



HAL
open science

NICOLAT : un système iNformatIque COmmunautaire et AdapTatif support d'une Communauté de Pratique pour un apprentissage basé sur la résolution de problèmes

Rachid Belmeskine

► **To cite this version:**

Rachid Belmeskine. NICOLAT : un système iNformatIque COmmunautaire et AdapTatif support d'une Communauté de Pratique pour un apprentissage basé sur la résolution de problèmes. Autre [cs.OH]. Université de Picardie Jules Verne; Université Sidi Mohamed ben Abdellah (Fès, Maroc). Faculté des Sciences et Techniques, 2015. Français. NNT : 2015AMIE0030 . tel-03653248

HAL Id: tel-03653248

<https://theses.hal.science/tel-03653248>

Submitted on 27 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse de Doctorat

Informatique

Ecole Doctorale en Sciences Technologie et Santé (ED 585)

Université de Picardie Jules Verne

Centre d'Etudes Doctorales : Sciences et Techniques de l'Ingénieur

Faculté des Sciences et Techniques Fès

Université Sidi Mohamed Ben Abdellah

présentée par

Rachid BELMESKINE

28 Décembre 2015

pour obtenir le grade de Docteur de l'Université de Picardie Jules Verne
et le grade de Docteur de l'Université Sidi Mohamed Ben Abdellah

***NICOLAT : Un système iNformatique Communautaire et
AdapTatif support d'une Communauté de Pratique pour un
apprentissage basé sur la résolution de problèmes***

M. Rachid Benslimane, Professeur

M^{me} Rachida Ajhoun, Professeur

M. Serge Garlatti, Professeur

M. El Mustapha Mouaddib, Professeur

M. Arsalane Zarghili, Professeur

M^{me} Ahlame Begdouri, HDR

M^{me} Dominique Lecllet Groux, HDR

Président

Rapporteur

Rapporteur

Examineur

Examineur

Codirectrice de thèse

Codirectrice de thèse

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude ainsi que mes sincères remerciements à mes encadrantes de thèse : Mme. Ahlame BEGDouri et Mme. Dominique LECLET-GROUX pour le soutien, l'aide, et la participation dans l'accomplissement de ce travail de thèse.

Je souhaite adresser également mes vifs remerciements aux membres du jury qui ont accepté d'examiner mon travail, en particulier à Mme. Rachida AJHOUN, M. Serge GARLATTI et M. Arsalane ZARGHILI de m'avoir fait l'honneur d'accepter d'être rapporteurs de ma thèse, enfin, à M. Rachid BENSLIMANE et M. El Mustapha MOUADDIB pour avoir accepté de participer à ce jury.

Je remercie aussi vivement l'équipe du laboratoire MIS (Modélisation, Information et Systèmes) et celle du laboratoire SIA (Systèmes Intelligents et Applications).

En dernier, mes remerciements aux personnes qui m'ont aidé de près ou de loin à réaliser ce travail.

Résumé

Un Système Informatique Communautaire (SIC) est un système qui permet d'assister l'activité d'une communauté d'acteurs qui partagent leur pratique et qui ont un objectif d'apprentissage en commun (Communauté de Pratique). L'apprentissage via un tel système se fait via les interactions entre ses différents acteurs. Cet apprentissage étant communautaire, il est important que la communauté puisse vivre et soit active. Cependant, l'activité de la communauté peut être freinée par la démotivation de ses membres aboutissant à l'abandon de l'utilisation du système. Dans un SIC support d'une Communauté de Pratique (CoP) dont les membres interagissent pour résoudre leurs problèmes, les facteurs de démotivation peuvent être liés à la sollicitation régulière des membres pour des résolutions de problèmes similaires, à la difficulté de prise en main de nouveaux outils d'interaction par les membres de la communauté, au temps de réponse long à une demande d'aide ou à la réception des solutions non adaptées au contexte du problème.

Dans ce cadre, nous nous sommes intéressés à la conception et au développement d'un système iNformatIque COmmunautaire mobiLe et AdapTatif, appelé NICOLAT. Ce dernier vise à supporter une CoP dans laquelle l'apprentissage s'effectue via la résolution communautaire de problèmes en offrant des solutions qui permettent de limiter les facteurs qui peuvent aboutir à la démotivation des membres de la CoP.

Pour expérimenter et valider les solutions que nous proposons à travers ce système, nous ciblons la CoP des enseignants usagers de la méthode pédagogique MAETIC, qui peuvent rencontrer, en classe, des problèmes dans l'usage de celle-ci.

Ainsi, nos principales contributions se résument dans les points suivants : 1) La mise en place du noyau communautaire du système NICOLAT. Ce dernier est sous forme d'un réseau social supportant la résolution communautaire de problèmes, 2) La mise en place d'une couche de résolution de problèmes ayant pour objectif d'aider le membre à résoudre son problème par exploitation de l'historique des problèmes résolus dans le passé. Ceci pour minimiser le nombre de demandes d'aide répétitives. Le cycle du raisonnement RàPC (Raisonnement à Partir de Cas) a été utilisé pour guider ce processus, 3) La mise en place de deux couches d'adaptation permettant de supporter les interactions des membres dans les outils d'interaction qu'ils préfèrent ou avec lesquels ils sont familiarisés le plus. L'objectif visé par cette adaptation est, d'une part, de dépasser les problèmes de prise en main de nouveaux outils d'interaction. D'autre part, de permettre l'accès au système en cas de mobilité et minimiser ainsi le temps de réponse, 4) La mise en place d'une approche de sélection des membres qui peuvent contribuer positivement à résoudre un problème à qui faire aboutir la demande

d'aide. L'objectif est de permettre à un membre cherchant à résoudre son problème de recevoir une réponse pertinente.

Abstract

A Community Computer System (CCS) is a system that allows assisting the activity of a community of members who share their practice and have a common learning objective (Community of Practice). In such system, learning occurs through interactions between its different actors. As the learning is based-community, it is important that the community is alive and active. However, the community activity may be hampered by the demotivation of its members resulting in the abandonment of the system. In a CCS support of a Community of Practice (CoP), whose members interact to solve their problems, the demotivation factors can be related to the regular solicitation of members for the solving of similar problems, to the problem of new interaction tools' manipulation, to the long response time to a help request or to the obtaining of non-adapted solutions to the context of the problem.

In this context, we focused on design and development of an adaptive and mobile community system, called NICOLAT (iNformatIque COmmunautaire mobiLe et AdapTatif). The latter aims to support a Community of Practice (CoP) in which learning is done through community problem solving by providing solutions that limit the factors that can lead to the demotivation of the CoP members.

To experiment and validate the solutions we provide through this system, we target the CoP of teachers users of the MAETIC pedagogical method, who can meet, in classroom, problems in the use of it.

Thus, our main contributions are summarized in the following points: 1) Establishment of community kernel of the NICOLAT system. The latter is as a social network supporting the community solving of problems, 2) Implementation of problems resolution layer that aims to help the member solve his problem through the exploitation of the history of problems solved in the past. This is to minimize the number of repetitive help requests. The cycle of the CBR (Case-Based Reasoning) was used to guide this process, 3) Establishment of an interactions adaptation layer to support the members' interactions in the interaction tools they prefer or with which they are most familiar. The purpose of this adaptation is, firstly, to exceed the problems of interaction tools manipulation. On the other hand, to enable access to the system in case of mobility and thereby minimize response time, 4) Establishment of a dynamic approach of selection of members who can contribute positively to solve a problem, to whom bring the help requests. The objective is to enable a member seeking to solve his problem to receive a relevant answer.

Table des matières

Remerciements	1
Résumé	2
Abstract	4
Table des matières	5
Liste des figures	10
Liste des tableaux	13
Chapitre 1 : Contexte des recherches	14
1. Domaine	14
2. Questions de recherche.....	15
3. Objectifs des recherches	17
4. Organisation du manuscrit	17
Chapitre 2 : Le système NICOLAT et sa Communauté de Pratique	19
1. Communautés de Pratique : Etat de l'art.....	19
1.1 Définitions et spécificités	19
1.2 CoP comme structure sociale	20
1.3 Cycle de vie d'une CoP	22
1.4 Intérêt des CoPs.....	23
1.4.1 Intérêts pour les organisations	24
1.4.2 Intérêts pour les membres.....	24
1.5 Exemples de CoPs : Success Story	25
1.6 Différences entre les CoPs et autres structures sociales.....	26
1.7 Communautés de Pratiques virtuelles	28
2. CoP MAETIC.....	32
3. NICOLAT comme support communautaire de la CoP MAETIC	35
3.1 Structure du système	35

3.2 Fonctions du noyau	36
3.3 Développement du noyau	38
3.4 Interfaces	39
4. Généricité de NICOLAT	43
5. généricité de la couche communautaire	44
5.1 Fonctionnalités	44
5.2 Structure générique de problèmes	45
5.3 Proposition d'implémentation	47
6. Conclusion.....	49
Chapitre 3 : La couche « Résolution de Problèmes »	50
1. Problématique et objectif de la couche RP.....	50
2. Raisonnement à Partir de Cas.....	51
2.1 Principe.....	51
2.2 Cycle du raisonnement RàPC.....	52
2.3 Modèles du RàPC.....	55
2.3.1 Modèle structurel.....	55
2.3.2 Modèle conversationnel	55
2.3.2 Modèle textuel.....	56
2.4 Avantages et inconvénients du RàPC.....	56
3. Conception de la couche RP.....	57
3.1 Conception fonctionnelle	57
3.2 Structure d'un cas.....	60
4. Processus de remémoration de la couche RP	62
4.1 Recherche de cas similaires.....	63
4.1.1 Techniques de calcul de la similarité textuelle.....	64
4.1.2 Notre approche de calcul de la similarité textuelle	64
4.1.3 Indexation.....	65
4.1.4 Similarité entre les cas.....	67
4.2 Tri des cas sélectionnés	68

5. Validation de la couche RP	69
5.1 Base initiale de cas	70
5.2 Etape d'élaboration	71
5.3 Etape de remémoration.....	73
6. Généricité de la couche RP	74
6.1 Conception fonctionnelle	74
6.2 Structure d'un Cas	75
7. Conclusion.....	76
Chapitre 4 : Les interactions dans le système NICOLAT	77
1. Adaptation dans les EIAHs	77
1.1 Définition	77
1.2 Adaptation dans les Systèmes de Tutorat Intelligents.....	78
1.3 Adaptation dans les Systèmes Hypermédias	79
1.4 Adaptation des scénarios pédagogiques	79
1.5 Adaptation dans les systèmes support de l'apprentissage collaboratif	80
1.6 Adaptation dans les systèmes de l'apprentissage mobile.....	81
2. Gestion du contexte.....	83
2.1 Définition	83
2.2 Sensibilité au contexte.....	84
2.3 Architecture d'un système sensible au contexte.....	85
2.4 Dimensions du contexte	86
2.5 Approches de modélisation du contexte.....	86
3. Modélisation du contexte	89
3.1 Modèle générique du contexte	89
3.2 Modélisation du contexte pour la CoP MAETIC.....	91
4. Adaptation des interactions dans le système NICOLAT	93
4.1 Problématique et objectif de l'adaptation.....	93
4.2 Architecture globale des couches d'adaptation.....	95
4.3 Couche AI « Adaptation des Interactions »	97

4.4 Couche Interface	99
5. Fonctionnement du système	100
5.1 Application Email	100
5.2 Application Facebook	104
5.3 Application Android.....	108
5.4 Application SMS	113
6. Expérimentation	117
6.1 Contexte	117
6.2 Hypothèses	117
6.3 Population cible.....	117
6.4 Description du protocole	118
6.5 Analyse des questionnaires	118
6.5.1 Hypothèse 1	118
6.5.2 Hypothèse 2.....	119
7. Généricité de l'adaptation des interactions	121
8. Conclusion.....	121
Chapitre 5 : La sélection des membres dans le système NICOLAT	122
1. Formation des groupes	122
1.1 Formation des groupes dans les systèmes d'apprentissage collaboratif	122
1.2 Formation de groupes dans le système NICOLAT	125
2. Approche de sélection	128
2.1 Principe de l'approche de sélection.....	128
2.2 Sélection statique.....	130
2.3 Généricité de la sélection statique	133
3. Sélection dynamique	134
3.1 Représentation des données	134
3.2 Mesure de similarité	136
3.3 Représentation des indices contributions	137
3.4 Algorithme de filtrage	139

3.5 Représentation des traces	141
3.6 Algorithme d'analyse de traces	142
3.7 Généricité de la sélection dynamique.....	144
4. Expérimentation	144
4.1 Scénario de validation	144
4.2 Gain en termes de qualité de la sélection	145
4.3 Gain en termes de temps d'exécution	147
6. Conclusion.....	149
Conclusion.....	150
Bibliographie.....	153
Publications et Communications	161
Annexe 1 : Questionnaire d'évaluation de l'adaptation des interactions	162
Annexe 2 : Profils des membres de la communauté	165
Annexe 3 : Traces des contributions des membres de la communauté	168
Annexe 4 : Variations des gains pour différents types de difficultés.....	174

Liste des figures

Figure 1. Niveaux de participation dans une CoP (Wenger, McDermott et al. 2002)	21
Figure 2. Le cycle de vie d'une CoP (Langelier 2005)	22
Figure 3. Cycle de vie de MAETIC (Lecllet and Talon 2008).....	32
Figure 4. Classification des catégories de difficultés liées à une étape de MAETIC.....	34
Figure 5. Classification des catégories de difficultés transversales	34
Figure 6. Diagramme de cas d'utilisation du système communautaire.....	37
Figure 7. Formulaire de description d'une difficulté MAETIC	40
Figure 8. Page d'actualités du noyau	41
Figure 9. Page permettant de visualiser une difficulté MAETIC.....	42
Figure 10. Diagramme de cas d'utilisation du noyau spécifiant les fonctionnalités liées aux spécificités de MAETIC	45
Figure 11. Structure générique du problème	46
Figure 12. Modèle de classification de problèmes	46
Figure 13. Classification réduite des problèmes liés au développement informatique	47
Figure 14. Classification des difficultés MAETIC liées à l'étape 2 (Cadrage).....	47
Figure 15. Classification des problèmes de développement informatique en fichier XML.....	48
Figure 16. Classification des difficultés MAETIC liées à l'étape 2 en fichier XML	48
Figure 17. Algorithme du script permettant de créer la taxonomie représentant la classification de problèmes	49
Figure 18. Fonctionnement d'un système RàPC	54
Figure 19. Le cycle du RàPC implémenté par la couche RP	58
Figure 20. Processus de résolution de difficultés MAETIC.....	59
Figure 21. Le diagramme de cas d'utilisation de la couche RP	60
Figure 22. La partie problème de la structure d'un cas représentant une difficultés MAETIC	61
Figure 23. Structure d'un cas d'une difficulté MAETIC	62
Figure 24. Résumé du processus d'indexation.....	67

Figure 25. Etape de remémoration	69
Figure 26. Difficulté cible saisie dans le formulaire de description de difficulté	73
Figure 27. Liste de cas retournés par la couche RD.....	74
Figure 28. Processus de résolution de problèmes	75
Figure 29. Structure générique de la solution.....	76
Figure 30. Architecture en couches d'un système sensible au contexte (Coutaz, Crowley et al. 2005).....	85
Figure 31. Modèle générique du contexte	91
Figure 32. Modélisation du contexte dans le système NICOLAT	93
Figure 33. Structuration des couches d'adaptation	96
Figure 34. Cycle de vie d'une interaction sans adaptation.....	97
Figure 35. Cycle de vie d'une interaction adaptée	97
Figure 36. Composants de la couche AI	98
Figure 37. Composants de la couche Interface	99
Figure 38. Processus d'envoi d'un contenu par Email.....	101
Figure 39. Extrait du code permettant de préparer et d'envoyer l'email décrivant une difficulté	102
Figure 40. Processus de publication d'un contenu via un client email	103
Figure 41. Formulaire de description de difficultés MAETIC via l'Email.....	104
Figure 42. Intégration d'une application dans Facebook	105
Figure 43. Processus de récupération d'un contenu via Facebook.....	106
Figure 44. Extrait du code permettant d'envoyer une notification à un utilisateur de l'application Facebook	106
Figure 45. Processus de publication d'un contenu via l'application Facebook	107
Figure 46. Page d'actualités de l'application Facebook.....	108
Figure 47. Processus d'enregistrement d'un dispositif	110
Figure 48. Code permettant de récupérer une notification par un dispositif Android	111
Figure 49. Processus de récupération d'un contenu via un dispositif Android.....	112
Figure 50. Réception d'une notification sur un dispositif Android.....	112
Figure 51. Processus de publication d'un contenu via un dispositif Android.....	113

Figure 52. Quelques interfaces de l'application Android.....	113
Figure 53. Processus d'envoi d'un contenu par SMS	114
Figure 54. Extrait de code permettant d'envoyer des SMS aux membres concernés par cet outil et de tracer l'envoi	115
Figure 55. Processus de récupération d'un commentaire publié par SMS.....	116
Figure 56. Code permettant de recevoir un SMS envoyé par un membre	116
Figure 57. Taux d'usage de chaque outil par les membres de la CoP.....	119
Figure 58. Types de sélections des membres offertes par NICOLAT	127
Figure 59. Les poids des attributs du contexte pour le type de difficulté « Réalisation du produit »	129
Figure 60. Structure de règles.....	131
Figure 61. Règle représentée selon le formalisme défini	132
Figure 62. Processus de fonctionnement du moteur d'exploitation des règles	133
Figure 63. Structure générique de règles.....	133
Figure 64. Règle liée au domaine de développement informatique	134
Figure 65. Les indices de contributions de l'enseignant E.....	138
Figure 66. Extrait d'un fichier de traces.....	142
Figure 67. Variation du gain global pour différents types de difficultés et différentes tailles de la CoP	149
Figure 68. La répartition des spécialités des membres de la communauté	166
Figure 69. La répartition des niveaux de compétences des membres de la communauté.....	167
Figure 70. Nombres de contributions des membres de la communauté par rapport difficultés de type « Panne réseaux »	169

Liste des tableaux

Tab 1. Différences en les CoPs et les autres structures sociales en termes d'objectifs (Wenger, McDermott et al. 2002)	28
Tab 2. Points communs entre les questions de notre problématique de recherche et les barrières identifiées par BSR Stars (Stars 2012).....	31
Tab 3. Base initial de cas.....	71
Tab 4. Indices de similarités textuelles entre les cas sources et le cas cible	73
Tab 5. Calcul de des indices de tri des cas : « CAS-6 ».....	73
Tab 6. Calcul de des indices de tri des cas : « CAS-10 ».....	74
Tab 7. Evaluation des approches de la modélisation de contexte par rapport aux exigences (Strang and Linnhoff-Popien 2004)	89
Tab 8. Résumé des résultats obtenus concernant les préférences des enseignants et la facilité d'usage de chaque outil.	119
Tab 9. Résumé des résultats obtenus concernant les réponses immédiates des membres aux demandes d'aide via chaque outil	120
Tab 10. Contextes des difficultés liées aux contributions du membre E	140
Tab 11. Suivi du processus de sélection.....	146
Tab 12. Résultats de la première expérience	148
Tab 13. Profils pédagogique des membres de la communauté	166
Tab 14. Base des contextes utilisés pour générer les traces	168
Tab 15. Traces et indices de contributions des membres par rapport aux difficultés de type « pannes réseau ».....	173
Tab 16. Variation du gain filtrage pour différents types de difficultés et différentes tailles de la CoP (+F-T / -F-T).....	174
Tab 17. Variation du gain global pour différents types de difficultés et différentes tailles de la CoP (+F+T / -F-T).....	174
Tab 18. Variation du gain tri pour différents types de difficultés et différentes tailles de la CoP (+F+T / -F-T).....	174

Chapitre 1 : Contexte des recherches

La première section de ce chapitre introduit le contexte de nos travaux de recherche. La section 2 expose notre problématique. La section 3 aborde les objectifs de nos travaux. Enfin, la dernière section présente la structure de ce manuscrit.

1. Domaine

Nos travaux de recherche se situent dans le domaine des EIAH (Environnements Informatiques pour l'Apprentissage Humain) et se placent dans le contexte d'un apprentissage de savoir-faire par l'intermédiaire d'un Système Informatique Communautaire (SIC). Ces systèmes permettent d'assister l'activité d'une communauté d'acteurs qui partagent leur pratique et qui ont un objectif d'apprentissage en commun. Le terme SIC a été choisi en référence aux travaux menés sur les Communautés de Pratiques (CoP) et sur les Communautés d'Apprentissage ([Charlier, Boukottaya et al. 2007](#)) ([Garrot 2008](#)) ([Dillenbourg, Poirier et al. 2003](#)) ([Wenger, McDermott et al. 2002](#)) ([Vaast 2002](#)).

Au sein d'un tel système, l'apprentissage se fait via les interactions entre ses différents acteurs. Ainsi, nos questions de recherche portent sur la manière avec laquelle ce système peut orchestrer, médiatiser et/ou faciliter les interactions entre ses différents acteurs.

La Communauté de Pratique cible est celle des enseignants utilisateurs de la méthode pédagogique MAETIC (Méthode pédAgogique instrumEntée par les Technologies de l'Information et de la Communication) ([Lecllet and Talon 2008](#)). Cette méthode vise l'acquisition, par des étudiants, de savoir-faire professionnel selon une approche fondée sur une pédagogie de groupe par projet. MAETIC se décompose en cinq étapes (issues du cycle de la conduite de projet) que les étudiants doivent suivre. Chacune de ces étapes produit un ou plusieurs livrables et se déroule en une ou plusieurs séances de TD.

Plus précisément, nos travaux se placent au sein des recherches sur les systèmes adaptatifs supports de l'apprentissage collaboratif ([Magnisalis, Demetriadis et al. 2011](#)). En effet, les membres d'une Communauté de Pratique peuvent avoir des profils, des préférences et des contextes de mise en place de leurs pratiques, très variés. Ces variétés exigent que le système support de cette Communauté de Pratique propose des fonctionnalités adaptatives pour répondre aux besoins spécifiques de chaque membre.

2. Questions de recherche

Dans le but de définir les fonctions de notre système support de la CoP MAETIC, nous nous sommes basés sur un ensemble de constats qui nous ont permis d'identifier une architecture fonctionnelle pour ce système.

Au cours d'expérimentations menées en 2010 auprès des enseignants utilisant la méthode MAETIC, nous avons constaté que lorsqu'un enseignant déploie cette méthode dans sa classe, il peut rencontrer des difficultés dues au manque d'expérience dans l'usage de celle-ci. Il peut alors avoir besoin d'aide. Ces expérimentations ont également souligné que des enseignants plus expérimentés pouvaient, à ce moment-là, lui apporter cette aide. En effet, les enseignants usagers de MAETIC ont besoin d'assistance tout au long de la phase de déploiement de leurs dispositifs, et cette assistance peut être apportée par des enseignants plus expérimentés. Ces enseignants constituent une CoP qui s'intéresse à la résolution des difficultés MAETIC rencontrées par ses membres et, au partage de pratiques liées à cette méthode. Nous appelons cette CoP : la CoP MAETIC.

Dans ce contexte, nous ambitionnons le développement d'un système informatique support de la CoP MAETIC à travers un SIC, afin de permettre aux membres d'échanger sur leurs pratiques et de formuler des demandes d'aide à la résolution de difficultés. Ce système peut être considéré comme un système d'apprentissage qui se base sur la résolution communautaire de problèmes. Ainsi, les problèmes traités dans ce cadre sont des difficultés MAETIC. L'apprentissage, à travers ce système, est principalement un accroissement de savoir-faire à travers les échanges et les interactions entre les membres de la CoP visant à résoudre les problèmes rencontrés.

Cet apprentissage étant communautaire, il est important que la communauté puisse vivre et soit active. Cependant, l'activité de la communauté peut être freinée par la démotivation de ses membres aboutissant à l'abandon de l'utilisation du système. Dans ce cadre, nous soulevons quatre facteurs de démotivation des membres d'une CoP que nous souhaitons limiter à travers nos travaux :

1. Les membres de la communauté pourraient être sollicités trop régulièrement pour des résolutions de problèmes similaires.
2. La difficulté de prise en main de nouveaux outils informatiques par les membres. En particulier, si l'interface de notre SIC n'est pas proche des interfaces des outils communautaires et sociaux les plus utilisés actuellement, la démotivation d'une partie des membres utilisant le système peut voir le jour.
3. Le temps de réponse de la communauté aux messages postés par ses membres peut être un facteur de démotivation lorsqu'il est assez long.

4. L'obtention, par un membre, de propositions de solutions qui sont inutiles pour le contexte de son problème.

Ainsi, nous décomposons notre problématique en cinq questions :

- Question 1 : Comment réduire les demandes de résolution de problèmes qui ont été résolus dans le passé ? Poster des demandes de résolution de problèmes qui ont été résolus dans le passé peut être un facteur de démotivation des membres et aboutir à l'inondation du système par des problèmes répétitifs.
- Question 2 : Comment dépasser les problèmes de prise en main et de familiarisation avec les nouveaux outils informatiques ? Les membres de certaines CoPs peuvent ne pas être habitués avec les technologies et peuvent donc trouver des problèmes pour se familiariser avec les nouveaux outils informatiques et les manipuler facilement. La faiblesse en matière de manipulation des technologies pourraient influencer les interactions entre les membres lors de la résolution communautaire d'un problème.
- Question 3 : Comment assurer l'accès au système en cas de mobilité ? Dans une ère où l'utilisation des dispositifs mobiles est en augmentation exponentielle, la proposition d'outils qui assurent la continuation des interactions en cas de mobilité des membres est devenue essentielle.
- Question 4 : Comment minimiser le temps de réponse ? Dans un système qui supporte la résolution communautaire de problèmes, un membre intéressé par la résolution de son problème doit le publier et attendre que les autres membres consultent le système. Ce qui peut engendrer un grand délai avant qu'il reçoive la réponse.
- Question 5 : Comment assurer l'obtention, par le membre cherchant à résoudre son problème, d'une solution qui soit la plus adaptée à son contexte à travers ses interactions avec les autres membres ? Les profils des membres d'une CoP peuvent être hétérogènes, la solution d'un problème donné peut changer selon le contexte dans lequel il a été rencontré. Ainsi, l'envoi d'une demande d'aide à la résolution d'un problème à des membres, dont le champ d'expertise et d'intérêt ne convient pas, pourrait donner lieu à des solutions inefficaces, à l'augmentation du temps de résolution du problème et à l'inondation des fils de discussion par des commentaires inutiles.

Nous cherchons par rapport à ces questionnements à obtenir des solutions génériques qui puissent être appliquées à la CoP MAETIC mais également à d'autres communautés.

3. Objectifs des recherches

Pour répondre à la question 1, concernant la réduction des demandes de résolution de problèmes qui ont été résolus dans le passé, nous visons offrir au membre de la CoP un outil d'aide à la résolution de problèmes permettant d'exploiter les anciennes expériences de résolutions de problèmes passés. Pour répondre à la question 2, concernant les problèmes de familiarisation avec les nouveaux outils informatiques, nous proposons d'offrir l'accès à notre système à partir de différents types d'outils d'interaction existants selon les préférences des membres. En particulier, ceux qu'ils utilisent au quotidien.

Pour répondre à la question 3, concernant l'accès dans les cas de mobilité, nous développons des outils d'accès au système qui soient adaptés aux terminaux mobiles des membres. En particulier, nous offrons l'accès mobile à travers le web et à travers les services de la téléphonie mobile.

Pour répondre à la question 4 concernant la minimisation de temps de réponse, nous offrons la possibilité de solliciter chaque membre par son outil préféré qu'il a utilisé au moment de la publication d'un problème.

Pour répondre à la question 5 concernant l'obtention d'une solution adaptée au contexte du problème, nous proposons une approche pour la sélection des membres. Cette approche permet de ne solliciter, lors de la publication d'un problème, que les membres ayant des profils adaptés par rapport au problème à résoudre et au profil de son émetteur, pour participer à sa résolution.

Le système support de la CoP MAETIC et l'ensemble des outils associés aux solutions proposées pour répondre aux questions de la problématique forment ce que nous appelons le système NICOLAT (système iNformatIque COmmunautaire mobiLe et AdapTatif).

Ainsi, pour chacune des solutions proposées, nous présentons une approche générique qui permet de s'adapter à d'autres communautés.

4. Organisation du manuscrit

Ce manuscrit de thèse se compose de 5 chapitres qui se présentent comme suit :

- Dans le chapitre 1, nous avons commencé par une présentation du contexte des recherches. Nous avons, alors, exposé notre problématique de recherche et ses questions ainsi que les objectifs de notre travail par rapport à ces questions.
- Dans le chapitre 2, nous abordons le système communautaire avec lequel nous supportons la CoP des enseignants usagers de la méthode MAETIC. Ainsi, nous commençons par présenter un état de l'art sur les CoP. Puis, nous présentons la méthode MAETIC et la CoP-MAETIC. Nous introduisons, ensuite, la structure du

système NICOLAT et nous détaillons son noyau communautaire. Enfin, nous abordons la généralité du système NICOLAT et de son noyau communautaire.

- Dans le chapitre 3, nous présentons la couche « Résolution de Problèmes » (RP) du système NICOLAT qui a pour objectif d'aider un membre de la CoP à exploiter les problèmes résolus dans le passé. Ainsi, nous commençons par présenter un état de l'art sur le Raisonnement à Partir de Cas (RàPC) sur lequel nous nous sommes basés pour exploiter les expériences de résolutions passées. Ensuite, nous détaillons la conception de la couche RP et son processus de remémoration. Après, nous abordons la validation expérimentale de cette couche. La dernière partie de ce chapitre traite sa généralité.
- Dans le chapitre 4, nous précisons notre approche visant à supporter les échanges/interactions entre les membres de la CoP via leurs outils d'interaction préférés et utilisés quotidiennement. Nous commençons tout d'abord par présenter un état de l'art sur l'adaptation dans les EIAH. Des notions sur la gestion du contexte sont ensuite introduites, car les notions d'adaptation et de sensibilité au contexte, sont étroitement liées dans les systèmes informatiques. Puis, nous présentons notre modélisation du contexte. Après, nous détaillons les deux couches d'adaptation : la couche « Adaptation des interactions » et la couche « Interface » et présentons le fonctionnement de notre système par rapport aux outils que nous supportons. Nous présentons ensuite une expérimentation en milieu écologique ayant pour but d'évaluer notre approche. Enfin, nous discutons la généralité de l'approche.
- Dans le chapitre 5, nous présentons notre approche de sélection des membres qui peuvent contribuer positivement à résoudre un problème. Nous débutons tout d'abord par la présentation des travaux existants dans le cadre de la formation de groupe. Cette présentation permet de situer notre approche de sélection des membres, que nous considérons comme un problème de formation de groupes d'apprenants autour d'un problème posté. Puis, nous détaillons notre approche de sélection des membres avec ses deux composantes statique et dynamique. La généralité de chacune de ces composantes est discutée. Enfin, nous terminons par un exposé des résultats de l'expérimentation par simulation que nous avons menée pour valider notre approche.

Chapitre 2 : Le système NICOLAT et sa Communauté de Pratique

Dans ce chapitre, nous présentons, dans un premier temps, les Communautés de Pratique. Puis, nous introduisons la méthode pédagogique MAETIC et la CoP des enseignants usagers de cette méthode. Après, nous abordons la structure du système NICOLAT. Nous détaillons, ensuite, le noyau communautaire de NICOLAT, support des interactions entre les membres de la CoP MAETIC. Enfin, nous concluons par la généralité du système NICOLAT et de son noyau communautaire.

1. Communautés de Pratique : Etat de l'art

1.1 Définitions et spécificités

Les Communautés de Pratique (CoPs) sont dérivées des travaux de Lave et de Wenger ([Lave and Wenger 1991](#)). Wenger ([Wenger 1998](#)) a défini les CoPs comme "*des groupes de personnes qui partagent un intérêt commun ou une passion qu'ils pratiquent et apprennent à la faire d'une meilleure façon en s'interagissant régulièrement*". C'est une forme de groupement de professionnels qui se rassemblent afin de partager et d'échanger des connaissances explicites et tacites liées à leur domaine d'intervention.

Wenger ([Wenger 2005](#)) affirme que toutes les Communautés de Pratique ont trois caractéristiques qui les distinguent : le domaine, la communauté et la pratique.

- Le domaine ou l'entreprise commune : Les membres de la CoP ont un domaine d'intérêt partagé qui les distingue des autres personnes.
- La communauté ou l'engagement mutuel : Les membres de la CoP doivent s'entraider et s'engager dans des activités, des discussions, etc.
- La pratique ou le répertoire partagé : C'est un répertoire partagé de ressources qui représente la mémoire de la communauté (expériences, outils, histoires, techniques pour la résolution des problèmes, etc.).

Il est important de détailler, ces trois caractéristiques pour pouvoir détecter les spécificités des CoP et, de préciser ce qui les distingue des autres structures communautaires. En effet, avoir un domaine est essentiel pour une CoP, sinon elle ne constitue seulement qu'un groupe d'amis ([Wenger 2005](#)). Le domaine détermine également l'identité d'une CoP puisque tous les membres sont engagés dans un domaine d'intérêt partagé. Il crée le sens de responsabilité et de participation dans la communauté, il définit ce que la communauté est, et attire les nouveaux arrivants. Ainsi, le domaine crée l'identité commune utilisée pour interagir avec le

monde extérieur. Le domaine n'est pas statique, ni immuable, il évolue avec la CoP et reflète toujours l'identité de ses membres. Wenger ([Wenger 2005](#)) a indiqué que c'est le domaine qui oriente l'apprentissage dans une CoP. Nous pouvons dire qu'il est le cadre général de la CoP, qui se détermine en fonction de l'intérêt commun, et détermine les thématiques de discussions et les sujets principaux d'apprentissage. Oguz ([Oguz 2007](#)) indique que le domaine commun ne veut pas dire que les membres ont les mêmes arrière-plans, compétences et perspectives. En effet, une certaine homogénéité peut accélérer la construction de la CoP au début mais elle n'est pas une composante essentielle pour son fonctionnement.

La « communauté » est également importante pour l'apprentissage qui se produit au sein de la CoP. Le terme 'communauté' signifie plutôt la 'communauté sociale'. Pour développer les compétences liées à leur domaine d'intérêt, les membres d'une CoP doivent s'engager dans des activités communes, des discussions, s'entraider, échanger des informations et construire des relations qui les aident à apprendre l'un de l'autre. En fait, nous ne pouvons pas parler d'une CoP sans parler d'interaction entre ses membres dans le but d'un apprentissage et d'un développement personnel dans le domaine d'intérêt commun. Peers ([Peers 2007](#)) indique à ce sujet que la communauté signifie la structure sociale de l'apprentissage, qui comprend les relations entre les membres et la nature de leurs interactions, y compris les niveaux de confiance, d'appartenance, et de réciprocité.

Enfin, une CoP n'est pas seulement un ensemble d'individus rassemblés par un intérêt commun, ce sont des praticiens qui échangent leurs expériences, leurs histoires, leurs outils, leurs moyens de résolutions de problèmes. En fait, ils partagent la pratique. Wenger ([Wenger 2005](#)) précise que la pratique est "*un ensemble de plans, outils, connaissances et documents que la communauté développe et partage*". Ainsi, la pratique change d'une communauté à une autre. Nous pouvons conclure que la pratique correspond à la production des activités effectuées par les membres de la CoP.

1.2 CoP comme structure sociale

L'idée de CoP n'est pas nouvelle. Elle se considère parmi les premières structures sociales de connaissances dans l'histoire de l'humanité. En effet, nous pouvons appartenir à une CoP sans être conscients. C'est le cas pour la plupart d'entre nous. L'appartenance à une CoP n'est pas un but, c'est un moyen pour atteindre un but d'apprentissage. Selon Wenger ([Wenger 2005](#)), l'apprentissage au sein d'une CoP est un apprentissage social. Ce dernier est assuré par l'équilibre entre deux processus : la participation et la réification.

- La participation concerne l'engagement des membres dans les activités de la CoP, le partage, la communication, la confrontation des idées et les échanges amenant à la création de connaissances.

- La réification concerne la formalisation des connaissances construites par la CoP dans des artefacts concrets : texte ou autres types de productions.

La CoP tolère 3 niveaux de participation ([Wenger, McDermott et al. 2002](#)). Comme le souligne la figure 1, ci-dessous, nous retrouvons :

- Le « noyau dur » d'individus qui participent très activement aux activités de la communauté et qui, pour certains, en assurent l'animation. D'après Wenger et al ([Wenger, McDermott et al. 2002](#)), ce groupe est généralement petit et, ne représente souvent, que 10 à 15% d'une CoP prise dans son ensemble.
- Les « membres actifs » qui participent occasionnellement aux activités collectives mais moins régulièrement et intensément que les membres du noyau dur. Ce groupe n'est pas très important non plus et représente en général 15 à 20% de l'effectif de la CoP.
- Les « membres périphériques » qui observent le plus souvent les interactions entre le noyau et les membres actifs. Ce mode de participation qui peut sembler opportuniste est au contraire essentiel à la vie de la communauté et, ne doit pas être découragé d'autant que cet accès sans participation active n'empêche pas ces membres périphériques d'utiliser les connaissances ainsi acquises dans le reste de l'organisation.

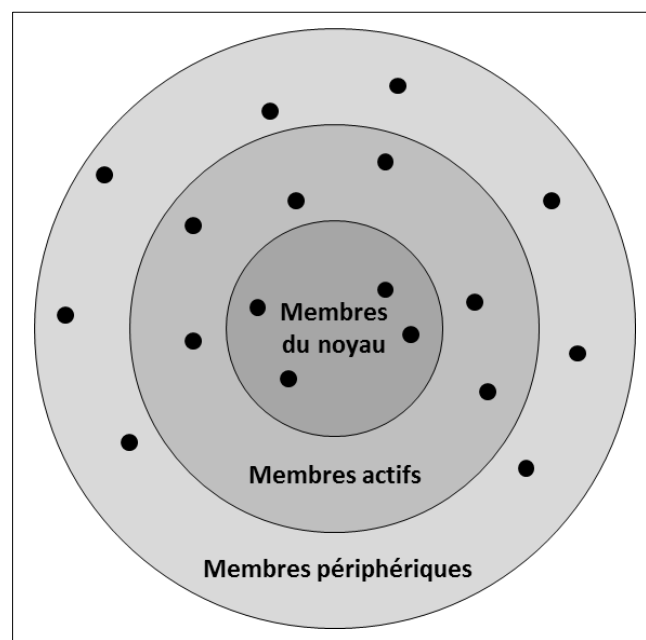


Figure 1. Niveaux de participation dans une CoP ([Wenger, McDermott et al. 2002](#))

Les membres d'une communauté se déplacent au cours du temps entre ces différents niveaux. Certains membres du noyau dur peuvent s'éloigner quand le thème de la CoP change ou évolue. Des membres actifs peuvent s'engager très intensément pour une courte période puis se désengager de nouveau.

Cependant, une CoP n'est pas uniquement une structure de partage, de collaboration et de construction de connaissances. Elle représente également une structure de développement identitaire pour ses membres ([Wenger 2005](#)). La négociation et le changement de pratiques peuvent être vues comme le commencement d'un parcours de transformation identitaire ([Charlier, Nizet et al. 2005](#)). Selon Wenger ([Wenger 2005](#)), au sein d'une CoP, l'apprentissage n'est pas uniquement un processus de développement de connaissances et de compétences, il peut être vu comme parcours identitaire au cours duquel un membre développe également sa carrière, ses projets personnels et son appartenance à divers groupes professionnels. Ainsi, les diverses trajectoires individuelles des membres se croisent et s'influencent pour les « membres actifs » et les « membres périphériques ». Ils négocient et développent ensemble, une identité commune, ce qui les amène à négocier et développer leur identité individuelle et à se positionner collectivement vis-à-vis du monde extérieur.

Une CoP se construit, se développe et possède alors un cycle de vie qui se compose d'un ensemble de stades.

1.3 Cycle de vie d'une CoP

La CoP évolue selon un cycle de vie allant de sa naissance jusqu'à sa mort ou sa transformation radicale. La connaissance de l'état dans lequel se situe la CoP, facilite la planification des activités qui peuvent aider à son développement. Wenger et al. ([Wenger, McDermott et al. 2002](#)) ont défini le cycle de vie d'une CoP en 5 stades (figure 2) : Potentiel, Unification, Maturité, Momentum et Transformation. C'est Langelier qui a introduit la figure 2, ci-dessous, représentant le cycle de vie d'une CoP selon les 5 stades de Wenger et al.

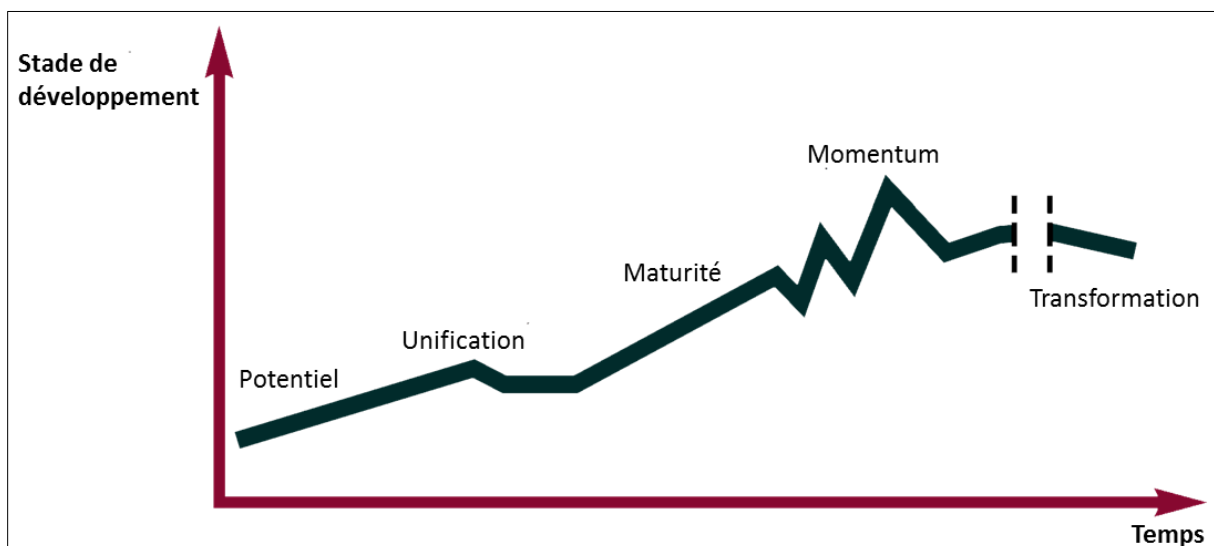


Figure 2. Le cycle de vie d'une CoP ([Langelier 2005](#))

- Le potentiel : c'est l'état embryonnaire de la CoP. La communauté correspond à un petit groupe d'individus ayant un thème partagé. Au sein de ce groupe, un domaine

partagé commence à se développer. A ce stade, la communauté a besoin d'un animateur ou d'un coordinateur, qui joue un rôle très important dans le succès de la CoP. Les activités de ce stade tournent autour la définition de l'objet de la communauté.

- L'unification : ce stade consiste à une augmentation du nombre des membres et des connaissances partagées. La communauté est fragile et doit être soutenue. Il est alors important de proposer des activités permettant de construire des relations entre les membres, de les sensibiliser et d'établir un climat de confiance.
- Maturité : à ce stade, la communauté alterne entre des cycles d'activités, qui sont à la fois, intenses et faibles. Il faut clarifier l'objet, le rôle et les frontières de la communauté. Cette dernière passe, à ce stade, du simple partage d'idées au développement de nouvelles connaissances.
- Momentum : à ce stade, la communauté doit maintenir sa vitalité. En effet, par rapport aux changements qui s'opèrent au sein de la CoP, les membres doivent maintenir leurs rythmes en diversifiant leurs actions, en recherchant des ressources, et en développant la coordination.
- Transformation : ce stade se caractérise par une baisse d'activités de la CoP. Cette baisse peut être due au fait que les problèmes sont résolus, les pratiques sont précisées ou les membres ont développé d'autres centres d'intérêt. Toutes communautés, même les plus actives, arrivent à ce stade. La CoP peut alors mourir ou se diviser en d'autres communautés, se fusionner avec d'autres, s'institutionnaliser, ou même se transformer à un club social.

L'explosion croissante des connaissances, la révolution technologique et le changement accéléré des différents aspects de la vie, imposent aux organisations, aux groupes et aux individus, de travailler d'une façon collaborative avec des méthodes nouvelles et innovantes. Ainsi, il est nécessaire de trouver des mécanismes de collaboration entre les individus au sein de différentes organisations. Dans le paragraphe suivant, nous voyons l'intérêt des CoPs.

1.4 Intérêt des CoPs

Les CoPs peuvent représenter un nouveau modèle pour la communication entre les individus, pour le partage de connaissances et d'expériences, ainsi que pour la collaboration et le développement individuel et organisationnel. Pourtant, Wenger et al. ([Wenger, McDermott et al. 2002](#)) indiquent qu'une CoP peut apporter des bénéfices tant pour l'organisation que pour

ses membres. Dans ce qui suit nous présentons ces bénéfiques, en se basant sur les travaux de Hautdidier ([Stenuit Hautdidier 2006](#))

1.4.1 Intérêts pour les organisations

Une CoP permet à une organisation de **gagner en productivité**. En effet, elle permet de pallier le manque de relations ou d'échanges entre les collaborateurs de la même organisation, en offrant des possibilités d'échanges. Elle diminue, ainsi, le risque de perdre de temps, pour développer des solutions identiques, tout en permettant aux collaborateurs de différents groupes de travail de se réunir autour d'un centre d'intérêt commun. La performance de l'organisation peut être améliorée grâce à la définition de nouveaux standards et de nouvelles procédures naissant à partir des pratiques échangées et, des confrontations de points de vue entre les collaborateurs.

Une CoP permet également de **favoriser l'innovation** au sein d'une organisation. La CoP peut être le terrain d'expérimentation de nouvelles pratiques qui peuvent être déployées plus largement en cas de valorisation par le groupe. De plus, elle facilite l'activité de la veille du domaine et permet d'avoir une bonne vision du marché et des nouveaux produits. Elle offre, également, des possibilités pour initier des projets transversaux.

Enfin, une CoP facilite **la capitalisation de connaissances et de pratiques**. Elle favorise la culture de partage, celle de la formalisation de connaissances tacites en connaissances explicites et permet la construction d'une base de connaissances partagée.

