005). Nous obtenons ainsi des fonctionnalités complémentaires à la création d'objets pédagogiques réutilisables dans des cours et des questionnaires. L'éditeur QTI est développé comme un objet externe (iDevice). C'est un *plug-in* chargé à la demande par l'auteur au moment où il désire associer un questionnaire à son cours, ou créer un test d'évaluation de fin de cours. Pour la réalisation, nous avons eu l'idée de séparer le contenu des questionnaires du contenu des cours tout en conservant le format XML, selon le standard IMS-QTI. Ainsi, nous poursuivons l'idée de la portabilité et l'interopérabilité des ressources pédagogiques, étant donné que le contenu est enregistré dans son format d'origine dans l'arborescence des collections d'une base de données *XML native*. Sur demande, ces collections ou des parties sélectionnées peuvent être transformées dans un format de présentation et ensuite publiées sur un serveur Web. Pour intégrer par la suite un questionnaire à un cours, il suffit de renseigner la balise <question> avec son URL. Dans le contenu du cours, le questionnaire apparaît comme un hyperlien qui pourrait être suivi par l'étudiant.

L'exécution d'un questionnaire passe par la présentation de chaque question en page unitaire. Ainsi, l'apprenant est concentré sur la réponse de la question en cours sans la possibilité de consulter les questions suivantes ou précédentes. C'est une approche pédagogique restrictive, mais elle contribue à la réalisation de tests adaptatifs, où la question suivante apparaît en fonction de la réponse de l'apprenant et/ou du temps écoulé. Nous avons vu déjà que le standard IMS-SS (IMS Global Learning Consortium, 2003) contribue dans une certaine mesure à l'enchaînement dynamique des questions, mais nous pensons que grâce à notre concept, nous pouvons le renforcer. Cette approche demande une stratégie pédagogique dynamique. La réalisation de cette stratégie demande un modèle et des algorithmes plus poussés que ce que nous avons en partie étudié et réalisé en (Madjarov *et al.*, 2004), (Madjarov et Betari, 2008). Dans un QCM par exemple, une question peut proposer plusieurs réponses possibles, dont une bonne, une mauvaise, et d'autres approximatives. La stratégie pédagogique doit déterminer quelle autre question serait en mesure de déterminer à partir de la réponse obtenue, s'il s'agit d'une erreur, d'une incompréhension, ou d'une frappe aléatoire. Un tel enchaînement des questions a un double effet, celui d'évaluer les connaissances de l'apprenant, et celui de les enrichir avec des questions instructives et ciblées.

La présentation des questions en séquences dynamiques demande une solution technique compatible avec l'environnement de l'apprenant. Le format XHTML présente la solution simple, mais privée d'interactivité. L'ajout d'objets téléchargeables tels que les composants ActiveX ou Flash apporte des solutions dynamiques, mais ne résout pas entièrement les problèmes à cause des défauts de sécurité et une incompatibilité avec tous les systèmes et navigateurs Web. En plus, ce sont des solutions non standardisées par le consortium W3C. Les applets Java sont des objets téléchargeables et portables beaucoup plus sécurisés qui assurent une interactivité parfaite, mais demandent plus d'efforts de développement et sortent hors du cadre des formats choisis. Les techniques ISAPI telles que JSP, ASP ou PHP fournissent du dynamisme certes, mais du côté serveur et non pas du côté client. Le résultat transféré

est toujours une page en format HTML. Il est évident que nous avons besoin d'une solution technique qui regroupe plusieurs de ces atouts et élimine les défauts déjà cités.