1.4.2 Intérêts pour les membres

Une des principales raisons de participer à une CoP est **l'apprentissage**. La participation à une CoP permet à un membre d'apprendre dans son contexte de travail. Le fait d'échanger avec des collègues sur des problèmes rencontrés en situation de travail permet d'apprendre des pratiques, de nouvelles techniques, ou bien encore de clarifier un concept. Elle favorise la construction d'une mémoire partagée sur un domaine et facilite la transmission de connaissances aux nouveaux arrivants.

L'appartenance à une CoP représente pour un individu un **support pour la résolution à ses problèmes et la prise de décision**. En effet, elle facilite la recherche d'interlocuteurs partageant les mêmes centres d'intérêt et l'obtention de réponses qualifiées. La participation à une CoP permet alors, de **développer de nouvelles attitudes ou d'en améliorer d'autres**. Un membre peut apprendre à exprimer et à défendre ses idées, apprendre à confronter des points de vue, et finalement apprendre à collaborer. Dans la section suivante, nous présentons quelques exemples de réussite de CoPs.

1.5 Exemples de CoPs : Success Story

Cette partie présente trois exemples de « success story ». Le premier exemple, concerne la très célèbre agence gouvernementale « National Aeronautics and Space Administration », plus connue sous son abréviation NASA, et qui a en charge la majeure partie du programme spatial civil des Etats-Unis ([Topousis, Dennehy et al. 2012](#)). Historiquement, les ingénieurs de la NASA avaient peu d'occasions ou d'incitations pour partager leur expertise au sein de l'agence. Souvent leur connaissance ne dépassait pas les frontières géographiques et organisationnelles. Suite aux missions qui devenaient de plus en plus complexes et l'arrivée de nouveaux ingénieurs, la nécessité de développer une culture de partage est devenue essentielle. Pour répondre à ce besoin, le bureau d'ingénieurs de la NASA a établi une CoP en se servant du réseau d'ingénieurs de la NASA. Ce dernier est une suite d'outils de recherche d'information et de partage de connaissances visant spécifiquement à faciliter la communication entre les personnels de la main-d'œuvre technique. Les CoPs établies concernent notamment l'avionique, la mécanique du vol, le soutien de la vie, la propulsion, la navigation. Le développement de ces CoPs avait pour principal objectif de favoriser le partage de connaissances. Au début, ces CoPs se sont concentrées sur la collection des informations telles les spécifications, les listes de contact, les offres de formations. Avec l'augmentation des membres des CoPs, elles se sont orientées vers la suggestion de contenus, la recherche des membres pouvant aider à résoudre un problème urgent et le partage de connaissances. Les informations clés ont été collectées et les ingénieurs ont pu échanger et travailler ensemble afin d'améliorer les différentes disciplines d'ingénierie de la NASA. Les nouveaux ingénieurs ont eu accès aux experts disponibles pour répondre à leurs questions. Ces CoPs continuent à se développer et elles sont intrinsèques à la NASA.

Le deuxième exemple, concerne l'entreprise suédoise Ericsson, entreprise de télécommunications fondée en 1876 par Lars Magnus Ericsson, et se rapporte aux trois sites délocalisés en Finlande, en Hongrie et aux Etats Unis ([Paasivaara and Lassenius 2014](#)). Ces sites comporte environ 400 employés répartis sur 40 équipes, qui travaillent à la production de systèmes de gestion de trafic des réseaux de télécommunications. En 2010, suite à de nombreux problèmes dans les délais de réalisation de ces systèmes (gestion, de cout/qualité, de motivation des personnels), la direction a souhaité introduire une nouvelle méthode de conduite de projets, la méthode agile Scrum. De plus, pour mener à bien cette conduite du changement, la société a développé plus de 20 CoPs dédiées au partage de connaissances et à l'apprentissage, à la coordination, au travail technique et au développement organisationnel. Parmi les plus grandes réussites de ces CoP, nous retrouvons : une CoP de coaching (qui a pour objectif de partager les expériences et de pratiques liées à Scrum) ; une CoP de coordination (qui vise la coordination entre les équipes) ; une CoP de bout-en-bout (qui assure

le suivi des progrès du travail) et enfin une CoP de développeurs (qui vise à partager les bonnes pratiques de développement).

Enfin, le troisième et dernier exemple, concerne l'entreprise DRV qui se situe aux pays bas et dédiée à l'ingénierie et au conseil dans l'architecture et la construction ([de Vrij and Helms 2006](#)). Cette entreprise compte environ 3900 employés dans le monde. En 2005, elle a délocalisé ses bureaux d'études de son principal bureau situé aux pays bas à un nouveau bureau à New Delhi (Inde). Dans un cadre de cette délocalisation, des problèmes de gestion de projet sont apparus notamment lors des dernières phases : réalisation du produit et intégration. Les employés distants n'arrivaient pas à maintenir une réalisation de qualité dans les délais impartis. L'entreprise DRV a alors développé une CoP afin de faciliter et de favoriser le transfert des connaissances. Cette CoP est supportée par un portail web. L'étude de cette CoP a montré qu'elle peut améliorer le processus de réalisation dans un contexte de délocalisation ([de Vrij and Helms 2006](#)).

Nous avons donc vu, qu'il existait différents types de structures sociales qui peuvent se confondre avec les CoP. En effet, nous ne pouvons pas appeler n'importe quelle structure sociale une CoP. Dans ce qui suit, nous expliquons la différence entre les CoPs et les autres structures sociales notamment, **l'équipe projet**, **l'unité fonctionnelle** ou bien encore les **réseaux informels**. Cette catégorisation se base sur les travaux de Wenger ([Wenger, McDermott et al. 2002](#)).

1.6 Différences entre les CoPs et autres structures sociales

Le cœur d'une **équipe de projet** correspond aux tâches interdépendantes qui contribuent à atteindre un objectif partagé et prédéfini. Une équipe possède souvent un leader qui veille à la concentration des membres du groupe, sur la production des livrables, et qui coordonne les contributions individuelles de ses membres, pour atteindre des objectifs globaux. En revanche, le cœur d'une CoP correspond à l'investissement des membres dans leur domaine. Le domaine est différent d'une tâche. Ainsi, les membres d'une CoP sont connectés par des connaissances interdépendantes et non pas par des sous-tâches interdépendantes. Le leader d'une CoP ne la dirige pas au sens traditionnel, mais rassemble les membres et permet à la communauté de trouver sa direction. La CoP peut s'engager dans des tâches spécifiques et des projets, au cours du développement de sa pratique mais elle n'est pas définie par un de ses projets. Elle est définie par son engagement, fondamental, d'explorer son domaine, de développer et de partager des connaissances pertinentes.

L'unité fonctionnelle d'une entreprise a la responsabilité de gérer des objectifs comme par exemple, l'accomplissement d'une fonction administrative ou bien encore la construction d'un produit. Cette responsabilité peut concerner l'allocation de ressources, la gestion des processus de l'entreprise, l'attribution de rôles formels ou l'évaluation des résultats de

l'entreprise. Nous pouvons prendre comme exemples le service des « Ressources Humaines » ou le service « comptabilité » d'une organisation. Comme dans une CoP, les membres d'une unité fonctionnelle ont souvent le même domaine. Par contre, les CoPs ont pour principal objectif de développer la connaissance et non pas de livrer un produit au marché. Les CoPs sont faiblement connectées, informelles et autogérées par rapport aux unités fonctionnelles. En effet, ces dernières sont fortement institutionnalisées. Les CoPs sont également basées sur la collégialité et non pas sur les liens hiérarchiques. L'appartenance à une CoP dépend de la participation plutôt que sur les affiliations institutionnelles. La reconnaissance d'un membre de la CoP dépend de ses contributions et non pas d'une quelconque autorité formelle.

Toute organisation peut avoir des **réseaux informels** de personnes qui communiquent, partagent l'information et construisent des relations et des réputations. Une CoP est différente de ce type de réseaux dans le sens où elle s'intéresse à « quelque chose ». Elle n'est pas qu'un ensemble de relations. Son domaine lui permet d'avoir une identité. Ainsi, un intérêt partagé tout seul ne produit pas nécessairement une CoP. La CoP comporte aussi le développement d'une pratique partagée qui affecte directement les comportements et les compétences des membres.

Savoir si un groupe constitue ou non une CoP, n'est pas quelque chose qui peut être déterminée à partir de son nom ou des caractéristiques de ses membres. Il faut regarder comment le groupe fonctionne et comment il combine les trois éléments de la CoP : domaine, communauté et pratique.

Le tableau 1 ci-dessous résume la différence entre les CoPs et les autres communautés selon les finalités, les participants, l'engagement et la durée. Cette classification est issue de Wenger et al. ([Wenger, McDermott et al. 2002](#)).

	Communauté de pratique	Unité fonctionnelle	Équipe de projet	Réseau informel
Finalité	Développer les aptitudes des membres; Créer, étendre et échanger la connaissance	Réaliser un produit ou un service	Accomplir une tâche définie	Collecter et faire circuler l'information
Participants	Les participants qui le souhaitent Auto-sélection basée sur l'expertise ou la passion par quelque chose	Toute personne dépendant du chef du groupe	Salariés ayant un rôle direct dans l'accomplissement du projet.	Les amis et connaissances d'affaires
Engagement	La passion, l'implication et	Les besoins du travail, la définition	Les exigences du travail, les	Les besoins

	l'identification à l'expertise du groupe	des tâches et les objectifs communs	différentes étapes du projet et leurs objectifs	mutuels
Durée	Aussi longtemps qu'il existe un intérêt à maintenir le groupe	Jusqu'à la réorganisation suivante	Jusqu'à l'achèvement du projet	Aussi longtemps que les membres ont des raisons de rester en contact

Tab 1. Différences en les CoPs et les autres structures sociales en termes d'objectifs ([Wenger, McDermott et al. 2002](#))

Historiquement, les Communautés de Pratique étaient délimitées par la géographie. Sans des formes de communication efficaces, les membres du groupe de différents emplacements ne bénéficiaient pas de façon optimale de la dispersion de la connaissance ([OEAS 2012](#)). La communication se faisait grâce à l'interaction en personne ou, alternativement, la conversation. Ces formes de communication limitaient la portée des CoPs à l'environnement entourant ([OEAS 2012](#)). Dans la section suivante nous traitons les CoPs virtuelles qui permettent de dépasser les limitations géographiques en exploitant les TIC.

1.7 Communautés de Pratiques virtuelles

Le concept de CoPs virtuelles n'est différent du concept de CoPs classiques. Ce sont les outils de communication qui les diffèrent. Pour les CoPs virtuelles, les technologies s'utilisent pour la communication et l'interaction sociale au lieu des interactions directes, en face à face. La CoP s'appelle virtuelle lorsque ses membres utilisent les TIC comme principal mode d'interaction ([Dubé and Bourhis 2003](#)). Ainsi, la virtualité n'exclut pas l'utilisation de réunions face-à-face, mais plusieurs facteurs, y compris la dispersion géographique et les horaires chargés rendent la communication à travers les TICs beaucoup plus efficace ([Dubé and Bourhis 2003](#)). En effet, les TICs "*permettent de transcender les barrières du temps et de l'espace, et de rassembler des individus en fonction de leurs intérêts, et non pas de leurs localisations géographiques*" ([Bourhis, Dubé et al. 2004](#)). Elles leur offrent de nouveaux moyens d'interaction à distance.

Lilleorg et al. ([Lilleorg, Tammets et al. 2014](#)) ([Stars 2012](#)) ont effectué une étude concernant les outils technologiques utilisés pour supporter les CoPs. Les outils identifiés sont classés en deux catégories : les outils traditionnels et les outils sociaux. Les outils traditionnels sont : l'email, l'intranet, les espaces de gestion de projets (Alfresco, Projectplace, BaseCamp, Trello), les outils de conférence en ligne (WebEx, Skype, GoToMeeting) et les newsletters. Les outils sociaux sont : Facebook, Twitter, LinkedIn, les outils de création collaborative de documents, les wikis et les weblogs. Dans ce qui suit, nous présentons les formes d'usage de ces outils par les CoPs en se basant sur les travaux de BSR Stars ([Stars 2012](#)).

L'email : L'email est l'outil le plus simple à utiliser pour partager les informations, les connaissances et les ressources entre les membres de la communauté ou avec d'autres contacts. Les listes de diffusion sont utilisées pour envoyer les emails directement pour tous les membres d'une communauté. L'inconvénient de cet outil réside dans le fait que les membres n'ont pas assez de temps pour se concentrer sur les emails. Les emails d'information sont ignorés dans la plupart des cas. Les gens n'ont pas assez de temps pour les lire et pour y répondre.

L'intranet est un système qui permet aux membres d'une communauté de communiquer entre eux. Différentes solutions peuvent être utilisées pour développer un intranet. Souvent, le site externe de la communauté est combiné avec l'intranet. Dans ce cas, la communauté utilise un CMS. Dans un tel système, il y a des pages web publiques dédiées à la communication externe et un espace privé dédié aux membres de la communauté.

Les espaces de gestion de projets permettent de gérer les projets et leurs différents processus. Ils peuvent être utilisés pour la création et le stockage de la documentation du projet, des rapports et des présentations. L'inconvénient de ces solutions est qu'elles ne supportent pas la communication. Elles ont besoin d'être combinées avec des outils de communication.

Les outils de conférence en ligne supportent la communication, le partage de connaissances et de l'information instantanés entre les membres de la CoP ce qui permet d'augmenter le sentiment de la communauté. Un des membres met en place la réunion et partage le lien avec les autres. Les membres intéressés rejoignent la réunion via leurs propres dispositifs. Ces outils offrent la possibilité de communiquer aussi souvent que nécessaire et permettent de construire le sens de la communauté. Ainsi, les membres qui n'ont pas la chance de participer peuvent réécouter la discussion plus tard. Cependant, l'inconvénient de l'usage de ces outils est lié aux barrières personnelles plutôt qu'aux outils eux-mêmes. En fait, certains membres peuvent être des participants passifs à la réunion.

Les newsletters sont publications distribuées régulièrement. Généralement, elles concernent un thème principal qui intéresse les abonnés. Les communautés qui visent à partager certaines informations et nouveautés préparent leurs newsletters régulièrement et les envoient par email aux abonnés. L'avantage des Newsletters est la facilité d'usage et la possibilité d'envoyer rapidement les dernières nouvelles à tous ceux qui peuvent être intéressés.

Facebook est utilisé par les CoPs pour supporter deux modes de communication : interne et externe. Pour supporter la communication interne, la CoP crée un groupe Facebook pour faciliter le partage et l'échange entre ses membres. Cependant, pour supporter la communication externe, la CoP crée une page Facebook qui s'utilise principalement pour partager les informations informatives, les nouveautés et les réalisations, etc. Dans ce cas, Facebook joue le rôle de la page web de la communauté. Le principal avantage de l'usage de

Facebook est sa popularité et la diversité des dispositifs qu'il supporte. Cependant, les membres peuvent préférer de séparer leurs vies personnelles de leurs vies professionnelles et de ne pas utiliser le même compte pour les deux.

Twitter est un outil de microblogage qui permet aux utilisateurs de poster des messages limités à 140 caractères. Les communautés utilisent Twitter pour la communication externe. Son avantage se rapporte à la facilité de mettre les nouveautés à disposition d'un large public. Par contre, il ne supporte pas la communication interne. Un autre point faible de Twitter est la taille courte des messages. Il doit être complété par un autre outil tel qu'un blog ou un wiki.

LinkedIn permet de mettre en contact des utilisateurs ayant des intérêts professionnels en commun, en formant un réseau social professionnel. Les communautés peuvent créer leurs propres pages et groupes pour faciliter la collaboration entre leurs membres. L'avantage de LinkedIn est sa popularité. Par contre, la communauté utilisant LinkedIn dépend des services offerts car il est difficile d'y intégrer de nouveaux.

Les outils de création collaborative de documents sont des outils, qui ont pour objectif de supporter la création des artefacts de connaissances. Ils sont utilisés principalement pour partager les documents sans passer par l'envoi d'emails. Les membres ayant accès à un document peuvent le modifier et y apporter des commentaires. L'avantage est de pouvoir stocker les documents sur des serveurs partagés. Cependant, l'inconvénient de ces outils est qu'ils ne supportent pas la communication.

Les weblog sont des sites qui permettent de publier des articles et d'y apporter des commentaires. L'usage des weblogs par les communautés est orienté vers la communication externe. Parmi les avantages des weblogs, nous trouvons la facilité d'utilisation. Par contre, ils ont besoin des membres volontaires pour s'occuper de leurs gestions.

L'usage **des wikis** est orienté vers la création des connaissances. En effet, les wikis facilitent l'édition collaborative des contenus. Par contre, un contenu risque de devenir incompréhensible lorsque le nombre de contributeurs est élevé.

L'étude exposée par BSR Stars ([Stars 2012](#)) a montré qu'il existe plusieurs barrières qui entravent l'utilisation des technologies par les communautés pour atteindre des objectifs liés au partage des connaissances. Les barrières détectées sont résumés dans les points suivants :

- La faible utilisation. Il existe des outils pouvant supporter le partage des connaissances. Cependant, les membres ne les utilisent qu'en cas d'obligation, ce qui peut induire leur abandon.
- L'inadéquation entre les besoins et les systèmes. Il y a souvent une inadéquation entre les besoins des membres et les systèmes existants.

- La faible intégration des systèmes. Les outils peuvent être plus utilisés par les membres des communautés si ce ne sont pas des "nouveaux" outils, mais plutôt des outils existants avec de nouvelles perspectives. Ainsi, il est préférable d'avoir un système regroupant un ensemble d'outils parmi ceux utilisés par les membres au quotidien.
- Le manque de transparence et de contrôle. Il concerne le problème de définition des rôles et de responsabilités par rapport à l'utilisation d'un outil. Qui fait quoi ? Qui est responsable de quoi ? Vers qui se tourner pour chaque genre de questions ?

Les points exposés ci-dessus supportent les questions de notre problématique de recherche. En effet, le point abordant l'aversion des membres des *nouveaux outils* est lié à la 2^{ème} question de notre problématique de recherche qui s'intéresse aux problèmes de familiarisation des usagers avec les nouveaux outils informatiques. Le point intégrant l'identification des membres vers lesquels se tourner pour chaque question peut être lié à la 5^{ème} question de notre problématique de recherche qui s'intéresse à l'obtention d'une solution adaptée au contexte de la difficulté et à celui de son émetteur.

Le point abordant la faible utilisation peut être lié aux 2^{ème}, 3^{ème} et 4^{ème} questions de notre problématique. Ils se rapportent, respectivement, à la familiarisation avec les nouveaux outils, à l'accès en cas de mobilité et à la minimisation de temps de réponse à une demande d'aide. Enfin, le point abordant l'inadéquation des outils peut être lié à l'objectif global de nos travaux visant à concevoir et à développer un outil personnalisé aux spécificités de notre CoP. Le tableau 2 ci-dessous présente les points communs entre les barrières entravant l'utilisation des technologies par les communautés et les questionnements de notre problématique de recherche.

	Q2 : Problèmes de familiarisation	Q3 : Minimisation du temps de réponse	Q4 : Accès en cas de mobilité	Q5 : Obtention d'une solution adaptée
Faible utilisation	X	X	X	-
Faible intégration des systèmes	X	-	-	-
Manque de transparence et de contrôle	-	-	-	X

Tab 2. Points communs entre les questions de notre problématique de recherche et les barrières identifiées par BSR Stars ([Stars 2012](#))

La section suivante présente la Communauté de Pratique MAETIC regroupant les usagers de la méthode pédagogique MAETIC.

2. CoP MAETIC

MAETIC ([Leclet and Talon 2008](#)) est une méthode pédagogique fondée sur une pédagogie de groupe par projet. Cette méthode est dédiée principalement aux formations professionnelles. Elle vise l'acquisition, pour les étudiants, de savoir-faire professionnel selon une pédagogie de groupe par projet. Pour ce faire, elle permet à l'enseignant de guider ses étudiants à réaliser un projet collectivement en suivant un ensemble d'étapes organisées selon un cycle de vie similaire à celui utilisé dans la gestion de projet. Les étapes de ce cycle sont (figure 3) : lancement, cadrage, planification, pilotage et clôture.

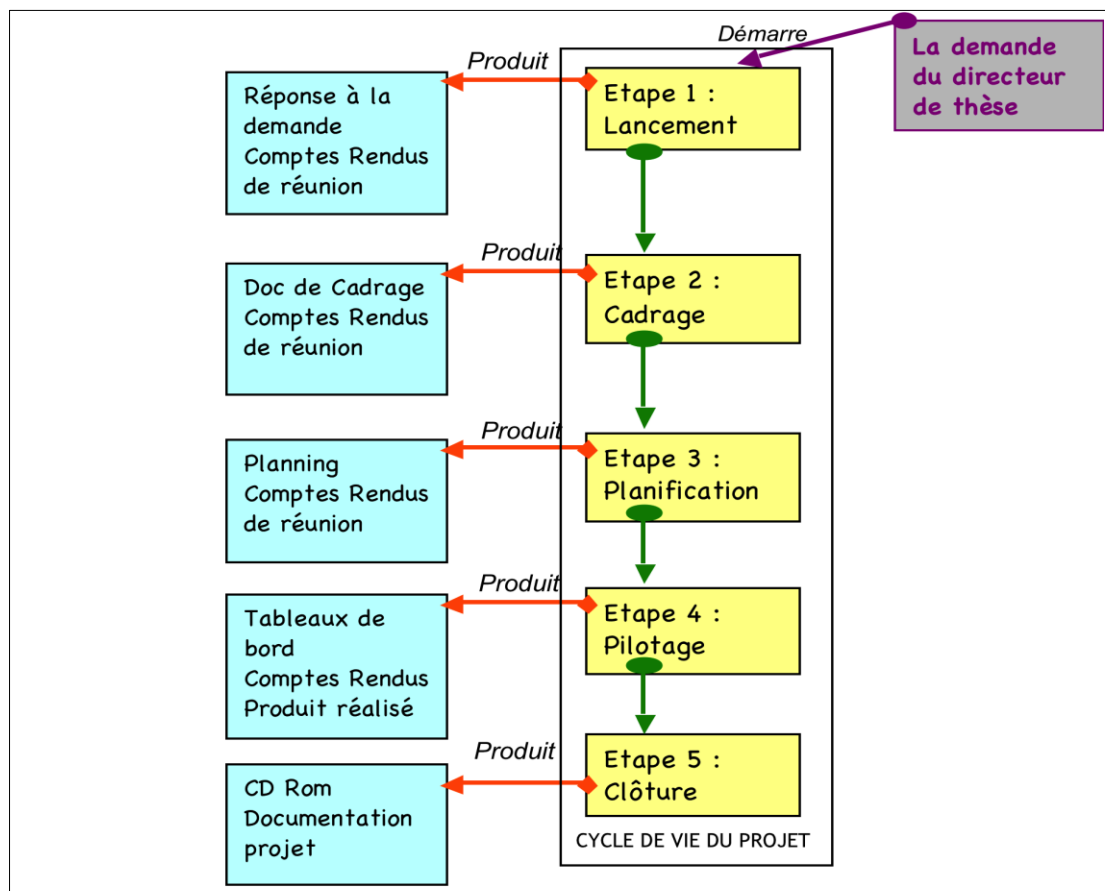


Figure 3. Cycle de vie de MAETIC ([Leclet and Talon 2008](#))

Chaque étape produit un ou plusieurs livrables et se déroule en une ou plusieurs séances de Travaux Dirigés (TDs). Chaque étudiant joue un rôle dans le projet : chef du projet, responsable du journal du bord du projet, chargé de communication et chargé de la documentation.

- L'étape de lancement a pour objectif de faire produire, par les étudiants, un livrable appelé « réponse à la demande » qui permet de définir le périmètre du projet. Cette étape permet de définir les acteurs du projet et leurs rôles, de réaliser une charte

graphique symbole du groupe et de créer un journal du bord, généralement un blog, qui permettra de suivre la vie du projet.

- L'étape de cadrage consiste à produire le cadre du projet qui répond à deux questions principales : "pourquoi et comment réaliser le projet ?". Elle permet également d'élaborer un plan de communication.
- L'étape de planification produit un document appelé le « planning ». Ce document définit les tâches à réaliser et leur déroulement dans le temps. Cette étape peut donner lieu à produire des comptes rendus de réunions.
- L'étape de pilotage permet de réaliser le produit du projet. Elle permet aussi de produire des comptes rendus de réunions ainsi que des tableaux de bord. Ces derniers permettent d'assurer le suivi du projet.
- L'étape de clôture a pour objectif de produire la documentation associée à la création du produit et à réaliser un document appelé « bilan du projet ». Elle consiste aussi à soutenir le projet et à livrer le produit final à l'enseignant.

Des retours d'expérimentations menées en 2010 ont montré que lorsqu'un enseignant déploie la méthode MAETIC dans sa classe, il peut rencontrer des difficultés dues, principalement, au manque d'expérience dans l'usage de celle-ci. En particulier, le manque de familiarisation avec les notions de gestion de projet ou de gestion des blogs qui sont considérés comme le principal outil informatique support aux interactions entre l'enseignant et ses étudiants. Il peut alors avoir besoin d'aide. Ces expérimentations ont également souligné que des enseignants plus expérimentés peuvent lui apporter cette aide.

Une classification des types de difficultés qui pouvaient être rencontrées, a été élaborée au cours d'un recueil d'expertise mené en 2010 auprès des experts de cette méthode. Ceux-ci ont identifié deux grandes classes de difficultés : les difficultés dites « transversales » et les difficultés spécifiquement liées à une étape de la méthode MAETIC.

La figure 4 ci-dessous présente les types de difficultés liées à une étape de MAETIC. Cette figure montre qu'une difficulté de ce type est liée à une étape spécifique de MAETIC : liée à l'étape 1 : Lancement (définition des rôles des étudiants, réalisation de la charte graphique, création du journal de bord etc.), liée à l'étape 2 : Cadrage (élaboration du cadre du projet, élaboration du plan de communication, etc.), liée à l'étape 3 : Planification (élaboration de diagramme de Gantt), liée à l'étape 4 : Pilotage (élaboration d'un tableau de bord, réalisation du produit, etc.), et liée à l'étape 5 : Clôture (élaboration du bilan de projet, production du manuel utilisateur, etc.).

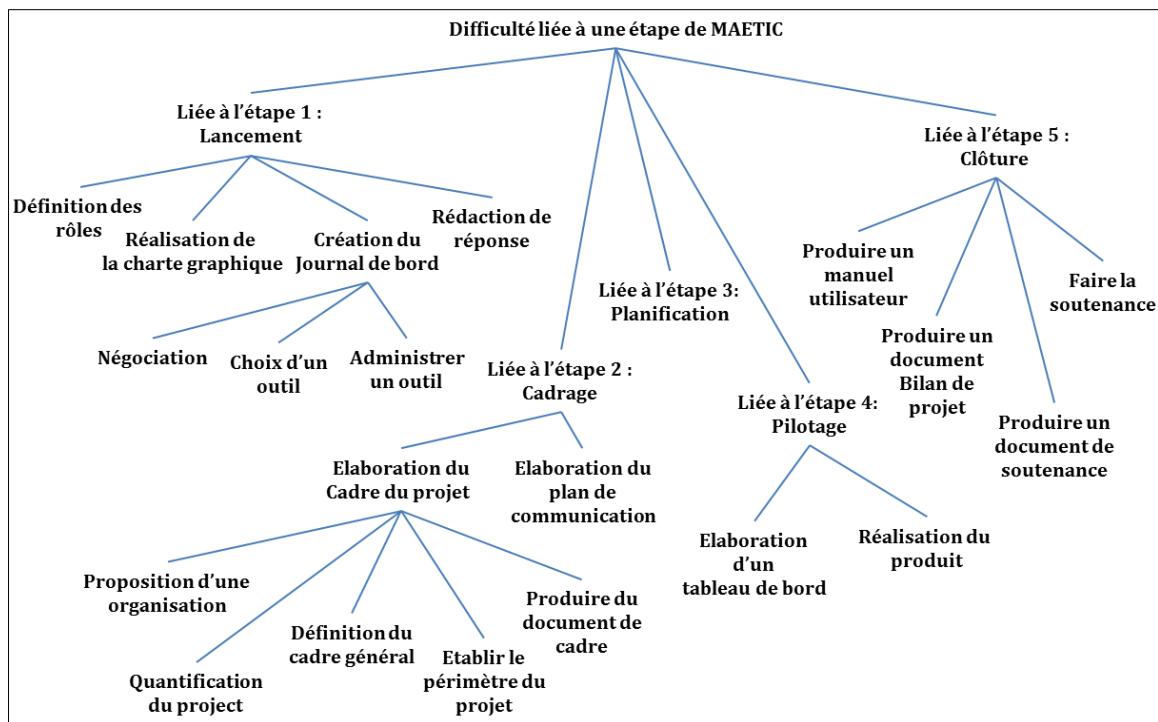


Figure 4. Classification des catégories de difficultés liées à une étape de MAETIC

La figure 5 ci-dessous présente les types de difficultés transversales. Les difficultés « transversales » ne sont pas liées à une étape spécifique de MAETIC mais plutôt à d'autres aspects. En particulier, au manque de compétences de l'enseignant dans la manipulation de blogs ou bien encore, dans la gestion de projet puisque MAETIC n'est pas seulement destinée aux enseignants familiarisés avec les technologies ou avec les notions de gestion de projet. Ces difficultés transversales peuvent aussi être liées à la gestion des étudiants, à l'évaluation des livrables ou à des problèmes d'infrastructure TIC disponible comme la panne réseau, configuration de salle, manque d'ordinateurs, etc.

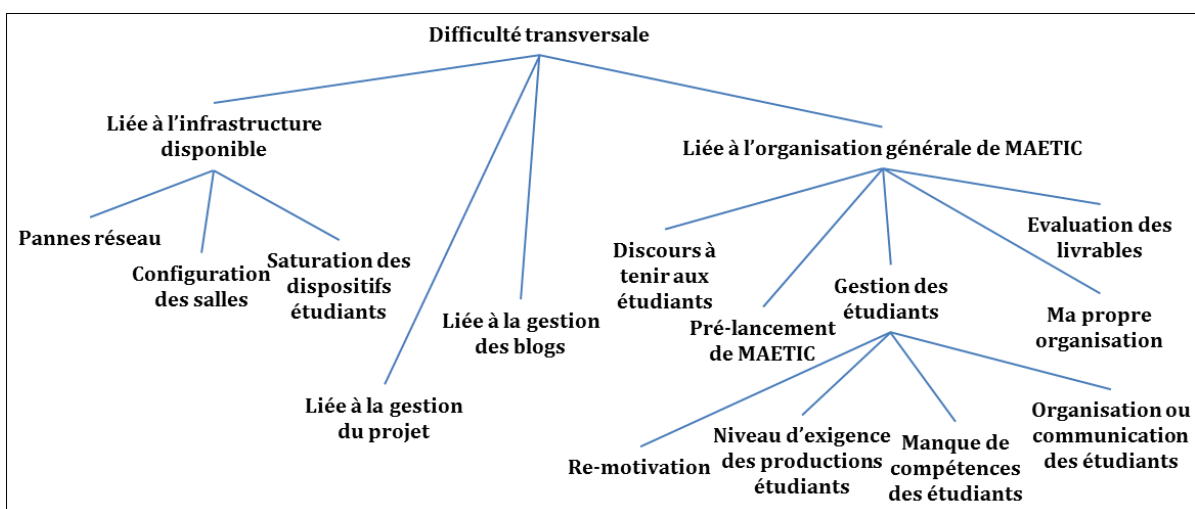


Figure 5. Classification des catégories de difficultés transversales

Pour ces deux classifications, transversales ou non, les difficultés peuvent avoir, ou non, un aspect lié au domaine enseigné.

Les enseignants usagers de la méthode MAETIC constituent donc une Communauté de Pratique dont les principaux objectifs sont : 1) **résoudre les difficultés liées à l'usage de MAETIC** et 2) **échanger leurs pratiques liées à cette méthode**.

En effet, d'une part, les membres de la CoP MAETIC sont dispersés géographiquement et d'autre part, les membres appartenant au même contexte géographique n'ont pas assez de temps pour se réunir en face à face et discuter des difficultés qu'ils rencontrent lors du déploiement de leurs dispositifs pédagogiques. Ceci soulève le besoin d'un outil technologique pour faciliter les échanges et la collaboration entre les membres de cette CoP.

Dans ce cadre, nous supportons la CoP MAETIC par un système communautaire dont le principal objectif est la résolution communautaire des difficultés MAETIC. Il doit donc prendre en considération les spécificités de ces difficultés. Nous avons appelé ce système NICOLAT (système iNformatIque COmmunautaire mobiLe et AdapTatif). Les principales fonctionnalités de NICOLAT sont présentées dans la section suivante.

3. NICOLAT comme support communautaire de la CoP MAETIC

3.1 Structure du système

Le système NICOLAT a pour objectif de supporter la CoP MAETIC et de faciliter les échanges entre ses membres. Il se compose de 4 couches : La couche communautaire (le noyau), la couche RP (Résolution de Problèmes), la couche AI (Adaptation des Interactions) et la couche Interface. Nous avons conçu ces 4 couches pour répondre à un aspect particulier de l'apprentissage via une CoP :

- La couche « **communautaire** » constitue le noyau du système. Elle correspond à un environnement web 2.0 dédié à la résolution communautaire de problèmes. Elle offre des fils de discussion similaires à ceux des forums, dont le but est de faciliter les interactions entre les membres autour de difficultés ou de pratiques MAETIC.
- La couche « **Résolution de Problèmes** » (RP). En observant les échanges entre les membres de la CoP, il a été constaté qu'il existait des problèmes (difficultés MAETIC) similaires postés de façon répétitive par plusieurs membres. Cet état de fait pouvait aboutir à l'inondation de l'espace communautaire par des fils de discussions similaires et par conséquent engendrer une démotivation des membres. La couche RP a donc pour objectif de proposer aux membres de la CoP les solutions de problèmes similaires qui ont été résolus dans le passé en se basant sur le Raisonnement à Partir des Cas (RàPC). Lorsque le membre n'est satisfait par aucune

de ces solutions, le problème est alors posté, minimisant ainsi le nombre de demandes d'aide sur l'espace communautaire.

- La couche « **Adaptation des Interactions** » (AI). Après la publication d'une demande d'aide par un membre, ce dernier doit alors attendre que les autres membres consultent l'environnement pour que sa demande soit prise en compte. La consultation fréquente de l'espace communautaire n'est pas assurée et peut être fastidieuse lorsqu'il n'y a pas souvent de nouveautés sur l'environnement. La couche AI a pour objectif d'implémenter des passerelles technologiques entre plusieurs outils d'interaction pour accéder à l'environnement communautaire. Cette couche permet ainsi aux membres de la CoP de consulter les fils de discussion de NICOLAT en choisissant l'un des outils suivants : Facebook, Gmail, SMS ou terminaux Android et iOS. La couche AI permet ainsi de minimiser le temps de réponse à une demande d'aide dans divers cas d'inaccessibilité comme les problèmes d'accès à Internet (utiliser les SMSs), les membres en forte mobilité (utiliser l'application Android ou SMS). La couche AI permet aussi d'augmenter le confort d'utilisation du système en proposant aux membres de supporter leurs interactions dans leurs outils préférés, notamment, en cas de non familiarisation avec l'interface par défaut de l'environnement communautaire dédié.
- La couche « **Interface** ». Le but de cette couche est d'assurer la communication entre la couche AI et le membre de la CoP. Elle fournit des composants logiciels au service de la couche AI capables de transmettre et de récupérer les données depuis et vers le système. Ces derniers sont groupés en : i) composants communautaires (application Facebook et composant dédié à l'interaction avec les clients mail) ; et ii) composants mobiles (application Android et un composant logiciel capable d'envoyer et recevoir des SMSs).

Dans les paragraphes suivants, nous détaillons le noyau du système. Les autres couches seront détaillées dans les chapitres suivants.

3.2 Fonctions du noyau

Le noyau du système NICOLAT présente un ensemble de fonctionnalités que nous retrouvons souvent dans les réseaux communautaires et qui tournent autour de la gestion d'utilisateurs, de leurs contenus et de notifications. Certaines de ces fonctionnalités sont adaptées aux spécificités de la CoP MAETIC. La figure 6 ci-dessous présente le diagramme des cas d'utilisation du noyau. Elle permet de visualiser ses principales fonctionnalités.

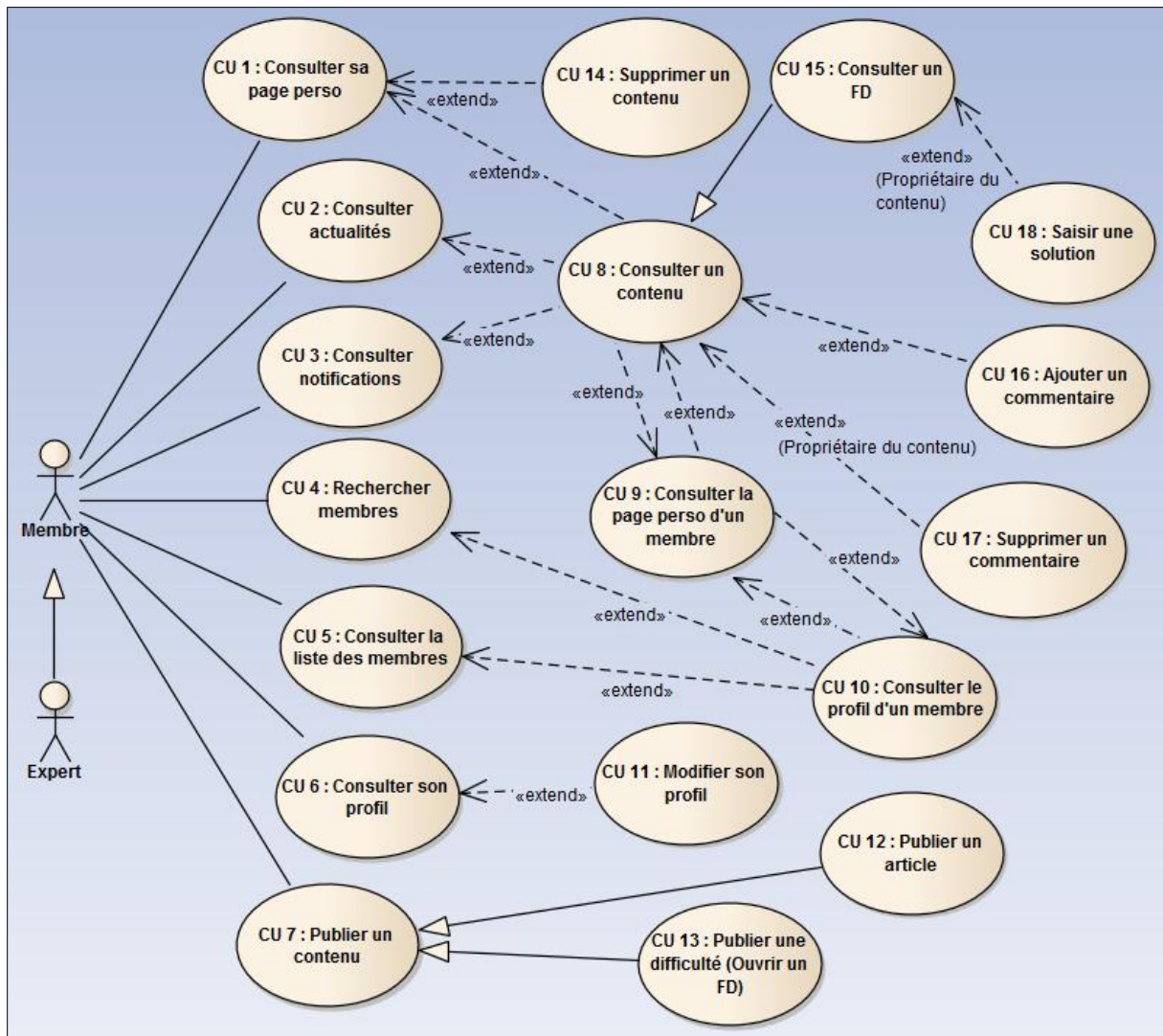


Figure 6. Diagramme de cas d'utilisation du système communautaire

Ainsi, à travers le noyau NICOLAT, un membre peut :

- Consulter sa page personnelle (CU 1) à travers laquelle il consulte tous les contenus qu'il a publiés sur le système (CU 8). Un contenu posté peut être une difficulté ou une pratique MAETIC.
- Consulter le fil d'actualité (CU 2), qui est une page synthétisant toutes les publications récentes intervenues sur les pages personnelles des autres membres de la communauté. A partir de la consultation des actualités, il a accès aux contenus récemment publiés.
- Consulter les notifications (CU 3) qu'il reçoit lorsqu'une nouvelle difficulté est publiée ou lorsqu'un nouveau commentaire est apporté à un contenu qu'il a publié. A partir de la consultation d'une notification, il a alors le droit de consulter le contenu duquel est issue la notification (CU 8).

- Consulter la liste des membres de la communauté (CU 5), ou encore effectuer la recherche d'un membre (CU 4) dans la communauté. A partir de ces deux cas d'utilisation, il a la possibilité de consulter le profil d'un membre (CU 10) et à partir de là consulter la page personnelle de ce membre (CU 9).
- Consulter son propre profil (CU 6) et à partir de là, le modifier (CU 11).
- Publier un contenu (CU 7) qui peut être un article, ou une difficulté pour initier un FD (Fil de discussion).

Par ailleurs :

- Lorsqu'un membre consulte un contenu particulier (CU 8) il aura alors la possibilité, d'y apporter un commentaire (CU 16) ou d'en supprimer s'il s'agit de son propre contenu (CU 17).

3.3 Développement du noyau

Pour développer le noyau, nous avons utilisé le CMS Drupal qui est un CMS libre, open source et populaire, permettant la création de sites web et des applications. Notamment, il est dédié à la création des sites web collaboratifs et ceux axés sur le contenu ([Persistent 2012](#)). Il est utilisé pour développer des sites web dans des contextes variés allant des blogs personnels aux sites d'entreprise et de gouvernements. Nous notons que le site de la maison blanche (WhiteHouse.gov) et celui d'accès aux données gouvernementales en UK (data.gov.uk) sont développés sous Drupal ainsi que 2% des sites web existants ([Sabo 2015](#)). C'est le 3eme CMS le plus utilisé après Joomla et Wordpress ([Sabo 2015](#)). De plus, Drupal est supporté par une communauté de plus de 180000 développeurs ([Sabo 2015](#)).

Drupal possède un noyau constitué de modules offrant les fonctionnalités essentielles pour la création de site web comme la gestion d'utilisateurs, ou bien encore de contenus. Ce noyau est extensible par des modules dont un grand nombre est disponible sur le site officiel de Drupal (plus de 17000). Le grand nombre de ces modules fait la puissance de Drupal. Ils permettent ainsi de créer des sites personnalisables et modulables. En effet, il est possible de créer des sites web complexes par des configurations et des ajouts de modules.

Notre choix d'utiliser DRUPAL a été effectué donc en raison de sa puissance dans l'implémentation des systèmes communautaires en offrant des modules utiles pour la création de ce type de systèmes, mais aussi en raison de sa communauté très active et son aspect modulaire qui permettra de faire évoluer le système réalisé facilement. De plus, un ensemble d'outils liés à la méthode MAETIC ont déjà été développés en DRUPAL.

Ainsi, en plus des modules faisant partie du noyau de Drupal, nous avons utilisé un certain nombre de ces modules parmi lesquels nous citons :

- Views, qui permet de créer des pages, des blocs ou des flux listant des contenus.
- Wysiwyg, qui offre un outil d'édition de texte, qui sert à produire du code HTML sans pour autant avoir à connaître les balises du langage.
- Chaos tools, qui permet la gestion avancée des formulaires, l'ajout de l'Ajax pour les formulaires, la création de simples contenus utilisables dans les panels et bien d'autres.
- Hierarchical Taxonomy, qui organise des taxonomies sur plusieurs niveaux. Dans Drupal, les taxonomies sont des termes/vocabulaires permettant de classifier les contenus.
- Hierarchical select, qui permet de créer des champs de sélection en cascade.
- Services, qui offre des services web permettant d'accéder aux différentes fonctionnalités d'un site Drupal.
- Best Reply, qui permet de marquer les meilleurs commentaires.

3.4 Interfaces

Dans le présent paragraphe, nous présentons quelques interfaces du noyau NICOLAT. La figure 7 ci-dessous, montre une interface de saisie d'une difficulté MAETIC.

Type

Type de la difficulté *

Difficulté Transversale ▼

Liée à l'organisation générale de MAETIC ▼

Discours à tenir aux étudiants ▼

Description

Titre de votre difficulté *

Description de la difficulté *

informations sur la classe

Niveau :

Effectif :

Nombre de groupes :

Nombre d'heures

Nombre d'heures par séance :

Figure 7. Formulaire de description d'une difficulté MAETIC

Cette interface permet aux membres de la CoP, de saisir les informations de la difficulté en renseignant 3 onglets :

1. Une liste de sélection hiérarchique permettant de choisir le type de difficulté par rapport à la classification des types de difficultés MAETIC présentée dans la section précédente.
2. Deux champs permettant de saisir le titre de la difficulté et sa description textuelle.

3. Des champs de sélection optionnels permettant de saisir des informations sur la classe où la difficulté est rencontrée représentant le contexte de la difficulté.

La figure 8 ci-dessous, présente la page d'accueil représentant le fil d'actualité permettant à un membre de la CoP de visualiser la liste des difficultés postées sur le noyau.

The screenshot shows a web page titled 'Actualités' with two tabs: 'Difficultés MAETIC' (selected) and 'Pratiques MAETIC'. The page displays three posts:

- Post 1:** Title: 'Publications s'affichent comme brouillons'. Submitted by 'admin' on 'le mer, 02/18/2015 - 08:59'. Status: 'Difficulté résolue' and 'Solution validée'. Type: 'Type de difficulté' with 'Administrer un outil'. Description: 'Lors de l'utilisation de wordpress, mes publications s'affichent comme brouillons, comment je peux résoudre ce problème ?'. Link: 'Lire la suite'.
- Post 2:** Title: 'temps insuffisant'. Submitted by 'abegdouri' on 'le mar, 02/17/2015 - 14:51'. Status: 'Difficulté résolue' and 'Solution validée'. Type: 'Type de difficulté' with 'Faire la soutenance'. Description: 'Je n'ai pas assez de temps pour faire la soutenance orale. Que faire?'. Link: 'Lire la suite' and '1 commentaire'.
- Post 3:** Title: 'Activités MAETIC'. Submitted by 'rachid' on 'le lun, 02/16/2015 - 23:18'. Type: 'Type de difficulté' with 'Liée à l'étape 3 : Planification'. Description: 'Quelles sont les étapes des activités et comment les planifier ?'. Link: 'Lire la suite' and '1 commentaire'.

Figure 8. Page d'actualités du noyau

Juste en dessous du titre de la page, nous trouvons deux onglets : « Difficultés MAETIC » et 'Pratiques MAETIC'. L'onglet « Difficultés MAETIC » représente la page qui permet de visualiser les difficultés postées par les membres de la CoP alors que l'onglet « Pratiques MAETIC » permet d'accéder à la liste de pratiques MAETIC postées. Sur cette page, chaque difficulté est illustrée par son titre, son type et sa description textuelle.

La figure 9 ci-dessous présente l'interface permettant de visualiser le détail d'une difficulté MAETIC.

The screenshot shows a web interface for 'Activités MAETIC'. At the top, the title 'Activités MAETIC' is displayed in a large, bold font. Below the title, the submission information reads 'Soumis par rachid le lun, 02/16/2015 - 23:18'. There are two tabs: 'Type de difficulté' (selected) and 'Liée à l'étape 3 : Planification'. The 'Description' section contains the text 'Quelles sont les étapes des activités et comment les planifier ?'. The 'Informations sur la classe' section lists: 'Niveau de la classe : Licence | Effectif : Entre 21 et 30 | Nombre de groupes : 5' and 'Nombre d'heures par seance : 2 | Nombre d'heures : Entre 41 et 50'. The 'Informations sur l'enseignant' section lists: 'Spécialité : Informatique | Pratique MAETIC : Expert' and 'Experiences Gestion de projet : Expert | Experiences Blog : Expert'. The 'Enseignants recommandés à résoudre cette difficulté' section lists 'admin'. A 'Solution' section is partially visible. At the bottom, there is a section for 'Ajouter un commentaire' with a form for 'Votre nom' (filled with 'rachid') and 'Comment *' (empty).

Figure 9. Page permettant de visualiser une difficulté MAETIC

Sur cette figure, est présenté le détail d'une difficulté de type planification dont le titre est « Activités MAETIC ». Nous trouvons donc le titre de difficulté suivi par des informations de

la soumission (émetteur de la difficulté, date et heure de publication). Ensuite, le type de la difficulté, sa description, des informations sur la classe et sur l'enseignant qui l'a posté. Enfin, la liste des commentaires ainsi que des champs permettant d'ajouter un nouveau commentaire.

Nous notons que le noyau a été utilisé par les membres de la CoP MAETIC. Ces derniers ont apprécié ses aspects fonctionnel et ergonomique.

4. Généricité de NICOLAT

Certaines fonctionnalités du système NICOLAT sont liées à la CoP MAETIC, ce qui peut représenter un problème si l'on souhaite l'utiliser pour d'autres communautés. Pour pallier ce problème, nous proposons de reconcevoir les composants de NICOLAT qui sont fortement attachés aux spécificités de la CoP MAETIC pour qu'ils soient génériques, est donc réutilisables par d'autres CoP.

Dans ce cadre, nous commençons tout d'abord par reprendre la structure du système NICOLAT et redéfinir ses couches, présentées dans la section 3.1, pour qu'elles soient génériques. Ainsi, nous redéfinissons ces couches comme suit :

- La couche « **Communautaire** » constitue le noyau du système. Elle correspond à un environnement web 2.0 qui proposent deux types de fonctionnalités : i) des fonctionnalités communes pour les systèmes communautaires support de l'apprentissage via la résolution communautaire de problèmes, et ii) des fonctionnalités qui nécessitent une personnalisation selon le domaine et la communauté cible.
- La couche « **Résolution de Problèmes** ». Afin de minimiser le nombre de demandes d'aide sur l'espace communautaire, cette couche propose au membre cherchant à résoudre un problème, les solutions de problèmes similaires résolues dans le passé. Lorsque le membre n'est satisfait par aucune de ces solutions, le problème est alors posté au sein de la communauté.
- La couche « **Adaptation des Interactions** ». Cette couche a pour objectif d'implémenter des passerelles technologiques entre plusieurs outils d'interaction pour accéder à l'environnement communautaire. Elle permet ainsi aux membres de la CoP de poster un problème ou de consulter les fils de discussion en utilisant un de leurs outils d'interaction préférés. Cette couche s'occupe aussi de sélectionner les membres à participer à la résolution d'un problème donné.
- La couche « **Interface** ». Le but de cette couche est d'assurer la communication entre la couche AI et les membres de la CoP. Elle fournit des composants logiciels au

service de la couche AI capables de transmettre et de récupérer les données depuis et vers le système NICOLAT.

Dans la suite de ce manuscrit, nous présenterons pour chaque chapitre, nos solutions par rapport à la CoP MAETIC puis la généralité des solutions proposées et comment elles peuvent être réutilisées par d'autres communautés. Dans le paragraphe suivant nous abordons la généralité de la couche communautaire.

5. généralité de la couche communautaire

5.1 Fonctionnalités

La couche/noyau communautaire du système NICOLAT présente un ensemble de fonctionnalités que nous retrouvons souvent dans les réseaux communautaires. Certaines de ces fonctionnalités peuvent être utilisées directement par d'autres communautés, alors que d'autres ont été adaptées aux spécificités de la CoP MAETIC. Dans la figure 10 ci-dessous nous reprenons le diagramme de cas d'utilisation du noyau communautaire en marquant en rouge les fonctionnalités liées aux spécificités de la CoP MAETIC et qui doivent donc être généralisées.

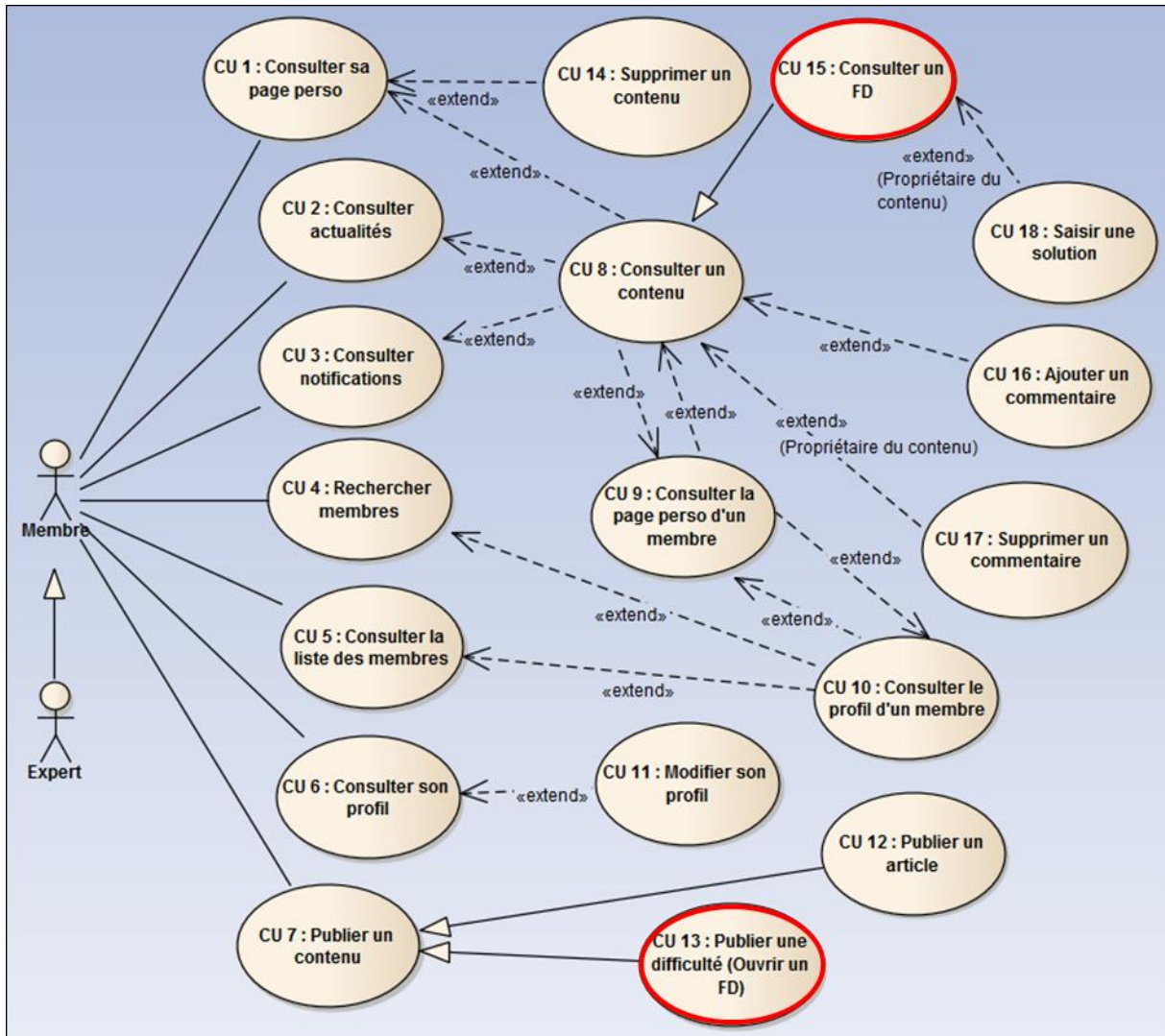


Figure 10. Diagramme de cas d'utilisation du noyau spécifiant les fonctionnalités liées aux spécificités de MAETIC

Il y a donc deux fonctionnalités à généraliser : la fonctionnalité « CU 13 » qui devra permettre de poster des problèmes dont la structure change selon la communauté cible et la fonctionnalité « CU 19 » qui permet de consulter un problème et ses commentaires (Fil de discussion). Les autres fonctionnalités sont utilisables par n'importe quelle communauté sans besoin de personnalisation.

Ces deux fonctionnalités, à généraliser, concernent respectivement la saisie et la consultation de problèmes, leur généralisation revient donc à proposer une structure de problème adaptable selon les spécificités du domaine cible. Dans la section suivante, nous proposons une structure générique de problèmes.

5.2 Structure générique de problèmes

Nous considérons que n'importe quel problème peut être décrit à travers trois types d'entités :

- Type de problèmes : permettant de classer les problèmes par thème.
- Description du problème : permettant de spécifier la description textuelle du problème.
- Attributs de contextes : permettant de décrire le contexte où le problème a été rencontré.

La figure 11 ci-dessous représente le formalisme UML décrivant ces trois types d'entités.

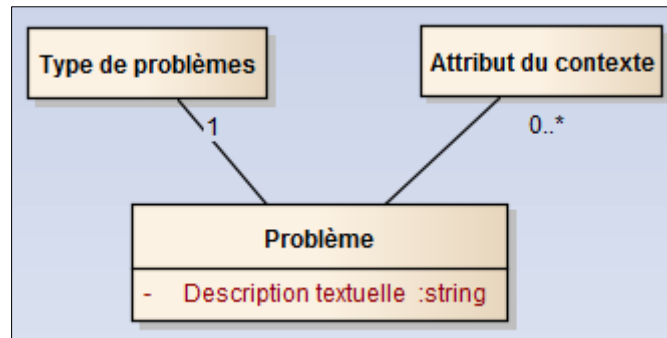


Figure 11. Structure générique du problème

Sur cette figure, la classe « Type de problèmes » définit le type du problème par rapport à la classification des types de problèmes du domaine cible. Dans ce cadre nous proposons un modèle permettant de concevoir une classification hiérarchique des types de problèmes. Ce modèle est représenté dans la figure 12 qui montre une classification de type de problèmes se composant tout d'abord de classes de problèmes, et chaque classe peut avoir des sous-classes. Finalement, chaque type de problèmes appartient à une classe.

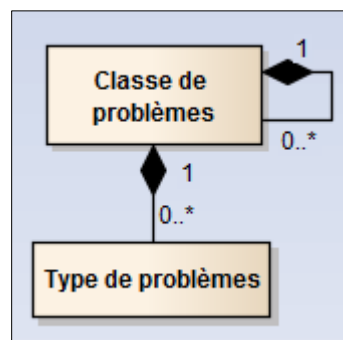


Figure 12. Modèle de classification de problèmes

La figure 13 ci-dessous présente un exemple de classification réduite des problèmes liés au domaine du développement informatique implémentant le modèle que nous avons proposé. Dans cette classification, la classe « Problème lié au développement informatique » regroupe trois sous-classes : « Java », « .NET » et « Mobile ». Chacune de ces classes regroupe des types de problèmes. La classe « Java » par exemple, regroupe quatre types de problèmes : « Java web », « Spring » et « Eclipse ».

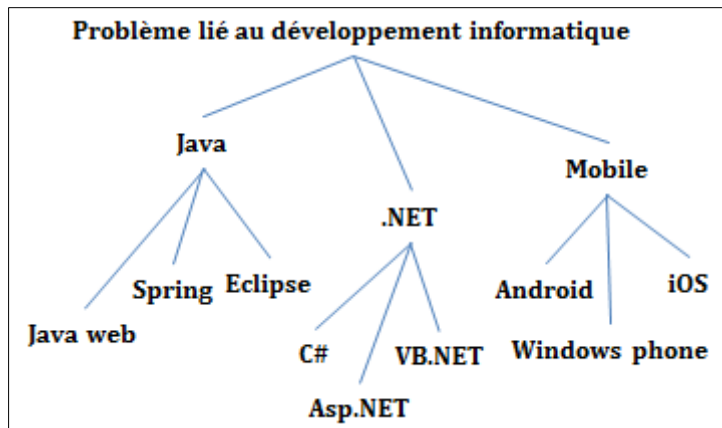


Figure 13. Classification réduite des problèmes liés au développement informatique

Ainsi, la figure 14 ci-dessous présente la classification des difficultés MAETIC liées à l'étape 2, implémentant le modèle que nous avons proposé. Sur cette figure, la classe « liée à l'étape 2 : Cadrage » regroupe le type « Elaboration du plan de communication » et la sous-classe « Elaboration du cadre du projet ». Cette dernière regroupe cinq types de difficultés MAETIC.

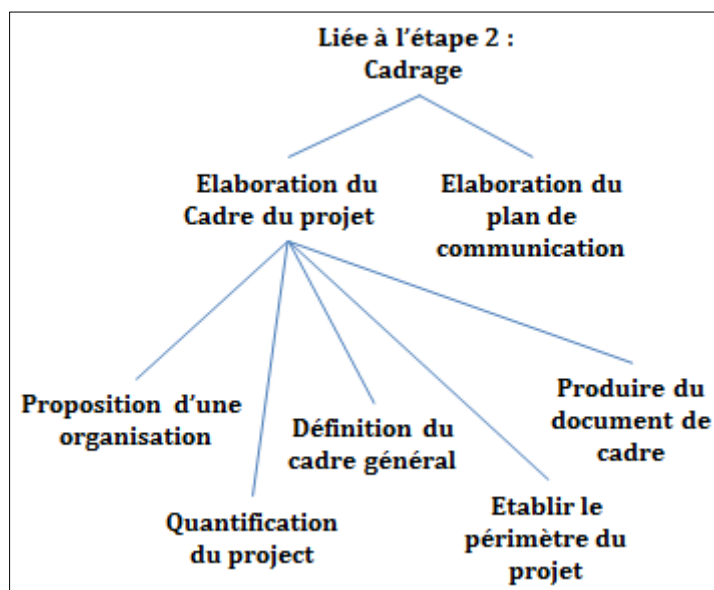


Figure 14. Classification des difficultés MAETIC liées à l'étape 2 (Cadrage)

5.3 Proposition d'implémentation

Pour implémenter notre structure générique de problèmes à la couche communautaire afin de la rendre générique, nous proposons :

- D'utiliser un fichier XML pour spécifier la classification des problèmes du domaine selon la classification présentée dans le paragraphe précédent.
- D'utiliser un second fichier XML pour spécifier les attributs du contexte.

- De développer un script PHP qui permet de personnaliser la couche communautaire en utilisant les informations fournies dans ces deux fichiers. Ce script crée, au niveau de la base de données du système, de nouvelles données décrivant les types de problèmes et les attributs comme ils sont spécifiés dans les deux fichiers XML.

L'utilisation de fichiers XML pour décrire la structure de problèmes, rend donc la couche communautaire personnalisable pour supporter différentes communautés.

La figure 15 ci-dessous expose le fichier XML décrivant la classification des problèmes informatique présentée dans le paragraphe précédent.

```
<problem-class name="Problème lié au développement informatique" >
  <problem-class name="Java" >
    <problem-type>Java web</problem-type>
    <problem-type>Spring</problem-type>
    <problem-type>Eclipse</problem-type>
  </problem-class>
  <problem-class name=".NET" >
    <problem-type>C#</problem-type>
    <problem-type>Asp.NET</problem-type>
    <problem-type>VB.NET</problem-type>
  </problem-class>
  <problem-class name="Mobile" >
    <problem-type>Android</problem-type>
    <problem-type>Windows phone</problem-type>
    <problem-type>iOS</problem-type>
  </problem-class>
</problem-class>
```

Figure 15. Classification des problèmes de développement informatique en fichier XML

Ainsi, la figure 16 ci-dessous présente le fichier XML décrivant la classification des difficultés MAETIC liées à l'étape 2.

```
<problem-class name="Liée à l'étape 2 : Cadrage" >
  <problem-type>Elaboration du plan de communication</problem-type>
  <problem-class name="Elaboration du Cadre du projet" >
    <problem-type>Proposition d'une organisation</problem-type>
    <problem-type>Quantification du projet</problem-type>
    <problem-type>Définition du cadre général</problem-type>
    <problem-type>Etablir le périmètre du projet</problem-type>
    <problem-type>Produire du document de cadre</problem-type>
  </problem-class>
</problem-class>
```

Figure 16. Classification des difficultés MAETIC liées à l'étape 2 en fichier XML

La CMS Drupal propose une structure appelée 'Taxonomie' permettant de classer le contenu en utilisant des vocabulaires. Ces derniers peuvent être organisés hiérarchiquement. Nous utilisons donc une taxonomie pour représenter la classification de problèmes où chaque classe/type correspond à un vocabulaire. La figure 17 ci-dessous présente l'algorithme du

script permettant de créer les vocabulaires d'une manière hiérarchique selon la classification décrite dans le fichier XML de types de problèmes. Cet algorithme est présenté sous forme d'une fonction récursive.

```
Fonction creervocabulaire(vocabulaireParent vp, balise b)
  Si b est 'classe'
    Créer un vocabulaire b.nom descendante de vp
    Retourner;
  Sinon
    Créer un vocabulaire b.nom descendante de vp
    Pour chaque sous balise de b : sb
      creervocabulaire(b.nom, sb)
    Fin pour
  Fin si
Fin fonction
```

Figure 17. Algorithme du script permettant de créer la taxonomie représentant la classification de problèmes

6. Conclusion

Ce chapitre a fait l'objet d'un état de l'art sur les Communautés de Pratiques qui nous a permis de présenter leur caractéristique, leur intérêt et leur cycle de vie. Nous avons également présenté leur distinction des autres structures sociales. Puis, nous avons abordé les communautés virtuelles et les outils pour les supporter ainsi que les barrières qui entravent l'utilisation des technologies par les CoPs.

Nous avons ensuite introduit la méthode pédagogique MAETIC et la CoP des enseignants utilisateurs de cette méthode. Rappelons que l'objectif de cette CoP est de résoudre les difficultés rencontrées par ses membres lors du déploiement de MAETIC en classe. Les fonctionnalités du système NICOLAT, support de la CoP MAETIC, et sa structure ont été alors détaillées. Nous avons ensuite exposé les fonctionnalités communautaires du noyau, son développement, ainsi que quelques-unes de ses interfaces. Enfin, nous avons abordé la genericité du système NICOLAT et de son noyau communautaire.

Pour la résolution de nouveaux problèmes rencontrés, nous exploitons les problèmes déjà résolus dans le passé, et nous combinons, le noyau communautaire avec la couche RP. Ce principe vise à aider le membre de la CoP à résoudre son problème lui-même sans solliciter la communauté. Cette couche est détaillée dans le chapitre suivant.

Chapitre 3 : La couche « Résolution de Problèmes »

Ce chapitre décrit la couche de Résolution de Problèmes (RP) de notre système qui se base sur le Raisonnement à Partir des Cas (RàPC). Nous commençons donc par l'exposition de la problématique à l'origine de la mise en place de cette couche, suivie d'une présentation détaillée du RàPC. Nous présentons ensuite le fonctionnement de la couche RP par rapport à la CoP MAETIC, notre conception d'un cas et l'implémentation du cycle RàPC que nous avons adopté. Dans une avant dernière partie, nous exposons une validation expérimentale de la couche RP. Enfin, nous concluons par la généralité de cette couche.

1. Problématique et objectif de la couche RP

L'apprentissage à travers une CoP dédiée à la résolution de problèmes est un apprentissage collaboratif basé sur la résolution communautaire de problèmes. Notre première contribution à rendre cet apprentissage plus efficace, est de mettre à disposition des membres de la CoP, un réseau social afin de supporter leurs interactions lors du processus de résolution de problèmes (le noyau communautaire du système NICOLAT cf. chapitre 1).

Or, la pratique de résolution de problèmes dans une CoP a montré que :

- Les problèmes rencontrés par les nouveaux membres se répètent. En effet, les mêmes questionnements se posent par ces membres.
- Les fils de discussion à travers le noyau communautaire se voient surchargés par des demandes d'aide pour des problèmes similaires et déjà résolus dans le passé.
- Les membres, conscients que leurs problèmes (ou des problèmes similaires) peuvent avoir eu des solutions dans le passé, ne sont pas prêts à chercher ces anciennes solutions dans les fils de discussion. En effet, ils se retrouveraient vite noyés dans un grand nombre de commentaires qui peuvent ne pas être utiles dans le contexte de leur nouveau problème.
- Enfin, les membres actifs dans la CoP peuvent être découragés s'ils reçoivent des demandes d'aide similaires. Ils peuvent ne plus répondre aux difficultés nouvellement postées qui ont été déjà résolues dans le passé. Ceci entraîne donc la démotivation de ces membres actifs, mais aussi des membres émetteurs du problème s'ils ne reçoivent pas (ou peu) de retour par rapport à leur demande d'aide.

Nous concevons donc la couche « Résolution de Problèmes » comme réponse à la première question de notre problématique de recherche : **Comment éviter des demandes de résolution de problèmes similaires ?**

Ainsi, nous visons à travers la couche RP à aider le membre à résoudre son problème lui-même en se basant sur des problèmes déjà résolus dans le passé sans faire intervenir les autres membres de la CoP, et en minimisant ainsi le nombre de demandes d'aide postées sur l'espace communautaire. En effet, avant de poster un problème, cette couche intervient pour assister le membre à trouver une solution en lui proposant des problèmes résolus et similaires au sien pour qu'il s'inspire de leurs solutions afin de concevoir la sienne. Si les solutions proposées, par la couche RP, ne le satisfont pas, il poste alors une demande d'aide dans l'espace communautaire.

Pour atteindre cet objectif, nous nous inspirons du cycle du Raisonnement à Partir de Cas, pour implémenter cette couche. En effet, cette méthode de raisonnement est très adaptée pour les applications cherchant à exploiter l'historique des expériences passées afin de les réutiliser dans la résolution d'un nouveau problème. La présentation du Raisonnement à Partir des Cas fait l'objet du paragraphe suivant.

2. Raisonnement à Partir de Cas

2.1 Principe

Le Raisonnement à Partir de Cas est un paradigme de raisonnement par analogie qui utilise des expériences précédentes pour résoudre de nouveaux problèmes ([Aamodt and Plaza 1994](#)). L'objectif donc est de permettre de trouver une solution à un problème en exploitant les solutions de problèmes similaires qui ont été résolus dans le passé.

Le RàPC est datée des années 1980 et issue des domaines des sciences cognitives et de l'intelligence Artificielle ([Kolodner and Leake 1996](#)). Il est inspiré des observations du raisonnement humain ([Jaczynski 1998](#)). En effet, dans notre vie quotidienne, lorsque nous rencontrons un problème et voulons le résoudre, nous faisons appel à nos expériences. Nous nous remémorons les situations similaires que nous avons vécues. Ensuite, nous les comparons à la situation actuelle afin de pouvoir concevoir une nouvelle solution que nous ajoutons ainsi à nos expériences. Le RàPC simule donc ce comportement humain. Il permet de résoudre un problème en cherchant des problèmes similaires, déjà résolus, dans une base de connaissances et en adaptant leurs solutions au problème actuel.

Dans un système RàPC, un **problème** et sa **solution** forment une expérience appelée un **cas**. L'ensemble des cas forme ce qui est appelé une base des cas. La description du problème peut regrouper les objectifs à atteindre, les contraintes sur ces objectifs et le contexte dans lequel le problème a été rencontré. La solution peut contenir la description de la solution effectivement apportée, les étapes qui ont mené à cette solution, à son évaluation et les effets de son application ([Jaczynski 1998](#)). Les représentations de ces deux éléments, le problème et sa

solution, dépendent des connaissances du domaine, de données disponibles et des objectifs du système.

Les cas sont des expériences passées qui permettent à un système RàPC de résonner. En effet, ils permettent de résoudre de nouveaux problèmes. Il faut donc extraire les caractéristiques des cas, appelées indices, qui permettent de déterminer si un cas est applicable/utile ([Jaczynski 1998](#)). Ces indices doivent être donc prédictifs. Ils sont censés permettre de décider dans quelles conditions une solution est réutilisable pour un autre problème.

Dans le but de trouver une solution à un problème donné, un système RàPC se base sur des processus et des connaissances. Dans la section suivante, nous présentons ceux-ci.

2.2 Cycle du raisonnement RàPC

Le RàPC se compose de processus qui doivent s'exécuter lors de la résolution d'un nouveau problème. Le nombre de ces processus varie selon les différentes sources bibliographiques entre trois et cinq processus. Fuchs et al. ([Fuchs, Lieber et al. 2006](#)) ont défini trois processus : la remémoration, l'adaptation et la maintenance. Aamodt et Plaza ([Aamodt and Plaza 1994](#)) qui sont les premiers auteurs à avoir décrit les processus du RàPC ont déterminé quatre processus : la remémoration, l'adaptation, la validation et la maintenance. Mille ([Mille 2006](#)) a ajouté un processus préliminaire qui précède les autres et qui permet d'élaborer les cas cibles, ce processus s'appelle l'élaboration. En résumé, les processus qui peuvent être exploités dans un système RàPC sont :

- **L'élaboration.** Ce processus permet de construire le cas cible en complétant et/ou filtrant la description d'un problème fourni par un utilisateur.
- **La recherche/remémoration.** Ce processus permet d'identifier et d'extraire les cas de la base dont les problèmes sont similaires au problème cible (à résoudre).
- **L'adaptation/Réutilisation.** Ce processus consiste à modifier et réutiliser les cas extraits par la phase précédente dans le but de résoudre le problème courant. Selon Lamontagne and Lapalme ([Lamontagne and Lapalme 2002](#)), peu de systèmes effectuent ce processus de manière automatique. Souvent, la génération de la solution à partir des cas extraits, nécessite une intervention humaine. Le degré de cette intervention est variable d'un système à un autre.
- **La révision.** Ce processus consiste à vérifier, à évaluer et éventuellement à corriger la solution. Elle s'effectue aussi dans la plupart des cas de façon manuelle.
- **La maintenance.** Ce processus consiste à ajouter le cas résolu dans la base de cas dans le but d'enrichir cette base et d'en profiter dans le future pour résoudre d'autres problèmes. Il permet d'assurer l'évolutivité du système.

Un autre processus appelé « Authoring » ([Lamontagne and Lapalme 2002](#)) s'ajoute à ces cinq processus. Celui-ci s'exécute lors de la phase d'implémentation du système afin de construire la base initiale de cas et les autres connaissances du système. Il se base sur différents types de ressources telles que des documents, des bases de données ou des entretiens avec des experts et des praticiens du domaine. Souvent, ce processus s'effectue manuellement par les concepteurs du système.

Selon Lamontagne et Lapalme ([Lamontagne and Lapalme 2002](#)), les différentes connaissances exploitées par un système RàPC sont regroupées en quatre catégories :

- Les **vocabulaires d'indexation**, sont des attributs qui permettent de décrire le problème et sa solution. Elles constituent le modèle utilisé pour représenter les cas.
- La **base de cas**, cœur du système RàPC, se compose des cas (problèmes avec leurs solutions).
- Les **mesures de similarité**, sont les mesures et les fonctions permettant de mesurer la similarité entre deux cas. Généralement, ces mesures s'utilisent lors de la phase de recherche pour identifier les cas de la base de cas qui sont similaires au cas cible.
- Les **connaissances d'adaptation** sont les règles qui permettent de modifier et d'exploiter les solutions retournées par la phase de remémoration dans le but de construire une solution pour le problème courant.

La figure 18 ci-dessous, basée sur le modèle générique d'un système RàPC présenté par Lamontagne et Lapalme ([Lamontagne and Lapalme 2002](#)), présente les différents processus et connaissances d'un système RàPC. Cette figure montre que le processus de « Authoring » permet de construire les différentes connaissances nécessaires pour le fonctionnement d'un système RàPC, alors que les autres processus (élaboration, recherche, adaptation, révision, maintenance) s'exécutent lors de l'introduction d'un nouveau problème. Ces processus exploitent les connaissances du système pour atteindre leurs objectifs.

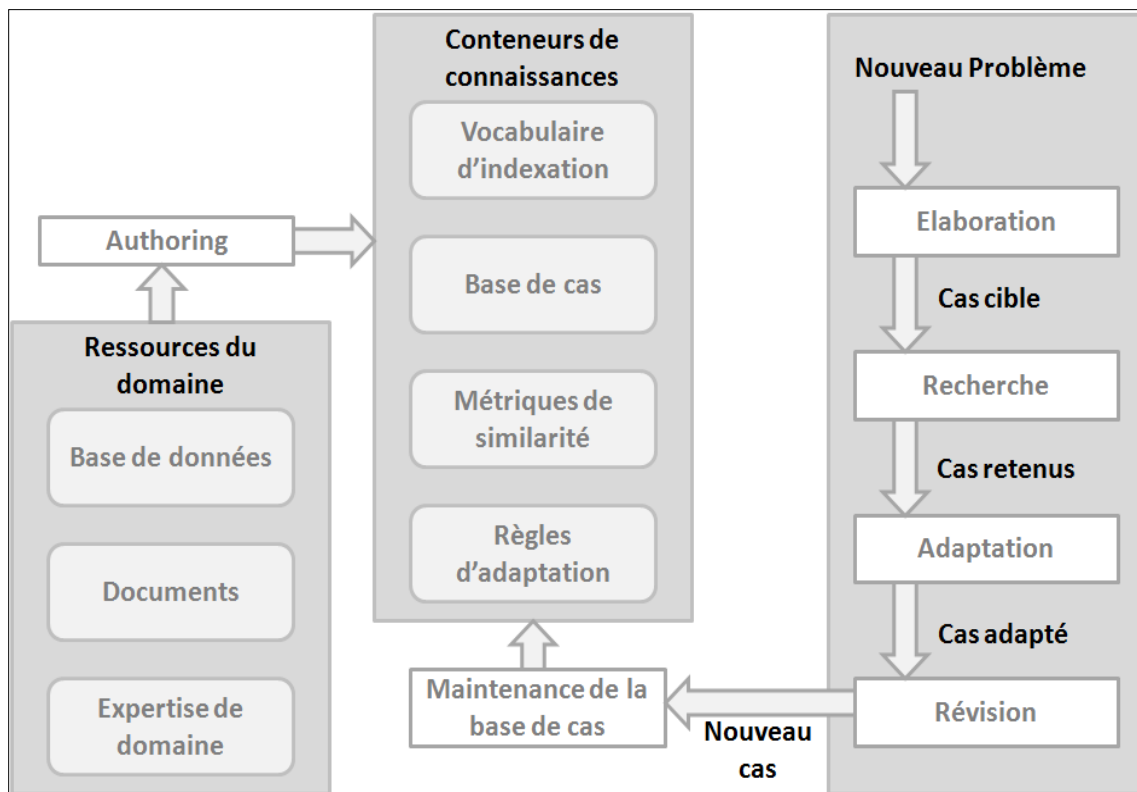


Figure 18. Fonctionnement d'un système RàPC

L'objectif de l'étape de recherche est d'identifier les cas utiles pour le raisonnement. L'utilité est mesurée par rapport à une mesure de similarité entre les indices du cas cible et ceux des cas sources ou/et sur des critères d'adaptabilité (Jaczynski 1998). Généralement, l'étape de recherche se compose d'une succession de sous-étapes de filtres (Jaczynski 1998). Chacune de ces sous-étapes prend des cas sources issus de la sous-étape d'avant et sélectionne parmi eux les cas les plus utiles. Pour atteindre ses objectifs, les sous-étapes de filtrage se base sur la notion d'index qui définit une organisation des cas sources au sein de la mémoire en utilisant leurs indices (Jaczynski 1998). Un index est une structure de données exploitée par les algorithmes de filtrage pour assurer une recherche plus efficace. Les algorithmes de filtrage exploitent également des connaissances du domaine ou/et les connaissances permettant d'identifier les cas les plus utiles. Comme exemple d'index, nous pouvons prendre celui par plus proches voisins qui s'appuie sur une structure linéaire de cas et sur une mesure de similarité.

Il est peu probable de trouver un cas qui correspond exactement au problème courant. Une adaptation des cas extraits par l'étape de recherche est nécessaire. L'étape de réutilisation permet donc d'adapter les solutions des cas extraits, afin de construire une solution au problème courant. Les méthodes d'adaptation à appliquer sont choisies par rapport aux différences entre les cas sources extraits et le problème courant.

La phase de révision consiste à évaluer la solution déterminée par l'étape de la réutilisation. Cette évaluation peut être effectuée manuellement, par un expert du domaine par exemple, ou automatiquement par un programme dédié. Si la solution est non-valide, des recommandations peuvent être fournies afin d'améliorer cette solution.

Dans l'étape de la maintenance, le système enrichit sa base de connaissances en exploitant le problème résolu et sa solution. En effet, si le cas cible est considéré comme intéressant, une méthode d'apprentissage détermine ses indices afin de définir dans quelles situations il sera applicable.

2.3 Modèles du RàPC

Selon Lamontagne et Lapalme ([Lamontagne and Lapalme 2002](#)) et Bergmann ([Bergmann 2003](#)), les modèles RàPC sont regroupés en trois grandes familles : structurelle, conversationnelle et textuelle.

2.3.1 Modèle structurel

Le modèle structurel est considéré comme le modèle le plus simple. Dans ce modèle, les cas sont structurés et représentés en attributs de type simple tel que les entiers, les réels, les booléens ou bien encore les caractères. Cette structuration peut être effectuée sur un seul niveau ou d'une façon hiérarchique ([Lamontagne and Lapalme 2002](#)).

La mesure de similarité entre deux cas correspond à une somme pondérée des distances entre les valeurs de chaque attribut. La pondération est nécessaire car généralement les attributs de la partie problème n'ont pas la même importance. Ainsi, la mesure euclidienne et celle de Hamming ([Lamontagne and Lapalme 2002](#)) sont utilisées souvent pour mesurer les distances.

2.3.2 Modèle conversationnel

Le modèle conversationnel permet de décrire le problème progressivement en se basant sur des interactions, entre le système et l'utilisateur, guidées par des questions prédéfinies, à l'opposé du modèle structurel qui exige une description complète du problème avant de commencer l'étape de remémoration ([Lamontagne and Lapalme 2002](#)).

La capacité à définir un problème progressivement fait de ce modèle le choix le plus pertinent pour les applications dont les problèmes sont difficiles à déterminer à l'avance. Parmi ces applications, nous trouvons celles dédiées aux services clientèle ([Lamontagne and Lapalme 2002](#)).

En plus de la partie problème et la partie solution, un cas conversationnel contient une autre partie qui se compose d'une série de questions et de réponses, et qui permettent d'obtenir plus d'informations sur le problème.

2.3.2 Modèle textuel

Nous parlons du modèle textuel lorsque les cas sont décrits textuellement. Il existe 2 types des cas textuels ([Lamontagne and Lapalme 2002](#)) :

1. Les cas semi-structurés, lorsque le cas est découpé en un ensemble d'attributs dont au moins un attribut pertinent pour la remémoration est de type textuel.
2. Les cas non-structurés, qui contiennent un seul attribut de type textuel qui englobe le problème et la solution.

Pour la recherche de cas similaires au problème courant, les systèmes qui implémentent le modèle textuel se basent sur : 1) la structuration de cas textuels en un ensemble d'attributs (concepts, mots-clés, catégorie, etc.) afin de profiter des techniques attachées au modèle structurel, ou sur 2) des modèles de recherche d'information ([Lamontagne and Lapalme 2002](#)). La première approche est valide pour les applications dont le domaine est restreint, alors que la deuxième est intéressante pour les applications qui veulent conserver une indépendance par rapport au domaine d'application.

2.4 Avantages et inconvénients du RàPC

Le RàPC a des nombreux avantages. Ces avantages portent essentiellement sur l'acquisition des connaissances, leur maintenance, l'efficacité et la qualité du raisonnement ainsi que son explication ([Jaczynski 1998](#)).

Dans le cadre de l'acquisition des connaissances, le RàPC exige en général moins d'efforts que l'établissement d'une base de règles. En effet, les acteurs de certaines applications raisonnent explicitement en utilisant des expériences passées ([Jaczynski 1998](#)). Ainsi, les informations constituant les cas sont plus faciles à expliciter que des règles générales.

De plus, le RàPC offre la possibilité de maintenance des connaissances en permettant l'ajout et le retrait des cas de la base de cas ([Jaczynski 1998](#)). Par exemple, une base initiale de cas doit être élaborée avant de mettre en place le système, après et selon les besoins de raisonnement, de nouveaux cas peuvent être ajoutés pour enrichir les connaissances du système.

Le RàPC peut assurer également une grande efficacité du processus de résolution de problème ([Jaczynski 1998](#)). En effet, dans un système à base de règles par exemple, les mêmes règles doivent être déclenchées. Cependant, dans un système RàPC, les échecs passés et l'utilisation des expériences passées réelles permettent d'améliorer le raisonnement.

Enfin, comme les cas sources sur lesquels se base un système RàPC sont des expériences vécues, l'explication ou la justification du raisonnement dans ces systèmes est beaucoup plus facile que dans des systèmes à base de règles comme les systèmes experts ou des systèmes

apprentis comme les réseaux de neurones ([Jaczynski 1998](#)). Ainsi, le raisonnement de ces systèmes est compréhensible par l'utilisateur, de ce fait, l'implication de ce dernier est possible dans un système RàPC interactif.

Cependant, le RàPC a également quelques inconvénients et conditions d'application. Tout d'abord, l'espace des problèmes ne peut pas être entièrement couvert, le système ne peut alors pas, proposer la solution optimale ([Jaczynski 1998](#)). Parfois, il ne peut même pas trouver une solution déterminée, il peut alors seulement proposer un ensemble de solutions possibles ([Devèze and Fouquin 2004](#)). De plus, la capitalisation de connaissances et d'expériences nécessitent l'intervention et la mobilisation d'experts du domaine cible. La principale difficulté, ici, réside dans le fait de motiver les experts à partager leurs connaissances surtout dans des entreprises où les experts considèrent leurs savoir-faire comme personnels ([Dieng, Corby et al. 2001](#)). Ainsi, l'application du RàPC repose sur la satisfaction de quelques conditions ([Jaczynski 1998](#)) :

- Régularité : les mêmes actions effectuées dans les mêmes conditions doivent avoir des effets proches voir identiques.
- Typicalité : les problèmes se répètent et les expériences passées peuvent être utiles pour résoudre les futurs problèmes.
- Cohérence : de petits changements dans la structure des cas ou des indices ne nécessitent que de petites modifications au niveau du raisonnement.
- Facilité d'adaptation : les différences entre les solutions de problèmes proches doivent être petites et peuvent être facilement prises en compte.

Une modélisation trop simpliste des connaissances d'un système RàPC ne permet pas souvent de satisfaire ces conditions. Il faut élaborer une modélisation suffisamment abstraite pour les cas et pour les indices ([Jaczynski 1998](#)). La section suivante présente notre conception de la couche RD.

3. Conception de la couche RP

Dans cette section, nous présentons la conception de la couche RP appliquée à la CoP MAETIC. Dans ce cadre les problèmes à résoudre sont appelés des difficultés MAETIC et les membres de la CoP sont des enseignants usagers de la méthode MAETIC.

3.1 Conception fonctionnelle

Le cas, selon le RàPC, est un couple (problème, solution). Dans le cadre de la couche RP, nous concevons la partie « problème » d'un cas comme une difficulté MAETIC et la partie

« solution » comme sa solution. La figure 19 ci-dessous présente le cycle du RàPC implémenté par la couche RD.

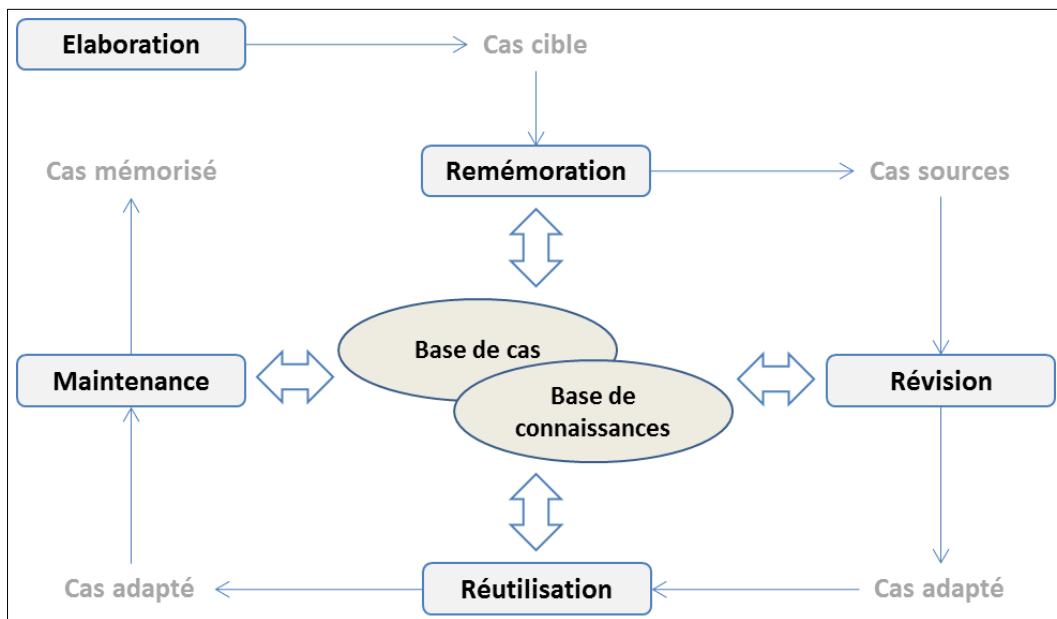


Figure 19. Le cycle du RàPC implémenté par la couche RP

En effet, la couche RP exploite les cinq processus du RàPC :

- L'étape **d'élaboration**, consiste à construire le cas cible et à initialiser sa partie « problème » à partir des informations fournies par l'enseignant sur la difficulté rencontrée.
- L'étape de **remémoration**, consiste à extraire les cas sources représentant des difficultés similaires à celle du cas cible.
- L'étape de **réutilisation**, permet à l'enseignant de concevoir une solution, soit en utilisant les cas extraits, soit en provoquant une discussion avec les membres de la communauté, si les cas extraits ne satisfassent pas ses besoins. La discussion entre les membres de la communauté est assurée par le noyau communautaire.
- L'étape de la **révision** (l'évaluation), implique des experts de la méthode MAETIC qui évaluent et valident la solution conçue par l'enseignant.
- L'étape de **maintenance**, permet d'ajouter la difficulté et sa solution validée dans la base des cas.

L'intégration de la couche RP au noyau du système NICOLAT permet d'implanter un nouveau processus de résolution de difficultés en plus de l'échange communautaire fourni par le noyau. Ce processus intègre les différentes étapes du RàPC citées ci-dessus, qu'il implémente en se basant sur les interactions entre les membres, et ce, pour assurer une

meilleure réutilisabilité des difficultés résolues dans le passé. La figure 20 ci-dessous présente le diagramme de séquence décrivant ce processus.

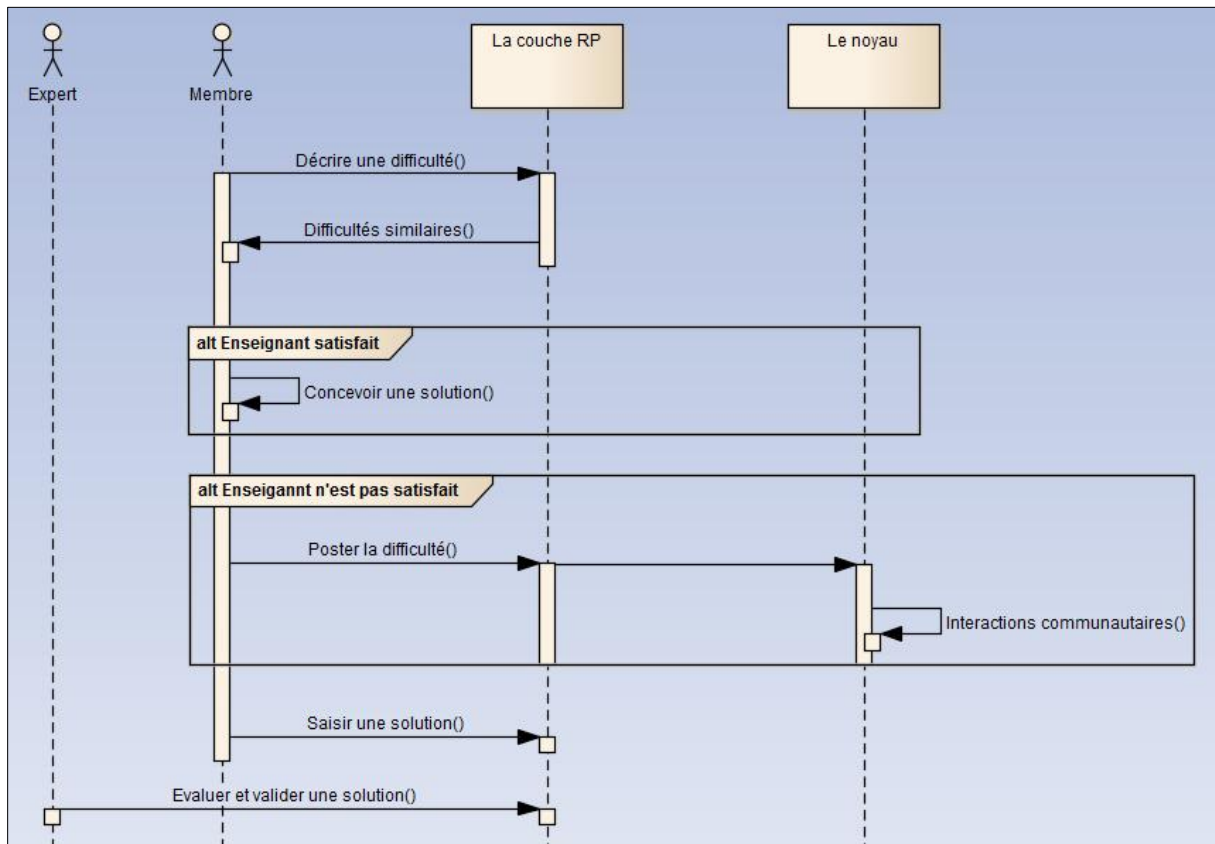


Figure 20. Processus de résolution de difficultés MAETIC

En effet, le membre cherchant à résoudre sa difficulté initie le processus de résolution de cette difficulté en la décrivant auprès de la couche RP. Cette dernière lui propose, par la suite, un ensemble de difficultés similaires à la sienne avec leurs solutions. Si l'enseignant est satisfait, il réutilise ces solutions pour concevoir la sienne, sinon, il poste sa difficulté sur le noyau communautaire et échange avec les autres membres pour arriver à une solution satisfaisante. Une fois la difficulté résolue, l'enseignant saisit la solution auprès de la couche RP. Cette solution sera évaluée et validée par un des experts de la méthode MAETIC. La validation de la solution mène à l'ajout du cas cible (la difficulté cible et sa solution) à la base de cas.