Nos études dans le domaine des solutions libres et *open source* nous ont amenés à la découverte du ZK Framework (2006). C'est un cadre programmable basé sur la technique AJAX (Asynchronous JavaScript and XML) (Madjarov et Boucelma, 2006) pour la création d'interfaces riches dédiées à un environnement Web. ZK propose un langage événementiel et un script Java en utilisant un ensemble de composants : XUL (XML-based User interface Language) pour la description des interfaces graphiques, XHTML et ZUML (ZK User Interface Markup Language) pour la structuration et la présentation du contenu. ZK manipule les composants XUL et XHTML au sein des pages Web, en se basant sur les événements déclenchés par les activités de l'utilisateur (AJAX). La technique AJAX (Snell, 2005) s'ajoute parfaitement aux services Web qui représentent un élément important de notre conception d'intégration des données et des applications. En se fondant sur cette technique, nous sommes en mesure d'assurer la réalisation de notre conception pour les questionnaires. Ainsi, suivant la réaction (réponse) de l'apprenant, les événements engendrés et traités peuvent apporter de nouveaux éléments ou en retirer d'autres de la page présentée à l'écran, suivant la stratégie pédagogique choisie. L'implantation ne demande pas de configuration particulière ou des ressources complémentaires du côté client. Le concept des questionnaires est présenté en Figure 5.

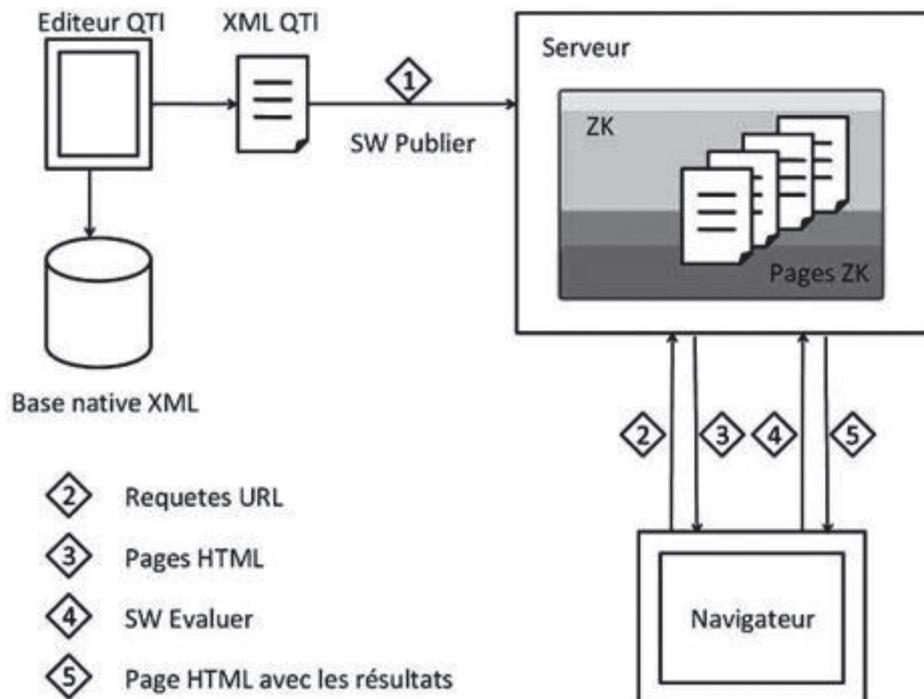


Figure 5. Le gestionnaire des questions

5. Conclusion

Notre travail de recherche s'inscrit dans les initiatives d'interopérabilité et de réutilisation non seulement des objets pédagogiques, mais aussi des applications qui les animent (Madjarov et Betari, 2008). Dans cet article, nous avons présenté en particulier nos travaux de recherche concernant la conception et le développement des composants «QTI editor» et «QTI player» intégrés dans notre plate-forme Xesop pour la création de tests et d'exercices de type questionnaire et leur présentation du côté de l'apprenant.

Les solutions existantes telles que (Joosten ten Brinke *et al.*, 2005), (Giacomini Pacurar *et al.*, 2005) se fondent sur des techniques d'éditeurs XML pour la création des «manifest» et des collections QTI. L'interprétation basée Web est confiée à des modèles linéaires et interactifs certes, mais qui n'assurent pas un dynamisme adaptatif dans la gestion des séquences. Un des *players* proposés est Q-Player qui permet l'interprétation de tout test décrit selon IMS-QTI, à l'aide d'un lecteur programmé en Flash. C'est une solution en format propriétaire qui requiert l'installation sur le navigateur de l'apprenant d'un *plug-in* Flash. Le *player*, lui-même, n'est ni libre, ni *open source*.

L'implantation de la technique AJAX dans notre conception fondée sur les services Web assure la possibilité de modifier *inline* les données pour maintenir l'interactivité et le dynamisme dans la gestion des séquences de pages. L'implication du JavaScript et de l'objet XMLHttpRequest permet d'effectuer des requêtes vers le serveur d'applications Web de manière asynchrone, qui de son côté introduit le résultat au point de l'appel, tout en restant dans un cadre standardisé Web.

La technique AJAX présente des avantages en termes d'interactivité et de souplesse de publication de contenus. Cette technique évite un aller-retour de requêtes entre le navigateur de l'apprenant et le serveur du site Web qui soutient l'application ou le service. Elle n'implique plus l'installation de «plug-in» ou de programmes spécifiques et non standardisés.

Référence

- Global Learning Consortium (2011) *IMS, Instructional Management Systems*, <http://www.imsglobal.org/>
- IEEE (2002) *LTSC - LOM (Learning Object Metadata)*. <http://ltsc.ieee.org/wg12/>
- Anderson T., McKell M. (2001) *IMS Content Packaging Best Practice Guide*, Final Specification, Version 1.1.2, IMS
- Michel C., Rouissi S. (2003) e-learning: normes et spécifications. Étude des spécifications LOM et IMS-QTI caractérisant des documents numériques inter-échangeables et réutilisables pour l'acquisition et l'évaluation des connaissances. *Revue Document Numérique «Les nouvelles facettes du document électronique dans l'éducation»*
- IMS (2005) Question & Test Interoperability (QTI) specification. *QTI 2.1 v2 release*, <http://www.imsglobal.org/question/#version2.1>
- Madjarov I., Betari A., Bakkoury Z. (2005) Web Service Based Remote Development Environment for an e-learning System. *Proceedings ICHSL5/CAPS5 Fifth International Conference on Human System Learning*, Morocco. Vol. Human System Learning Who is in control, edited by Khaldoun Zreik *et al.*, Europia, Paris, pp 79-95
- Joosten ten Brinke, D., Gorissen, P., and Latour I. (2005) Integrating Assessment into e-learning Courses, Chapter 10. *Learning Design: A handbook on Modelling and Delivering Networked Education and Training*. Rob Koper & Colin Tattersall (Eds.), ISBN 3-540-22814-4 Springer.
- Giacomini Pacurar E., Philippe Trigano P., Alupoae S. (2005) A QTI editor integrated into the netUniversité Web portal using IMS LD. *Journal of Interactive Media in Education (Advances in Learning Design. Special Issue*, eds. Colin Tattersall, Rob Koper)
- Madjarov I., Boucelma O. (2006) Data and Application Integration in Learning Content Management Systems: a Web Services Approach. *Proceedings of First European Conference on Technology Enhanced Learning*. Crete, Greece, W. Nejdil and K. Tochtermann (Eds.): EC-TEL 2006, LNCS 4227, pp. 272 – 286, Springer-Verlag Berlin Heidelberg
- SCORM (2004) *ADL Sharable Content Object Reference Model*, <http://www.adlnet.org>.
- Madjarov I., Boucelma O., Betari A. (2004) An Agent- and Service-Oriented, e-learning Platform. *LNCS, Vol. 3143: Advances in Web-Based Learning - ICWL 2004, Third International Conference*. Wenyin Liu, Yuanchun Shi, Qing Li (Eds.) 2004, pages 27-34. Springer-Verlag Berlin Heidelberg
- W3C (2004) *XML Schema*, <http://www.w3.org/XML/Schema>.
- IMS Global Learning Consortium (2003) *IMS Simple Sequencing Specification, Version 1.0 Final Specification*, <http://www.imsglobal.org/ss/>.
- ZK (2006) *ZK- Ajax Web framework*. <http://www.zkoss.org>
- Jesse James Garret (2005) *Ajax: A New Approach to Web Applications*. <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
- Snell J. (2005) *Call SOAP Web Services with Ajax*. IBM Press
- Madjarov I., Betari A. (2008) Adaptive Learning Sequencing for Course Customization : a Web Service Approach. *Proceedings IEEE-APSCC 2008*, IEEE Asia-Pacific Services Computing Conference. pp.530-535,
- Madjarov I., Boucelma O. (2011) Multimodality and Context-adaptation for Mobile Learning, , *Social Media Tools and Platforms in Learning Environments*, pp. 257-276, 1st Edition, White B., King I., Tsang P. (Eds.) ISBN 978-3-642-20391-6, Springer. DOI: 10.1007/978-3-642-20392-3_15.
- Madjarov I. (2010) Adaptation de contenu pédagogique pour téléphone mobile, Actes du 2^{ème} colloque: *Les Cahiers pédagogiques des R&T*, pp.67-75, Guyanne Française.