Pour assurer ce scénario, la couche RD offre des fonctionnalités qui complètent celles du noyau communautaire. Celles-ci sont présentées par le diagramme de cas d'utilisation illustré par la figure 21 ci-dessous.

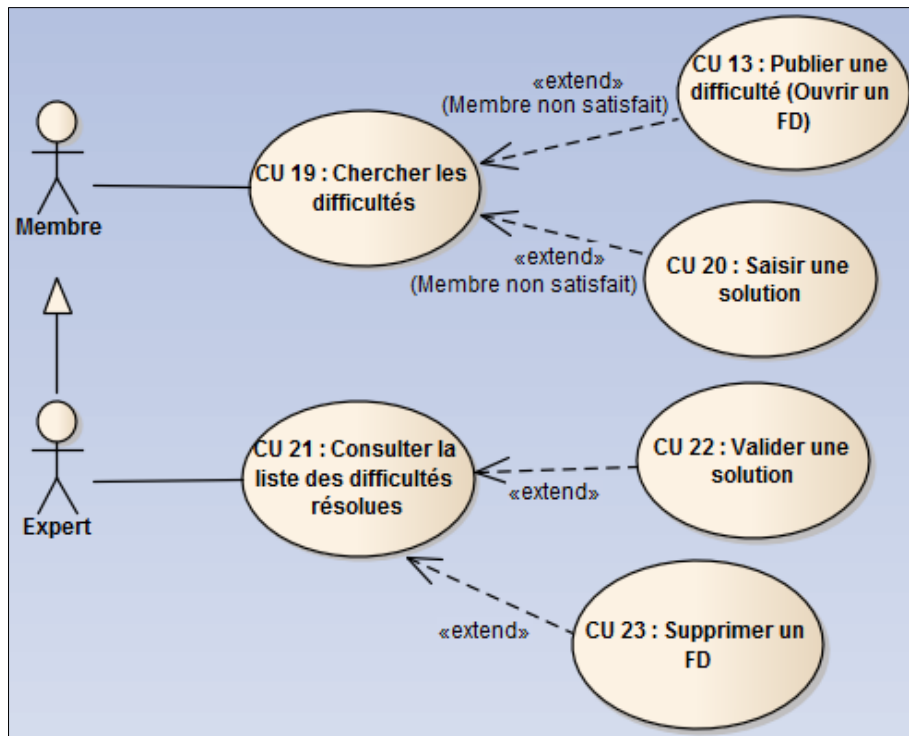


Figure 21. Le diagramme de cas d'utilisation de la couche RP

Ainsi, comme présenté sur la figure, un membre ou un expert a la possibilité de :

- Décrire une difficulté et recevoir comme résultat les difficultés similaires à sa difficulté (CU 19). Ce cas regroupe les étapes d'élaboration et de remémoration du RàPC.
- Poster sa difficulté sur le noyau s'il n'est pas satisfait (CU 13).
- Saisir une solution à sa difficulté dans le cas de satisfaction (CU 20). Ce cas d'utilisation correspond à l'étape de réutilisation, et la difficulté sera marquée comme résolue.

Un membre expert a en plus la possibilité de consulter la liste des difficultés résolues (CU 21) et les valider (CU 22) ou les supprimer (CU 23). Ces 3 cas d'utilisation (CU 21, CU 22, CU, 23) correspondent à l'étape d'évaluation. Après la validation, la difficulté et sa solution s'ajoutent automatiquement dans la base de cas (étape de maintenance).

Dans le paragraphe suivant, nous présentons la structure de cas que nous adoptons dans notre implémentation du RàPC.

3.2 Structure d'un cas

La description d'une difficulté MAETIC peut contenir des remarques, des explications, des questions composées, etc. Il est donc nécessaire d'utiliser un attribut textuel comme principal attribut décrivant la difficulté. En effet, il permettra de faciliter la compréhension de la

difficulté et d'éviter de perdre l'information. Cependant, les informations représentant la classe où la difficulté a été rencontrée, peuvent être structurées et représentées par des attributs indépendants. Chacun de ces attributs représente une information sur cette classe.

Ainsi, la classification des types de difficultés MAETIC, présentée dans le chapitre 1, peut être exploitée pour définir un attribut représentant le type de la difficulté par rapport à cette classification. Une difficulté peut donc être décrite par rapport à 3 types d'information : le type de la difficulté, sa description textuelle et des informations sur la classe. Nous regroupons ces informations en deux parties :

- **Pb-specif** : cette partie représente la spécification de la difficulté et contient, le type de la difficulté et sa description textuelle.
- **Pb-context** : cette partie représente le contexte où la difficulté a été rencontrée et contient des attributs caractérisant les informations sur la classe où la difficulté a été rencontrée.

Suite à un recueil d'expertise mené auprès d'experts de la méthode MAETIC afin d'identifier les caractéristiques des difficultés MAETIC, nous avons identifié 5 attributs qui peuvent représenter la classe où la difficulté a été rencontrée. Ces attributs sont :

- **Effectif** : le nombre d'étudiants dans la classe.
- **Nombre de groupes** : le nombre de groupes d'étudiants défini pour déployer MAETIC.
- **Niveau** : Le niveau d'enseignement (DUT, Licence, Master, Ingénierie).
- **Nombre d'heures global** : le nombre total d'heures dédiées au module.
- **Nombre d'heures par séance** : le nombre d'heures dédiées à chaque séance.

Ainsi, la figure 22 ci-dessous présente la partie problème de la structure d'un cas représentant les difficultés MAETIC.

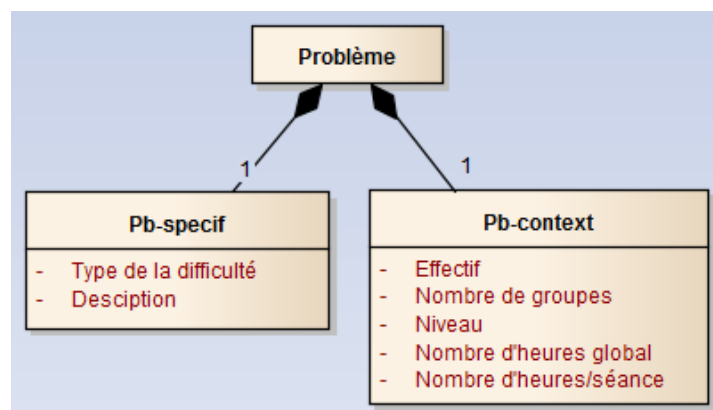


Figure 22. La partie problème de la structure d'un cas représentant une difficultés MAETIC

La solution d'une difficulté MAETIC consiste en un ensemble d'actions à effectuer. Ainsi, elle peut contenir des remarques, des explications comme elle peut contenir plusieurs alternatives de solutions dont chacune est adaptée à une situation spécifique. Nous représentons cette solution par un seul attribut textuel que nous appelons « actions solution ». Ainsi, pour permettre à l'enseignant qui a mis en place cette solution d'exprimer sa satisfaction de la solution et de décrire ses effets, nous ajoutons un attribut textuel que nous appelons « effets solution ». Enfin, pour permettre aux experts de la méthode MAETIC d'évaluer la solution proposée, donner des remarques ou proposer des améliorations, nous utilisons un attribut textuel que nous appelons « eval-sol ». La partie solution de la structure de cas représentant les difficultés MAETIC se compose donc de 3 attributs textuels :

- Sol-actions : cette partie représente les actions à effectuer par l'enseignant pour résoudre la difficulté.
- Sol-effets : cette partie représente les effets des actions de la solution.
- Sol-eval : cette partie représente l'évaluation de la solution par des experts.

La figure 23 ci-dessous présente la structure d'un cas d'une difficulté MAETIC.

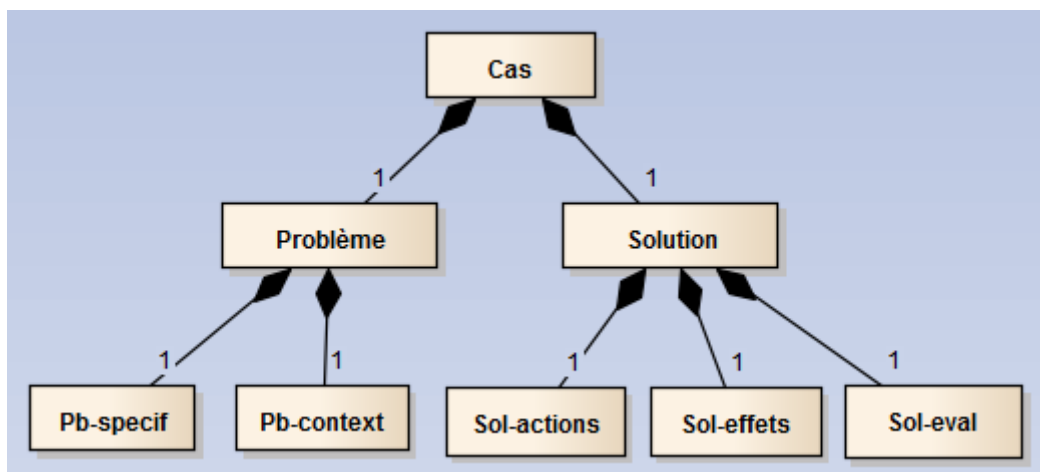


Figure 23. Structure d'un cas d'une difficulté MAETIC

Lors de la publication d'une difficulté, sa partie « Problème » est utilisée pour chercher les difficultés similaires résolues dans le passé. Cette opération de recherche correspond au processus de remémoration du cycle RàPC. Ce dernier représente le principal processus du RàPC effectué automatiquement par notre système. Nous présentons ce processus dans la section suivante.

4. Processus de remémoration de la couche RP

La partie « Problème » de la difficulté se compose d'un attribut « type de difficulté », de sa description textuelle et d'un ensemble d'attributs de type simple caractérisant la classe où la

difficulté a été rencontrée. En fait, chacun de ces attributs caractérisant la classe possède une liste prédéfinie de valeurs possibles. L'exploitation des attributs de la partie « Problème » par le processus de remémoration consiste à utiliser une combinaison entre les modèles textuel et structurel du RàPC. En effet, ce processus de remémoration se compose de deux phases :

- **La recherche de cas similaires** : dans notre cas, cette première phase exploite l'attribut « type de la difficulté » et sa description textuelle pour la recherche de cas similaires. L'utilisation d'un attribut textuel pour la recherche nous met devant le modèle textuel du RàPC. Ainsi, un modèle de recherche d'information est utilisé pour évaluer la similarité entre les valeurs de cet attribut.
- **Le tri des cas sélectionnés** : cette seconde phase consiste à trier les cas retournés par la première étape afin de les présenter à l'enseignant. Elle exploite tous les attributs de la partie « Problème ». L'exploitation des attributs de la classe, qui sont des attributs de type simple peut être vue comme une implémentation du modèle structurel du RàPC. En fait, le tri se base sur un calcul de distances entre les valeurs des attributs représentant les difficultés retournées et celles représentant la difficulté cible.

Nous argumentons notre choix de n'utiliser que les attributs « Pb-specif » pour la recherche, par le fait que l'utilisation de tous les attributs de « Problème » peut retourner des cas sources qui sont très proches du cas cible au niveau de contexte, et très loin au niveau de la difficulté décrite. Par exemple, si D1 et D2 deux difficultés différentes rencontrées dans le même contexte, ces deux difficultés ont alors des valeurs similaires concernant les attributs « Pb-context » et des valeurs différentes concernant l'attribut « Description de la difficulté ». Si la couche RD se base sur tous les attributs de « Problème » pour comparer D1 et D2, elle les juge comme similaires car la différence au niveau de l'attribut « Description de la difficulté » entre ces 2 cas sera ignorée devant les similitudes des valeurs des autres attributs qui sont les attributs du contexte.

4.1 Recherche de cas similaires

Nous proposons que la première phase du processus de remémoration, celle de la recherche des cas similaires, sélectionne les cas similaires en n'utilisant que les attributs de « Pb-specif » : type de la difficulté et sa description textuelle. En effet, la description de la difficulté exprimée en langage naturel est plus pertinente pour comparer deux cas.

Pour rechercher les cas similaires, le système commence par sélectionner les cas ayant le même type de difficulté que celui de la difficulté cible. Ensuite, il mesure la similarité textuelle entre la valeur de l'attribut « Description de la difficulté » de ces cas source et du cas cible.

4.1.1 Techniques de calcul de la similarité textuelle

L'évaluation de similarité entre les textes se base sur les modèles de Recherches d'Information (RI) ([Baziz 2005](#)). Les principaux modèles de RI sont :

- Le modèle booléen : ce modèle est considéré comme le modèle de RI le plus ancien et le plus simple. Il propose de représenter une requête sous forme d'une expression logique dans laquelle des termes d'indexation sont reliés par des connecteurs logiques ET, OU et NON. C'est un modèle exact, c'est à dire que la notion de pertinence est binaire. Un document est soit pertinent soit non pertinent. L'inconvénient est qu'il ne sélectionne que les documents qui correspondent idéalement à la requête. Dans ce modèle, les termes ont la même importance et il est donc incapable de présenter le degré de pertinence de chaque document par rapport à la requête. Il est aussi difficile d'exprimer des requêtes longues.
- Le modèle vectoriel : dans ce modèle, la requête et les documents sont représentés dans un espace vectoriel dont les dimensions sont les termes d'indexation. L'espace est de N dimensions, avec N le nombre de termes d'indexation. Chaque document ou requête est représenté par un vecteur où chaque élément représente le poids d'un terme dans ce document/requête. La pertinence d'un document par rapport à une requête se définit par rapport à une mesure de similarité entre leurs vecteurs. Parmi les mesures de similarité qui peuvent être utilisées, nous trouvons : la mesure cosinus, la mesure de Jaccard, le produit scalaire ([Baziz 2005](#)), etc. Ce modèle se considère comme le plus répandu en recherche d'information. En effet, il présente le même modèle de représentation pour la requête et les documents. L'inconvénient de ce modèle est qu'il considère que tous les termes sont indépendants.
- Le modèle probabiliste : ce modèle utilise un modèle mathématique basé sur la théorie de la probabilité ([Baziz 2005](#)). Le processus de recherche consiste à chercher la probabilité de pertinence d'un document par rapport à une requête. Ce modèle se base sur le calcul des probabilités conditionnelles. Le principal inconvénient de ce modèle est qu'il nécessite l'estimation des probabilités initiales. Ainsi, il considère que tous les termes sont indépendants.

4.1.2 Notre approche de calcul de la similarité textuelle

Dans notre cas, la requête et le document sont des descriptions de difficultés MAETIC. Dans ce cadre, nous adoptons le modèle vectoriel qui présente une uniformité entre les représentations de requête et du document. Ce modèle ne nécessite pas l'estimation de valeurs de certains paramètres pour son fonctionnement et permet donc de garder une certaine indépendance par rapport au domaine d'application.

La mesure de similarité textuelle consiste à représenter des textes sous forme de vecteurs et à définir une fonction mathématique qui permet de calculer la similarité entre ces vecteurs. Les dimensions de l'espace vectoriel utilisé pour représenter ces vecteurs, sont associées à des unités linguistiques. Trois types de représentations des unités linguistiques sont possibles : celle par mots, par N-gramme ([Shannon 1948](#)) ou par groupe de mots ([Lewis 1992](#)).

Nous adoptons la représentation des unités linguistiques par des mots. En fait, une unité linguistique correspond à un mot. Nous avons éliminé celles par groupe de mots parce que les descriptions de nos difficultés sont de petites tailles ce qui engendrerait des fréquences trop faibles pour être exploitables. Celle par N-gramme a été éliminée à cause du grand temps d'exécution qu'elle engendre en comparaison avec la représentation par mots que nous adoptons.

Notre approche pour le calcul de similarités entre des descriptions de difficultés se base, donc, sur le mot comme unité linguistique et le modèle vectoriel. Dans ce qui suit, nous présenterons les deux principaux processus du calcul de similarité textuelle qui sont : l'indexation, et la mesure de similarité.

Avant d'aborder ces deux processus, nous définissons :

- $BC = \{c_1, c_2, \dots, c_k\}$: la base des cas initiale, où k est le nombre de cas.
- $BD = \{d_1, d_2, \dots, d_k\}$: l'ensemble des descriptions textuelles de BC , tel que d_i est la description textuelle du cas c_i .

4.1.3 Indexation

Nous décomposons le processus d'indexation en 2 sous-processus : **l'extraction des termes d'indexation** et la **vectorisation des descriptions de difficultés**. Le premier sous-processus modifie alors les descriptions de manière à faciliter leur usage lors du second.

L'extraction des termes d'indexation se compose de :

- Tokénisation, qui consiste à découper les textes en unités linguistiques (mots).
- Élimination des « stop words », qui consiste à éliminer les termes qui sont inutiles à indexer ou à utiliser dans une recherche. Par exemple : « le », « la », « de », « du », « ce », « ça », etc.
- Lemmatisation, qui permet de réduire les formes d'un mot en un seul, appelé lemme. Par exemple : « petit » est le lemme de : « petit », « petite », « petits » et « petites ». Elle permet un gain important de temps de calcul.

Ce premier sous-processus produit un ensemble de termes. Cet ensemble sera finalisé manuellement en ajoutant ou en éliminant certains termes. L'ensemble final se compose d'un

ensemble de termes qu'on appelle : les termes d'indexation. Nous notons cet ensemble de termes :

- $T = \{t_1, t_2, \dots, t_n\}$, où n est le nombre de termes d'indexation.

La vectorisation des descriptions de difficultés (second sous-processus) représente les descriptions de manière à faciliter leur exploitation pour mesurer les similarités entre elles. Dans celui-ci, nous représentons les descriptions de difficultés par des vecteurs v_i dans un espace vectoriel dont les dimensions sont les termes d'indexation retenus après le premier sous processus. La dimension de cet espace correspond au nombre de termes d'indexation. À chaque élément de v_i correspond un poids w_{ji} qui représente le poids du terme t_j dans d_i . Ce poids peut se représenter de plusieurs manières : représentation booléenne, fréquentielle ou représentation TF-IDF ([Salton and Buckley 1988](#)).

La représentation booléenne marque la présence ou l'absence d'un mot. Elle ne prend pas en compte le nombre d'apparition d'un terme dans une description. Or, certains mots peuvent avoir plus d'importance par rapport à leur intérêt alors que d'autres peuvent avoir un intérêt réduit.

La représentation fréquentielle permet de mesurer l'importance d'un mot en se basant sur son nombre d'apparitions dans une description. La représentation TF-IDF garde le même principe de la représentation fréquentielle en mettant en jeu un facteur IDF qui sert à donner plus d'importance aux mots qui apparaissent moins fréquemment. En effet, un terme qui revient trop souvent dans les descriptions de difficultés n'est pas significatif. C'est pourquoi, dans notre cas, nous choisissons d'utiliser la représentation TF-IDF par rapport à la représentation fréquentielle.

Selon cette représentation, un poids w_{ji} se calcule comme suit (1) :

$$w_{ji} = tf_{ji} * idf_j \quad (1) \text{ où}$$

tf_{ji} est la fréquence d'apparition du terme t_j dans d_i .

$idf_j = \log(N/n_j)$, où N est le nombre de d_i (dans notre cas $N=k$), et n_j le nombre des descriptions d_i qui contiennent t_j . idf_j représente l'inverse de la fréquence de t_j dans D .

La Figure 24 ci-dessous résume le processus d'indexation.

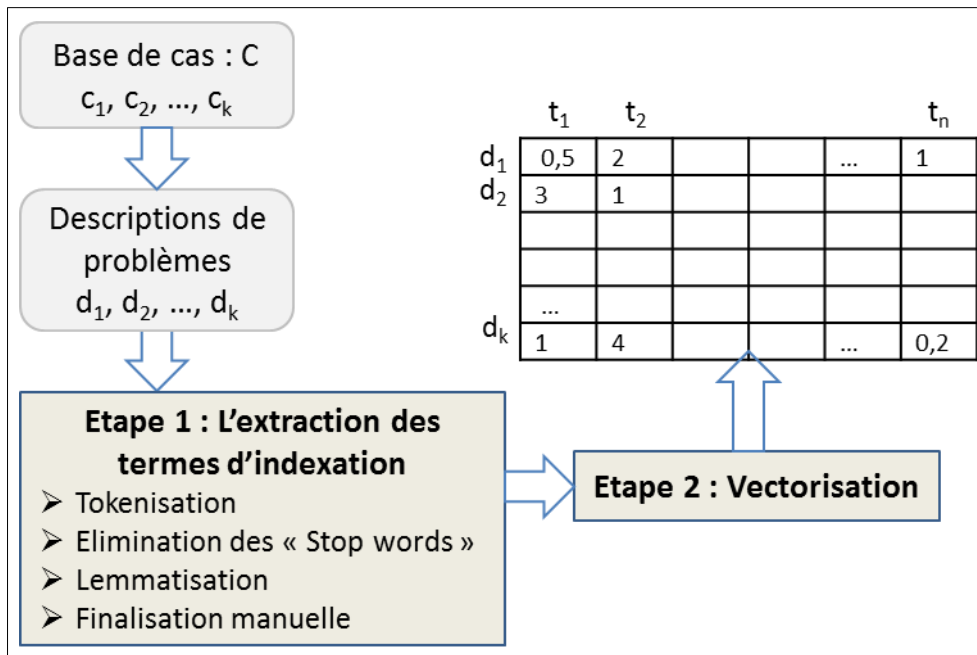


Figure 24. Résumé du processus d'indexation

Dans le paragraphe suivant, nous présentons la mesure de similarité que nous utilisons pour comparer les vecteurs décrivant les descriptions de difficultés.

4.1.4 Similarité entre les cas

Pour mesurer la similarité entre deux descriptions de difficultés données, il faut calculer la similarité entre leurs vecteurs associés. Plusieurs mesures qui permettent de calculer la similarité entre deux vecteurs sont possibles (la mesure de cosinus, la mesure de Jaccard, etc.) ([Baziz 2005](#)). Nous avons choisi la mesure cosinus ([Salton 1971](#)), qui, au contraire des autres mesures, ne tient pas compte de la longueur des vecteurs. Ce qui voudrait dire dans notre cas, que deux descriptions de la même difficulté seront jugées similaires indépendamment de leurs nombres de mots.

Soient d_x et d_y deux descriptions de difficultés, et v_x et v_y leurs vecteurs associés. Pour le calcul de similarité entre d_x et d_y , nous calculons la similarité cosinus entre v_x et v_y par la formule suivante (2) :

$$\text{simText}(d_x, d_y) = \cos(v_x, v_y) = v_x \cdot v_y / \|v_x\| \cdot \|v_y\| \quad (2)$$

Plus $\text{simText}(d_x, d_y)$ est grande, plus les descriptions sont similaires. Au contraire, plus $\text{simText}(d_x, d_y)$ est petite, plus elles sont différentes. On nomme $\text{simText}(d_x, d_y)$ la similarité textuelle entre d_x et d_y .

La seconde phase de l'étape de remémoration consiste à trier les cas retournés par la phase de recherche des cas similaires. Le tri s'effectue en se basant sur les attributs représentant la classe et celui de la description textuelle.

4.2 Tri des cas sélectionnés

Les attributs de la classe sont de types simples, ce qui rend facile le calcul de la distance entre les valeurs de ces attributs. Pour calculer la distance entre deux valeurs d'un de ces attributs, nous utilisons une distance binaire, donc si les 2 valeurs sont égales, la distance est égale à 1 sinon la distance est égale à 0. Nous notons chacun de ces attribut a_i . La valeur de l'attribut a_i pour une difficulté D_x est notée : $val(D_x, a_i)$. La distance entre la valeur de l'attribut a_i de la difficulté D_x et celle de la difficulté D_y est comme suit :

$$dsBin(val(D_x, a_i), val(D_y, a_i)) = 1 \text{ si } val(D_x, a_i) = val(D_y, a_i)$$

$$dsBin(val(D_x, a_i), val(D_y, a_i)) = 0 \text{ si } val(D_x, a_i) \neq val(D_y, a_i)$$

Pour l'attribut « Description de la difficulté » qui est du type textuel, nous considérons les similarités textuelles calculées dans la phase 1 comme distances entre les valeurs de cet attribut :

$$dsText(D_x, D_y) = simText(d_x, d_y)$$

avec d_x est la description textuelle de D_x , et d_y est la description textuelle de D_y .

Pour calculer la similarité entre le cas cible et un cas source ou ce que nous appelons d'indice de tri, nous commençons par le calcul de la distance entre les valeurs des mêmes attributs. Ensuite, nous calculons la similarité globale entre ces deux cas par une somme pondérée de la similarité de chacun des attributs. Comme les attributs n'ont pas tous la même importance, nous attribuerons un poids à chaque attribut de la partie « Problème » du cas en fonction de propositions qui sont données par les experts de MAETIC. Nous définissons donc un vecteur P englobant le poids de chaque attribut :

$$P = \{p_0, p_1, \dots, p_n\}$$

P_0 est le poids de l'attribut 'description de la difficulté'.

$\{p_1, \dots, p_n\}$ sont les poids des attributs de la classe, avec n le nombre de ces attributs.

Ainsi, la formule permettant de calculer l'indice de tri est la suivante :

$$\begin{aligned} & indiceTri(D_x, D_{cible}) \\ &= \frac{p_0 \cdot dsText(D_x, D_{cible}) + \sum_{i=1}^n p_i \cdot dsBin(val(D_x, a_i), val(D_{cible}, a_i))}{\sum_{i=0}^n p_i} \end{aligned}$$

Nous trions les cas selon un ordre décroissant de leurs indices de tri avec le cas cible.

La Figure 25 ci-dessous résume l'étape de remémoration employée dans la couche RP.

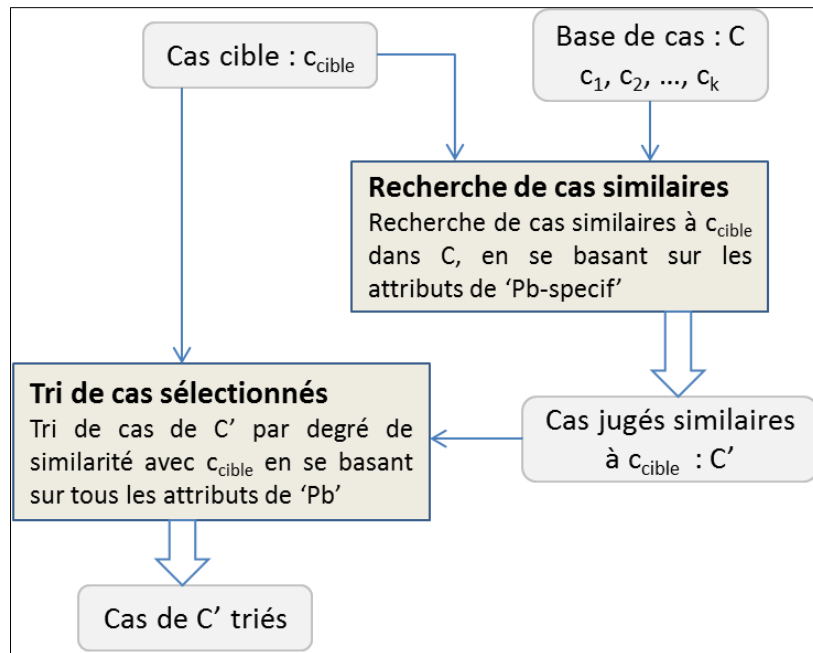


Figure 25. Etape de remémoration

Cette figure montre que le processus de remémoration se compose de deux phases. La première consiste à extraire, de la base de cas, les cas dont les difficultés sont jugées similaires à la difficulté cible. Cette phase se base sur les attributs de la partie « Pb-spécif » : le type de difficulté et la description textuelle. La deuxième phase consiste à trier les cas extraits avant de les présenter à l'enseignant cherchant à résoudre sa difficulté. Le tri se base sur l'attribut « Description de la difficulté » et les attributs de la classe.

5. Validation de la couche RP

La couche RP de notre système permet à un enseignant de construire une requête en décrivant sa difficulté. Cette couche répond à la requête en retournant une liste de difficultés résolues dans le passé et jugées satisfaisantes pour la difficulté cible avec leurs solutions. L'utilisateur peut soit choisir une solution à sa difficulté, soit décider de poster sa difficulté s'il n'est pas satisfait. Cette couche a été développée sous forme d'un module Drupal que nous avons intégré au noyau communautaire déjà réalisé.

Pour valider la couche RP, nous avons procédé à un test sur une base initiale de 10 cas. Nous nous focalisons sur les étapes d'élaboration et de remémoration. En fait, les étapes restantes sont manuelles. Dans ce qui suit, nous commençons par présenter la base initiale de cas sur laquelle nous avons effectué notre test. Nous détaillons, ensuite, l'étape d'élaboration. Enfin, nous précisons l'étape de la remémoration.

5.1 Base initiale de cas

Le tableau 3 ci-dessous présente la liste des cas, sans information de la classe, constituant notre base initiale de cas. Ce sont des cas réels et ils peuvent donc contenir des fautes de français ou de frappe.

Libellé	Type	Description / Actions de la solution
Cas-1	Pannes réseau	<p>Description : Des fois il y a des paramètres externes qui freinent le bon déroulement des séances: l'internet de qui ne marche pas de fois, certains sites notamment des blogs bloqués pour cause de mauvais paramétrage du logiciel de filtrage au niveau du serveur.</p> <p>Actions de la solution : J'ai rencontré ce problème plusieurs fois. Je conseille alors en effet soit la journalisation papier, soit un travail de préparation sous traitement de texte. Dès que l'Internet est accessible, ils injectent leurs travaux déjà préparés sur leur machines.</p>
Cas-2	Manque de compétences des étudiants	<p>Description : Les étudiants n'ont pas une très bonne culture en informatique. A la première étape c'était difficile de leur expliquer un Blog en plus qu'il savait créer un blog. Cela s'explique peut-être du fait que certains d'entre eux ne voient l'intérêt de faire tout ce travail puis qu'ils vont s'orienter vers les math en 3eme année.</p> <p>Actions de la solution : Lorsque les étudiants ne sont pas spécialistes en info... il faut s'abord se concentrer sur l'activité 1 et 2. Charte graphique et les rôles. Ensuite tu fais un ex de blog et comment on le gère. Cela peut te prendre 3h. Il faut se focaliser dessus. Ensuite à la séance suivante, tu fais créer le blog et seulement le blog et comme ils ont fait la charte graphique tu peux faire des recommandations sur la couleur et les catégories... et ensuite tu leurs fait faire la réponse à la demande. Il faut que tu gères au mieux la création du blog et insister sur la journalisation et laisser une trace de ce qu'ils font et ils s'en serviront pour la gestion du projet. et c'est important.</p>
Cas-3	Pré-lancement de MAETIC	<p>Description : Le nombre d'étudiants est très élevé pour une pédagogie de projet: jusqu'à 60 étudiants. J'étais obligé de scinder en 2 groupes, j'ai trouvé un peu fatiguant cette méthode. Dès demain je commencé une nouvelle: pour le groupe de mardi je déploie MAETIC, pour l'autre sans MAETIC. Pour voir après la différence entre les 2 méthodes.</p> <p>Actions de la solution : Oui. MAETIC peut être supporté pour 10 à 11 groupes d'étudiants de 4 étudiants donc au max 40 étudiants. C'est bien de les avoir coupés en deux. C'est ce que je t'avais dit. Fatiguant oui... au début, il faut être structuré. et ne pas laisser les étudiants se dispersés.</p>
Cas-4	Discours à tenir aux étudiants	<p>Description : Des fois les étudiants ne voient l'intérêt de blogs, de la constitution des groupes, du bilan de la séance. Comment leur expliquer d'une manière concise ces inquiétudes tout en se mettant à leur niveau.</p> <p>Actions de la solution : l'intérêt de blogs : => Traçabilité du projet et journalisation. on s'oblige à rendre compte et à se situer dans le temps et par rapport au travail des autres grâce aux bilans de séance : obligation professionnelle. => Partage des documents même à distance. => Suivi personnalisé et continu : car l'enseignant contrôle régulièrement le travail. Il peut détecter au plus tôt les erreurs dans la pratique.</p>
Cas-5	Liée à l'étape 3 : Planification	<p>Description : Dans la planification, j'ai demandé aux étudiants de détailler l'étape 4 en sous étapes correspondant au travail demandé dans le projet (élaborer un MCD, puis proposer un modèle relationnel, etc.). Maintenant je me demande si ces détailles ne vont pas plutôt dans le tableau de bord.</p> <p>Actions de la solution : Non, en étape 3 tu fais bien planifier les activités</p>

		<i>liées au projet : élaborer le MCD, traduire en relationnel, ... (qui seront faites en étape 4) : qui va le faire, durée, quand. Les tableaux de bord visent à "piloter" c'est à dire à contrôler l'avancement des tâches par le chef de projet par rapport au planning : retard pris, décisions par recaler le travail, % de travail fait, etc. : Cf. Dossier Pratique 4.</i>
Cas-6	Pré-lancement de MAETIC	Description : <i>je ne pouvais pas savoir en détail les objectifs visés du cours, demandés dans la fiche du scénario.</i> Actions de la solution : <i>Les objectifs visés du cours sont personnels à l'enseignant. Il suffit de vous posez la question « qu'est ce que je veux transmettre comme connaissance à mes étudiants ». On parle ici de connaissances domaines. S'il s'agit des connaissances liées à MAETIC.</i>
Cas-7	Pré-lancement de MAETIC	Description : <i>je ne pouvais pas prévoir à l'avance tout les séances avec précision ainsi que leurs activités et comment ces activités pouvaient être ajustées avec un temps prédéfinis.</i> Actions de la solution : <i>Globalement, une règle générale serait de L'étape 1 (se réalise en maximum 6h de présence). L'étape 2 et 3 se réalise en maximum 4h de présence. Si les étudiants n'ont pas fini, ils devront le faire en dehors des séances. L'étape 5 se réalise en 4h de présence (le reste en dehors des séances de présence). On prend donc : le nombre d'heures associés au TD. On retire + ou - 18h environ pour les étapes 1,2,3,5. On attribue le reste à l'étape 4. Si cela ne passe pas dans la totalité des heures de TD, on fait faire des activités de MAETIC en dehors des séances de présence attribuées pour le TD.</i>
Cas-8	Liée à une étape = ETAPE 1	Description : <i>L'étape 1 est très lourde à suivre dans son intégralité.</i> Actions de la solution : <i>Je proposerais dans certains cas de limiter cette étape en allégeant certaines parties (ex : charte graphique) pour se concentrer plus vite sur le produit à élaborer.</i>
Cas-9	Liée à l'organisation générale de MAETIC – Réalisation d'un Cr de réunion	Description : <i>J'ai demandé à mes étudiants de réaliser au moins un compte rendu de réunion. Cependant, je viens de parcourir le guide MAETIC en ligne et je n'ai pas vu de fiche concernant la rédaction de compte rendu réunion.</i> Actions de la solution : <i>La réalisation du compte rendu se situe dans le dossier pratique n°2 – Fiche Pratique n°3. Il suffit de leur laisser lire ce qui est écrit page 38 du livre MAETIC.</i>
Cas-10	Pré-lancement de MAETIC	Description : <i>Je dois faire du cours magistral en amphitheâtre mais je ne veux pas les faire suivre en cours magistral. Parce que les étudiants ne m'écoutent pas.</i> Actions de la solution : <i>0. On répartit les étudiants par groupe de 4 ou 5. Quand même car après ils auront à effectuer du MAETIC. En plus le groupe permet l'émulation et l'entraide.</i> <i>1. On peut leur faire lire des PPT en amphitheâtre et ensuite leur faire faire un résumé (une synthèse de ce qu'ils ont compris) et aussi des QCM sur ce qu'ils ont lu (20 questions par chapitre lu). 2. On fait un peu de cours magistral sur les notions importantes et le reste on fait lire par groupe... Ensuite on mixte et on fait refaire les QCM et on corrige les synthèses.</i>

Tab 3. Base initial de cas

5.2 Etape d'élaboration

Cette étape commence par la saisie d'une difficulté à travers le formulaire de description d'une difficulté. Nous saisissons, comme exemple, la difficulté suivante :

- Type : Pré-lancement de MAETIC
- Description : Mon module est divisé en cours magistraux amphi et des TDs, qu'est ce que je peux faire pendant les cours magistraux?
- Niveau : Master

Une seule information sur le contexte est saisie : le niveau. Les autres champs sont laissés vides. La figure 26 ci-dessous présente ces informations saisies à partir desquelles le système construit le « cas cible ». Dans ce cas, les attributs correspondants aux champs vides du formulaire auront la valeur 'null'.

Type

Type de la difficulté *

Difficulté Transversale ▼

Liée à l'organisation générale de MAETIC ▼

Pré-lancement de MAETIC ▼

Description

Titre de votre difficulté *

problème avec les cours magistraux

Description de la difficulté *

Mon module est divisé en cours magistraux amphi et des TDs, qu'est ce que je peux faire pendant les cours magistraux?

▼ informations sur la classe

Niveau :

Master ▼

Effectif :

▼

Nombre de groupes :

▼

Nombre d'heures

▼

Nombre d'heures par seance :

▼

Chercher une solution

Figure 26. Difficulté cible saisie dans le formulaire de description de difficulté

5.3 Etape de remémoration

Après la validation du formulaire par l'utilisateur, le système commence par éliminer les difficultés ayant un type différent à celui du cas cible. Il ne garde alors que les cas : CAS-3, CAS-6, CAS-7 et CAS-10.

Puis, le système mesure les similarités textuelles entre la description du cas cible et celles des cas CAS-3, CAS-6, CAS-7 et CAS-10. Les indices de similarité retournés sont présentés dans le tableau 4 ci-dessous :

Description du cas	Cas-1	Cas-6	Cas-7	Cas-10
Indice de similarité avec le cas cible	0	0.204	0	0.34

Tab 4. Indices de similarités textuelles entre les cas sources et le cas cible

Le système élimine ensuite, les cas dont les descriptions textuelles ont un indice de similarité égal à 0 avec la description du cas cible. Il ne garde alors que les cas : CAS-6, CAS-10 qui sont les cas à retourner à l'utilisateur. Le système doit ensuite effectuer son tri.

Pour effectuer l'opération de tri, le système utilise tous les attributs, sauf ceux qui ont une valeur 'null' dans le cas cible ou dans le cas source. Les attributs de contexte n'ont pas la même importance que l'attribut « Description de la difficulté ». Nous avons attribué à chaque attribut du contexte un poids égal à 0.2 et à l'attribut 'description de la difficulté' un poids égal à 1 : $P = \{0.5, 0.1, 0.1, 0.1, 0.1, 0.1\}$. Concernant la distance entre les valeurs d'un attribut de contexte, elle sera 1 si les valeurs sont les mêmes, sinon elle sera 0. Pour l'attribut « Description de la difficulté », nous utilisons l'indice de similarité déjà calculé comme distance. Les deux tableaux 5 et 6 ci-dessous présentent le calcul de l'indice de tri des cas : CAS-6 et CAS-10.

Attribut	Valeur de l'attribut		Distance	Poids	Distance * Poids
	Cas cible	CAS-10			
Description	-	-	0.34	1	0.34
Domaine	Master	Master	1	0.1	0.1
Indice de tri :					0.44

Tab 5. Calcul de des indices de tri des cas : « CAS-6 »

Attribut	Valeur de l'attribut		Distance	Poids	Distance * Poids
	Cas cible	CAS-6			
Description	-	-	0.204	1	0.204
Domaine	Master	Licence	0	0.1	0
Indice de tri :					0.204

Tab 6. Calcul de des indices de tri des cas : « CAS-10 »

Après le calcul des indices de tri, le système présente les cas à l'utilisateur en ordre décroissant selon cet indice. La figure 27 ci-dessous présente la liste des cas retournés par la couche RD.

The screenshot displays search results within a white-bordered container. At the top, the heading 'Résultats de la recherche' is followed by a blue link 'Cours magistral en amphi'. Below this, a text block reads: 'Je dois faire du cours magistral en amphi mais je ne veux pas les faire suivre en cours magistral.. Parce que les étudiants ne m'écoutent pas.' To the right of this text is a blue link 'Lire la suite'. A horizontal line separates this from the next section, which has the heading 'les objectifs visées du cours' and the text 'Je ne pouvais pas savoir en détaille les objectifs visées du cours, demandés dans la fiche du scénario'. To the right of this text is another blue link 'Lire la suite'. At the bottom of the container, there are four rounded rectangular buttons: 'Poster ma difficulté', 'taguer des membres', 'Saisir une solution', and 'Retour au formulaire de recherche'.

Figure 27. Liste de cas retournés par la couche RD

Ces résultats montrent que la couche RP permet de retourner un ensemble des cas (difficultés avec leurs solutions) dont les difficultés sont bien jugées similaires à celle saisie par l'enseignant. Le test de validation sur la base de 10 cas a permis de satisfaire nos attentes en matière de calcul de similarité.

6. Généricité de la couche RP

6.1 Conception fonctionnelle

Le processus de résolution de problème que nous avons proposé pour la CoP MAETIC implique des experts de cette méthode. Or, nous pouvons avoir des communautés dans lesquelles il n'y a pas de distinction entre les membres experts et non experts. Dans ce cadre, nous modifions ce processus pour assurer sa généricité. En effet, le membre cherchant à résoudre son problème initie le processus de résolution de ce problème en le décrivant auprès de la couche RP. Cette dernière lui propose, par la suite, un ensemble de problèmes similaires avec leurs solutions. Si le membre est satisfait, il réutilise ces solutions pour concevoir la

sienne, sinon, il poste son problème sur le noyau communautaire et échange avec les autres membres pour arriver à une solution satisfaisante. L'étape de la validation par des experts est optionnelle et dépend de la communauté cible. En effet, s'il n'est pas possible d'identifier les experts du domaine ou si les experts sont non-capables de juger les solutions proposées, cette étape est supprimée. Dans ce cas le problème et sa solution sont ajoutés directement dans la base de cas après la saisie de la solution. La figure 28 ci-dessous résume ce processus.

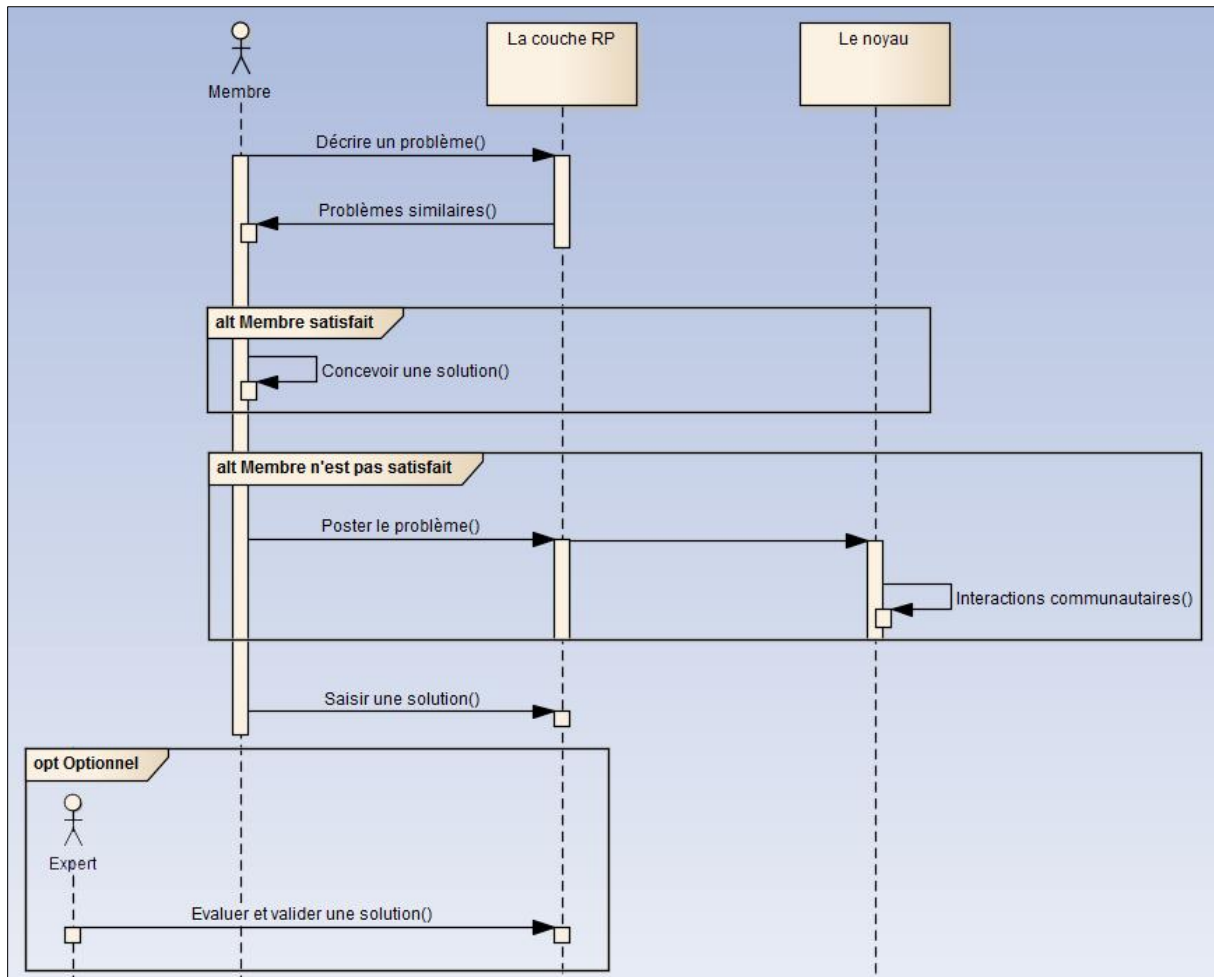


Figure 28. Processus de résolution de problèmes

6.2 Structure d'un Cas

Dans la section 5 du chapitre 2, nous avons proposé une structure générique de problèmes qui représente un problème par rapport à trois types d'entités : le type de problème, sa description textuelle et les attributs représentant son contexte. Ainsi, la couche RP présentée dans ce chapitre manipule des cas dont la partie problème (difficulté MAETIC) respecte cette structure. Elle utilise le type du problème et sa description textuelle au cours de la phase de sélection de l'étape de remémoration et les attributs du contexte pour la phase de tri. La

structure de problèmes et l'étape de remémoration du cycle RàPC ne nécessitent donc aucune modification si une autre CoP souhaite réutiliser cette couche.