Fiche équipe

L'association GREENTIC

Aawatif Hayar

ENSEM, université Hassan-II, Casablanca, Maroc
aawatif.hayar@hotmail.fr

Résumé

GREENTIC est une organisation à but non lucratif créée en avril 2010 dans le domaine des technologies vertes de l'information et des communications (*GREEN Information and Communication Technologies*). Cette initiative a pour but de renforcer les activités des laboratoires marocains dans le secteur des technologies de télécommunication vertes qui respectent l'environnement et la qualité de vie des citoyens. À travers ses multiples activités de collaboration avec l'Europe, GREENTIC vise à établir un partenariat scientifique euro-méditerranéen équilibré qui favorise l'innovation technologique respectueuse de l'environnement et au service de sa protection dans l'espace méditerranéen grâce à l'utilisation intelligente des technologies vertes de l'information.

Mots-clés

technologies vertes

1. Présentation

GREENTIC est une organisation à but non lucratif créée en avril 2010 grâce aux soutiens de l'université Hassan II de Casablanca (UH2C), la Fondation Hassan II pour les Marocains résidents à l'Étranger et des experts marocains, du Maroc et de l'étranger, dans le domaine des technologies vertes de l'information et des communications (*GREEN Information and Communication Technologies*).

Cette initiative a pour but de renforcer les activités des laboratoires marocains dans le secteur des technologies de télécommunication vertes qui respectent l'environnement et la qualité de vie des citoyens. Ceci est réalisé grâce à un fort partenariat entre les laboratoires des Technologies de l'Information et de la Communication marocains et centres de recherche de renommée en Europe et aux États-Unis. À travers ses multiples activités de collaboration avec l'Europe, GREENTIC vise à établir un partenariat scientifique euro-méditerranéen équilibré qui favorise l'innovation technologique respectueuse de l'environnement et au service de sa protection dans l'espace méditerranéen grâce à l'utilisation intelligente des technologies vertes de l'information. L'organisation vise aussi à encourager le partenariat académie-entreprise local et international à travers des contrats de recherche et de conseil développés conjointement avec l'industrie. En parallèle avec ce partenariat qui vise à répondre à des besoins liés à des exigences d'applications diverses, GREENTIC encourage une recherche d'excellence à travers son programme doctoral mis en place en partenariat avec de grands centres mondiaux comme la chaire Alcatel Lucent à Supelec, le LAAS et l'école d'ingénieur ISITV de l'Université de Toulon en France, le centre Athens Information Technology en Grèce, le CTTC à Barcelone en Espagne ainsi que l'université Northeastern University à Boston aux États-Unis.

2. Motivation et objectifs

Le secteur des Technologies de l'Information et de la Communication (TIC) est responsable de 2% d'émission Carbone en Europe par exemple sans oublier que ces technologies sont aussi intégrées dans tous les secteurs de l'économie. Un rapport de la commission européenne a montré que l'amélioration de l'efficacité énergétique des TIC pourrait contribuer à réduire de 15% des émissions du CO₂.