Pour la structure de la solution, nous gardons l'entité « actions de la solution ». Cependant, nous rendons les entités « effets de la solution » et « évaluation de la solution » optionnelles. L'utilisation de ces dernières pour spécifier la solution dépend de la communauté cible. Ainsi, la figure 29 ci-dessous présente le formalisme UML décrivant la structure générique de la solution que nous adoptons.

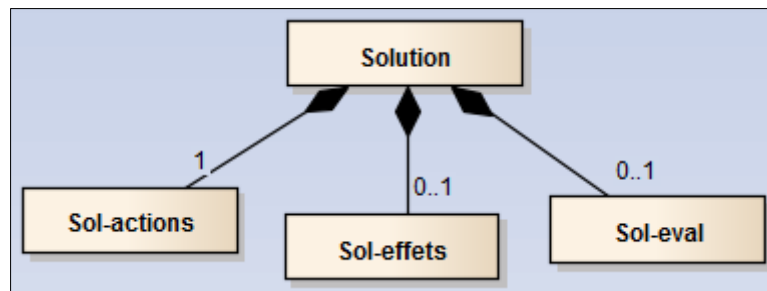


Figure 29. Structure générique de la solution

Pour pouvoir réutiliser cette couche par une autre communauté, une simple configuration au niveau du module qui l'implémente est suffisante afin de spécifier si les deux entités « effets de la solution » et « évaluation de la solution » font partie de la solution et si l'étape de validation fait partie du processus de résolution de problème.

7. Conclusion

Dans ce chapitre, nous avons commencé par présenter la problématique de la couche de Résolution de Problèmes de notre système. Nous avons ensuite exposé une présentation détaillée du RàPC, technique du raisonnement sur laquelle se base l'implémentation de la couche RP. Puis, nous avons décrit le fonctionnement de la couche RP par rapport à la CoP MAETIC et comment elle implémente le RàPC. Nous avons alors, exposé sa conception fonctionnelle, sa structure du cas et nous avons détaillé l'étape de remémoration de son implémentation du RàPC. Nous avons présenté par la suite une validation expérimentale de l'implémentation que nous avons réalisée de la couche RP. Enfin, nous avons exposé la généralité de cette couche.

Rappelons que l'objectif de cette couche est d'aider le membre d'une CoP à résoudre son problème lui-même en se basant sur des problèmes qui ont déjà été résolus dans le passé, et de ce fait, de minimiser le nombre de demandes d'aide postées sur le noyau communautaire. Les couches suivantes du système NICOLAT sont celles assurant l'adaptation des interactions des membres de la CoP. Cette adaptation fait l'objet du chapitre suivant.

Chapitre 4 : Les interactions dans le système NICOLAT

Ce chapitre décrit notre proposition visant à supporter les échanges/interactions entre les membres de la CoP via leurs outils d'interaction préférés et utilisés au quotidien. Cela revient à réaliser des adaptations en fonction de chaque outil d'interaction supporté. Nous commençons ce chapitre par présenter un état de l'art sur l'adaptation dans les EIAH. Puis, des notions sur la gestion du contexte sont introduites, car les notions d'adaptation et de sensibilité au contexte, sont étroitement liées dans les systèmes informatiques. Nous présentons ensuite notre modélisation du contexte. La problématique d'adaptation dans le système NICOLAT et notre approche pour l'adaptation des interactions entre les membres de la CoP sont également exposées. Nous détaillons les deux couches d'adaptation : la couche AI « Adaptation des interactions » et la couche « Interface ». Nous présentons ensuite le fonctionnement de notre système par rapport aux outils que nous supportons : Email, Facebook, SMS et les dispositifs Android. Nous parlons par la suite de la généralité de l'approche d'adaptation proposée. Nous terminons ce chapitre par une expérimentation en milieu écologique ayant pour but d'évaluer notre approche.

1. Adaptation dans les EIAHs

1.1 Définition

Le mot adaptation signifie une modification du comportement d'une partie d'un système en fonction du changement de son contexte pour atteindre un nouveau point de fonctionnement plus satisfaisant ([Alhalabi 2009](#)). Cette notion d'adaptation est caractérisée par la capacité d'un système à changer son comportement, automatiquement ou suite à une intervention humaine, dans le but d'assurer ses fonctions et, si possible, d'améliorer ses performances. Elle est liée à deux concepts ([Krogseter, Oppermann et al. 1994](#)) ([Oppermann and Rasher 1997](#)) :

- L'adaptabilité, caractérisée par la possibilité de modifier certains comportements d'un système par l'utilisateur,
- L'adaptativité, qui concerne la capacité d'un système à s'adapter automatiquement au contexte, sans intervention de l'utilisateur.

Dans le domaine des EIAHs, l'intégration des technologies dans le processus d'apprentissage a pour objectif d'améliorer la qualité de ce processus. L'atteinte de cet objectif se freine par la diversité des profils des apprenants, leurs préférences, leurs centres d'intérêt, leurs contextes d'apprentissage, etc. Cette diversité soulève la nécessité d'intégrer l'adaptation dans les EIAH pour assurer un apprentissage personnalisé et adapté à chaque apprenant.

Ouraiba ([Ouraiba 2012](#)) mentionne que l'adaptation des systèmes d'apprentissage permet de satisfaire les besoins spécifiques des apprenants et d'accomplir leurs objectifs spécifiques. Selon Boticario et Santos ([Boticario and Santos 2007](#)), l'adaptation dans un EIAH consiste à « *créer une expérience d'apprenant ajustée volontairement aux différentes conditions (caractéristiques personnelles, centres d'intérêts, connaissances pédagogiques, contenus disponibles, etc.) sur une période de temps, avec une intention d'augmenter la réussite selon certains critères prédéfinis (l'efficacité de l'apprentissage, le temps écoulé, le coût économique, la participation et la satisfaction des utilisateurs, etc.)* ».

Les deux concepts d'adaptabilité et d'adaptativité qui caractérisent l'adaptation en réaction à une intervention humaine et l'adaptation automatique, se sont étendus à l'adaptation des EIAH ([Burgos 2008](#)). Dans la pratique, ces deux concepts ne sont pas opposés et il est difficile d'isoler l'un de l'autre, à cause de leur relation étroite ([Burgos 2008](#)). Ces deux concepts et toutes les combinaisons entre eux entrent dans le cadre de la personnalisation de l'apprentissage ([Burgos 2008](#)). Dans nos travaux, nous utilisons le mot adaptation pour décrire toute approche qui amène à la personnalisation de l'apprentissage.

Dans ce qui suit nous présentons un aperçu des différents travaux d'adaptation réalisés dans le cadre des EIAH : les STI (Systèmes de Tutorat Intelligents), les SHA (Systèmes Hypermédias Adaptatifs), l'adaptation des scénarios pédagogiques, l'adaptation dans les CSCL (Computer Supported Collaborative Learning) et l'adaptation dans les systèmes d'apprentissage mobile (mobile learning).

1.2 Adaptation dans les Systèmes de Tutorat Intelligents

Freedman ([Freedman, Ali et al. 2000](#)) considère un Systèmes de Tutorat Intelligent (STI) comme tout programme informatique qui abrite une certaine intelligence et qui peut être utilisé pour l'apprentissage. L'objectif principal des STI est de fournir une assistance adaptée à la situation de l'apprenant. Ils permettent aux apprenants de développer leurs compétences par la réalisation de tâches adaptées à leurs besoins en matière d'apprentissage.

Selon Phobun et Vicheanpanya ([Phobun and Vicheanpanya 2010](#)), un STI est capable de personnaliser ses stratégies d'enseignement en termes de contenu, de style d'apprentissage, d'explications appropriées, de conseils, d'exemples, de démonstrations et de problèmes. Cette capacité de personnalisation des stratégies d'enseignement représente l'adaptation dans les STI.

Pour assurer cette adaptation, un STI se base sur quatre modèles de données ([Nwana 1990](#)) ([Freedman, Ali et al. 2000](#)) ([Nkambou, Bourdeau et al. 2010](#)) :

- Le modèle du domaine (expert), qui est une représentation des connaissances, des compétences, des concepts, etc. liés au domaine cible.

- Le modèle de l'apprenant, qui permet de modéliser les caractéristiques personnelles de l'apprenant.
- Le modèle pédagogique (d'enseignement), qui se base sur les modèles du domaine et de l'apprenant afin de prendre des décisions et d'effectuer des choix concernant les stratégies à exécuter.
- Le modèle de l'interface, qui représente le composant communicant et qui contrôle les interactions entre l'apprenant et le système.

1.3 Adaptation dans les Systèmes Hypermédias

Les Systèmes Hypermédias Adaptatifs (SHA) sont des systèmes qui permettent de présenter notamment, des contenus adaptés aux connaissances de l'apprenant, ses objectifs, ou bien encore ses préférences, en lui offrant une navigation souple entre de grandes quantités de données. L'adaptation dans ce type de systèmes est caractérisée par la capacité de présentation de ces contenus adaptés.

Dans le contexte pédagogique, l'adaptation de contenus se base sur un modèle qui représente les aspects pertinents de l'apprenant comme ses préférences, ses connaissances et ses intérêts ([Brusilovsky 1996](#)) ([Phobun and Vicheanpanya 2010](#)). Selon Brusilovsky, un SHA construit un modèle des objectifs, des préférences et des connaissances de chaque apprenant. Il utilise ce modèle tout au long des interactions afin d'adapter l'hypertexte aux besoins de cet apprenant ([Brusilovsky 2001](#)). Pour atteindre leur objectif d'adaptation de contenus, ces systèmes supportent deux types d'adaptation ([Phobun and Vicheanpanya 2010](#)) :

- L'adaptation de la présentation, qui concerne l'adaptation du contenu d'une page hypermédia.
- L'adaptation de la navigation, qui concerne l'adaptation du chemin d'apprentissage.

1.4 Adaptation des scénarios pédagogiques

Un scénario pédagogique est une description d'une situation d'apprentissage en termes de rôles, d'activités, d'environnements et de connaissances manipulées ([Pernin 2003](#)). Cette situation peut être en présentiel ou à distance ([Ouraiba 2012](#)).

Une situation d'apprentissage dans la réalité est difficile à scénariser à l'avance car il y a souvent des événements nouveaux et imprévus qui imposent des déviations par rapport au scénario préétabli ([Zarraonandia, Fernández et al. 2006](#)) ([Miao 2005](#)). Ainsi, un scénario réel d'apprentissage est unique et situé dans son propre contexte ([Ouraiba 2012](#)). Cette difficulté de concevoir des scénarios d'apprentissage qui répondent aux besoins réels des situations d'apprentissage soulève la nécessité de l'adaptation de ces scénarios.

L'adaptation d'un scénario pédagogique peut s'effectuer lors de sa conception ou lors de son exécution ([Ouraiba 2012](#)). Dans le cas de l'adaptation pendant la phase de conception, les concepteurs doivent prendre en considération tous les événements susceptibles d'être déclenchés lors de l'exécution du scénario. Elle peut être implémentée par l'utilisation des propriétés et des conditions de type « if condition : action » pour décrire les règles d'adaptation. Plusieurs travaux ([Miao 2005](#)) ([Burgos 2008](#)) ([Towle and Halm 2005](#)) qui se basent sur IMS-LD (IMS Learning Design : une spécification permettant de modéliser les processus d'apprentissage) ont implémenté ce type d'adaptation. L'adaptation pendant l'exécution, est caractérisée par la capacité de répondre aux événements déclenchés par les apprenants lors du déroulement de leurs sessions d'apprentissage. La réponse à ces événements s'effectue par la prise de décisions en temps réel dans le but d'orienter le déroulement de ces sessions. Ces décisions peuvent être prises en compte, par le système en se basant, par exemple, sur des agents intelligents ([Van Rosmalen, Vogten et al. 2007](#)), ou par les enseignants qui peuvent effectuer les ajustements nécessaires ([Ouraiba 2012](#)).

1.5 Adaptation dans les systèmes support de l'apprentissage collaboratif

Stahl et al. ([Stahl, Koschmann et al. 2006](#)) définissent l'apprentissage collaboratif assisté par ordinateur (ACAO) comme une branche des sciences d'apprentissage qui concerne l'étude du « *comment les individus apprennent ensemble à l'aide des ordinateurs* ». C'est une approche pédagogique dans laquelle l'apprentissage se fait via les interactions sociales en utilisant l'ordinateur ou via l'internet. Ce type d'apprentissage est caractérisé par le partage et la construction de connaissances entre les participants utilisant la technologie comme principal moyen de communication ou comme ressource commune ([Stahl, Koschmann et al. 2006](#)). Cependant, les systèmes support de l'apprentissage collaboratif n'impliquent pas seulement l'utilisation de la technologie dans le but de la communication. Ils visent à capturer et modéliser les informations et les connaissances qui concernent l'activité du groupe et les utiliser pour assurer un contrôle et un support plus efficace du groupe ([Caballé, Daradoumis et al. 2007](#)).

Dans le cadre du développement des systèmes support de l'apprentissage collaboratif, deux domaines de recherches ont émergés ([Magnisalis, Demetriadis et al. 2011](#)) :

- Les systèmes adaptatifs support de l'apprentissage collaboratif (ACSCL : Adaptive Computer Supported Collaborative Learning)
- Les systèmes intelligents support de l'apprentissage collaboratif (ICSCL : Intelligent Computer Supported Collaborative Learning).

Ces deux domaines ont un champ important d'intersection. En effet, il y a des systèmes qui combinent les caractéristiques de l'intelligence et la capacité d'adaptation. Ainsi, les frontières entre ces deux domaines sont imprécises et peuvent être unifiées sous le nom des

systèmes intelligents et adaptatives support de l'apprentissage collaboratif (AICLS) ([Magnisalis, Demetriadis et al. 2011](#)). De notre point de vue, nous considérons les systèmes ICSCS comme une sous-catégorie des systèmes ACSCL qui implémentent des techniques d'Intelligence Artificielle afin de personnaliser la collaboration et donc nous utilisons le terme ACSCL pour nous référer aussi aux ICSCS.

Magnisalis et al. ([Magnisalis, Demetriadis et al. 2011](#)) ont utilisé le terme cible d'intervention pour désigner les objectifs de l'adaptation dans les ACSCL. Ainsi, ils ont mentionné que la cible d'intervention des systèmes ACSCL concerne trois dimensions :

1. La formation des groupes, qui consiste à créer des groupes d'apprenants d'une façon qui permet de favoriser et améliorer l'apprentissage.
2. Le support du domaine, qui concerne l'aide aux apprenants pour acquérir des connaissances du domaine cible.
3. Le support des interactions, qui vise à améliorer les interactions entre les apprenants.

Les différents ACSCL traitent au moins une de ces dimensions afin d'atteindre l'objectif final qui est l'amélioration de l'apprentissage.

1.6 Adaptation dans les systèmes de l'apprentissage mobile

L'informatique mobile connaît depuis les années 90 un essor important, grâce aux évolutions importantes réalisées dans le domaine des réseaux sans fil et des terminaux mobiles. En effet, les réseaux sans fil sont disponibles actuellement selon une multitude de normes et de standards avec une offre diversifiée de couverture et de débit. La distance couverte est en fonction du type de réseau allant des réseaux corporels, personnels, boucle locale radio aux réseaux cellulaires étendus. L'offre en terme de débit est prometteuse car une large gamme de débits est disponible allant de quelques kb/s à des centaines de Mb/s laissant aux concepteurs des applications la liberté de choisir le débit le mieux adapté à leur contexte d'usage.

Les terminaux mobiles connaissent de plus en plus de baisse des prix, ce qui les rend accessibles au grand public. Ils sont disponibles sur le marché selon une grande diversité de configurations et de capacités techniques comme la taille de l'écran, poids, vitesse du processeur, etc. En termes de performances, un terminal mobile de pointe de nos jours peut être considéré comme ayant le même ordre de performances qu'un ordinateur du bureau de fin des années 90.

Dans ce contexte, de nouvelles applications mobiles ont été développées pour des fins aussi diversifiées que le commerce, le tourisme, jeux, publicité, etc. Le domaine de l'apprentissage ne peut pas échapper à cette tendance et le concept de l'apprentissage mobile, communément appelé dans la littérature mobile learning ou m-learning a été apparu pour référer à l'usage des technologies mobiles pour des fins d'apprentissage.

Généralement, l'apprentissage mobile se caractérise par l'accès aux ressources d'apprentissage en situations de mobilité. Dans ce cadre, Quinn ([Quinn 2000](#)) l'a défini comme l'apprentissage qui s'effectue à travers des terminaux mobiles. Keeygen ([Keegan 2007](#)) précise que l'apprentissage mobile peut être vue comme la proposition de formations sur terminaux mobiles. Valk et al. ([Valk, Rashid et al. 2010](#)) et Park ([Park 2011](#)) confirment que l'apprentissage mobile se rapporte à l'utilisation de terminaux mobiles pour faciliter l'apprentissage. Ces définitions traitent de la mobilité du point de vue des technologies utilisées.

O'Malley ([O'Malley, Vavoula et al. 2005](#)) a proposé la définition la plus adoptée dans la littérature. Il définit l'apprentissage mobile comme « *toute sorte d'apprentissage qui se produit lorsque l'apprenant n'est pas dans un endroit fixe ou prédéterminé, ou l'apprentissage qui se produit lorsque l'apprenant profite des possibilités d'apprentissage offertes par les technologies mobiles* ». Dans cette définition, la mobilité est traitée au sens plus large. En fait, la mobilité n'est pas limitée à la technologie utilisée mais englobe aussi l'apprenant.

De même, Göth et Schwabe ([Göth and Schwabe 2008](#)) considèrent l'apprentissage mobile comme l'apprentissage des acteurs mobiles. Dans ce cadre, Yin ([Yin 2010](#)) a précisé que les chercheurs se basent sur deux aspects pour définir l'apprentissage mobile :

- Mobilité grâce à l'utilisation du terminal mobile où l'apprentissage est caractérisé par l'utilisation d'un terminal mobile même si l'apprenant est dans un endroit fixe.
- Mobilité du contexte d'apprentissage de l'apprenant, l'apprentissage mobile est caractérisé par la mobilité de l'apprenant même s'il interagit avec des technologies fixes.

Ainsi, Nguyen ([Nguyen 2010](#)) a considéré que l'apprentissage mobile permet d'effectuer un apprentissage en contexte et au travers de contextes :

- L'apprentissage en contexte concerne la production de l'apprentissage dans des endroits et des temps pertinents pour l'apprenant. En fait, l'apprentissage mobile permet à l'apprenant d'apprendre dans des situations réelles.
- L'apprentissage au travers des contextes concerne l'apprentissage dans des contextes différents. En changeant le contexte, le rôle de l'apprentissage mobile est d'assurer la continuité de l'activité d'apprentissage.

Le contexte de l'apprentissage en mobilité, peut changer tout le temps ([Yin 2010](#)). Les solutions d'apprentissage mobile doivent être capables de s'adapter aux différentes situations d'apprentissage et donc aux changements du contexte. La prise en compte du contexte de l'apprentissage dans le cadre de l'apprentissage mobile est nommée l'apprentissage mobile contextuel (CML-Contextuel Mobile Learning). Un système CAML (Context-Aware Mobile

Learning) ([Yin 2010](#)) est un système d'apprentissage mobile capable de capturer le contexte d'apprentissage et de s'adapter en conséquence.

2. Gestion du contexte

2.1 Définition

Dans la littérature, les chercheurs n'ont pas encore abouti à une définition de la notion de contexte qui soit générique. En effet, les définitions proposées sont très spécifiques à un domaine, ce qui rend la formalisation du contexte difficile, par manque de précisions ou par manque de généralité. En effet, la notion de contexte a été abordée dans les travaux de Schilit et al. ([B. Schilit, Adams et al. 1994](#)) qui l'associent à trois questions : *Où es-tu ? Avec qui es-tu ? De quelles ressources disposez-vous à proximité ?*. Ils définissent alors le contexte en se basant sur 3 aspects : la localisation de l'utilisateur (*Où es-tu ?*), les personnes qui sont à proximité de lui (*Avec qui es-tu ?*), et les dispositifs dont il a l'accès (*De quelles ressources disposez-vous à proximité ?*).

Pour Brown ([Brown 1995](#)), le contexte est l'ensemble des éléments de l'environnement d'un utilisateur dont l'ordinateur a connaissance. Cependant, Pascoe et al ([Pascoe, Ryan et al. 1998](#)) considèrent que les éléments du contexte sont : la localisation de l'utilisateur, l'environnement, l'identité et le temps. La première définition est assez large car elle considère toute information détectable par l'ordinateur comme élément de contexte, ce qui rend difficile l'exploitation de cette définition dans la pratique. La deuxième définition est plus précise car elle restreint le contexte à une liste d'éléments spécifique à l'application cible, cette définition reste moins flexible.

Dey et Abowd ([Dey and Abowd 1999](#)) ont donné la définition la plus adoptée par les chercheurs dans le domaine de la sensibilité au contexte. Ils considèrent le contexte comme : *« toute information qui peut être utilisée pour caractériser la situation d'une entité. Toute entité est une personne, ou un objet qui est considéré significatif à l'interaction entre l'utilisateur et l'application, incluant l'utilisateur et l'application eux-mêmes »*. Dans cette définition, Dey et Abowd insistent sur la notion de pertinence de l'information et essaient de préciser la nature des entités relatives au contexte ([Chaari 2007](#)).

Truong et al. ([Truong, Lee et al. 2005](#)) ont considéré le contexte comme une collection d'informations qui caractérise l'interaction entre l'utilisateur et l'application. Cette définition se rapproche de celle de Dey, à la différence que les auteurs mettent plus l'accent sur le rôle du contexte dans l'interaction.

L'analyse de l'état de l'art des différentes définitions du contexte montre que nous avons soit des définitions qui correspondent à des tentatives de formaliser cette notion, soit des

définitions spécifiques à une application, qui correspondent à l'énumération des éléments du contexte. Les définitions spécifiques étant les plus présentes.

2.2 Sensibilité au contexte

La première définition de la sensibilité au contexte a été proposée par Schilit et Themeir ([B. N. Schilit and Theimer 1994](#)) qui considèrent un système sensible au contexte comme « *tout système qui s'adapte selon son endroit d'utilisation, la collection de personnes et objets voisins aussi bien que le changement de ces objets* ». Cette définition traite la sensibilité au contexte du point de vue de la localisation.

Brown et al. ([Brown, Bovey et al. 1997](#)) considèrent les applications sensibles au contexte comme « *celles dont le comportement peut varier en fonction du contexte* ». Selon Dey et al. ([Dey, Abowd et al. 2001](#)), un système sensible au contexte est un système qui utilise le contexte pour fournir des informations et/ou des services pertinents à un utilisateur, où la pertinence dépend de la tâche de l'utilisateur. Dans la définition de Brown, la sensibilité au contexte est liée à la capacité du changement du comportement. Ainsi, selon la définition de Dey, même une application capable d'afficher des informations du contexte sans réagir selon ces informations est considérée comme une application sensible au contexte.

Hull et al. ([Hull, Neaves et al. 1997](#)) et Pascoe et al. ([Pascoe, Ryan et al. 1998](#)) considèrent la sensibilité au contexte comme la capacité de détection, capture, interprétation et de réaction à un aspect changeant de l'environnement de l'utilisateur et des dispositifs qu'il utilise. Xiaohang ([Xiaohang 2003](#)) définit la sensibilité au contexte d'un système logiciel comme sa capacité à acquérir, gérer, interpréter et répondre aux changements du contexte afin de fournir les services appropriés. Ces deux définitions intègrent les différentes étapes du processus de gestion du contexte (acquisition, gestion, interprétation, réponse aux changements).

Viana ([Viana 2010](#)) a classé, en quatre catégories, les systèmes sensibles au contexte selon les fonctions qu'ils fournissent. Il considère les systèmes sensibles au contexte comme ceux qui utilisent les informations du contexte pour :

- L'adaptation, qui concerne l'adaptation du comportement, des interfaces et des contenus selon le contexte d'utilisation.
- La suggestion du contenu, qui vise à suggérer les contenus les plus ajustés au contexte d'utilisation.
- L'annotation, qui s'intéresse à l'annotation des documents par des informations du contexte qui peuvent être exploitées pour effectuer des requêtes ou pour élaborer des mécanismes de partage de contenus.

- La proposition de services, qui concerne la proposition et l'exécution de services en fonction du contexte d'utilisation.

2.3 Architecture d'un système sensible au contexte

Les systèmes sensibles au contexte ont besoin de moyens pour capturer, interpréter et exploiter les informations du contexte afin de fournir des services adaptés à l'utilisateur.

Dans ce cadre, Coutaz ([Coutaz, Crowley et al. 2005](#)) a défini quatre fonctions de la gestion du contexte : la capture, la transformation, l'identification et l'exploitation du contexte. Ainsi, elle a associé à chacune de ces fonctions, une couche (Figure 30). Ces couches sont complétées par des couches transversales qui peuvent concerner la gestion de l'historique, la sécurité des données ou encore la gestion des connexions aux capteurs.

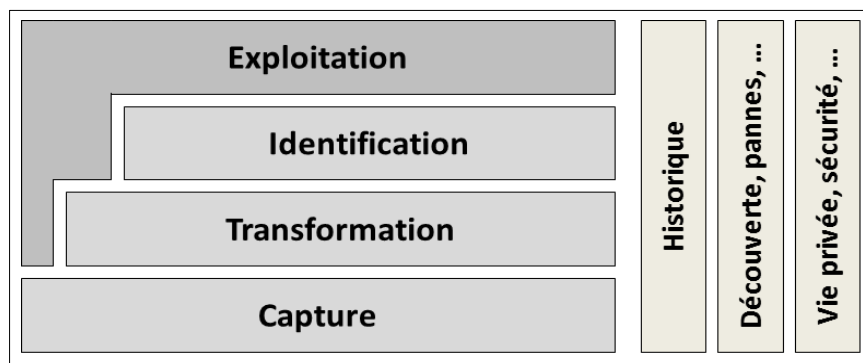


Figure 30. Architecture en couches d'un système sensible au contexte ([Coutaz, Crowley et al. 2005](#))

La capture concerne l'acquisition/collection des informations brutes du contexte à partir de l'environnement. Cette acquisition s'effectue de deux façons en fonction du type de contexte considéré :

- La capture explicite concerne généralement le contexte environnemental. Elle s'effectue via des capteurs physiques (température, vitesse, lumière, pression, etc.).
- La capture implicite se fait à travers un accès direct aux informations stockées dans des fichiers contenant, par exemple, des données des profils utilisateurs, ou à travers une inférence de certaines informations par l'usage des techniques de fouille de données (historique d'interaction, cookies, etc.).

La transformation concerne l'interprétation des données brutes capturées afin d'extraire les informations pertinentes pour le système. En effet, les données capturées ne sont pas souvent appropriées pour une exploitation directe par le système.

L'identification concerne l'identification de la situation actuelle de l'utilisateur en détectant les changements de contexte.

L'exploitation concerne la fourniture des informations du contexte nécessaires au système afin qu'il puisse exploiter le contexte.

2.4 Dimensions du contexte

Les dimensions du contexte sont les catégories d'informations qui composent le contexte. Il n'est pas possible de les présenter d'une façon exhaustive car elles dépendent des objectifs du système ([Nguyen 2010](#)).

Chari ([Chari 2007](#)) a utilisé les dimensions *terminal*, *localisation* et *utilisateur* dans ses travaux, qui s'intéressent à la conception d'une architecture de la gestion du contexte et de l'élaboration d'une stratégie d'adaptation.

Dans le travail d'Azouaou et de Desmoulin ([Azouaou and Desmoulin 2006](#)) concernant un outil d'annotation sensible au contexte, le contexte est composé des éléments suivants : le profil des terminaux (Computing Software, Computing Hardware), l'enseignant, l'activité de l'apprenant, l'activité de l'enseignant, la place et le temps.

Dans les travaux de Nguyen ([Nguyen 2010](#)) qui concernent la conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario, les dimensions du contexte identifiées sont : le scénario, le terminal, la localisation, l'heure, les outils pédagogiques, le réseau, l'environnement physique, la ressource et l'utilisateur. Dans ce système, la dimension de l'utilisateur se décompose en 4 sous-dimensions : le rôle, les compétences, les préférences et la carte de fidélité.

Dans le cadre de la conception du système d'apprentissage appelé SAMCCO (Système d'Apprentissage Mobile Contextuel et Collaboratif) ([Yin 2010](#)), six dimensions ont été utilisées : l'acteur, le terminal, l'environnement, l'activité et la collaboration.

Certaines dimensions sont souvent utilisées comme la localisation, le terminal, et l'utilisateur, d'autres sont rarement utilisées. Ces dernières sont très liées aux domaines d'application. Nous pouvons aussi trouver une même dimension dans plusieurs systèmes alors que les informations qui la constituent changent d'un système à l'autre.

2.5 Approches de modélisation du contexte

L'utilisation des informations de contexte dans des applications, Framework, middleware, etc. nécessite un modèle de contexte dont le but est de structurer ces informations, et donc de faciliter leur stockage, leur interprétation et leur exploitation.

Strang and Linnhoff-Popien ([Strang and Linnhoff-Popien 2004](#)) ont proposé une classification des différentes approches utilisées pour modéliser le contexte. Cette classification a, également, été adoptée dans les travaux de Hoareau et Satoh ([Hoareau and Satoh 2009](#)) et

dans ceux du Popovici ([Popovici 2012](#)). Elle se base sur la structure de données utilisée pour décrire et pour échanger les informations du contexte ([Chaari 2007](#)).

Cette classification se compose de six modèles de contexte qui sont : le modèle clef-valeur, le modèle à balises, le modèle graphique, le modèle orienté objet, les ontologies, et le modèle basé sur la logique.

- Le modèle clef-valeur où les paires « clef/valeur » s'utilisent pour décrire les informations du contexte. La « clef » représente le type de l'information, alors que la « valeur » représente sa valeur. Par exemple, <vitesse, 20> représente l'information de vitesse avec le modèle clef-valeur. Ce modèle se considère comme le plus simple et le plus facile à entretenir, mais peut s'avérer peu flexible. Ce modèle a été utilisé dans le canevas logiciel « contextToolkit » ([Dey, Abowd et al. 2001](#)) dont le but est d'offrir une boîte à outils et des composants logiciels qui permettent le développement et le déploiement d'applications sensibles au contexte.
- Le modèle à balises utilise des balises dans le but d'organiser les données dans des structures hiérarchiques dont les profondeurs dépendent des contextes décrits. Parmi les implémentations de ce modèle, nous trouvons le modèle CC/PP (Composite Capabilities / Preference Profiles) qui est une spécification pour définir les capacités des dispositifs et les préférences des utilisateurs. Le modèle à balises a été utilisé dans les travaux de Indulska et al. ([Indulska, Robinson et al. 2003](#)), qui concernent un système de gestion du contexte, à travers l'utilisation du modèle CC/PP pour le stockage des informations du contexte.
- Le modèle graphique concerne l'utilisation des modèles graphiques pour la modélisation du contexte. Parmi ces modèles, nous trouvons le modèle UML connu par ses fortes composantes graphiques. Ainsi, nous trouvons les modèles formels fondés sur le modèle E/A (Entity/ Association) ([Nguyen 2010](#)). Le modèle graphique est plus adapté pour les cas où la gestion du contexte est assurée par des bases de données ([Nguyen 2010](#)). Ce modèle a été utilisé dans les travaux de Bauer ([Bauer 2003](#)) qui s'est basé sur les diagrammes UML (Unified Modeling Language) pour modéliser les contextes relatifs à la gestion du trafic aérien.
- Le modèle orienté objet. Dans ce modèle, les données du contexte sont structurées et stockées dans des objets. Il permet de profiter des aspects de la conception orienté/objet, tel que la réutilisabilité assurée par l'héritage, la sécurité assurée par l'encapsulation, etc. Ce modèle a été utilisé dans l'outil CASS (Context-awarenesssub-structure) ([Fahy and Clarke 2004](#)) qui est un middleware dont le but est de supporter le développement des applications mobiles sensibles au contexte. Il est utilisé aussi dans Hydrogen ([Hofer, Schwinger et al. 2003](#)) qui est un Framework logiciel développé pour appuyer la sensibilité au contexte.

- Le modèle ontologique. Les ontologies sont utilisées pour la représentation et le traitement du contexte. Les ontologies sont plus adaptées dans le cas où l'aspect sémantique des données du contexte est important. Parmi les projets qui ont utilisé les ontologies pour modéliser le contexte, nous trouvons CoBrA (Context Broker Architecture) ([H. Chen, Finin et al. 2003](#)) et SOCAM ([Gu, Pung et al. 2004](#)). CoBrA est une architecture orientée agents pour les systèmes sensibles au contexte dans des environnements intelligents (par exemple, maisons intelligentes, véhicules intelligents, salles de réunion intelligentes, etc.). Le middleware SOCAM fournit une infrastructure pour la création de services sensibles au contexte.
- Le modèle basé sur la logique. Ce type de modèle se base sur des clauses logiques. Il permet d'utiliser des prédicats, des règles, des opérateurs logiques et des faits pour représenter le contexte ([Nguyen 2010](#)). L'exploitation du contexte basé sur ce type de modèle peut s'effectuer par l'application des méthodes de la logique de premier ordre ([Popovici 2012](#)).

Strang and Linnhoff-Popien ([Strang and Linnhoff-Popien 2004](#)) ont identifié six exigences des systèmes sensibles au contexte en termes d'approches de modélisation :

- La composition distribuée (notée dc), qui concerne la possibilité de gestion distribuée du modèle de contexte.
- La validation partielle (notée pv), qui concerne la capacité de la validation partielle des connaissances contextuelles, quand un modèle du contexte est en cours d'utilisation.
- La richesse et qualité d'information (notée qua), qui concerne la capacité de supporter l'indication de la richesse et de la qualité.
- L'incomplétude et ambiguïté (notée inc), qui concerne la capacité de gérer des informations de contexte qui sont incomplètes ou ambiguës.
- Le degré de formalité (noté for), qui concerne la capacité de description des faits contextuels et des interrelations entre les données.
- L'applicabilité aux environnements existants (notée app), qui concerne l'implémentation du modèle dans des systèmes sensibles aux contextes existants.

Strang and Linnhoff-Popien ([Strang and Linnhoff-Popien 2004](#)) se sont basés sur ces exigences pour évaluer les six modèles du contexte présentés ci-dessus. Le tableau 7 ci-dessous montre le résumé des résultats de cette évaluation.

Approche	dc	pv	qua	inc	for	app
Modèle clé-valeur	-	-	-	-	-	+
Modèle schéma de balisage	+	++	-	-	+	++
Modèle Graphique	-	-	+	-	+	+
Modèle orienté objet	++	+	+	+	+	+
Modèle basé sur des logiques formelles	++	-	-	-	++	-
Modèle basé sur l'ontologie	++	++	+	+	++	+

Tab 7. Evaluation des approches de la modélisation de contexte par rapport aux exigences (Strang and Linnhoff-Popien 2004)

Ce tableau montre que le modèle orienté objet et celui basé sur les ontologies sont les plus répondants à ces exigences et que ces approches sont les mieux adaptées pour modéliser le contexte. Cela ne veut pas dire que les autres modèles sont inutiles ou inutilisables. En effet, l'efficacité d'un modèle dépend des besoins de l'application cible.

3. Modélisation du contexte

3.1 Modèle générique du contexte

Comme nous avons pu le remarquer dans le paragraphe 2.1, les différentes définitions du contexte présentées dans la littérature (B. Schilit, Adams et al. 1994) (Brown 1995) (Pascoe, Ryan et al. 1998) (Dey and Abowd 1999) (Truong, Lee et al. 2005) (Truong, Lee et al. 2005), montrent que nous avons soit des définitions qui correspondent à des tentatives de formaliser la notation du contexte et qui sont très abstraites et manquent de précision, soit des définitions spécifiques à une application ou à un domaine particulier et qui manquent de généralité.

Ainsi, la définition de Dey (Dey and Abowd 1999) : « **toute information qui peut être utilisée pour caractériser la situation d'une entité. Toute entité est une personne, ou un objet qui est considéré significatif à l'interaction entre l'utilisateur et l'application, incluant l'utilisateur et l'application eux-mêmes** », reste la définition la plus adoptée par les chercheurs dans le domaine de la sensibilité au contexte. Dans cette définition Dey insiste sur la notion de pertinence de l'information et essaie de préciser la nature des entités relatives au contexte (Chari 2007).

En cherchant à proposer une définition du contexte par rapport à l'apprentissage communautaire qui se base sur les interactions entre les membres d'une CoP via un SIC, nous reprenons la définition de Dey et proposons d'inclure l'aspect communautaire (interactions entre les acteurs du système). Nous définissons donc le contexte comme **un ensemble d'informations qui peuvent aider le système à stimuler les activités de ses acteurs et à favoriser leurs interactions.**

Le cœur du système NICOLAT est une application web basé sur le CMS DRUPAL dont les données sont structurées sous forme d'objets. Pour assurer la compatibilité entre les données du contexte et l'application, nous retenons le modèle Orienté/Objet pour modéliser ce contexte. Nous visons aussi à élaborer une représentation qui soit extensible et facilement réutilisable dans d'autres applications. Ces deux exigences peuvent être assurées par le modèle Orienté/Objet via les relations de généralisation et de spécialisation. Nous avons vu dans l'état de l'art sur la modélisation du contexte (Paragraphe 2.5) que ce modèle et le modèle ontologique sont les modèles les plus adaptés pour modéliser le contexte. Le modèle Orienté/Objet répond aux différentes exigences des systèmes sensibles au contexte en terme d'approche (la composition distribuée, la validation partielle, la richesse et qualité d'information, l'incomplétude et ambiguïté, le degré de formalité, l'applicabilité aux environnements existants) comme présenté dans ce paragraphe.

Comme nous l'avons énoncé dans chapitre 1, nous souhaitons offrir deux types d'adaptation :

- La première adaptation concerne le support des échanges/interactions entre les membres de la CoP via leurs outils d'interaction préférés et utilisés quotidiennement. Cette adaptation fait l'objet de ce chapitre. Elle sera détaillée à partir de la section suivante. Elle vise principalement à permettre à chaque membre de la CoP d'utiliser son outil préféré pour poster un contenu et à solliciter chaque membre dans son outil utilisé au moment de la publication d'une demande d'aide ou d'un commentaire. Pour effectuer cette adaptation, il faut avoir des **informations sur les outils d'interaction des membres de la communauté**.
- La deuxième adaptation concerne la sélection des membres qui peuvent contribuer positivement à résoudre un problème. Cette adaptation fait l'objet du chapitre suivant. L'objectif est de sélectionner des membres ayant des profils adaptés par rapport au contexte du problème à résoudre. Il y a donc deux aspects dont nous avons besoin : **le profil du membre** et **le contexte du problème**. Les éléments constituant ces deux entités dépendent du domaine cible.

Pour pouvoir guider les fonctionnalités d'adaptation souhaitées, nous représentons le contexte de la situation d'apprentissage en considérant deux aspects essentiels : **le contexte du problème** et **le profil du membre**. Ce dernier peut être :

- Le membre émetteur du problème lorsqu'on traite le contexte du problème objet de l'apprentissage ou problème cible,
- ou bien, le membre candidat à être sélectionné pour résoudre le problème cible lorsqu'on évalue son profil.

Ainsi, nous regroupons les informations sur les outils d'interaction du membre dans des sous entités du profil du membre dont chacune est appelée : **profil technologique**. Un membre

peut avoir plusieurs profils technologiques dont chacun correspond à un de ses outils d'interaction. Un profil technologique d'un membre permet donc de préciser son outil d'interaction, des informations techniques permettant de communiquer avec cet outil et sa disponibilité par rapport à cet outil. La figure 31 ci-dessous présente le modèle générique du contexte.

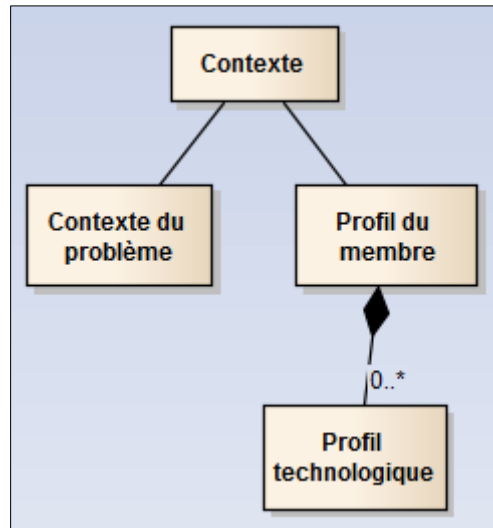


Figure 31. Modèle générique du contexte

3.2 Modélisation du contexte pour la CoP MAETIC

Dans ce paragraphe, nous personnalisons le modèle générique du contexte pour la CoP MAETIC. Dans ce cadre, les membres de la CoP sont des enseignants et les problèmes sont des difficultés MAETIC. **Nous notons donc le contexte du problème par le contexte de la difficulté et le profil du membre par le profil de l'enseignant.**

Pour identifier les différentes dimensions constituant le contexte de la difficulté et celles constituant le profil de l'enseignant, nous avons effectué des entretiens avec les experts de la méthode MAETIC. Comme résultat, nous avons retenu une dimension appelée « Profil pédagogique » qui rejoint la dimension « Profil technologique » pour constituer le profil de l'enseignant. La dimension « Profil pédagogique » regroupe la spécialité de l'enseignant et trois attributs liés à l'utilisation de MAETIC. Les attributs de cette dimension sont :

1. Sa spécialité.
2. Son niveau d'expertise par rapport à l'utilisation de MAETIC (noté compétence MAETIC),
3. Son degré de familiarisation avec la Gestion de Projet (noté compétence GP),
4. Son degré de familiarisation avec l'utilisation des blogs (noté compétence blog).

Chacun des trois derniers attributs peut prendre trois valeurs possibles : novice, intermédiaire et expert.

Le contexte de la difficulté se compose de trois dimensions :

1. La dimension « classe » : regroupe les caractéristiques de la classe où la difficulté a été rencontrée. Elle se compose de cinq attributs :
 - Effectif : le nombre d'étudiants dans la classe,
 - Nombre de groupes : le nombre de groupes d'étudiants définis pour déployer MAETIC,
 - Nombre d'heures global : le nombre total d'heures dédiées au module,
 - Nb d'heures par Séance : le nombre d'heures dédiées à chaque séance,
 - Niveau : le niveau d'enseignement de la classe : DUT, Licence, Master, Ingénierie.
2. La dimension « spécifique » : regroupe des informations liées directement à la difficulté. Elle se compose de deux attributs :
 - Type de la difficulté : le type de la difficulté par rapport à la classification des types de difficultés MAETIC présentée dans la section 2 du chapitre 2,
 - Liée au domaine : caractérise si la difficulté concerne la matière enseignée ou concerne un aspect complémentaire comme la gestion de projet, la gestion des blogs, etc.
3. La dimension « profil pédagogique » de la difficulté est en réalité le profil pédagogique de l'enseignant émetteur de la difficulté. Au niveau modélisation, nous reprenons la dimension pédagogique du contexte de l'enseignant.

La figure 32 ci-dessous présente le diagramme de classes du contexte.

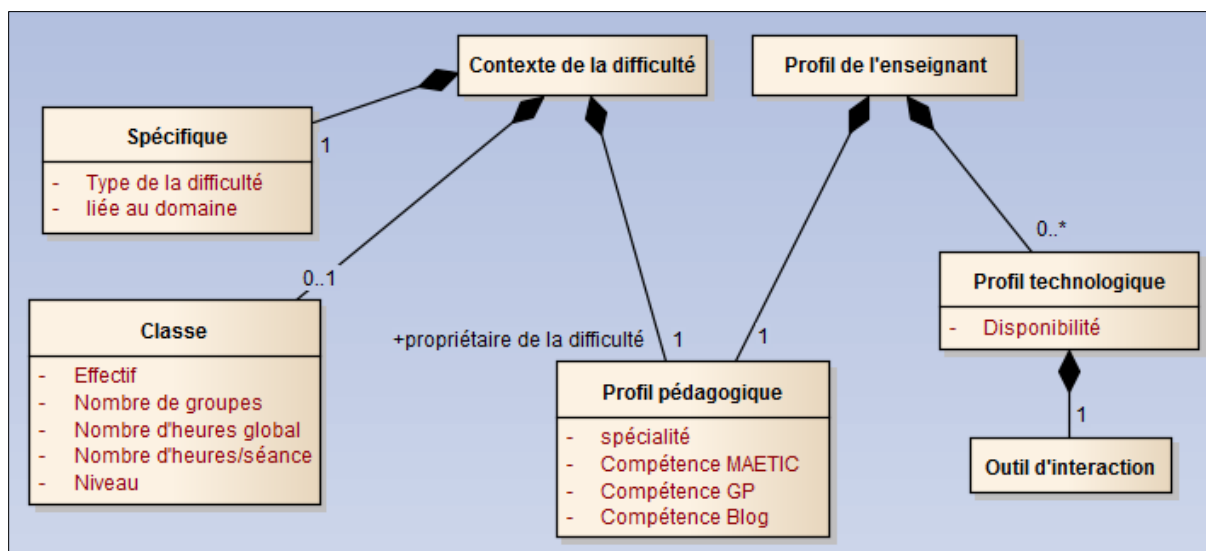


Figure 32. Modélisation du contexte dans le système NICOLAT

Notons finalement que les données liées au profil pédagogique sont stockées dans le système. Les données liées à la dimension spécifique de la difficulté et à la dimension de la classe sont collectés auprès de l'enseignant émetteur de la difficulté au moment où il publie sa difficulté à travers un formulaire en ligne.