Le niveau des ondes électromagnétiques utilisées dans les télécommunications est aussi un sujet de santé publique. C'est pour cela que nous devons encourager et renforcer la recherche scientifique et technologique dans le domaine des télécommunications vertes afin de concevoir des solutions technologiques innovantes et efficaces qui respectent un cadre de vie sain pour les citoyens.

Notre initiative GREENTIC s'inscrit complètement dans cette démarche et vise à mettre en place un centre de R&D spécialisé dans les télécommunications vertes pour développer et proposer des solutions technologiques performantes qui respectent l'environnement en garantissant en même temps, une meilleure efficacité énergétique des réseaux télécom et une qualité de vie non dégradée des citoyens, à travers une gestion optimisée et intelligente

des interférences et des radiations électromagnétiques. Nous visons aussi à développer en collaboration avec les partenaires marocains et internationaux institutionnels, industriels et académiques une recherche d'excellence au Maroc dans ce domaine permettant de contribuer à réduire l'effet des technologies TIC sur l'environnement et sur la santé.

3. Partenariats

Plusieurs partenaires institutionnels, industriels ou académiques au Maroc et à l'étranger ont déjà exprimé leur soutien à l'initiative GREENTIC à travers un financement de sujets de recherche, des projets de co-encadrements de thèses, accueil de thésards ou pour assurer des séminaires en technologies de l'information et des télécommunications vertes au centre GREENTIC.

Établissements marocains :

- Université Hassan II de Casablanca, Maroc,
- La Fondation Hassan II pour les marocains résidents à l'étranger,
- Direction de la Technologie au Ministère l'Education Nationale, de l'Enseignement Supérieur, de la Formation des Cadres et de la Recherche Scientifique au Maroc,
- Maroc Numeric Cluster,
- ENSIAS, Rabat Maroc,
- EMI, Rabat Maroc,
- FST Settat Maroc,
- Faculté des Sciences Université Moulay Ismail Meknès Maroc,
- Faculté des Sciences Université Abdelmalek Essaadi Tétouan Maroc
- EST Salé,
- Lydec Casablanca.
-

Centres de recherche étrangers :

- Chaire Radio Flexible Alcatel Lucent à Supelec en France,
- AIT centre international de recherche en Télécommunications en Grèce,
- L'Université Northeastern, Boston États-Unis
- CTTC Centre des Technologies des Télécommunication, Barcelone, Espagne,
- ISITV Université Sud Toulon Var, France.

L'équipe de recherche Siweb

Système d'Information et Web

Ounsa Roudiès

École Mohammadia d'Ingénieurs (EMI), université Mohammed V-Agdal, Rabat, Maroc
roudiès@emi.ac.ma

Résumé

L'équipe Siweb (<http://siweb.jimdo.com/>) relève du Centre d'études doctorales «Sciences et techniques de l'Ingénieur» et regroupe des enseignants-chercheurs et des doctorants de l'École Mohammadia d'Ingénieurs (EMI), à Rabat (Maroc).

Notre objectif est d'offrir un espace de recherche dédié aux systèmes d'information (SI) complexes dans un contexte marqué par le Web. Les SI considérés selon deux dimensions :

- verticalement lorsque le SI est perçu comme un continuum de systèmes conçus et manipulés à différentes échelles, allant du SI urbanisé (groupe structuré) au SI spécifique d'une entreprise.
- horizontalement quand le SI est considéré comme une nébuleuse d'informations accessibles à travers le Web.

Les activités phares de ses membres sont la gestion du bureau W3C à Rabat par Pr. Tounsi, la publication de la revue e-TI (<http://www.revue-eti.net>). L'équipe participe à l'organisation de plusieurs manifestations: JRSI, W3C days, Inet (Internet Society), Agile Tour (Rabat), etc.

Mots-clés

système d'information, réutilisation, pattern, composant, composition de services, MDA, méta modélisation, sémantique, urbanisation, assurance Qualité, Sécurité, accessibilité, personnalisation, W3C, revue e-TI

1. Présentation

L'équipe Siweb (<http://siweb.jimdo.com>) s'inscrit dans la continuité de l'équipe accréditée en 2008 dans le cadre du laboratoire Système d'Information et Répartition (SIR).