4. Adaptation des interactions dans le système NICOLAT

4.1 Problématique et objectif de l'adaptation

Dans le système NICOLAT, tous les enseignants membres de la CoP sont considérés comme des apprenants qui apprennent à travers la résolution collaborative de problèmes qu'ils rencontrent. Les interactions, entre les membres, pour l'entre-aide qu'ils offrent les uns aux autres, représentent le seul moyen d'apprentissage à travers la CoP.

Le support des interactions est offert par NICOLAT à travers le noyau communautaire présenté dans le chapitre 2. Il est représenté sous la forme d'un réseau social permettant aux membres de poster leurs problèmes et de les résoudre à l'aide de fils de discussions. Ce noyau est caractérisé par sa propre interface graphique, inspirée de celles des réseaux sociaux classiques.

Pour faire aboutir l'apprentissage, après publication d'une demande d'aide par un membre, il doit attendre que les autres membres consultent le noyau NICOLAT pour que sa demande soit prise en compte. Dans ce cas, nous avons relevé un certain nombre de constatations qui peuvent altérer la qualité de cet apprentissage. Nous les présentons comme suit :

- La consultation fréquente de l'espace communautaire par les membres n'est pas assurée et peut être fastidieuse lorsqu'il n'y a pas souvent de nouveautés sur

l'environnement. Ceci freine la résolution de problèmes qui peuvent avoir un caractère « urgent ».

- Les membres d'une CoP peuvent ne pas être familiarisés avec les outils sociaux du web 2.0 en général, et avec l'interface communautaire du noyau NICOLAT en particulier. Ceci peut être la cause des consultations éloignées dans le temps, ou même de l'abandon total.

Nous constatons aussi :

- Qu'à l'heure actuelle, et avec le développement des outils de communication, télécom ou à travers l'Internet, chacun des membres devrait avoir ses propres préférences pour les outils d'interaction usuels qu'il utilise pour interagir avec son entourage au quotidien : la messagerie électronique, les SMS, Facebook, etc. Avec ses outils préférés, le membre devrait être familiarisé et très habile dans leur manipulation technique.

Nous exploitons donc cette dernière opportunité afin d'offrir aux membres une nouvelle vision pour supporter leurs interactions et essayer de minimiser les abondons et les freins à l'apprentissage dû à l'utilisation de nouveaux outils technologiques. Nous proposons alors de réaliser des adaptations des interactions en permettant à chaque membre d'interagir avec la CoP en utilisant l'interface classique de son outil d'interaction préféré : i) avec lequel il est le mieux familiarisé, ou ii) qui soit le plus adapté à sa situation actuelle (en cas de mobilité physique par exemple).

Notre vision d'adaptation situe notre travail de recherche par rapport à deux axes de travaux d'adaptations dans les EIAH qui ont été présentés dans le paragraphe précédent :

- Les systèmes adaptatifs support de l'apprentissage collaboratif. En effet, l'adaptation concerne la formation des groupes, le support au domaine métier et support aux interactions. L'objectif primaire de l'adaptation dans NICOLAT est, alors, l'adaptation des interactions entre les apprenants. Comme deuxième étape, nous formerons des groupes d'apprenants appropriés, à qui faire aboutir les demandes d'aide lorsqu'elles sont publiées. La formation des groupes fait l'objet du chapitre suivant.
- Les systèmes adaptatifs support de l'apprentissage mobile. Dans notre travail d'adaptation, nous considérons deux types de mobilité : la Mobilité Spatiale (MS) en permettant aux membres de la CoP d'interagir via leurs terminaux mobiles, et la Mobilité Virtuelle (MV) que nous définissons par la possibilité pour un apprenant de naviguer entre plusieurs outils d'interactions : outils matériels (Smartphone, ordinateur, tablette, etc.) ou logiciels (client de messagerie, Facebook, etc.).

Nous supportons l'adaptation des interactions dans NICOLAT à travers deux couches :

- La couche « Adaptation des Interactions » (AI). Cette couche a pour objectif d'implémenter des passerelles technologiques entre plusieurs outils d'interaction utilisés pour accéder à l'environnement communautaire. Dans le cas de la CoP MAETIC, elle permet aux membres de la CoP de consulter les fils de discussion de NICOLAT en choisissant l'un des outils suivants : Facebook, clients Email, SMS ou terminaux Android. La couche AI permet ainsi de minimiser le temps de réponse à une demande d'aide dans divers cas d'inaccessibilité comme les problèmes d'accès à Internet (utiliser les SMSs) ou les membres en forte mobilité (utiliser l'application Android ou SMS). La couche AI permet aussi d'augmenter le confort d'utilisation du système en proposant aux membres de supporter leurs interactions dans leurs outils préférés, notamment, en cas de non familiarisation avec l'interface par défaut de l'environnement communautaire dédié.
- La couche « Interface ». Le but de cette couche est d'assurer la communication entre la couche AI et l'utilisateur final. Elle fournit des composants logiciels au service de la couche AI capables de transmettre et de récupérer les données depuis et vers le système. Ces derniers sont groupés en : i) composants communautaires (application Facebook et composant dédié à l'interaction avec les clients mail) ; et ii) composants mobiles (application Android et un composant logiciel capable d'envoyer et recevoir des SMSs).

Signalons enfin que notre approche d'adaptation des interactions permet de proposer des solutions à trois questions de notre problématique de cette thèse :

- Question 2 : " Comment dépasser les problèmes de prise en main et de familiarisation avec les nouveaux outils informatiques ?",
- Question 3 : "Comment assurer l'accès au système dans des cas de mobilité ?",
- Et à la question 4 : " Comment minimiser le temps de réponse ?".

4.2 Architecture globale des couches d'adaptation

Pour supporter les échanges entre les membres de la CoP via leurs outils d'interaction, la couche d'adaptation des interactions (AI) s'occupe de la gestion des interactions et des transformations de contenus alors que la couche Interface fournit les interfaces, les applications et les composants logiciels nécessaires au service de la couche AI afin de pouvoir communiquer avec les membres de la CoP. La couche AI joue donc le rôle d'un composant intermédiaire qui assure l'échange de données entre le système (le noyau communautaire et la couche RP) et les membres de la CoP via la couche Interface.

Les outils que nous supportons, dans le cas de la CoP MAETIC, sont : Email, Facebook, SMS, Dispositifs Android. Le choix de ces outils a été effectué par rapport aux réponses des

membres de la CoP à un questionnaire visant à identifier leurs habitudes d'usage en termes d'outils de web 2.0, de terminaux mobiles et de services de téléphonie mobile.

La figure 33 ci-dessous présente les couches d'adaptation par rapport au noyau communautaire et la couche RP d'un côté et les outils d'interaction supportés d'un autre.

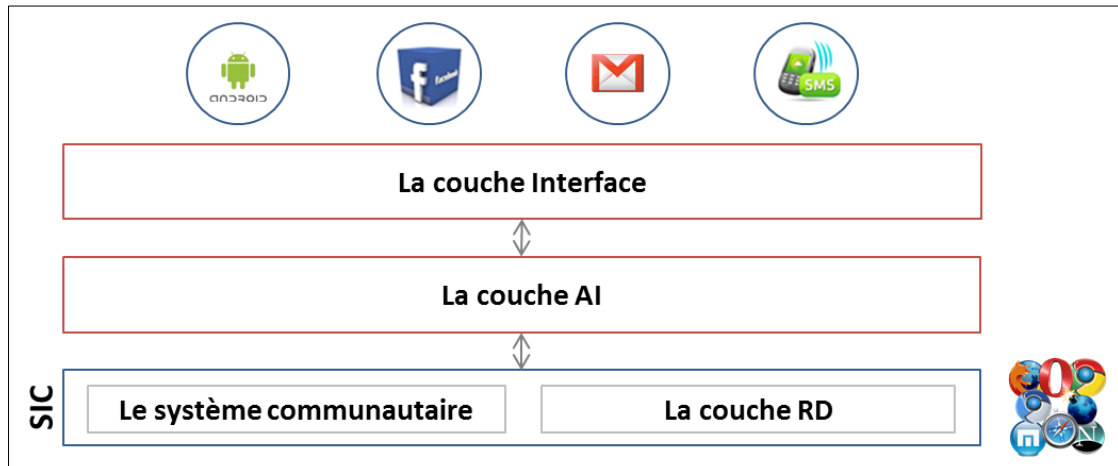


Figure 33. Structuration des couches d'adaptation

L'interaction entre l'utilisateur et le Système Informatique Communautaire (SIC) via un les outils supportés consiste à naviguer entre un ensemble de composants logiciels et éventuellement des services externes. Cette navigation génère des processus dont les extrémités sont le SIC et l'un des outils d'interaction.

Durant un processus d'interaction, les entrées d'un composant dépendent des sorties d'un autre. Chaque composant est modélisé sous forme d'une fonction F qui prend en entrée un ensemble de paramètres $P = \{p_1, p_2, \dots, p_n\}$ et retourne un ensemble de données $R = \{r_1, r_2, \dots, r_m\}$. Le nombre de paramètres et celui de données retournées sont spécifiques pour chaque composant. Ainsi, les entrées et les sorties d'un composant changent selon le sens de l'interaction (du SIC vers l'outil d'interaction ou le contraire).

La conception de notre système sous la forme de composants logiciels et l'interaction sous forme d'un processus impliquant un sous ensemble de ces composants a pour but d'assurer :

- L'évolutivité d'interopérabilité, qui concerne la capacité d'interconnecter avec de nouveaux outils. Pour assurer l'interconnexion avec un nouvel outil, il suffit d'ajouter le (ou les) composant(s) dédié(s) à l'interaction avec cet outil au niveau de la couche Interface et d'ajouter les composants de transformation de contenus dédiés à cet outil au niveau de la couche AI.
- La réutilisabilité, qui concerne la conception de système sous forme de composants indépendants pour faciliter la réutilisation de ces composants dans d'autres systèmes.

4.3 Couche AI « Adaptation des Interactions »

En ne considérant que le noyau NICOLAT dans le support des interactions entre les membres de la CoP, un membre est obligé de passer par l'interface par défaut du système communautaire pour publier ou consulter un contenu. Ainsi, une interaction entre 2 ou plusieurs membres s'effectue d'une manière asynchrone (figure 34).

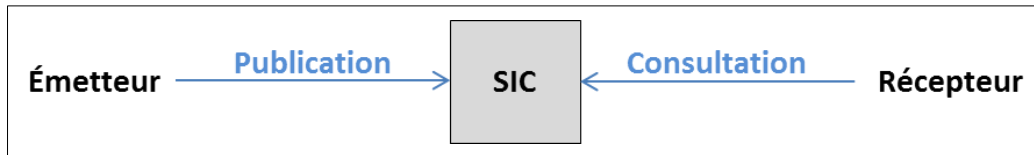


Figure 34. Cycle de vie d'une interaction sans adaptation

La figure 35 présente le cycle de vie d'une interaction adaptée par la couche AI. En effet, un membre de la CoP n'est pas obligé de consulter le noyau communautaire pour interagir avec les autres membres.

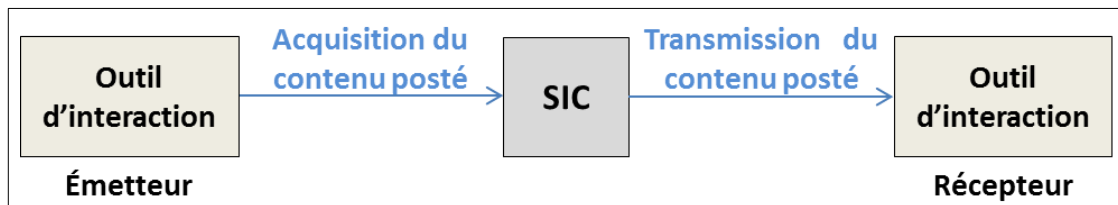


Figure 35. Cycle de vie d'une interaction adaptée

Nous avons séparé les tâches d'adaptation de la couche AI en deux types :

- L'adaptation du format des contenus échangés, depuis, et vers le SIC. C'est ce que nous avons appelé des « passerelles technologiques ».
- La prise des décisions d'adaptation qui concerne le choix des membres de la CoP à solliciter (Objet du chapitre 5) ainsi que le choix des outils d'interaction à travers lesquels ils sont sollicités.

Pour réaliser les adaptations concernant les outils d'interaction des membres (choix des outils et adaptation du format des contenus), la couche AI se base sur le « Profil technologique » du membre. Nous rappelons qu'un membre peut avoir plusieurs profils technologiques, chacun est lié à un outil d'interaction et contient des informations techniques permettant de communiquer avec cet outil (le numéro de téléphone pour interagir par SMS, l'ID Facebook pour interagir via Facebook, l'adresse email pour interagir par email, etc.) et spécifie la disponibilité de l'enseignant par rapport à cet outil.

Par conséquent, les quatre composants de la couche AI sont présentés dans la figure 36 :

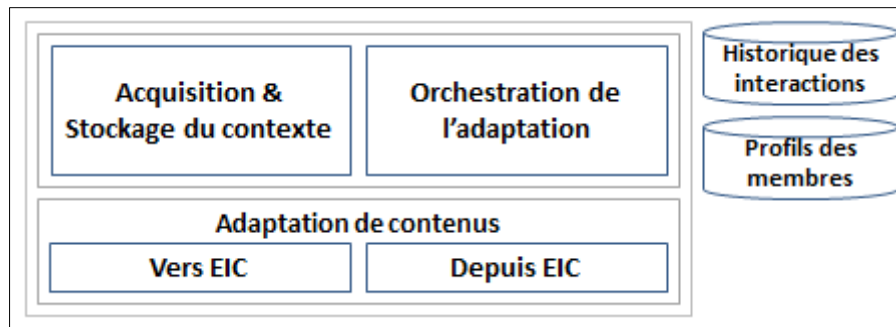


Figure 36. Composants de la couche AI

Le composant « Depuis SIC » : permet de transformer le format d'un contenu stocké sous forme d'un Objet Drupal à un format spécifique à un outil d'interaction. Il permet de transformer les problèmes stockés sous forme de nœuds DRUPAL et les commentaires stockés comme des objets associés à ces nœuds à des Objets JSON (pour pouvoir les transmettre via les services web), à des emails ou à des textes SMS. Nous notons que JSON est un format de représentation de données dérivé de la représentation des objets du langage JavaScript. Il permet de représenter les données structurées comme le permet XML.

Le composant « Vers SIC » : permet de transformer un contenu dont le format est spécifique à un outil d'interaction à un objet Drupal. Il permet de transformer un objet JSON récupéré via les services Web en un nœud DRUPAL pour être stocké dans le système ou à un Object associé à un nœud DRUPAL dans le cas d'un commentaire. Ainsi, il permet de transformer des problèmes et des commentaires transmis via les boites email des membres et des commentaires transmis en SMS pour être stockés dans le système.

Le composant « Acquisition et Stockage du contexte » : permet d'acquérir et de stocker les données du contexte qui seront utilisées par le composant « Orchestration de l'adaptation ».

Le composant « Orchestration de l'adaptation » : constitue le composant principal chargé de prendre des décisions par rapport à notre approche d'adaptation. Il prend en charge deux types de décisions :

- Les décisions concernant les membres à solliciter lors de la publication d'une difficulté. La sélection des membres à solliciter sera détaillée dans le chapitre suivant.
- Les décisions concernant l'outil d'interaction à utiliser pour solliciter les membres sélectionnés en exploitant les données des profils technologiques des membres. En effet, l'utilisateur spécifié dans l'interface de configuration de son profil, sur quels outils il est disponible. En se basant sur ces informations, le système le sollicite dans les outils appropriés.

4.4 Couche Interface

Cette couche se compose des interfaces, des applications et des composants qui permettent aux membres d'interagir avec le système par les différents outils d'interaction supportés. Pour atteindre son objectif, cette couche se base sur deux stratégies de communication avec l'utilisateur en fonction des outils cibles. Ces stratégies sont :

- **Push notification** : le système envoie une notification à l'utilisateur. Après consultation de la notification, le contenu se récupère via les services web. C'est la stratégie utilisée avec Facebook et les dispositifs Android.
- **Push content** : le système prépare et envoie le contenu à l'utilisateur. C'est la stratégie utilisée avec l'Email et le SMS.

Les composants de cette couche sont les suivants (Figure 37) : **Gestion Mail, Formulaires Mail, Gestion SMS, App. Facebook, App. Android, Services web, Gestion de notifications.**



Figure 37. Composants de la couche Interface

Le composant « Formulaires mail » se compose de formulaires HTML qui s'intègrent dans les emails et qui permettent de décrire des problèmes (difficultés MAETIC) et de poster des commentaires.

Le composant « Gestion mail » permet de recevoir les données transmises via les formulaires HTML de « Formulaires mail » et d'envoyer des emails.

Le composant « Gestion SMS » permet d'envoyer et de recevoir des SMSs. Il se base sur les services fournis par une plateforme d'envoi des SMS (<http://www.octopush-dm.com/>).

Le composant « App. Facebook » est une application Facebook qui permet d'interagir avec le système communautaire/La couche RP via Facebook.

Le composant « App. Android » est une application Android qui permet d'interagir avec le système communautaire via des dispositifs qui fonctionnent sous Android.

Le composant « Services web » est un composant qui offre des services web pour la publication et la récupération des problèmes et des commentaires. Le but est d'assurer la communication avec les applications Android et Facebook.

Le composant « Gestion de notifications » permet de notifier aux membres la publication d'un problème ou d'un commentaire sous le contrôle du composant « Orchestration de l'adaptation ». Pour envoyer une notification, ce composant exploite les services externes suivants :

- **Google Cloud Messaging (GCM)**, qui permet d'envoyer des notifications aux dispositifs des utilisateurs d'une application Android. En effet, GCM est un service de Google. C'est le seul service qui offre cette fonctionnalité d'envoi des notifications aux dispositifs des utilisateurs d'applications Android.
- **Facebook API Notifications**, qui permet d'envoyer des notifications vers les profils des abonnés d'une application Facebook. En effet, c'est un API qui permet d'envoyer des messages courts (jusqu'à 180 caractères) aux utilisateurs d'une application Facebook. Ces messages sont reçus par les utilisateurs sous forme de notifications Facebook. Lorsque l'utilisateur clique sur la notification, il sera redirigé vers un URL spécifié lors de génération de la notification.

Dans ce qui suit, nous détaillons comment nous offrons la possibilité d'interagir à travers les différents outils d'interaction avec lesquels nous supportons la CoP MAETIC : Email, Facebook, Android et SMS.

5. Fonctionnement du système

5.1 Application Email

Pour qu'un membre puisse interagir avec les autres membres de la CoP via un client email, nous avons développé des formulaires HTML qui s'intègrent dans les emails et qui permettent de poster des difficultés MAETIC et des commentaires. Ainsi, nous avons développé le composant « Gest. Email » qui permet :

- De récupérer les données transmises par les membres à travers ces formulaires.
- D'envoyer les difficultés et les commentaires par email.

Pour envoyer un email, ce composant utilise la fonction mail() de PHP. En effet, après la publication d'une difficulté ou d'un commentaire, le système l'envoie par Email aux utilisateurs intéressés par cet outil. La figure 38 ci-dessous présente le processus d'envoi d'un contenu par Email :

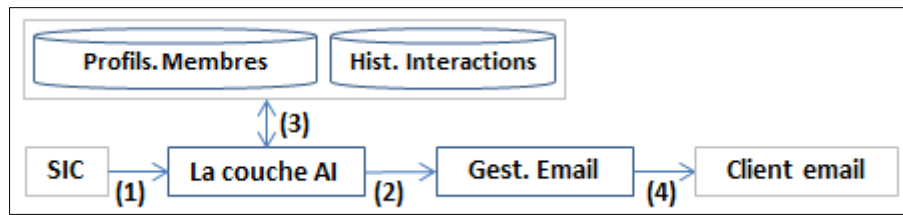


Figure 38. Processus d'envoi d'un contenu par Email

Comme le montre la figure 38, après la publication d'un contenu, le système fait appel à la couche AI (1). Cette couche sélectionne les membres à solliciter via l'outil email en se basant sur leurs profils et les traces de leurs interactions (2) et transmet la liste de ces membres au composant 'Gestion mail' (3). Ce dernier prépare l'email décrivant le contenu publié et y intègre un formulaire permettant à celui qui va recevoir la difficulté d'y apporter un commentaire à ce contenu. Enfin, il envoie cet email à l'utilisateur concerné (4). La figure 39 ci-dessous présente un extrait du code permettant de préparer et d'envoyer l'email décrivant une difficulté MAETIC.

```

$body.= '<h2><b>'. $node->title. '</b></h2>';
$body.= '<u>Type de la difficulté :</u>';
$body.= '<a href=" ../taxonomy/term/'. $type_diff_tid. '" >'. $type_diff_name. '</a><br>'; } (1)
$body.= '<u>description de la difficulté :</u><br>'. $strip_tags($desc). '<br>'; } (2)

$body.= '<div><u>Informations sur la classe :</u><br>';
if(isset($niveau_classe_tid)){
    $body.= '<span>Niveau de la classe :</span>';
    $body.= '<a href=" ../taxonomy/term/'. $niveau_classe_tid. '" >'. $niveau_classe_name. '</a>';
}
...
...
$body.= '</td></tr></div>'; } (3)

$body.= '<div><u>Informations sur l\'enseignant :</u><br>';
if(isset($specialite_name)){
    $body.= '<span>Spécialité :</span>';
    $body.= '<a href=" ../taxonomy/term/'. $specialite_tid. '" >'. $specialite_name. '</a>';
}
...
...
$body.= '</td></tr></div>'; } (4)

$body.= "<br><table style='padding: 2px,width: 550px;' ><tr><td><fieldset>";
$body.= "<legend>Ajouter un commentaire</legend>";
$body.= "<form action=' ../LIEN_REC_MAIL.' method='post' >";
$body.= "<input type='hidden' name='com_id_node' value='". $node->nid. "' />"; } (5)
$body.= "Commentaire :<br><textarea name='com_body'></textarea><br>";
$body.= "<input type='submit' value='valider' />";
$body.= "<input type='reset' value='annuler' />";

$headers = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=utf-8' . "\r\n";

foreach ($list_users as $user){
    $body_user = $body;
    $body_user.= "<input type='hidden' name='com_id_user' value='". $user->uid. "' /><br>"; } (6)
    $body_user.= "</form>";
    $body_user.= "</fieldset></td></tr></table><br>";
    $body_user.= "Accéder à l'EIC MAETIC pour ajouter un commentaire : ";
    $body_user.= "<a href='". $lien. "' >". $lien. "</a>";
    $body_user.= "</body></html>";
    mail($user->mail, $subject, $body_user, $headers);
}

```

Figure 39. Extrait du code permettant de préparer et d'envoyer l'email décrivant une difficulté

Dans ce code, Le composant « Gest. Email » structure les données à transmettre aux membres en utilisant le langage HTML. Il construit une structure HTML contenant le type de la difficulté (1), sa description (2), les informations sur la classe où elle a été rencontrée (3), les informations sur l'enseignant qui l'a posté (4) et un formulaire permettant d'ajouter un commentaire (5). Ensuite, il personnalise le formulaire pour chaque membre en ajoutant un champ caché contenant l'identifiant du membre ciblé par l'email (6). Si un membre répond en utilisant ce formulaire, son identifiant intégré dans le formulaire permettra de l'identifier.

Cependant, pour recevoir les données transmises via les formulaires que nous avons développés, le composant 'Gest. Email' offre un URL associé à une fonction capable de traiter des données issues de ces formulaires.

En effet, pour répondre à une demande ou poster une difficulté, le membre utilise nos formulaires HTML pour transmettre le contenu à posté via le composant ‘Gest. Email’ à notre système. La figure 40 ci-dessous présente le processus de publication d’un contenu.

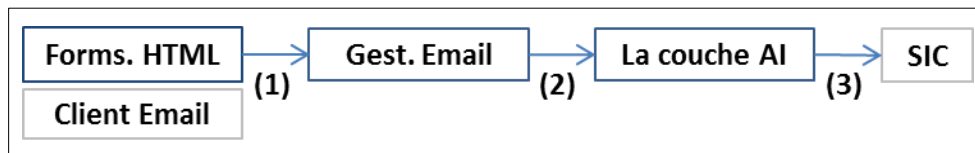


Figure 40. Processus de publication d’un contenu via un client email

Comme le montre la figure 40, après le remplissage du formulaire HTML dédié à décrire une difficulté MAETIC ou celui dédié à poster un commentaire, et sa validation, le composant « Gest. Mail » reçoit le contenu envoyé (1). Ce dernier identifie l’émetteur et fournit les données reçues à la couche AI (2). La couche AI prépare le contenu pour être stocké sur le système (3). La figure 41 ci-dessous illustre le formulaire permettant de décrire une difficulté via l’email.

Description *

Titre :

Description :

liée au domaine :

oui

non

Type de la difficulté *

⇒ Difficulté transversale :

<p>Liée à l'infrastructure disponible</p> <ul style="list-style-type: none"> <input type="radio"/> Pannes réseau <input type="radio"/> Configuration des salles <input type="radio"/> Saturation des dispositifs étudiants 	<p>Liée l'organisation générale de MAETIC</p> <ul style="list-style-type: none"> <input type="radio"/> Discours à tenir aux étudiants <input type="radio"/> Pré-lancement de MAETIC <input type="radio"/> Gestion des étudiants <input type="radio"/> Ma propre organisation <input type="radio"/> Evaluation des livrables
---	--

⇒ Difficulté liée à une étape de MAETIC :

<p>Lancement</p> <ul style="list-style-type: none"> <input type="radio"/> Définition des roles <input type="radio"/> Réalisation de la charte graphique <input type="radio"/> Création du journal de bord <input type="radio"/> Rédaction de réponse 	<p>Cadrage : Elaboration du cadre du projet</p> <ul style="list-style-type: none"> <input type="radio"/> Proposition d'une organisation <input type="radio"/> Qualification du projet <input type="radio"/> Définition du cadre général <input type="radio"/> Etablir le périmètre du projet <input type="radio"/> Produire le document de cadre 	<p>Cadrage : <input type="radio"/> Elaboration du plan de communication</p>
--	---	---

<p><input type="radio"/> Planification</p>	<p>Pilotage</p> <ul style="list-style-type: none"> <input type="radio"/> Elaboration d'un tableau de bord <input type="radio"/> Réalisation du produit 	<p>Cloture</p> <ul style="list-style-type: none"> <input type="radio"/> Produire un manuel utilisateur <input type="radio"/> Produire un document bilan de projet <input type="radio"/> Produire un document de soutenance <input type="radio"/> Faire la soutenance
--	--	--

Information sur la classe

Niveau :

Effectif :

Nombre de groupes :

Nombre d'heures :

Nombre d'heures par séance :

Figure 41. Formulaire de description de difficultés MAETIC via l'Email

5.2 Application Facebook

Pour offrir la possibilité d'interagir avec notre système via Facebook, nous avons développé une application Facebook. Pour ce faire, il faut développer une application externe avec un langage de programmation comme php, python, java, etc., et l'intégrer dans une iFrame sur Facebook dédiée à cette application et configurée auparavant. La figure 42 ci-dessous illustre cette solution.

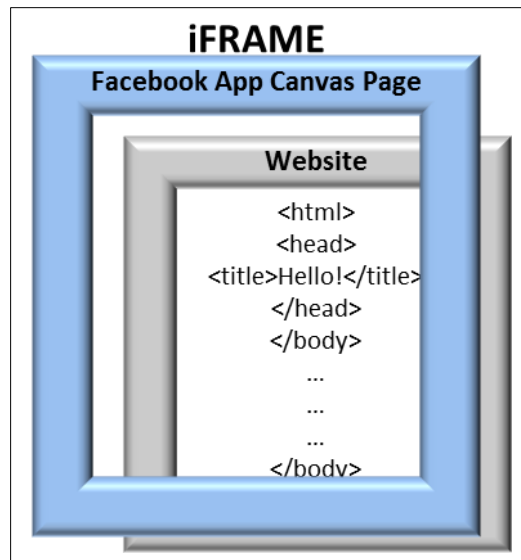


Figure 42. Intégration d'une application dans Facebook

Dans notre cas, nous développons une application basée sur le langage PHP, qui communique avec notre système via les services web. Ainsi, nous utilisons le PHP SDK qui est un outil permettant de manipuler les API de Facebook au sein de notre application.

Cette application assure l'accès aux principales fonctionnalités du SIC. En effet, elle offre les fonctionnalités suivantes :

- Publication d'une difficulté
- Publication d'un commentaire.
- Accès à l'ensemble des difficultés et des pratiques.
- Réception des notifications lors de la publication d'une difficulté ou d'un commentaire. La réception des notifications est opérationnelle même si l'application est non lancée.
- Accès à la couche RP.

Lorsqu'une difficulté ou un commentaire est publié, un membre intéressé par l'interaction via Facebook reçoit une notification dans son compte Facebook. Pour pouvoir envoyer des notifications aux utilisateurs de notre application Facebook, nous utilisons « Notifications API » qui est une API permettant d'envoyer des notifications aux utilisateurs d'une application Facebook. En consultant une notification, l'utilisateur récupère le contenu concerné via notre application Facebook. La figure 43 ci-dessous présente le processus de récupération d'un contenu via Facebook.

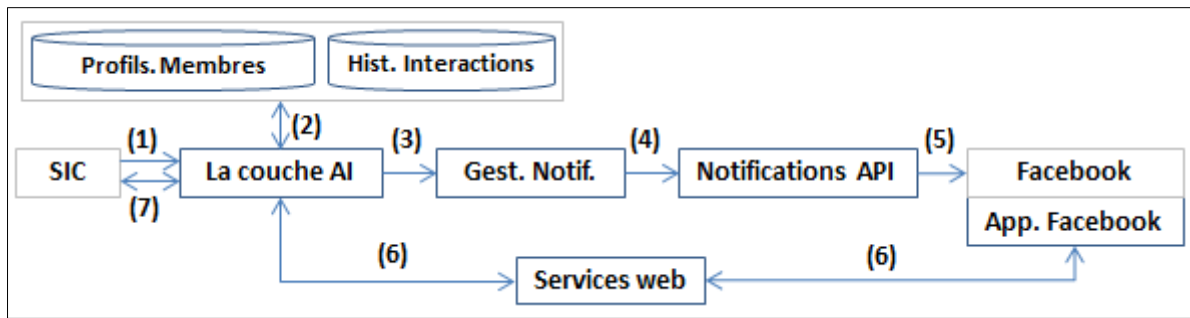


Figure 43. Processus de récupération d'un contenu via Facebook

Comme le montre la figure 43, après la publication d'un contenu, le système fait appel à la couche AI (1). Cette dernière sélectionne les utilisateurs à solliciter via Facebook en se basant sur leurs profils et les traces de leurs interactions (2), et transmet la liste de ces utilisateurs au composant 'Gest. notifications' (3). Ce dernier notifie les utilisateurs via 'Notifications API' (4)(5). En consultant la notification, l'utilisateur se redirige vers notre application qui s'occupe de lui afficher le contenu concerné par la notification. En effet, le contenu de la publication se transforme en objet JSON par la couche AI et se transmet via le composant 'services web' à l'application Facebook (6). La figure 44 ci-dessous montre un extrait du code permettant d'envoyer une notification à un utilisateur de l'application Facebook.

```

$facebook = new Facebook(array(
    'appId' => APPID,
    'secret' => APPSECRET,
)); } (1)

$access_token = $facebook->getAppId() . '|' . $facebook->getApiSecret();

foreach($list_fb_id as $fb_id){ } (2)
    try {
        $facebook->api('/' . $fb_id . '/notifications/', 'post', array(
            'access_token' => $access_token,
            'href' => $lien,
            'template' => $subject
        ));
    } catch (Exception $e) {
        echo $e->getMessage();
    }
} } (3)
  
```

Figure 44. Extrait du code permettant d'envoyer une notification à un utilisateur de l'application Facebook

Comme le montre cette figure, le composant « Gestion notifications » commence par créer une instance de la classe 'Facebook' en utilisant « APP. ID » et « APP. SECRET » qui permettent d'identifier notre application auprès de Facebook (1). La classe « Facebook »

fournit des fonctions permettant d'interagir avec Facebook. Après, il parcourt la liste des identifiants Facebook des membres concernés par cet outil (2). L'identifiant Facebook d'un membre est utilisé pour lui envoyer une notification (3) en spécifiant l'objet de la notification et le lien permettant de rediriger l'utilisateur vers la bonne page de l'application après avoir cliqué sur la notification.

Dans le sens inverse, c'est-à-dire pour répondre à une demande ou poster une difficulté via Facebook, l'application transmet le contenu à poster via le composant « services web » et la couche AI. La figure 45 ci-dessous présente le processus de publication d'un contenu.



Figure 45. Processus de publication d'un contenu via l'application Facebook

Comme le montre la figure 45, après avoir saisi et validé un contenu via l'application Facebook, il se transmet à la couche AI via le composant « Services web » (1) (2). La couche AI transforme le contenu (qui est sous forme d'un objet JSON) en objet Drupal pour être stocké (3). La figure 46 ci-dessous présente la page d'actualités de l'application Facebook.

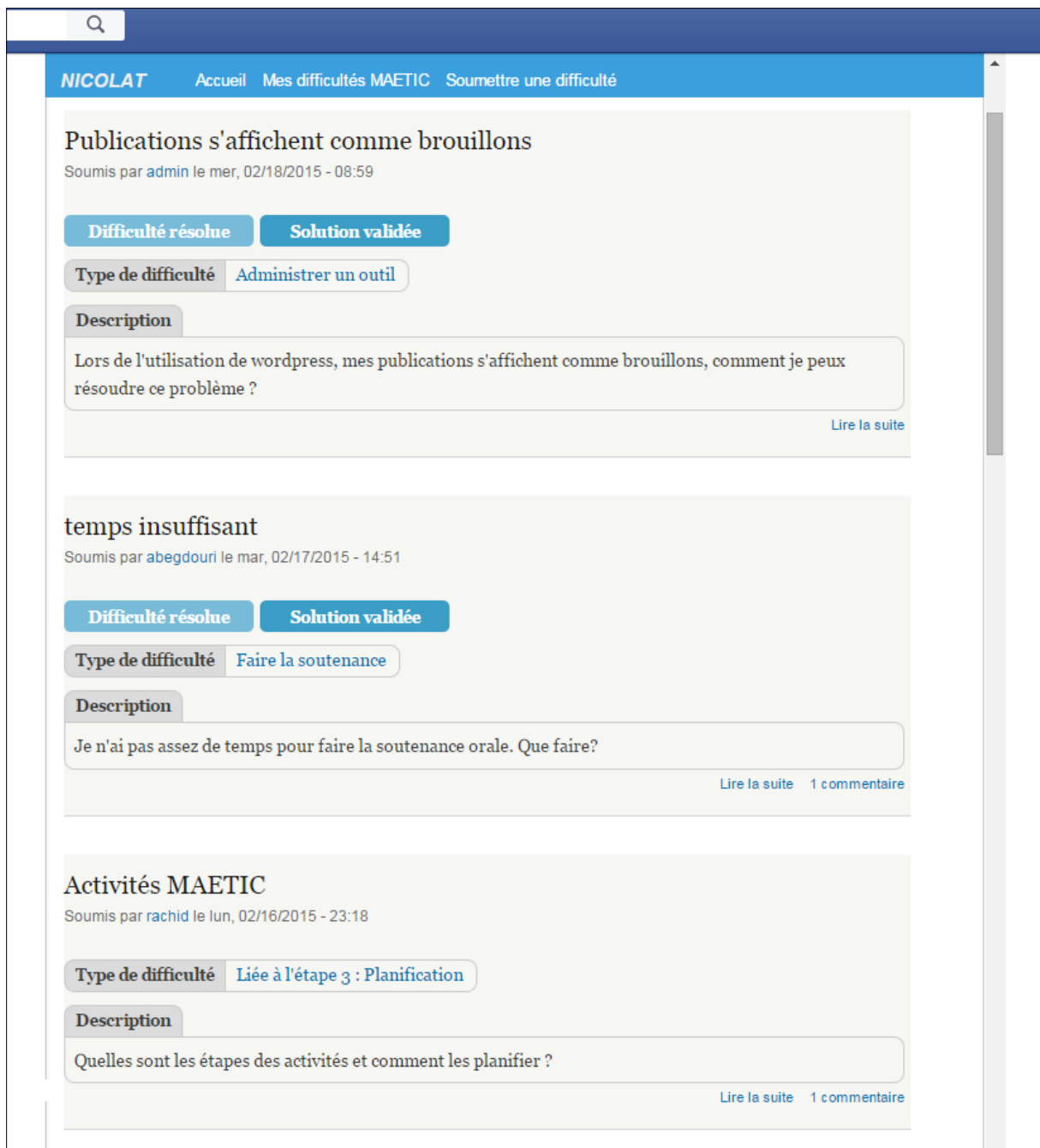


Figure 46. Page d'actualités de l'application Facebook

5.3 Application Android

Pour offrir la possibilité d'interagir avec notre système via un dispositif qui fonctionne sous Android, nous avons le choix entre trois solutions :

- Créer une interface web adaptative.
- Développer une application basée sur un Framework de développement mobile (comme Titanium Mobile ou Phone Gap)

- Développer une application Android native.

Dans notre contexte, nous avons choisi de développer une application Android native. En effet, les applications Android natives sont plus performantes et offrent une meilleure ergonomie puisqu'elles se basent sur des composants conçus spécifiquement pour les dispositifs Android. Pour développer notre application, nous avons utilisé le kit de développement (SDK) d'Android qui se base sur le langage Java. Les principales fonctionnalités auxquelles notre application offre l'accès sont :

- Publication d'une difficulté.
- Publication d'un commentaire.
- Accès à l'ensemble des difficultés et des pratiques.
- Réception des notifications lors de la publication d'une difficulté ou d'un commentaire. La réception des notifications est opérationnelle même si l'application est non lancée.

Lorsqu'une difficulté ou un commentaire est publié, chaque membre concerné reçoit une notification dans son outil préféré (ou utilisé au moment de la publication) si cet outil supporte la réception de notifications. Pour pouvoir envoyer des notifications aux membres utilisateurs des dispositifs Android, nous exploitons Google Cloud Messaging (GCM) qui est un service de Google permettant d'envoyer des notifications aux dispositifs des utilisateurs d'une application Android. GCM est le seul service qui offre cette fonctionnalité pour les dispositifs Android. En effet, pour utiliser ce service, il faut créer un projet sur « Google APIs Console » (<https://code.google.com/apis/console>), activer GCM sur l'interface d'administration du projet et récupérer le 'Sender Id' qui s'utilise pour identifier notre application et les dispositifs de ses abonnés auprès du serveur GCM.

Pour qu'un dispositif puisse recevoir des notifications après la publication d'une difficulté ou d'un commentaire sur notre système, il doit être enregistré auprès du GCM et de notre système. En effet, un utilisateur souhaitant utiliser notre application Android est obligé d'enregistrer, tout d'abord, son dispositif en cliquant sur le bouton nommé « Notifications » qui apparaît dans le menu principal de l'application si le dispositif est non enregistré. Après le clique sur ce bouton, le processus d'enregistrement commence. Les étapes de ce processus sont les suivantes (Figure 47) :

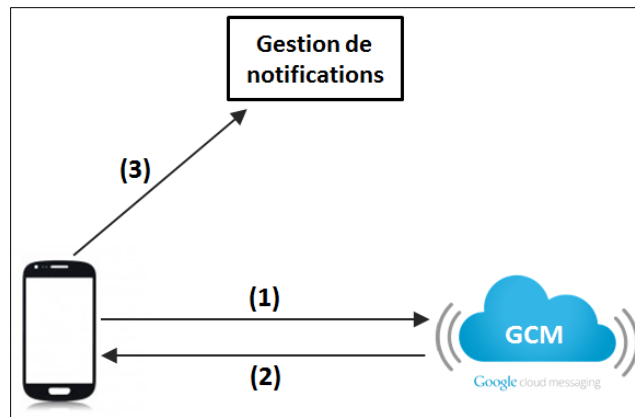


Figure 47. Processus d'enregistrement d'un dispositif

1. Le dispositif Android envoie le « Sender id » en utilisant la fonction `GCMRegistrar.register(Context context, String SENDER_ID)` de GCM API d'Android (c'est une fonction dédiée à l'enregistrement des dispositifs Android au niveau du serveur GCM), ce qui permet d'enregistrer le dispositif sur le serveur GCM.
2. Après l'enregistrement, un « registration id » sera envoyé au dispositif Android. Le « registration id » est un identifiant unique attribuer à un dispositif par GCM.
3. Après la réception de 'registration id', l'application utilise une requête POST pour l'envoyer au composant « Gest. notifications » qui le stocke avec les informations du profil du membre pour une utilisation ultérieure.

Après enregistrement, pour envoyer une notification à un utilisateur, le système envoie une requête au serveur GCM qui s'occupe de notifier le dispositif de l'utilisateur ciblé. En effet, le composant « Gest. Notifications » de notre système utilise une requête POST pour envoyer au serveur GCM un objet JSON contenant : le « registration id », le type du contenu publié (Difficulté ou commentaire) et son id (cet id sera utilisé après pour récupérer le contenu via les services web). Après, GCM envoie le type de contenu et son Id au dispositif de l'utilisateur en utilisant le « registration id ». Enfin, le dispositif ciblé récupère ces données par l'utilisation d'une implémentation de la classe `GCMBASEIntentService` de l'API GCM d'Android. Enfin, l'utilisateur récupère le contenu en consultant la notification reçue.

La figure 48 ci-dessous montre le code permettant de récupérer une notification par un dispositif Android.


```

public class GCMIntentService extends GCMBaseIntentService {

    @Override
    protected void onMessage(Context context, Intent intent) {
        String message = intent.getExtras().getString("maetic");
        String str[] = message.split(" ");
        String m = null;
        if(str[0].equals("d"))
            m = "Une difficulté a été rencontrée !!"; } (1)
        else
            m = "Un Commentaire a été publié !!";
        generateNotification(context, m ,str[1]); } (2)
    }

    private static void generateNotification(Context context, String message, String id) {
        NotificationManager notificationManager =
            NotificationManager(context.getSystemService(Context.NOTIFICATION_SERVICE)); } (3)

        Notification notification =
            new Notification(R.drawable.ic_launcher, message, System.currentTimeMillis()); } (4)
        String title = context.getString(R.string.app_name);

        Intent notificationIntent = new Intent(context, TabShowDiffActivity.class); } (5)
        notificationIntent.putExtra("id", id);
        PendingIntent intent = PendingIntent.
            getActivity(context, 0, notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT);
        notification.setLatestEventInfo(context, title, message, intent);

        notification.defaults |= Notification.DEFAULT_SOUND; } (6)
        notification.defaults |= Notification.DEFAULT_VIBRATE;

        notificationManager.notify(0, notification); } (7)
    }

    ...
    ...
}

```

Figure 48. Code permettant de récupérer une notification par un dispositif Android

Sur cette figure, la fonction « onMessage » permet de récupérer les informations sur la notification envoyée par le serveur GCM. Ainsi, elle prépare le message à afficher selon le type du contenu publié par la notification (difficulté ou commentaire) (1) et fait appel à la fonction « generateNotification » (2). Cette dernière crée une instance de la classe NotificationManger permettant de gérer les notifications (3). Après, elle crée la notification en spécifiant l'icône de la notification et le texte qui apparaît avec la notification (4) et elle définit la redirection au moment de la clique sur l'icône de la notification (5). Cette notification redirige vers l'activité permettant d'afficher la difficulté. Ensuite, elle spécifie que la notification génère un son et le dispositif vibre (6). Enfin, elle utilise la fonction « notify » de la classe « NotificationManager » pour lancer la notification (7).

La figure 49 ci-dessous présente le processus de récupération d'un contenu via un dispositif Android :

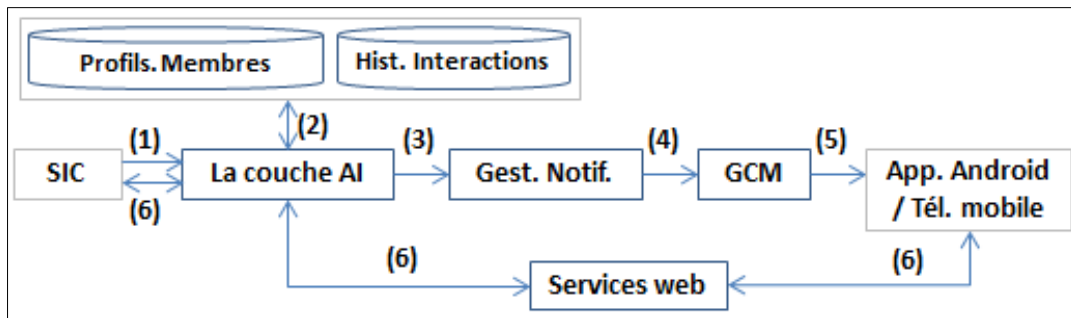


Figure 49. Processus de récupération d'un contenu via un dispositif Android

Comme le montre la figure 49, après la publication d'un contenu, le système fait appel à la couche AI (1). La couche AI sélectionne les utilisateurs à solliciter via leurs dispositifs Android en se basant sur leurs profils et les traces de leurs interactions (2) et transmet la liste de ces utilisateurs au composant « Gest. Notifications » (3). Le composant « Gest. Notifications » notifie les utilisateurs via GCM (4)(5). En consultant la notification, le contenu de la publication se transforme en objet JSON par la couche AI et se transmet via le composant « services web » à l'application Android qui s'occupe de le présenter à l'utilisateur (6). La figure 50 ci-dessous présente des captures d'écran montrant la réception d'une notification sur un dispositif Android.