Notre objectif est d'offrir un espace de recherche dédié aux systèmes d'information (SI) complexes dans un contexte marqué par le Web. Les SI considérés selon deux dimensions :

- verticalement lorsque le SI est perçu comme un continuum de systèmes conçus et manipulés à différentes échelles, allant du SI urbanisé (groupe structuré) au SI spécifique d'une entreprise.
- horizontalement quand le SI est considéré comme une nébuleuse d'informations accessibles à travers le Web.

Ces systèmes d'information se prolongent à travers le Web qui est resté un défi. L'exploitation de son potentiel est favorisée par l'implication des membres de Siweb dans le W3C et dans la revue électronique internationale eTI (<http://www.revue-eti.net>).

2. Axes de recherches

Les méthodes de conception de systèmes d'information ont atteint leur maturité dans le cas de SI centralisés correspondant à des organismes de taille moyenne, relativement stables. Le concept de référence est l'objet qui a été exploité intensivement de l'analyse au déploiement des applications.

Ces méthodes de conception restent à affiner dans le cas des SI complexes qui doivent être appréhendés à plusieurs échelles: SI urbanisé (groupes de sociétés, e-gouvernement, etc.), SI métiers mis en place par des solutions spécifiques telles que les progiciels ou les composants de granularité diverse servant de briques élémentaires pour l'élaboration

de SI composite, SI fondé sur des connaissances issues d'un Web évoluant en permanence ... La maîtrise de ce multi-échelle dépend en grande partie des méthodes et des outils utilisés pour leur construction ainsi que du degré de formalisation, à la fois des produits de la conception, mais aussi des processus et stratégies.

Pour compenser les limites de l'objet, nous utilisons des paradigmes comme le composant, le modèle (MDA), le service (SOA, Web services, m-services) ou le processus à travers les pistes de recherches suivantes :

- Formalisation de composants réutilisables et traçabilité (patterns, services, urbanisation...),
- Formalisation de processus qualité, sécurité et décisionnels (alignement, évaluation, risques),
- Développement du Web et standardisation.

3. Composition de l'équipe

Responsable

- Ounsa Roudiès (Systèmes d'information), roudiès@emi.ac.ma

Membres permanents

- Anwar Adil (ingénierie dirigée par les modèles, vérification), anwar@emi.ac.ma
- Fatima-Zahra Belouadha (composition de Web et m-services, BI), belouadha@emi.ac.ma
- Sara Bouchikhi (Sécurité des SI), bouchikhi@emi.ac.ma
- Najib Tounsi (Accessibilité du Web et Internationalisation), ntounsi@emi.ac.ma

Chercheurs

- H. Ouchetto (évolution et adaptation de composant)
- Noufissa Mouline, Imane Rhalem (processus Qualité et Sécurité)
- Aissam Mesrar (SI pervasifs)
- Ibrahim El Bitar, Hajar Omrana, (Composition de services sémantiques)
- Omar Barki, Reda Benkhouya, Safae Nassiri, Morad Ellali (urbanisation, proc. décisionnels).

4. Manifestations

L'organisation de JRSI, Journée de Recherches en Systèmes d'Information, qui est dédiée à un thème d'actualité des systèmes d'information nous semble importante pour faire connaître les limites de cette spécialité souvent confondue avec le génie logiciel. En effet, les SI qui sont à la croisée de plusieurs disciplines (informatique, sciences de l'organisation, ergonomie...) demeurent paradoxalement mal cernés dans le domaine de la recherche et n'intéressent que peu d'équipes, comparativement aux autres disciplines à caractère plus « mathématique ».

La publication de la revue eTI qui est notre activité phare reste bien entendu une priorité.

Les membres de l'équipe participent activement à plusieurs associations porteuses du Web: le W3C consortium dont le bureau de Rabat est sous la direction de Pr. Tounsi, Internet society Maroc (MISOC).