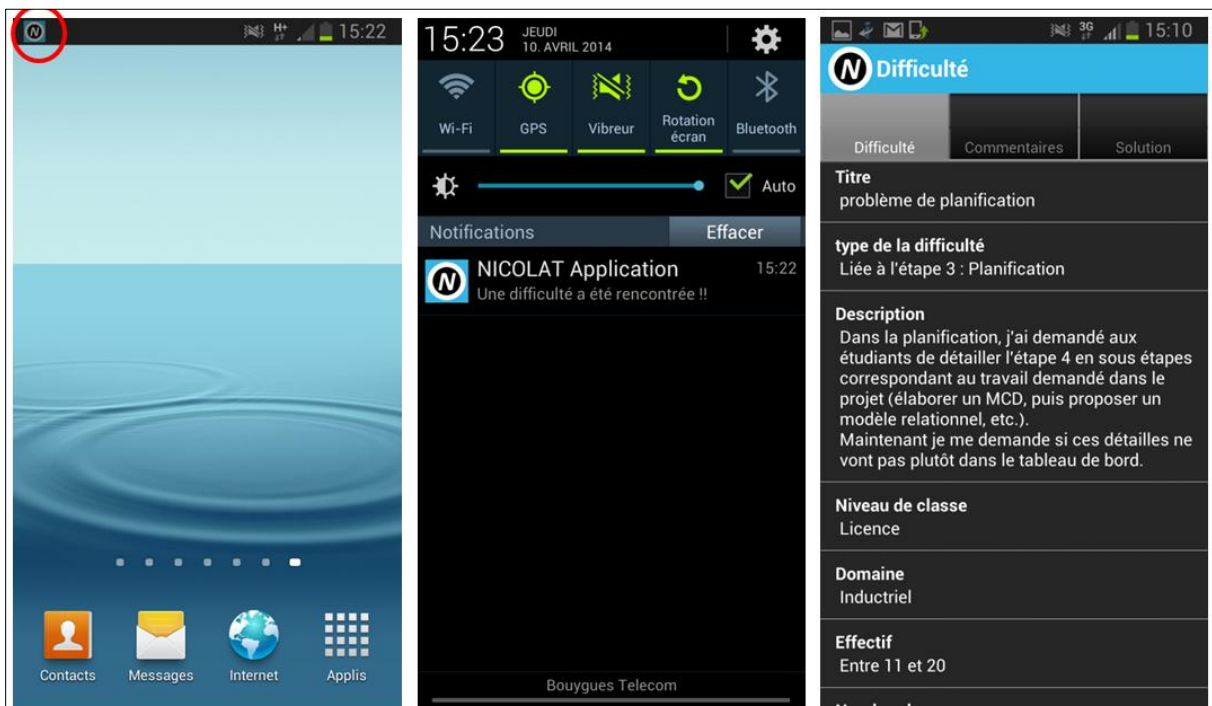


Figure 50. Réception d'une notification sur un dispositif Android

Dans le sens inverse, c'est-à-dire, pour répondre à une demande ou poster une difficulté, l'application transmet le contenu à poster via le composant « services web » et la couche AI. La figure 51 ci-dessous présente le processus de publication d'un contenu.



Figure 51. Processus de publication d'un contenu via un dispositif Android

Comme le montre la figure 51, après avoir saisi et validé un contenu via l'application Android, il se transmet à la couche AI via le composant « Services web » (1)(2). La couche AI transforme le contenu (qui est sous forme d'un objet JSON) en objet Drupal pour être stocké (3). La figure 52 ci-dessous montre des captures d'écran de l'application qui concerne le menu principal, le formulaire de description d'une difficulté et celui permettant de saisir les informations de la classe où la difficulté a été rencontrée.



Figure 52. Quelques interfaces de l'application Android

5.4 Application SMS

Pour qu'un membre puisse interagir avec les autres membres de la CoP via les SMS, nous avons développé le composant « Gest. SMS ». Ce dernier permet :

- D'envoyer les difficultés et les commentaires par SMS.
- De récupérer les SMS envoyés par les membres pour être stockés sur le système sous forme de commentaires.

Pour pouvoir envoyer et recevoir des SMS, ce composant exploite les services fournis par Octopush (<http://www.octopush-dm.com/>) qui est une plateforme d'envoi de SMS en mode SaaS (Software as Service). Le choix de cette plateforme a été effectué suite à une étude comparative de différentes solutions, d'envoi de SMS, existantes sur le marché. Cette étude s'est basée sur 3 exigences :

1. Celle des SMS à l'international. Les membres de notre CoP sont dispersés géographiquement (entre le Maroc et la France). Ainsi, nous avons besoin d'une solution qui supporte l'envoi des SMSs à l'international.
2. Celle de la réponse au SMS. Nous avons besoin d'une solution qui permet au membre de répondre à un SMS de demande d'aide par SMS. Cette fonction est importante pour la continuité des interactions avec les membres qui préfèrent ou qui sont obligés d'utiliser cet outil.
3. Celle de l'API PHP ou les services web. Notre système est développé sous le CMS DRUPAL (qui se base sur le langage PHP), donc une plateforme qui fournit un API PHP ou des services web permet de faciliter notre tâche en matière de développement et d'éviter les problèmes d'hétérogénéité avec le système.

Parmi les solutions qui répondent à ces 3 exigences nous avons choisi celle qui propose les meilleurs tarifs. La plateforme « Octopush » était donc la solution choisie.

Après la publication d'une difficulté, le système l'envoie par SMS aux membres intéressés par cet outil. La figure 53 ci-dessous présente le processus d'envoi d'une difficulté par SMS :

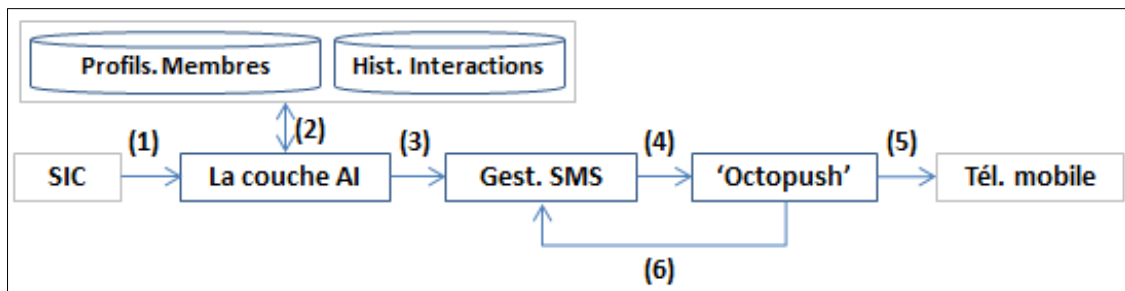


Figure 53. Processus d'envoi d'un contenu par SMS

Comme le montre la figure 53, le SIC notifie la couche AI de la publication d'une difficulté (1). La couche AI sélectionne les utilisateurs à solliciter par SMS en se basant sur leurs profils et les traces de leurs interactions (2). Ainsi, elle adapte le contenu pour qu'il soit possible de l'envoyer par SMS. Enfin, elle transmet le contenu adapté et la liste des utilisateurs sélectionnés au composant « Gest. SMS » (3). Le composant « Gest. SMS » fait appel à la plateforme « Octopush » en fournissant : la liste des numéros de téléphone des membres ciblés, le contenu à envoyer et le « API KEY » qui est l'identifiant de notre système auprès de cette plateforme (4). « Octopush » envoie les SMS (5) et transmet au composant « Gest.

SMS » (6) le « ticket_sms » qui sera utilisé lors de la réception des réponses. La figure 54 ci-dessous montre un extrait du code permettant d'envoyer des SMS aux membres concernés par cet outil et de tracer l'envoi.

```

if(count($sms_recipients)>0){ }- (1)

    $sms_type = "WWW";
    $sms_mode = INSTANTANE;
    $sms_sender = 'NICOLAT';

    $sms = new SMS(); }- (2)

    $sms->set_user_login($user_login);
    $sms->set_api_key($api_key);
    $sms->set_sms_mode($sms_mode);
    $sms->set_sms_text($sms_text);
    $sms->set_sms_recipients($sms_recipients);
    $sms->set_sms_type($sms_type);
    $sms->set_sms_sender($sms_sender); } (3)

    $xml = $sms->send(); }- (4)
    $start = '<ticket>';
    $end = '</ticket>';
    $ticket = strstr( substr( $xml, strpos( $xml, $start) + strlen( $start)), $end, true);

    db_insert('traces_sms')->fields(array(
        'text' => $sms_text,
        'sms_ticket' => $ticket,
        'node_id' => $node_id,
        'user_cible' => $id_user,
    ))
    ->execute();
}

```

Figure 54. Extrait de code permettant d'envoyer des SMS aux membres concernés par cet outil et de tracer l'envoi

Comme le montre cette figure, le composant « gestion SMS » commence par vérifier si la liste des numéros de téléphones des membres à solliciter par SMS « sms_recipients » n'est pas vide (1). Puis, il crée une instance de la classe SMS (2), fournit avec l'API permettant de gérer les SMS, et il spécifie ses paramètres (3) : le login, la clé, le type des SMS (international, national), le texte du SMS, la liste des numéros de téléphone, etc. Le SMS est envoyé avec la fonction « send » (4). Cette fonction retourne un fichier XML contenant des informations sur l'opération d'envoi. Parmi ces informations, nous trouvons « sms_ticket » qui permet d'identifier le SMS envoyé. Ce « sms_ticket » est stocké avec l'identifiant du contenu envoyé pour tracer cet envoi (5).

Après la réception d'une difficulté par SMS, l'utilisateur peut envoyer son commentaire en répondant au SMS reçu. La figure 55 ci-dessous présente le processus de récupération d'une réponse :

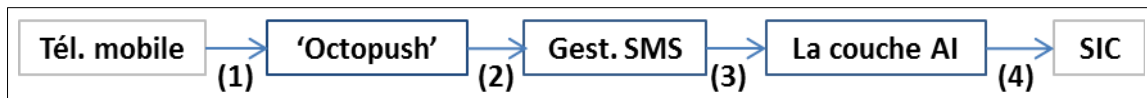


Figure 55. Processus de récupération d'un commentaire publié par SMS

Comme le montre la figure 55, l'utilisateur envoie un commentaire en répondant au SMS de demande d'aide (1). « Octopush » associe le « sms_ticket » à ce commentaire (réponse) et les transmet au composant « Gest. SMS » à travers une page web fournie par ce composant et dédiée à récupérer les réponses des SMS (2). Ce composant identifie l'utilisateur propriétaire du commentaire (en utilisant le numéro de téléphone) et le contenu (difficulté) concerné (en utilisant le « sms_ticket ») et fournit ce commentaire avec les identifiants de l'utilisateur et du contenu à la couche AI (3). La couche AI prépare alors le commentaire pour être stocké sur le système (4). La figure 56 ci-dessous montre le code permettant de recevoir un SMS envoyé par un membre.

```

$sms_ticket = $_POST['sms_ticket'];
$number = $_POST['number'];
$text = $_POST['text']; } (1)

$trace = db_select('traces_sms', 'n')
  ->fields('n')
  ->condition('sms_ticket', $sms_ticket, '=') } (2)
  ->execute()
  ->fetchAssoc();

$user = db_select('field_data_field_numero_de_tel', 'n')
  ->fields('n')
  ->condition('field_numero_de_tel_value', substr($number,1,strlen($number)) , '=') } (3)
  ->execute()
  ->fetchAssoc();

stocke_comment($trace["node_id"], $user["entity_id"], $text); } (4)
  
```

Figure 56. Code permettant de recevoir un SMS envoyé par un membre

Comme le montre cette figure, le composant « Gestion SMS » reçoit les informations sur le SMS de réponse (le commentaire) dans une requête POST effectuée par la plateforme « Octopush », ces informations sont (1) : le texte du SMS, le numéro du téléphone de son propriétaire et le ticket du SMS. Le ticket est utilisé pour connaître le contenu concerné (2) alors que le numéro de téléphone est utilisé pour connaître l'utilisateur propriétaire du SMS (3). L'identifiant de l'utilisateur et celui du contenu ainsi que le texte sont utilisés pour créer un nouveau commentaire (4).

Dans l'objectif de valider, notre adaptation des interactions, nous avons souhaité mesurer le degré de satisfaction des membres de la CoP par rapport aux passerelles technologiques que nous avons implémentées. Pour cela nous avons menée une expérimentation.

6. Expérimentation

Notre expérimentation s'est décomposé en six parties : le contexte, les hypothèses, la population cible, le protocole d'expérimentation, l'analyse des données recueillies ainsi que les préconisations ([Tricot 2007](#)). Par ailleurs, nous avons choisi des méthodes quantitatives ([Tricot and Lafontaine 2002](#)) ([Shute and Regian 1993](#)) en utilisant des outils d'analyse mathématiques et statistiques, mais également des méthodes qualitatives ([Taylor and Bogdan 1984](#)) qui produisent et analysent des données descriptives, tels les paroles écrites ou dites et le comportement observatoire des personnes.

6.1 Contexte

Notre expérimentation s'est déroulée à Fès au Maroc, dans les locaux de la Faculté des Sciences et Techniques, en Mars 2014, auprès de 13 enseignants de la FSTF. Au préalable, ces enseignants avaient reçu plusieurs formations sur la méthode MAETIC.

6.2 Hypothèses

L'objectif de l'expérimentation est de vérifier si la possibilité, offerte aux membres, d'interagir avec différents types d'outils augmente leur confort et contribue à la minimisation du temps de réponse à une demande d'aide.

Nous avons alors dégagé les deux hypothèses suivantes :

- Hypothèse 1 : la possibilité d'interagir via différents types d'outils augmente le confort des membres et permet de dépasser les problèmes de manipulation et de familiarisation avec les outils informatiques.
- Hypothèse 2 : le fait de supporter la mobilité et d'offrir la possibilité de solliciter chaque membre dans son outil de préférence permet de minimiser le temps de réponse à une demande d'aide à la résolution d'une difficulté et donc le temps de sa résolution.

6.3 Population cible

La population de cette expérimentation était composée de 13 enseignants de la FST de Fès au Maroc. Initialement, ces enseignants avaient été formés à la pédagogie active de groupe par projet et à l'usage de MAETIC lors d'une série des séances de formation débutées en 2012. Deux d'entre eux avaient déjà appliqué la méthode MAETIC dans l'un de leurs modules d'enseignement à la rentrée 2012. Cependant, ils confirmaient avoir rencontré des difficultés dans le respect de l'enchaînement des toutes les étapes de la méthode lors de l'implémentation de leurs e-mallettes MAETIC.

Parmi ces enseignants, six enseignaient d'informatique (l'un d'eux est spécialiste en réseaux et télécom), trois enseignaient de chimie, un enseignait de géologie, un de biologie, un de la gestion et un des langues et techniques de communication.

6.4 Description du protocole

Durant cette expérimentation, nous avons utilisé des méthodes quantitatives. Nous avons proposé d'administrer un questionnaire auprès de la population décrite ci-dessus. Ce questionnaire contenait des questions sur le confort et les habitudes d'usage par rapport aux outils d'interaction supportés (hypothèse 1) et la minimisation du temps de réponse à une demande d'aide (hypothèse 2).

Pour prouver les hypothèses 1 et 2, notre protocole d'expérimentation consistait à laisser les membres de la CoP utiliser les différents outils d'interaction supportés par le système. Ensuite, administrer les questionnaires à ces mêmes enseignants.

Le questionnaire est organisé en 4 parties. Chacune concerne un des outils d'interaction supportés : Email, Facebook, Android SMS. Par rapport à chaque outil, nous avons posé des questions concernant l'habitude du membre à utiliser cet outil, la facilité d'interaction avec notre système via cet outil et concernant la possibilité de répondre immédiatement lors de la réception d'une demande d'aide sur cet outil.

Ce questionnaire est donné dans l'Annexe 1.

6.5 Analyse des questionnaires

6.5.1 Hypothèse 1

Le questionnaire complété par les enseignants tend à confirmer la première hypothèse. En effet, 100% des enseignants affirment qu'ils ont l'habitude d'utiliser au moins deux des outils supportés :

- 100% utilisent l'email dont 92,30% trouvent que c'est facile d'interagir avec notre système via cet outil,
- 69,23% utilisent Facebook dont 66,66% trouvent que notre application Facebook est facile à manipuler (33,33% n'ont pas répondu),
- 53,84% utilisent des dispositifs Android ayant accès à Internet et trouvent que notre application Android est facile à manipuler,
- 84,61% utilisent des SMS pour communiquer.

La figure 57 ci-dessous présente le taux d'usage de chaque outil par les membres de la CoP.

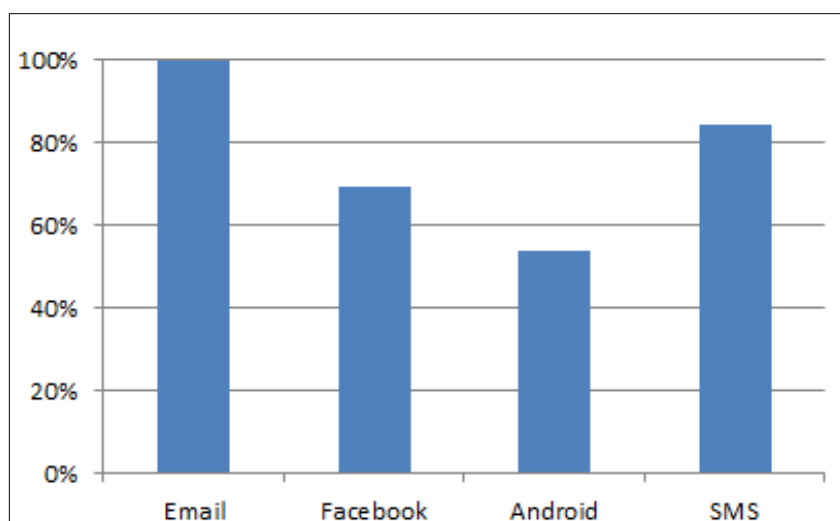


Figure 57. Taux d'usage de chaque outil par les membres de la CoP

Le tableau 8 ci-dessous résume les résultats obtenus concernant les préférences des enseignants et la facilité d'usage de chaque outil.

Email	<ul style="list-style-type: none"> • 92,30% trouvent que c'est facile d'interagir avec notre système via l'email. • 38,46% préfèrent l'usage de l'email par rapport à l'interface web, 46,15% préfèrent l'interface web. • 15,38% n'ont pas répondu.
Facebook	<ul style="list-style-type: none"> • 66,66% trouvent que notre applications Facebook est facile à manipuler alors que 33,33% non pas répondu. • 33,33% préfèrent l'application Facebook par rapport à l'interface web, • 33,33% préfèrent l'interface web et 33,33% n'ont pas répondu.
Android	<ul style="list-style-type: none"> • 100 % trouvent que l'application NICOLAT sur Android est facile à manipuler.
SMS	<ul style="list-style-type: none"> • 90,9% répondent par SMS, • 9,09% appellent le correspondant.

Tab 8. Résumé des résultats obtenus concernant les préférences des enseignants et la facilité d'usage de chaque outil.

5.5.2 Hypothèse 2

Le questionnaire complété par les enseignants tend à confirmer la deuxième hypothèse. En effet,

- 100% affirment qu'ils sont prêts à répondre immédiatement à une demande d'aide par rapport à au moins un outil :
 - 92,30% indiquent qu'ils répondent immédiatement s'ils reçoivent des demandes d'aide par email,

- 77,77% indiquent qu'ils répondent immédiatement s'ils reçoivent des notifications Facebook de la part de notre application,
- 11,11% n'ont pas répondu,
- 100% affirment qu'ils répondent immédiatement s'ils reçoivent une notification de la part de NICOLAT sur leurs dispositifs,
- Enfin 100% affirment qu'ils peuvent répondre immédiatement s'ils reçoivent une un SMS de la part de NICOLAT.

Le tableau 9 ci-dessous résume les résultats obtenus concernant les réponses immédiates des membres aux demandes d'aide via chaque outil.

Email	<ul style="list-style-type: none"> • 92,30% indiquent qu'ils répondent immédiatement s'ils reçoivent des demandes d'aide par email. • 69,23% consultent leurs boîtes email sur mobile, 66,66% d'eux répondent sur mobile.
Facebook	<ul style="list-style-type: none"> • 77,77% indiquent qu'ils répondent immédiatement s'ils reçoivent des notifications Facebook de la part de notre application. 11,11% ont indiqué non. 11,11% n'ont pas répondu. • 33,33% utilisent la version mobile de Facebook et indiquent qu'ils répondent immédiatement s'ils reçoivent une notification sur leurs dispositifs mobiles.
Android	<ul style="list-style-type: none"> • 100% affirment qu'ils répondent immédiatement s'ils reçoivent une notification de la part de NICOLAT sur leurs dispositifs Android.
SMS	<ul style="list-style-type: none"> • 100% affirment qu'ils peuvent répondre immédiatement s'ils reçoivent un SMS de la part de NICOLAT.

Tab 9. Résumé des résultats obtenus concernant les réponses immédiates des membres aux demandes d'aide via chaque outil

Les résultats de cette expérimentation sont encourageants. Ils ont montré que les outils que nous avons développés couvrent les préférences d'usage des membres en termes d'outils d'interaction. En fait, chaque membre a l'habitude d'utiliser au moins deux des outils supportés. Généralement, les utilisateurs de chaque outil trouvent que c'est facile d'interagir avec le système NICOLAT avec cet outil. Ils étaient donc satisfaits en termes de facilité d'usage. Ainsi, ils ont montré qu'ils sont motivés et prêts à répondre immédiatement aux demandes d'aide via les outils d'interaction qui correspondent à leurs préférences d'usage. En effet, la majorité des utilisateurs préférant l'usage d'un outil d'interaction affirment qu'ils répondent immédiatement s'ils reçoivent des demandes d'aide ou des commentaires sur cet outil. Ceci montre qu'offrir à chaque membre la possibilité d'interagir avec son outil de préférence amène à la minimisation du temps de réponse à une demande d'aide et ainsi, à un plus grand confort d'usage.

7. Généricité de l'adaptation des interactions

Notre approche d'adaptation des interactions vise à permettre aux membres de la CoP d'interagir entre eux via leurs outils d'interaction préférés. Cette approche ne se base sur aucune information liée au domaine de la communauté. Elle n'exploite que les informations liées aux outils d'interaction des membres (ce que nous avons appelé les profils technologiques des membres). Cette approche est donc générique. Ainsi, l'architecture d'adaptation que nous avons proposée est générique et ne nécessite aucune modification pour être utilisée par une autre communauté.

Du point de vue technique, les composants de ces couches d'adaptation sauf l'interface ne nécessitent pas de modification. En effet, ces composants s'occupent de transmettre et de récupérer des problèmes (et des commentaires) auprès de l'utilisateur selon la structure de problème prédéfinie au sein du noyau. Cependant, les membres d'une autre communauté peuvent avoir des préférences d'usage différentes de celles des membres de la CoP MAETIC. Il sera donc nécessaire de développer de nouvelles applications pour supporter ces nouveaux outils. Dans ce cadre, ces applications pourront exploiter le composant « services web » pour échanger les données avec le système.

8. Conclusion

Dans ce chapitre, nous avons décrit notre proposition visant à supporter les échanges/interactions entre les membres de la CoP via leurs outils d'interaction préférés et utilisés au quotidien. Nous avons commencé par présenter un état de l'art sur l'adaptation dans les EIAH et la gestion du contexte. Puis, nous avons présenté notre modélisation du contexte. Ensuite, nous avons présenté notre approche d'adaptation des interactions des membres. Nous avons détaillé deux couches d'adaptation : la couche « Adaptation des interactions » et la couche « Interface ». Puis, nous avons abordé le fonctionnement de notre système avec les outils que nous supportons : Email, Facebook, SMS et les dispositifs Android. Une expérimentation ayant pour but d'évaluer notre approche a ensuite été détaillée. Les résultats de cette expérimentation sont satisfaisants. Nous avons conclu ce chapitre par une discussion de la généricité de l'approche d'adaptation proposée.

Lors de la présentation des couches d'adaptation, précisément la couche AI, nous avons exposé le composant « Orchestration de l'adaptation ». C'est le composant responsable de la prise des décisions par rapport à l'adaptation. L'une des principales fonctions de ce composant est de sélectionner les membres à solliciter lors de la publication d'un problème. L'approche de sélection des membres implémentée par ce composant fait l'objet du chapitre suivant.

Chapitre 5 : La sélection des membres dans le système NICOLAT

Ce chapitre aborde notre proposition concernant la sélection des membres qui peuvent contribuer positivement à résoudre un problème afin de leur faire acheminer une demande d'aide. Dans l'état de l'art du chapitre précédent, nous avons pu constater que l'un des domaines de recherche clés dans les systèmes adaptatifs support de l'apprentissage collaboratif est la formation des groupes ([Magnisalis, Demetriadis et al. 2011](#)). Nous commençons donc ce chapitre par une présentation des travaux existants dans ce domaine. Cette présentation permettra de situer notre approche de sélection des membres, que nous considérons comme un problème de formation de groupes d'apprenants autour d'un problème posté. Puis, nous détaillons notre approche de sélection des membres avec ses deux composantes statique et dynamique. Enfin, nous concluons par un exposé des résultats de l'expérimentation par simulation que nous avons menée pour valider notre approche.

1. Formation des groupes

1.1 Formation des groupes dans les systèmes d'apprentissage collaboratif

Selon Lipponen ([Lipponen 2002](#)), l'apprentissage collaboratif supporté par les TICs est axé sur la manière avec laquelle la collaboration et les technologies peuvent, d'une part, améliorer les interactions et le travail de groupe, et d'autre part, faciliter le partage et la diffusion des connaissances et de l'expertise parmi les membres de la communauté. Dans ce contexte, l'apport des TICs peut être vu sous la forme de l'intégration, de fonctionnalités adaptatives et intelligentes, aux systèmes d'apprentissage collaboratifs.

En effet, un système d'apprentissage adaptatif est un système qui vise à adapter une partie de ses fonctionnalités, comme la présentation de contenu ou le support à la navigation, pour s'adapter aux besoins et aux préférences de l'apprenant. Un système d'apprentissage intelligent vise à fournir à l'apprenant un support pendant le processus de résolution de problème comme l'aurait fait un tuteur humain ([Brusilovsky and Peylo 2003](#)).

Cependant, il existe des systèmes d'apprentissage collaboratif qui combinent les deux fonctionnalités d'adaptation et d'intelligence. Ces dernières peuvent être utilisées pour la formation de groupes, pour fournir un support lié au domaine enseigné ou pour supporter les interactions entre les pairs ([Magnisalis, Demetriadis et al. 2011](#)).

La formation de groupes dans un système d'apprentissage collaboratif s'avère importante pour rendre l'apprentissage plus efficace, favoriser la collaboration et augmenter la

productivité des échanges. En effet, Chen considère la formation de groupes comme cruciale pour générer des interactions fructueuses ([W. Chen 2006](#)).

L'état de l'art concernant la formation de groupes montre que la plupart des systèmes proposent de former des groupes hétérogènes en se basant sur les styles d'apprentissage des apprenants. L'idée derrière cela est que l'hétérogénéité du groupe favorise des interactions efficaces entre les pairs ([Alfonseca, Carro et al. 2006](#); [Martín and Paredes 2004](#); [Paredes and Rodriguez 2006](#)). Les données de l'apprenant sont généralement représentées par rapport à un ensemble d'attributs et sont récupérées en utilisant son profil et/ou des questionnaires. Il existe 3 stratégies permettant de former des groupes par rapport aux valeurs de ces attributs :

1. Groupement homogène : les valeurs des attributs pour les apprenants du même groupe doivent être homogènes.
2. Groupement hétérogène : les valeurs doivent être hétérogènes.
3. Groupement mixte : les valeurs de certains attributs doivent être homogènes, pour d'autres elles doivent être hétérogènes.

Le groupement homogène est effectué en utilisant des techniques de classification ou des méthodes spécifiques à l'application basées sur des calculs de similarité. Ces mêmes techniques peuvent être adoptées dans le cadre d'un groupe hétérogène en créant tout d'abord des groupes homogènes, et en prenant ensuite un apprenant de chaque groupe pour former un groupe hétérogène. Généralement, les groupements, hétérogènes et mixtes, sont considérés comme des problèmes d'optimisation. Des méthodes d'optimisation, tel les algorithmes génétiques, sont alors utilisées pour trouver les meilleures combinaisons possibles.

Les travaux présentés ci-dessous exposent différents travaux qui traitent de la formation des groupes dans les systèmes d'apprentissage collaboratifs.

TANGOW ([Martín and Paredes 2004](#)) est un système d'apprentissage web qui propose aux apprenants des activités individuelles et collaboratives. Ce système exploite le modèle de style d'apprentissage de Felder-Silverman ([Felder and Silverman 1988](#)), et un questionnaire « Index of Learning Styles (ILS) » associé à ce modèle, pour définir la présence ou l'absence d'activités de collaboration chez certains types d'élèves. La formation de groupes ici a pour objectif d'augmenter la productivité du groupe et consiste à mettre, avec des pourcentages égaux, des apprenants ayant un style d'apprentissage « réflexive » avec des apprenants ayant un style « actif ».

NUCLEO ([Sancho, Fernandez et al. 2008](#)) est un système qui simule un monde virtuel présenté sous forme d'un jeu de rôle dans lequel les apprenants sont groupés en équipes pour réaliser des missions de travail. Ce système se base sur le modèle de style d'apprentissage de Vermunt ([Vermunt 1992](#)) pour former les groupes d'apprenants. L'objectif est de rassembler au sein d'un même groupe des apprenants ayant des styles d'apprentissage complémentaires.

AUTO-COLLEAGUE ([Tourtoglou and Virvou 2010](#)) est un système qui permet l'apprentissage du langage UML. Le système utilise un modèle de l'apprenant fondé sur : 1) le niveau d'expertise qui décrit la connaissance d'UML, 2) les performances du comportement de l'apprenant et 3) la personnalité, qui est en relation avec des caractéristiques qui influencent le comportement de l'apprenant. Le système se base alors sur un algorithme génétique appelé « simulated annealing » pour trouver les meilleures compositions de groupes.

WikiClassroom ([Sawant, Singhal et al. 2011](#)) est un wiki collaboratif, qui offre à ses apprenants une interface, pour écrire et réviser des pages ainsi qu'un forum pour discuter de leurs idées durant la collaboration. Le système trace les activités de l'apprenant pour construire un modèle qui représente les contributions à l'égard du groupe. Ce modèle caractérise l'utilisation du système par l'apprenant et se compose de 4 attributs : 1) active-use : caractérisant les actions de l'apprenant, 2) passive-use : caractérisant la prise d'information, 3) interaction : caractérisant les échanges avec les autres, et 4) évaluation : représentant le score obtenu concernant des activités liées au wiki. L'apprenant possède un score par rapport à chacune de ces catégories. Dans le but de former des groupes hétérogènes, le système associe à chaque apprenant un agent-intelligent qui utilise les réseaux bayésiennes pour faire une estimation probabiliste de la contribution à l'égard d'un groupe.

MATHEMA ([Papadimitriou, Grigoriadou et al. 2012](#)) est un système hypermédia adaptatif qui vise à supporter l'apprentissage de l'électromagnétisme individuellement et/ou en collaboration. Pour former les groupes, ce système se base sur un style d'apprentissage prédéfini et un niveau de connaissance lié à l'objectif d'apprentissage. Il présente alors à l'apprenant une liste de collaborateurs (potentiels et les plus recommandés) triée par rapport aux styles d'apprentissage. Les apprenants ayant le même style d'apprentissage sont triés par rapport au niveau de connaissance lié à l'objectif de l'apprentissage. Les groupes sont constitués manuellement et c'est l'apprenant qui choisit avec qui collaborer.

Les travaux de Brauer et Schmidt ([Brauer and Schmidt 2012](#)) s'intéressent à un réseau social d'apprentissage dans lequel l'utilisateur décide d'initier un travail collaboratif. Le système recherche alors les candidats et fait des propositions de groupes. La formation de groupes se base sur : 1) la disponibilité, 2) le style d'apprentissage, et sur 3) le modèle de connaissances représenté par des tags pondérés. L'objectif est de sélectionner les membres ayant un style d'apprentissage commun, un score élevé par rapport aux connaissances liées au sujet proposé, et une distance faible dans le réseau par rapport à l'initiateur. Pour trouver les meilleures solutions possibles, un algorithme génétique est utilisé.

Dans les travaux de Jagadish ([Jagadish 2014](#)) un algorithme de classification K-NN a été utilisé pour classifier les apprenants dans des groupes, en se basant sur les traits de leurs personnalités et leurs styles d'apprentissage. Les groupes d'apprenants créés sont destinés à

échanger via le module de chat de MOODLE à propos d'un cours et/ou d'activités particulières. Les données sur lesquelles se base l'algorithme sont issues des profils des apprenants et de questionnaires complétés par eux-mêmes.

Enfin, Duque et Gómez-Pérez ([Duque, Gómez-Pérez et al. 2015](#)) proposent à l'enseignant une méthode lui permettant de former des groupes d'apprenants, homogènes ou hétérogènes, selon ses propres critères basés sur la manière avec laquelle ces apprenants résolvent les tâches académiques qu'il leur a confié. Cette méthode utilise un ensemble d'indicateurs représentant comment ces apprenants résolvent les tâches académiques. Le système analyse leur collaboration et interactions et utilise le concept de « Data depth » pour mesurer la similarité entre les valeurs de ces indicateurs dans le but de former des groupes. Il faut avoir spécifié des indicateurs hétérogènes et/ou homogènes par l'enseignant avant de pouvoir utiliser la méthode. Les valeurs des indicateurs utilisées peuvent être construites à partir d'actions effectuées par les apprenants ou déduites à partir des questionnaires qu'ils ont complétés et saisis manuellement.

1.2 Formation de groupes dans le système NICOLAT

La minimisation du temps de réponse et l'augmentation de l'accessibilité à l'environnement à travers l'adaptation des interactions que nous avons réalisée, dans le cadre de la couche AI, représente un point fort de NICOLAT. En effet, il permet d'encourager la collaboration et d'augmenter la motivation pour l'entre-aide et donc pour l'apprentissage parmi les membres de la CoP.

Cependant, selon l'expérimentation de NICOLAT menée auprès des membres de la CoP MAETIC, nous avons collecté un certain nombre de constatations qui auraient un effet négatif sur la collaboration entre les membres de la CoP :

- Tous les membres ne sont pas capables de résoudre tout type de problèmes, principalement, lorsqu'il s'agit de problèmes spécialisés lorsqu'ils sont postés par des membres plus expérimentés.
- Un membre de la CoP est démotivé lorsqu'il est sollicité pour résoudre un problème alors qu'il n'en est pas capable. Ainsi, un membre peut souhaiter recevoir une réponse le plutôt possible afin de ne pas être bloqué. Or, dans une CoP, tous les membres ont accès à tous les échanges sur l'espace communautaire. Ainsi, au lieu de recevoir des réponses par rapport à la demande d'aide qu'il a posté, il se voit impliqué dans la résolution de problèmes auxquels il ne peut pas donner une contribution positive.
- L'inondation de l'espace personnel d'un membre par des notifications inutiles. Selon le principe d'adaptation des interactions de NICOLAT, une simple publication d'un problème, ou d'un commentaire dans le fil de discussion correspondant, génèrent un

grand nombre de notifications à tous les membres de la CoP dans tous les outils d'interaction qu'ils utilisent. En effet, un membre en contact permanent avec les activités de la CoP va recevoir, au moment de la publication d'une difficulté, une notification via Facebook, via l'application appropriée sur son smartphone et sur sa tablette. Il recevra aussi une notification par SMS et par Email.

Nous avons proposé dans NICOLAT une interface qui permet aux membres de restreindre leur disponibilité selon leurs préférences par rapport aux outils d'interaction utilisés. Cependant, une grande restriction rend le système plus ou moins fermé, ce qui va à l'encontre du principe de base d'ouverture et de communication dans les CoPs.

Comme solution, gardant les avantages d'ouverture et d'accessibilité, et diminuant les effets de démotivation cités plus haut, nous proposons de former des groupes homogènes de membres de la CoP, à qui faire aboutir une demande d'aide lorsqu'elle est postée. Cet ensemble sera choisi parmi les membres, que nous jugeons, les plus aptes à contribuer positivement à résoudre la difficulté en question.

Notre approche de sélection des membres a donc pour objectif de répondre à la cinquième question de notre problématique de recherche : **Comment assurer l'obtention, par l'enseignant cherchant à résoudre son problème, d'une solution la plus adaptée à son contexte à travers ses interactions avec les autres membres ?**

Pour ce faire, nous utilisons, en premier lieu, des règles statiques modélisant les connaissances permettant de définir les membres capables, théoriquement, de résoudre un problème.

Cependant, ces membres peuvent ne pas être actifs dans la CoP alors que d'autres membres, non sélectionnés, peuvent être très actifs et auraient pertinemment contribué à résoudre des problèmes similaires postés dans le passé. De ce fait, ils peuvent être capables et très appropriés pour participer à la résolution du problème actuel. L'analyse des anciennes contributions des membres ne peut pas se faire à travers des règles statiques, une sélection dynamique s'impose.

En résumé, nous considérons un membre comme capable de contribuer positivement à résoudre un problème dans les cas suivants :

- Lorsqu'il est sélectionné à l'aide des règles statiques.
- Dynamiquement, par analyse de ses traces de contribution lorsqu'il n'est pas sélectionné par ces règles mais qu'il est actif dans la CoP et a participé à résoudre des problèmes similaires dans le passé.

Dans les travaux de formation de groupes existants et présentés dans le paragraphe 1 de cette section ([Martín and Paredes 2004](#)) ([Sancho, Fernandez et al. 2008](#)) ([Tourtoglou and Virvou](#)

2010) ([Sawant, Singhal et al. 2011](#)) ([Papadimitriou, Grigoriadou et al. 2012](#)) ([Brauer and Schmidt 2012](#)) ([Jagadish 2014](#)) ([Duque, Gómez-Pérez et al. 2015](#)), les données des membres sont représentées sous forme de vecteurs. Les groupes, sont formés de manière synchrone à l'aide d'algorithmes de clustering pour les groupes homogènes et d'algorithmes génétiques par exemple pour les groupes non homogènes.

Dans notre cas, nous nous intéressons à former un seul groupe à la fois autour d'un problème posté. Nos groupes sont donc formés de manière asynchrone alors qu'un membre de la CoP peut appartenir à plusieurs groupes en même temps. Les données de l'apprenant sur lesquelles nous nous basons pour la formation du groupe constituent alors une liste de traces de contributions et non pas des données représentées sous forme de vecteur de valeurs. Par conséquent, nous ne pouvons pas adopter l'une des solutions exposées dans les travaux précédents.

En plus de ces deux types de sélection, nous offrons au membre ayant posté le problème une sélection directe et basique. En effet, nous postulons que les interactions entre les membres de la CoP leur permettent d'avoir des aperçus sur les compétences et les points forts de chacun d'eux. Avec le temps le membre pourra connaître qui sont les membres qui peuvent aider à résoudre des types spécifiques de problèmes. Nous offrons donc au membre émetteur du problème, lors de la saisie de ce problème, la possibilité de tagguer un ensemble de membres à solliciter directement. Le membre émetteur du problème peut donc choisir, lors de la saisie de son problème, les membres à solliciter. La liste des membres sélectionnés sera complétée par les sélections statique et dynamique. La figure 58 ci-dessous illustre les étapes du processus de la sélection des membres à solliciter.

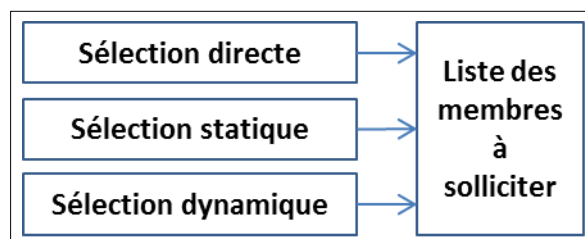


Figure 58. Types de sélections des membres offertes par NICOLAT

Comme le montre cette figure, le membre émetteur du problème commence par effectuer la sélection directe. Il choisit donc manuellement des membres à solliciter. Après, le système intervient pour effectuer les sélections statique et dynamique. Les membres sélectionnés pendant une de ces étapes sont ajoutés à la liste des membres à solliciter.

Nous présentons dans le paragraphe suivant le détail de notre approche de formation des groupes basée sur les deux types de sélection : statique et dynamique.

2. Approche de sélection

2.1 Principe de l'approche de sélection

La sélection statique consiste à faire correspondre les profils de tous les membres de la CoP au contexte du problème cible et au profil de son émetteur pour sélectionner seulement les membres qui ont des capacités permettant de résoudre le problème. Pour atteindre cet objectif, nous utilisons un ensemble de règles permettant de vérifier l'aptitude d'un membre à contribuer positivement à résoudre un problème particulier par rapport aux caractéristiques du problème et du profil de son émetteur.

La sélection dynamique consiste à analyser les anciennes activités des membres liées à des problèmes rencontrés dans des contextes similaires à celui du problème cible pour sélectionner seulement ceux qui étaient les plus pertinemment actifs dans des contextes similaires.

Les sélections, statique et dynamique, sont complémentaires. En effet, au moment de la publication du problème, la sélection statique permet de définir un ensemble de membres à solliciter en se basant sur leurs profils, alors que la sélection dynamique évalue les activités passées des membres non sélectionnés lors de la sélection statique dans le but de compléter la liste des membres à solliciter par des membres ayant des contributions pertinentes et des profils actifs.

Les contributions des membres sont sous forme de commentaires postés dans le fil de discussion suite à la publication d'un problème. Pour pouvoir mesurer la pertinence de ces commentaires, nous offrons au membre émetteur du problème la possibilité de distinguer les commentaires qui l'ont aidé à le résoudre. Dans des fichiers, que nous appelons fichiers trace, nous gardons les traces de ces contributions. Une trace de contribution correspond à une trace du contexte du problème auquel la contribution est liée.

Les traces de contribution ne concernent que les commentaires jugés pertinents par les membres émetteurs des problèmes. De plus, nous considérons une autre dimension de la pertinence liée aux paramètres du contexte du problème. En effet, nous considérons que l'importance de ces attributs change d'un type de problèmes à un autre.

Pour représenter ce changement d'importance des attributs, nous attribuons un poids à chaque attribut du contexte pour chaque type de problèmes. Un poids est un nombre binaire qui caractérise le degré d'importance d'un attribut par rapport aux problèmes d'un type donné. La valeur 0 signifie que l'attribut correspondant est inutile pour les difficultés de ce type alors que la valeur 1 signifie qu'il est pertinent.

Si nous prenons l'exemple de la CoP MAETIC, la figure 59 ci-dessous montre les poids attribués aux différents attributs du contexte de difficultés par rapport au type de difficultés

« Réalisation du produit ». Dans ce cas par exemple, les attributs « Compétence GP » et « Nb d'Heures global » sont pertinents pour les difficultés de ce type alors que l'attribut « Compétence Blog » est inutile. Nous notons que les attributs du contexte des difficultés MAETIC sont présentés dans le paragraphe 3.2 du chapitre 4.

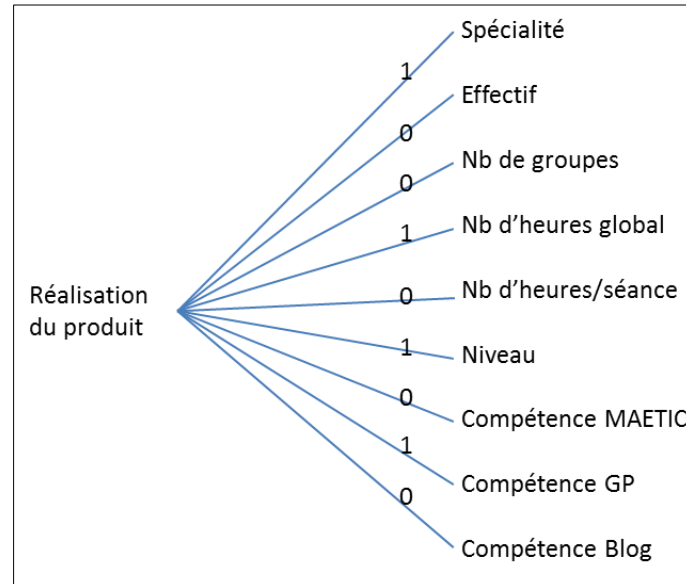


Figure 59. Les poids des attributs du contexte pour le type de difficulté « Réalisation du produit »

Lors de l'analyse des traces, un membre est sélectionné s'il a un nombre suffisant de contributions rattachées à des problèmes similaires au problème cible en termes de contexte. Ce nombre doit être supérieur ou égal à un minimum que nous appelons seuil de contributions (SeuilContributions). Ce Seuil est défini par les experts du domaine cible pour chaque type de problème. Par exemple, dans le cadre de la CoP MAETIC, si $\text{SeuilContributions}_{\text{SPlanification}} = 6$, le membre qui sera sélectionné pour participer à la résolution d'une difficulté de type « planification », doit avoir eu dans le passé au minimum 6 contributions à résoudre des difficultés similaires à la difficulté cible.

Ainsi, la sélection dynamique consiste à parcourir la liste des membres de la CoP pour extraire, pour chacun d'entre eux, la liste des problèmes similaires au problème cible et auxquels il a apporté des contributions pertinentes. Si le nombre de ces dernières est supérieur ou égal au Seuil de contribution lié au type du problème cible, le membre sera sélectionné.

Une solution naïve consiste à parcourir les traces de contribution linéairement pour détecter celles associées à des problèmes similaires au problème cible en termes de contexte, et de calculer leur nombre. Cette solution pose des problèmes liés essentiellement au temps d'exécution. En effet, dans un environnement dédié à la résolution communautaire de problèmes, le nombre de commentaires et donc le nombre de traces de contributions à

analyser augmentera avec le temps. Par conséquent, le temps d'exécution des algorithmes cherchant à analyser ces contributions, d'une façon linéaire, augmente considérablement.

Pour remédier à ceci, nous proposons de faire un traitement préliminaire permettant de limiter le nombre de traces à analyser avant d'exécuter l'algorithme d'analyse des traces proprement dite. Ainsi, nous utilisons deux algorithmes complémentaires :

- Un algorithme de filtrage, qui consiste à éliminer certains membres de la liste des candidats potentiels à être sélectionnés, que nous jugeons, non capables de participer à la résolution du problème cible. Nous réalisons cette analyse sans parcourir les fichiers trace mais en se basant sur des indices de contribution décrivant ces fichiers (voir paragraphe 3.3). L'algorithme utilise des opérations de recherche dans des arbres représentant ces indices de contribution, le nombre de contributions dans ce cas n'influence pas le nombre d'opérations de recherche.
- Un algorithme d'analyse des traces qui consiste à faire une analyse optimale des traces après filtrage en se basant sur la représentation numérique de ces dernières (voir paragraphe 3.5).

Dans ce qui suit, nous présentons, tout d'abord, l'implémentation de la sélection statique, avant d'aborder, dans la section 3, la sélection dynamique en présentant la représentation des données et la mesure de similarité que nous avons adoptées et en détaillant les deux algorithmes de la sélection dynamique : l'algorithme de filtrage et celui de l'analyse des traces. Les sélections, statique et dynamique, sont présentées par rapport à la CoP MAETIC. En effet, nous utilisons les contextes des difficultés MAETIC et les profils des enseignants MAETIC pour expliquer nos solutions. A la fin de chaque partie nous discutons la généralité des solutions proposées.

2.2 Sélection statique

Pour assurer l'obtention de propositions de solutions adaptées au membre cherchant à résoudre son problème, nous effectuons une comparaison entre les profils des membres de la communauté d'un côté, la difficulté cible et le profil de son émetteur d'un autre côté, dans le but de sélectionner ceux qui peuvent fournir de l'assistance. Pour atteindre cet objectif, nous concevons un ensemble de règles modélisant la connaissance permettant de vérifier l'aptitude d'un membre à contribuer positivement à résoudre une difficulté particulière par rapport aux caractéristiques de la difficulté et du profil de son émetteur.

Théoriquement, une règle s'exprime sous forme d'un triplet <événement, condition, action>. Lorsque l'évènement se produit, la condition est évaluée et si elle est vraie, l'action est effectuée. Dans notre cas, les éléments de ce triplet sont :

- L'événement, qui déclenche la vérification de la condition. Il n'y a qu'un seul événement qui est la publication d'une difficulté. En effet, toutes les règles sont associées à cet événement.
- Condition, qui correspond à des critères définis par rapport à la difficulté cible et au profil pédagogique de son émetteur. Nous organisons ces critères dans des ensembles. Nous notons chaque ensemble un élément « IN ». Si la difficulté cible et le profil de son émetteur satisfont l'ensemble des critères d'un seul élément de type « IN », la règle est applicable.
- Action, qui correspond à la décision de sélection des membres à solliciter. Nous définissons des critères qui guident cette sélection sous forme d'ensembles. Nous notons chaque ensemble de critères « OUT ». Ces critères concernent les profils pédagogiques des enseignants.

La figure 60 ci-dessous montre la structure générale de nos règles.

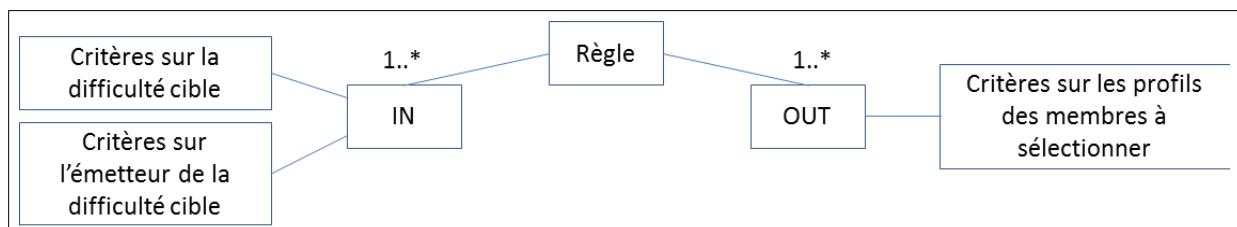


Figure 60. Structure de règles

Pour représenter les règles selon notre structure (Figure 60), nous avons défini un formalisme basé sur des balises XML. En effet, nous définissons une balise parente nommée « rule » qui regroupe les éléments nécessaires pour décrire une règle. Elle contient des balises « in » et « out » qui correspondent, respectivement, aux éléments « IN » et « OUT » de la règle. Chaque balise « in » ou « out » contient un groupe de critères. Un critère correspond à une balise portant le nom d'un attribut du contexte et contenant la valeur qu'il doit avoir.

Dans les balises « in », les critères sont regroupés dans deux sous-balises « problème » et « membre ». Le but est de pouvoir distinguer entre les critères liés à la difficulté elle-même et ceux liés à son émetteur.

Le choix de stocker les règles de cette manière a été effectué dans le but de répondre à deux principales contraintes :

- L'évolutivité des règles. L'analyse des interactions entre les membres pourra donner lieu à l'identification de nouvelles règles. Ainsi, la nécessité de la prise en compte de nouveaux domaines s'impose.
- La maintenance des règles. Après l'analyse des interactions entre les membres, certaines règles pourront nécessiter des modifications.

A travers cette solution, nous pouvons ajouter de nouvelles règles ou modifier celles existantes facilement et sans avoir besoin de toucher au code du système exploitant ces règles. Nous pouvons même développer des interfaces de visualisation de ces règles pour permettre aux experts MAETIC de les manipuler facilement.

Prenons la règle suivante. *Si un enseignant rencontre une difficulté de type « Planification », alors les enseignants qui peuvent fournir de l'aide sont :*

- (1) Tout enseignant informaticien ayant un niveau de compétence en GP (Gestion de projet) supérieur à 2 (intermédiaire ou expert).
- (2) Tout enseignant ayant un niveau de compétence en GP égal à 3 (expert).

Cette règle se représente comme suit (Figure 61) :

```
<rule>
  <in>
    <difficulte>
      <type-difficulte>planification</type-difficulte>
    </difficulte>
  </in>
  <out>
    <specialite>informatique</specialite>
    <competence-gp sup="oui" >2</competence-gp>
  </out>
  <out>
    <competence-gp>3</competence-gp>
  </out>
</rule>
```

Figure 61. Règle représentée selon le formalisme défini

Pour pouvoir exploiter ces règles, nous développons un moteur d'exploitation de règles. La figure 62 ci-dessous présente le processus de fonctionnement de ce moteur. Il commence par récupérer les caractéristiques de la difficulté et celles du profil de son émetteur. Après, il exploite les éléments « IN » des différentes règles pour identifier les règles appropriées à ces caractéristiques. Après l'identification des règles à appliquer, le moteur se base sur les critères présents dans leurs éléments « OUT » pour sélectionner les membres à solliciter.

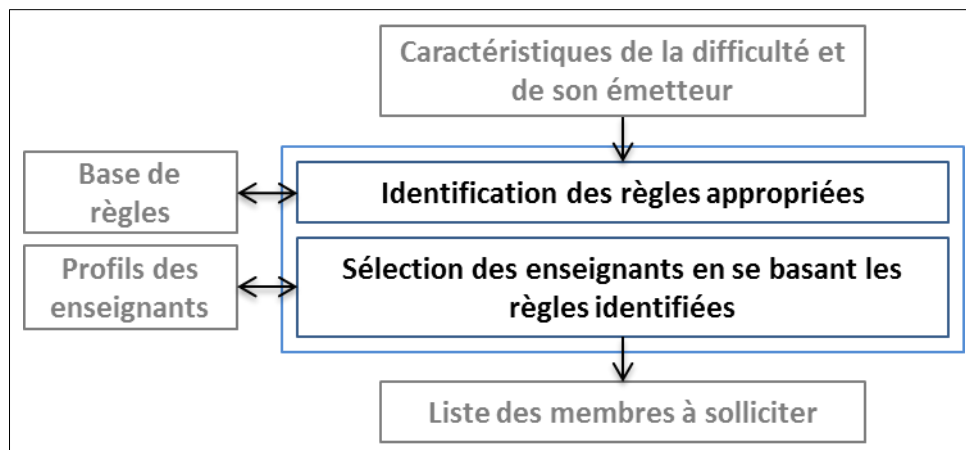


Figure 62. Processus de fonctionnement du moteur d'exploitation des règles

2.3 Généricité de la sélection statique

Pour rendre la sélection statique générique, nous reprenons la structure de règles que nous avons proposée pour les règles liées à la CoP MAETIC et la rendons générique. L'objectif est de proposer une structure permettant de représenter des règles de sélection quel que soit la communauté cible. La structure générique de règles est représentée par la figure 63 ci-dessous. Une règle se compose donc de deux types d'éléments :

- IN : qui regroupe des critères sur le problème et son émetteur. Si le problème et le profil de son émetteur satisfont les critères d'un seul élément de type « IN », la règle est applicable.
- OUT : qui englobe des critères sur les profils des membres à sélectionner. Si le profil d'un membre répond aux critères d'un seul élément de type « OUT », il sera sélectionné.

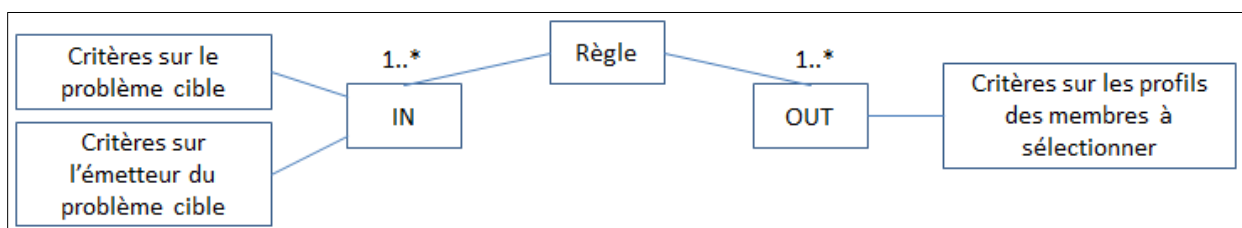


Figure 63. Structure générique de règles

Le moteur d'exploitation de règles que nous avons proposé pour exploiter les règles de la CoP MAETIC est réutilisable pour d'autres Communautés à condition de respecter la structure de règles proposée.

Prenons l'exemple d'une règle liée au domaine de développement informatique dans le cadre d'une communauté dédiée à la résolution de problèmes informatique comme suit : Si un membre rencontre un problème de type « VB.NET » (nous reprenons la classification

présenté dans le paragraphe 5.2 du chapitre 1) et que ce problème est lié à l'architecture de l'application, alors les membres qui peuvent participer à la résolution de ce problème sont :

- Tout membre spécialiste en « .NET », qui occupe une fonction d'expert technique et qui a 5 ans d'expérience.

Cette règle se présente, selon la structure que nous avons proposée, comme suit (Figure 64) :

```

<rule>
  <in>
    <probleme>
      <type-probleme>VB.NET</type-probleme>
      <aspect>architecture-application</aspect>
    </probleme>
  </in>
  <out>
    <specialite>.NET</specialite>
    <fonction>expert-technique</fonction>
    <nb-annee-experience>5</nb-annee-experience>
  </out>
</rule>

```

Figure 64. Règle liée au domaine de développement informatique

3. Sélection dynamique

3.1 Représentation des données

Nous considérons :

- $C = \{C_1, C_2, \dots, C_i, \dots, C_9\}$, l'ensemble des attributs du contexte d'une difficulté liés à : i) la dimension de la classe où a été rencontrée la difficulté, et à ii) la dimension pédagogique de l'enseignant émetteur de la difficulté. L'ensemble est donc constitué des 9 attributs suivants :

$C = \{C_1=\text{Spécialité}, C_2=\text{Effectif}, C_3=\text{NbGroupes}, C_4=\text{NbHeuresGlobal}, C_5=\text{NbHeuresSéance}, C_6=\text{Niveau}, C_7=\text{CompétenceMAETIC}, C_8=\text{CompétenceGP}, C_9=\text{CompétenceBlog}\}$

L'attribut « type de difficulté » n'appartient pas à cet ensemble car nous adoptons une organisation de traces basée sur cet attribut.

- $V_i = \{V_{i,1}, V_{i,2}, \dots, V_{i,m_i}\}$, l'ensemble des valeurs possibles pour un attribut C_i . m_i est le nombre de ces valeurs. Par exemple, pour l'attribut $C_7 = \text{'CompétenceMAETIC'}$ l'ensemble V_7 est comme suit :

$V_7 = \{1 \text{ (novice)}, 2 \text{ (intermédiaire)}, 3 \text{ (expert)}\}$

avec $m_7 = 3$

- $C_{xt}(D) = \{V_{1,k1}, V_{2,k2}, \dots, V_{i,ki}, \dots, V_{9,k9}\}$: le contexte de la difficulté D représenté par les valeurs des 9 attributs caractérisant le contexte dans lequel la difficulté D a été rencontrée.

Avec $V_{i,ki} \in V_i = \{V_{i,1}, V_{i,2}, \dots, V_{i,ki}, \dots, V_{i,mi}\}$ et $1 \leq k_i \leq m_i$.

- Par exemple, nous considérons une difficulté D rencontrée lors de l'enseignement d'une classe de niveau Licence qui se compose de 27 étudiants partagés en 4 groupes. Le nombre d'heures dédiées à l'enseignement est 44 partagé sur des séances de 2 heures. L'enseignant qui rencontre cette difficulté, est un informaticien, expert en gestion des blogs et qui a un niveau d'expertise intermédiaire en gestion de projet et en MAETIC. Par rapport à ces informations, le contexte de D est comme suit :

$C_{xt}(D) = \{ C_1 = \text{Spécialité} = \text{'Informatique'}$,
 $C_2 = \text{Effectif} = [21,30]$,
 $C_3 = \text{NbGroupes} = 4$,
 $C_4 = \text{NbHeuresGlobal} = [41,50]$,
 $C_5 = \text{NbHeuresSéance} = 2$,
 $C_6 = \text{Niveau} = \text{'Licence'}$,
 $C_7 = \text{CompétenceMAETIC} = 2$,
 $C_8 = \text{CompétenceGP} = 2$,
 $C_9 = \text{CompétenceBlog} = 3 \}$

- $VP_T = \{P_{T,1}, P_{T,2}, \dots, P_{T,i}, \dots, P_{T,n}\}$, le vecteur de pertinence du type de difficulté T où $P_{T,i}$ est le poids binaire associé à l'attribut du contexte C_i pour le type de difficulté T. Avec $P_{T,i} = 1$ ou $P_{T,i} = 0$ selon que C_i est pertinent ou non.

Si nous supposons que la difficulté de l'exemple précédent est de type « Réalisation du produit », et selon les connaissances des experts MAETIC, les poids des 9 attributs du contexte sont :

C_1 (Spécialité) => **1**
 C_2 (Effectif) => **0**
 C_3 (NbGroupes) => **0**
 C_4 (NbHeuresGlobal) => **1**
 C_5 (NbHeuresSéance) => **0**
 C_6 (Niveau) => **1**
 C_7 (CompétenceMAETIC) => **0**
 C_8 (CompétenceGP) => **1**
 C_9 (CompétenceBlog) => **0**

Le vecteur de pertinence de ce type de difficulté est donc comme suit :

$VP_{\text{RéalisationProduit}} = \{1,0,0,1,0,1,0,1,0\}$

- $R_Cxt(D) = \{V_{i,k_i}\} / poids(C_i) = 1$

$R_Cxt(D)$ est le contexte pertinent de la difficulté D qui est un sous ensemble du $Cxt(D)$ représenté juste par les valeurs des attributs pertinents pour le type de la difficulté D selon le vecteur de pertinence. i peut prendre toute valeur entre 1 et 9 selon que l'attribut C_i est pertinent ou non.

Le contexte pertinent de la difficulté de l'exemple précédent est :

$$R_Cxt(D) = \{ C_1 = \text{Spécialité} = \textbf{'Informatique'} , \\ C_4 = \text{NbHeuresGlobal} = [41,50], \\ C_6 = \text{Niveau} = \textbf{'Licence'} , \\ C_8 = \text{CompétenceGP} = \textbf{2} \}$$

- $R_Num_Cxt(D)$, une représentation numérique du contexte pertinent de la difficulté D qui facilitera le traitement des données par les algorithmes de sélection. En effet, les valeurs des 9 attributs du contexte peuvent être :
 - de type numérique comme $C_3 = \text{'NbGroupes'}$, $C_5 = \text{'NbHeuresSéance'}$, $C_7 = \text{'CompétenceMAETIC'}$, $C_8 = \text{'CompétenceGP'}$, $C_9 = \text{'CompétenceBlog'}$,
 - de type textuel comme $C_1 = \text{'Spécialité'}$ et $C_6 = \text{'Niveau'}$,
 - ou d'un autre type comme $C_4 = \text{'NbHeuresGlobal'}$ qui est sous forme d'intervalle.

Lorsqu'un attribut non numérique C_i prend une valeur $V_{i,j}$ cette dernière est remplacée par son indice j . Les indices sont ensuite collés pour que le contexte soit représenté par un nombre.

En reprenant l'exemple :

$$R_Cxt(D) = \{ \text{'Informatique'}, [41,50], \text{'Licence'}, 2 \}$$

$$R_Num_Cxt(D) = 1322$$

3.2 Mesure de similarité

La sélection dynamique est basée sur la sélection des membres de la CoP qui ont un nombre suffisant de contributions dans le passé liées à des difficultés rencontrées dans des contextes similaires à celui de la difficulté cible.

Nous considérons les contextes de deux difficultés $D1$ et $D2$ comme similaires si les deux difficultés sont du même type T et si les valeurs des attributs pertinents qui composent les deux contextes $R_Cxt(D1)$ et $R_Cxt(D2)$ sont similaires.

Nous proposons deux mesures de similarité pour les valeurs des attributs du contexte :

- La similarité absolue : deux valeurs sont similaires si elles sont identiques. Nous appliquons cette mesure de similarité pour les attributs : Type de difficulté, Spécialité, Effectif, NbGroupes, NbHeuresGlobal, NbHeuresSéance.
- La similarité relative : nous appliquons, ici, un raisonnement pédagogique qui prend en considération la sémantique des attributs du contexte par rapport à l'habilité à résoudre les difficultés MAETIC. Nous appliquons cette mesure aux attributs qui ont trait à caractériser les compétences d'un membre. Par exemple, un enseignant ayant un niveau de compétence MAETIC égal à 2 peut aider un membre qui a le niveau 1 ou 2 à résoudre sa difficulté. Par contre, il ne peut pas aider les membres ayant le niveau 3, qui sont considérés plus expérimentés, que lui. Pour représenter ce principe, nous considérons alors deux valeurs comme similaires si la valeur de l'attribut du contexte de la difficulté cible est inférieure ou égale de la valeur de l'attribut du contexte de la difficulté avec laquelle nous mesurons la similarité. Nous appliquons cette mesure aux attributs : Niveau, CompétenceMAETIC, CompétenceGP, CompétenceBlog.

3.3 Représentation des indices contributions

Comme énoncé dans le paragraphe 2.2, nous utilisons les indices de contributions pour permettre à l'algorithme de filtrage de prendre des décisions concernant l'élimination de certains membres de la liste des candidats potentiels à être sélectionnés sans avoir besoin de parcourir leurs traces.

L'indice de contribution est une valeur qui caractérise le nombre de contributions pertinentes d'un membre pour résoudre des difficultés d'un type donné, dans un contexte donné. Ainsi, nous définissons pour chaque membre E, Pour chaque type de difficulté T :

- $I_{C_E}(T)$: l'indice de contribution du membre E pour le type de difficulté T. Il représente le nombre global des anciennes contributions pertinentes du membre E pour résoudre des difficultés de type T. Dans l'exemple de la figure 65 représentant graphiquement l'arbre de contribution d'un membre E, $I_{C_E}(\text{« Réalisation du produit »})=7$. Ce qui signifie que ce membre a contribué de manière pertinente à résoudre 7 difficultés de type « réalisation du produit ».

Nous définissons aussi pour chaque valeur $V_{i,j}$ d'un attribut du contexte C_i :

- $I_{C_{E,T}}(V_{i,j})$: l'indice de contribution du membre E pour le type de difficulté T dans un contexte où la valeur de l'attribut C_i est similaire à $V_{i,j}$. Ici, la similarité dépend de l'attribut et peut être absolue ou relative. La figure 65 montre que :
 - $I_{C_{E,T}}(V_{8,2})=3$: le membre E a aidé 3 enseignants ayant pour l'attribut C8= « compétence GP » une valeur similaire à $V_{8,2} = 2$ (intermédiaire).

- $I_{C_{E,T}}(V_{8,1})=7$: le membre E a aidé 4 enseignants ayant pour l'attribut C8= « compétence GP » une valeur similaire à $V_{8,1}=1$ (novice). En effet, Puisque la similarité appliquée à C8 est une similarité relative, ce membre est considéré comme s'il avait aidé 7 membres ayant des niveaux de compétence GP similaires à 1 (4 novices et 3 intermédiaires).

Enfin, nous définissons pour chaque enseignant, pour chaque type de difficulté et pour chaque attribut de contexte le vecteur de contribution :

- $V_{C_{E,T}}(C_i)=\{I_{C_{E,T}}(V_{i,1}),I_{C_{E,T}}(V_{i,2}),\dots,I_{C_{E,T}}(V_{i,j}),\dots,I_{C_{E,T}}(V_{i,m})\}$: le vecteur de contribution caractérisant le nombre de contributions pertinentes du membre E pour les difficultés de type T par rapport à toutes les valeurs possibles pour l'attribut du contexte C_i .

Où $I_{C_{E,T}}(V_{i,j})$ représente l'indice de contributions de l'enseignant E à résoudre des difficultés de type T ayant comme valeur de l'attribut C_i , une valeur similaire à $V_{i,j}$.

- Selon la figure 65 :

$$V_{C_{E,T}}(C_8= \text{« compétence GP »})=\{7,3,0\}$$

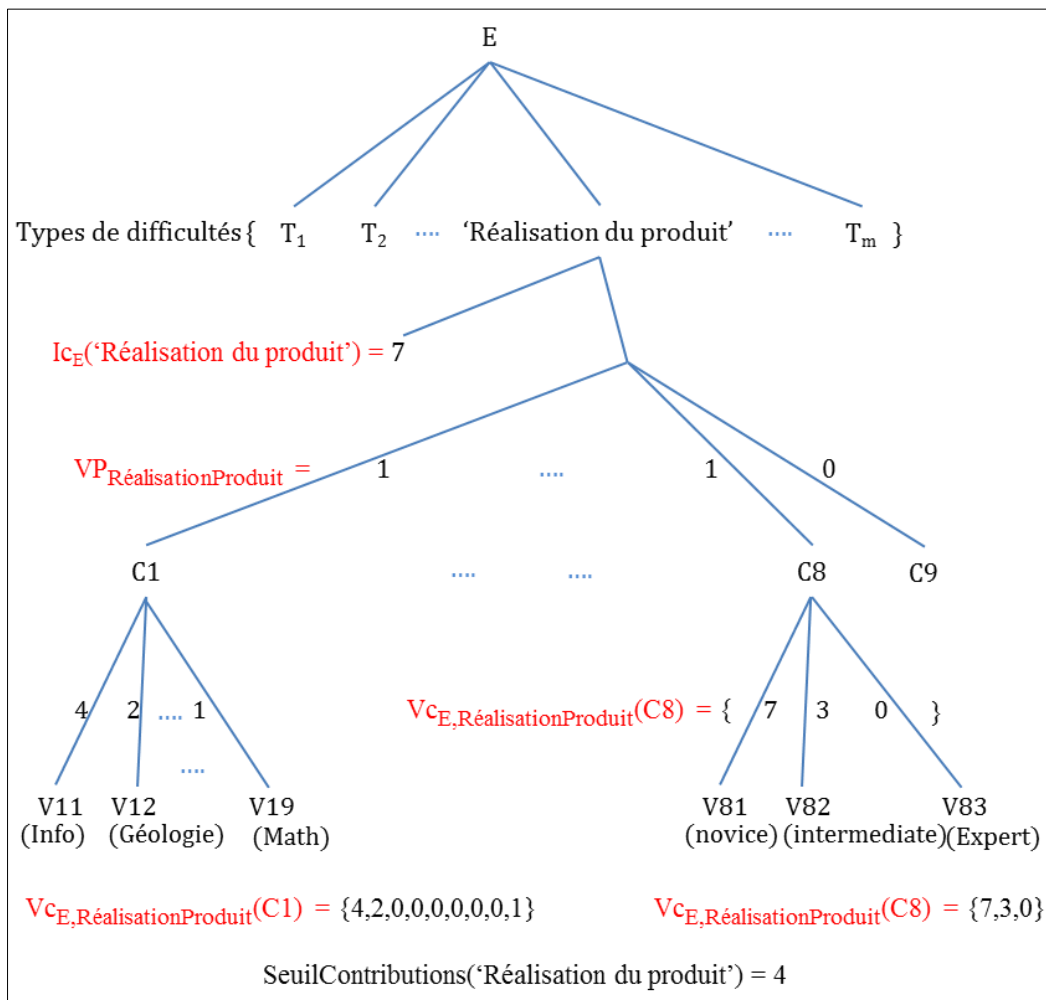


Figure 65. Les indices de contributions de l'enseignant E

Pour des besoins de proactivité et d'optimisation du temps de réponse, la mise à jour des indices de contribution d'un membre se fait implicitement au moment de l'évaluation de ses commentaires. En effet, après avoir déclaré un commentaire comme pertinent, les indices de contribution du sous-arbre du membre E correspondant au type de difficulté en question sont mis à jour.

3.4 Algorithme de filtrage

L'algorithme de filtrage a pour objectif de définir, parmi les membres de la CoP non sélectionnés par les règles statiques, un ensemble d'enseignants jugés non capables de participer à la résolution de la difficulté cible afin d'alléger la tâche de l'algorithme d'analyse des traces qui ne va pas parcourir leurs fichiers trace.

Au moment où une difficulté D de type T est publiée, caractérisée par un contexte pertinent $R_Cxt(D)$, l'algorithme de filtrage considère qu'un membre E est non capable de participer à résoudre D lorsque :

- Le nombre de ses anciennes contributions par rapport aux difficultés de type T est inférieur au seuil de contribution de T.

Ceci est vérifié **Si $IG_{E,T} < SeuilContributions(T)$**

En effet, $I_{E,T}$ représente les anciennes contributions de E par rapport aux difficultés de type T. Etant calculé en arrière-plan, l'algorithme décide de suite que le membre peut être éliminé si cet indice est inférieur au seuil de contribution de T.

- L'indice de contribution pour le type T est supérieur au seuil de contribution mais le nombre des contributions qui ont eu lieu dans des contextes similaires au contexte de la difficulté cible est inférieur au seuil de contribution de T.

Ceci est vérifié

Si $IG_{E,T} > SeuilContributions(T)$

Et Si $I_{E,T}(V_{i,j}) < SeuilContributions(T) / V_{i,j} \in R_Cxt(D)$

En effet, si $V_{i,j}$ est une valeur d'un attribut C_i pertinent pour D, et si $I_{E,T}(V_{i,j}) < SeuilContributions(T)$, c'est que le nombre de contributions passées du membre dans des contextes similaires à la difficulté cible est inférieur au seuil de contribution.

Prenons un exemple, nous avons une difficulté D de type « Réalisation du produit » et nous voulons évaluer les contributions d'un membre E, afin de se décider concernant son habilité à participer à la résolution de cette difficulté. Le contexte de D et le vecteur de pertinence du type « Réalisation du produit » et le $SeuilContributions$ de ce type sont comme suit :

- $VR_{Réalisation_Produit} = \{1,0,0,1,0,1,0,1,0\}$
- $Cxt(D) = \{Informatique, [21,30], 4, [41,50], 2, Licence, 2, 2, 3\}$

- $R_Cxt(D) = \{Informatique, [41,50], Licence, 2\}$
- $SeuilContributions('Réalisation du produit') = 4$

Le tableau 10 ci-dessous présente les contextes pertinents des difficultés de type « Réalisation du produit » auxquelles E a contribué.

	Spécialité	Nombre d'heures global	Niveau	Compétence GP
1	Informatique	[41,50]	Licence	2
2	Géologie	[41,50]	Master	2
3	Chimie	[31,40]	Master	1
4	Informatique	[31,50]	Master	3
5	Informatique	[41,50]	Licence	2
6	Géologie	[41,50]	Licence	1
7	Electronique	[41,50]	Master	1

Tab 10. Contextes des difficultés liées aux contributions du membre E

Ce tableau montre que E a contribué 3 fois à résoudre des difficultés ayant « Informatique » comme valeur de l'attribut C_1 (Spécialité). La valeur « Informatique » est la première valeur de la liste des valeurs possibles de C_1 , donc $Ic_{E,Realisation_Produit}(V_{1,1}) = 3$. Dans ce cas, $Ic_{E,Realisation_Produit}(V_{1,1}) < SeuilContributions('Réalisation du produit')$ et donc E sera éliminé. En effet, tous les contextes susceptibles d'être similaires au contexte de D doivent avoir la valeur « Informatique » comme valeur de C_1 (nous utilisons une similarité absolue pour cet attribut). Pourtant, $Ic_{E,RealisationProduit}(V_{1,1})$ montre que seulement 3 contextes ont la valeur « Informatique ». Donc, 3 contextes seulement peuvent être similaires à celui de D (1, 4, 5). Parmi ces trois contextes, il existe 2 qui sont similaires au contexte de D (1, 5).

Dans cet exemple nous pouvons éliminer le membre E en se basant sur la valeur de $Ic_{E,RealisationProduit}(V_{1,1})$ et sans parcourir les traces de contextes des difficultés auxquelles il a contribué à résoudre.

Par conséquent, l'algorithme de filtrage est le suivant :

Function Filtering(Cxt(D), $Ic_{E,T}$, $IGc_{E,T}$, VR_T , SeuilContributions(T))

If $IGc_{E,T} < SeuilContributions(T)$

Return false

End if

For i: 1 to n

If $VR_{T,i} = 1$

If $Ic_{E,T}(V_{i,ki}) < SeuilContributions(T)$

Return false

End if

End if
End for
Return true
Fin function

L'algorithme prend en paramètres le contexte de la difficulté cible, les indices de contributions du membre E par rapport au type de la difficulté cible ainsi que le vecteur de pertinence et le seuil de contributions de ce type.

Tout d'abord, nous vérifions si l'indice de contributions global de l'enseignant E à résoudre des difficultés de type T est inférieur à `SeuilContributions` du type T. Après, pour chaque valeur du contexte de la difficulté cible dont l'attribut est considéré comme pertinent pour le type T, nous vérifions si son indice de contributions est inférieur à `SeuilContributions(T)`.

Le choix de faire ce premier traitement a été effectué pour limiter la quantité de données à traiter lors de la publication d'une difficulté. En effet, après chaque publication de difficulté, nous évitons d'analyser la totalité des contributions des membres. Cet algorithme est basé sur la consultation des valeurs des vecteurs contenant les indices de contributions. Il permet de prendre des décisions concernant la non-sélection de certains membres. Cependant, nous ne pouvons rien dire par rapport aux membres dont il n'a pas considéré comme des membres à éliminer.

En résumé, jusqu'ici, dans notre approche de sélection des membres :

- Un ensemble de membres a été sélectionné à l'aide de la sélection statique.
- Parmi ceux non sélectionnés, un ensemble de membres ont été filtrés par l'algorithme de filtrage. Ils sont donc jugés non capables de résoudre la difficulté cible.
- Les membres restants font l'objet du processus d'analyse des traces que nous présentons dans les deux paragraphes suivants.

3.5 Représentation des traces

Pour un contexte `R_Num_Cxt(D)`, pour un membre E, la trace de ses contributions représente le nombre de ses contributions pertinentes (`Cont_Nb`) dans des contextes identiques à `R_Num_Cxt(D)`.

Nous représentons une trace de contribution par le couple :

Trace = (`R_Num_Cxt(D)` : `Cont_Nb`)

En reprenant notre exemple de difficulté et en supposant que le membre E a eu 3 contributions pour résoudre des difficultés ayant les mêmes contextes pertinents que celui de difficulté D :

$R_Num_Cxt(D) = 1322$

Trace = 1322:3

Les traces de contributions de chaque enseignant E par rapport aux difficultés de type T sont groupées dans un fichier. Nous avons donc pour chaque paire <Enseignant, Type de difficultés> un fichier regroupant les traces de contributions de cet enseignant à résoudre des difficultés de ce type. Ces traces sont triées par rapport aux représentations numériques des contextes. La figure 66 ci-dessous présente un extrait d'un fichier de traces :

1121:7
1312:9
1322:3
2121:1
2333:1
3111:6

Figure 66. Extrait d'un fichier de traces

3.6 Algorithme d'analyse de traces

L'objectif de l'analyse des traces est de trouver, parmi les membres non encore sélectionnés, ceux qui peuvent aider de manière pertinente à résoudre la difficulté cible.

Etant donnée le contexte d'une difficulté ($R_Num_Ctxt(D)$) de type T, l'algorithme d'analyse des traces parcourt les fichiers de trace du type de difficulté T de tous les enseignants non sélectionnés. L'algorithme décide de sélectionner un enseignant lorsque le nombre de ses contributions dans des contextes similaires au $R_Num_Cxt(D)$ est supérieur ou égal au seuil de contribution du type T ($SeuilContributions(T)$).

Les problèmes qui se posent à ce niveau sont liés au :

- Grand nombre de fichiers à parcourir (nombre d'enseignants non encore sélectionnés) qui dépend de la taille de la CoP,
- Grand nombre de lignes à parcourir dans chaque fichier. Ce nombre est lié au grand nombre de combinaisons possibles pour les contextes des anciennes contributions d'un enseignant.

Dans le but d'accélérer le parcours de ces fichiers, nous proposons une organisation particulière des fichiers de trace :

- Nous avons groupé les valeurs des attributs auxquels on applique la similarité relative à la fin de la représentation numérique du contexte, c'est-à-dire que les quatre dernières positions sont relatives aux valeurs des attributs : Niveau, CompétenceMAETIC, CompétenceGP et CompétenceBlog.

- Nous avons trié les lignes du fichier trace par ordre croissant des représentations numériques du contexte des traces (R_Num_Cxt(D)).

Ainsi, étant donné le R_Num_Cxt(D) d'une difficulté cible, l'algorithme d'analyse des traces parcourt le fichier trace d'un enseignant en utilisant la recherche dichotomique (puisque les lignes sont triées) afin d'identifier la position où il y a une trace ayant la même représentation ou la plus proche de R_Num_Cxt(D) de la difficulté cible. Le choix du groupement des attributs auxquels on applique la similarité relative à la fin de R_Num_Cxt(D) permet de minimiser le nombre de traces à parcourir après avoir trouvé cette position. En effet, les lignes ayant des contextes similaires à R_Num_Cxt(D), que nous cherchons, seront groupées entre 2 bornes :

- La ligne relative à la position trouvée.
- La ligne ayant dans sa représentation numérique du contexte :
 - a. La partie gauche (relative aux valeurs des attributs auxquels on applique la similarité absolue) identique à celle du contexte cible
 - b. La partie droite (relative aux valeurs des attributs auxquels on applique la similarité relative) est maximale.

En reprenant notre exemple où le contexte de la difficulté cible est : R_Num_Cxt(D) = 1322, les représentations numériques similaires à R_Num_Cxt (D) se trouvent parmi les lignes :

1322{'Informatique', [41,50], 'Licence', 2}
 1323{'Informatique', [41,50], 'Licence', 3}
 1331{'Informatique', [41,50], 'Master', 1}
 1332{'Informatique', [41,50], 'Master', 2}
 1333{'Informatique', [41,50], 'Master', 3}

Cette organisation des fichiers de trace permet de limiter le nombre de lignes à parcourir sans l'application d'aucune mesure de similarité aux 5 lignes ci-dessus. En appliquant la similarité relative à la partie droite du contexte de ces mêmes lignes, les représentations numériques similaires à R_Num_Cxt(D) sont représentées dans les 4 lignes suivantes :

1322{' Informatique', [41,50], 'Licence', 2}
 1323{' Informatique', [41,50], 'Licence', 3}
 1332{' Informatique', [41,50], 'Master', 2}
 1333{' Informatique', [41,50], 'Master', 3}

Après l'identification de ces deux bornes, l'algorithme d'analyse des traces calcule la somme des contributions dans des contextes similaires au contexte de la difficulté cible pour décider ou non de sélectionner le membre en fonction du seuil de contribution du type de la difficulté cible.

3.7 Généricité de la sélection dynamique

Les solutions que nous avons proposées pour effectuer la sélection dynamique exploitent un ensemble d'attributs dont chacun a un ensemble déterminé de valeurs possibles. Pour pouvoir réutiliser ces solutions par d'autres CoP, il faut donc spécifier la liste des attributs à exploiter et les valeurs possibles pour chaque attribut. Ainsi, il faut spécifier pour chaque attribut, le type de similarité à appliquer par rapport à ses valeurs (absolue ou relative). Enfin, il faut déterminer les vecteurs de pertinence de chaque type de problème.

4. Expérimentation

L'objectif des expérimentations que nous présentons dans cette section est de valider notre approche de sélection. En effet, nous mesurons le gain par rapport à plusieurs aspects :

- Le gain lié à l'utilisation d'un filtrage avant l'analyse des traces.
- Le gain lié à l'organisation des fichiers de trace que nous proposons (représentation numérique des contextes pertinents et tri de ces représentations dans les fichiers de traces)
- Le gain en termes de temps d'exécution des algorithmes de filtrage et d'analyse des traces pour différents types de difficultés et différentes tailles de la CoP.

4.1 Scénario de validation

Pour atteindre notre objectif, nous procédons par simulation. Notons qu'une communauté MAETIC est déjà en place et qu'elle utilise le système en milieu écologique. Sachant que le nombre des membres de cette CoP n'est pas très grand (~30 enseignants) et que nous avons besoin de beaucoup de paramétrage pour mener à bien nos expériences, nous avons opté pour la validation de notre approche par simulation.

Pour ce faire, nous avons simulé une communauté de 30 enseignants ayant des profils diversifiés en termes de spécialité, expertise en gestion de blogs et maîtrise de MAETIC, mais aussi par rapport à leur niveau d'activité dans la CoP. En effet, notre CoP se compose de 45% de novices, 30% de membres intermédiaires et 25% d'experts. La distribution des membres de la CoP par rapport à ces 3 catégories est basée sur les travaux de Wenger ([Wenger, McDermott et al. 2002](#)). Un novice a un nombre de contributions pertinentes entre 6 et 20, un intermédiaire entre 21 et 40, un expert entre 41 et 60. Les profils des membres sont détaillés en annexe 2, alors que les détails de leurs traces de contributions sont présentés en annexe 3.

Pour effectuer nos simulations, nous avons considéré 8 difficultés de différents types, qui sont rencontrées dans le contexte suivant : un enseignant, informaticien, expert en blog, et qui a un niveau d'expertise intermédiaire en MAETIC et en gestion de projet, rencontre une difficulté

lors de son enseignement d'une classe de niveau Licence, qui se compose de 27 étudiants partagés sur 4 groupes. Le nombre d'heures dédié au module est 44h partagées en séances de 2h.

Une de ces difficultés est de type « pannes réseau », étant donné le vecteur de pertinence de ce type :

$$VP_{\text{Pannes_reseau}} = \{1,0,1,0,0,1,1,0,0\}$$

La représentation numérique du contexte pertinent de cette difficulté est :

$$R_Num_Cxt(D) = \{1,4,2,2\}$$

4.2 Gain en termes de qualité de la sélection

Dans la première expérience nous nous sommes intéressés au nombre d'enseignants sélectionnés. En effet, nous suivons le processus de sélection des membres, étape par étape, afin d'analyser de plus près le gain de chacune des étapes de notre approche ainsi que son gain global.

En effet, la sélection statique commence par sélectionner un certain nombre d'enseignants, jugés théoriquement, capables de résoudre la difficulté cible. La sélection dynamique sélectionne ensuite les enseignants les plus aptes à résoudre la difficulté cible parmi les membres de la CoP non sélectionnés statiquement. En effet, l'algorithme de filtrage commence par éliminer un ensemble de ces derniers, jugés incapables de résoudre la difficulté cible en se basant sur leurs activités passées, afin d'alléger le travail de l'algorithme d'analyse des traces qui ne va analyser que les fichiers trace des membres restants. Les membres sélectionnés finalement seront ceux résultant de la sélection statique plus ceux résultant de l'analyse des traces.

Nous avons fait le suivi du processus de sélection sur notre CoP simulée de 30 membres par rapport à plusieurs types de difficultés (T1 : Panne réseau, T2 : planification, ..., T8 : Négociation). Chaque type est caractérisé par son propre vecteur de pertinence $VP(T_i)$. Dans le tableau 11, nous représentons les valeurs collectées suivantes :

- A : le nombre des membres sélectionnés par la sélection statique.
- B : le nombre des membres éliminés par l'algorithme du filtrage
- C : le nombre des membres dont les traces vont être analysées par l'algorithme d'analyse des traces.

$$\text{Avec } C = 30 \text{ (taille de la CoP)} - 1 \text{ (l'émetteur de la difficulté)} - A - B$$

- D : le nombre des membres sélectionnés par la sélection dynamique (après analyse des traces).

- E : le nombre total des membres sélectionnés.

Avec $E = A + D$

- F : le nombre total des membres éliminés.

Avec $F = 30 - 1 - E$

Nous représentons dans la dernière colonne, le gain lié à l'algorithme du filtrage que nous définissons comme le nombre des membres qui ont pu être éliminés par l'algorithme de filtrage divisé par le nombre total des membres éliminés ($\text{Gain} = B/F$). En effet, ce nombre montre la performance de l'algorithme de filtrage et sa capacité à minimiser le traitement de l'algorithme d'analyse des traces, étape suivante de la sélection dynamique. Nous représentons ce gain aussi en termes de pourcentage.

	VR(T)	A	B	C	D	E	F	Filtering gain
T1	101001100	4	17	8	5	9	20	17/20 85,00%
T2	000111110	2	13	14	3	5	24	13/24 54,17%
T3	100001110	7	9	13	7	14	15	13/15 86,67%
T4	000001110	2	9	18	15	17	12	9/12 75,00%
T5	100001100	0	15	14	5	5	24	15/24 62,50%
T6	001001100	0	14	15	3	3	26	14/26 53,85%
T7	100101110	2	13	14	2	4	25	13/25 52,00%
T8	000001101	6	11	12	7	13	16	11/16 68,75%

Tab 11. Suivi du processus de sélection

Par analyse de ces résultats, nous trouvons tout d'abord que E est plus grand que A pour tout type de difficultés ; c'est-à-dire que le nombre total des membres sélectionnés est toujours plus grand que le nombre de ceux sélectionnés statiquement. Ce qui montre la pertinence de notre sélection dynamique qui complète la liste des membres sélectionnés théoriquement à base de leurs profils, par des membres réellement actifs dans la CoP à base de leurs contributions pertinentes.

D'autre part, nous trouvons que le gain de l'algorithme de filtrage varie entre 52% et 85% ; c'est-à-dire, qu'il permet de filtrer entre 52% et 85% des membres qui ne doivent pas être sélectionnés et minimise ainsi le traitement de l'algorithme d'analyse des traces par le même pourcentage.

4.3 Gain en termes de temps d'exécution

Dans l'objectif de montrer l'intérêt de chacune des étapes de notre approche, nous avons mesuré les temps d'exécution des algorithmes de filtrage et d'analyse des traces dans différents cas de figure :

- Cas 1 : sélection dynamique sans filtrage et sans tri des représentations numériques des contextes dans les fichiers de trace. Nous notons le temps d'exécution dans ce cas : $T(-F-T)$
- Cas 2 : sélection dynamique avec filtrage et sans tri des fichiers traces. Nous notons le temps d'exécution correspondant : $T(+F-T)$.
- Cas 3 : sélection dynamique avec filtrage et tri des fichiers trace. Nous notons le temps d'exécution correspondant : $T(+F+T)$.

L'objectif de nos expérimentations est de mesurer les gains liés aux différentes étapes de l'approche. Ainsi, nous définissons :

- Le gain de l'algorithme du filtrage, comme le pourcentage d'amélioration du temps d'exécution de la sélection dynamique avec filtrage et sans tri par rapport à la sélection sans filtrage et sans tri : $T(+F-T) / T(-F-T)$
- Le gain lié au tri des fichiers de trace comme, le pourcentage d'amélioration du temps d'exécution de la sélection dynamique avec filtrage et avec tri par rapport à la sélection avec filtrage et sans tri : $T(+F+T) / T(+F-T)$
- Le gain global de notre approche, comme le pourcentage d'amélioration du temps d'exécution de la sélection dynamique avec filtrage et tri des fichiers trace par rapport à celui de la sélection sans filtrage et sans tri : $T(+F+T) / T(-F-T)$

De plus, nous avons refait tous les calculs pour les 8 types de difficultés de la première expérience et, pour des raisons de passage à l'échelle, pour différentes tailles de la CoP allant de 30 à 5000 membres.

Notons, que nous avons exécuté nos programmes sur un ordinateur de bureau Windows 7 à 64 bits, doté d'un processeur Intel(R) Core(TM) i7-3770 CRU @ 3.40 GHz et d'une RAM de 32 Go.

Nous regroupons les premiers résultats dans le tableau 12 ci-dessous où nous avons représenté, pour le type de difficulté T1, pour différentes tailles de CoP, les temps d'exécution correspondant aux 3 cas ci-dessus ainsi que les différents gains correspondants.

CoP Dimension	30	50	100	500	1000	5000
	Temps d'exécution en ms					
T(-F-T)	27	42	61	127	223	5376
T(+F-T)	15	22	40	92	137	588
T(+F+T)	8	11	22	68	86	228
	Gain en %					
Gain de filtrage	44,44%	47,62%	34,43%	27,56%	38,57%	89,06%
Gain de tri	46,67%	50,00%	45,00%	26,09%	37,23%	61,22%
Gain global	70,37%	73,81%	63,93%	46,46%	61,43%	95,76%

Tab 12. Résultats de la première expérience

En ne considérant que la première colonne du tableau représentant la taille initiale de notre CoP de 30 membres, nous remarquons bien une minimisation successive du temps d'exécution au fur et à mesure de l'intégration des différents algorithmes. En effet, le temps d'exécution correspondant à l'analyse des traces des membres sans filtrage préalable et avec des fichiers trace non triés est de 27ms (T(-F-T)). En ajoutant la fonctionnalité de filtrage, le temps d'exécution passe à 15ms (T(+F-T)), donc une amélioration de 44,44%. Par ajout de la fonctionnalité de tri des fichiers trace, le temps d'exécution passe à 8ms (T(+F+T)). C'est-à-dire, une amélioration additionnelle de 46,67% ; et donc une amélioration totale de 70,37% par rapport à une sélection dynamique qui analyse les traces sans filtrage et sans tri.

En considérant la variation du nombre des membres de la CoP pour le même type de difficulté, nous trouvons que les différents gains prennent des valeurs positives autour de 50% alors que le gain global prend des valeurs entre 70% et 95%.

En généralisant les mêmes calculs pour plusieurs types de difficultés, nous pouvons confirmer que les gains obtenus ne dépendent pas des types de difficultés puisque les mêmes intervalles pour les 3 gains considérés sont observés pour les 8 types de difficultés considérés, aussi bien pour les petites tailles de CoP que pour les grandes. En effet, comme le montre la figure 67 ci-dessous qui présente la variation du gain global, nous pouvons constater que l'écart, par rapport à ce gain, pour les difficultés (T1, T2, T3, T5, T6, T7, T8) est proportionnel pour des CoPs de 30 que pour des CoPs de 5000, ce qui n'est pas le cas pour le type de difficultés T4 qui a un gain moins élevé surtout pour des CoPs de petites tailles. Ceci est justifié par le fait que le type T4 n'a pas d'attributs pertinents parmi ceux auxquels nous appliquons la similarité absolue ($VR(T4) = 000001110$), ce qui rend la détection des membres à éliminer par l'algorithme de filtrage plus compliquée.

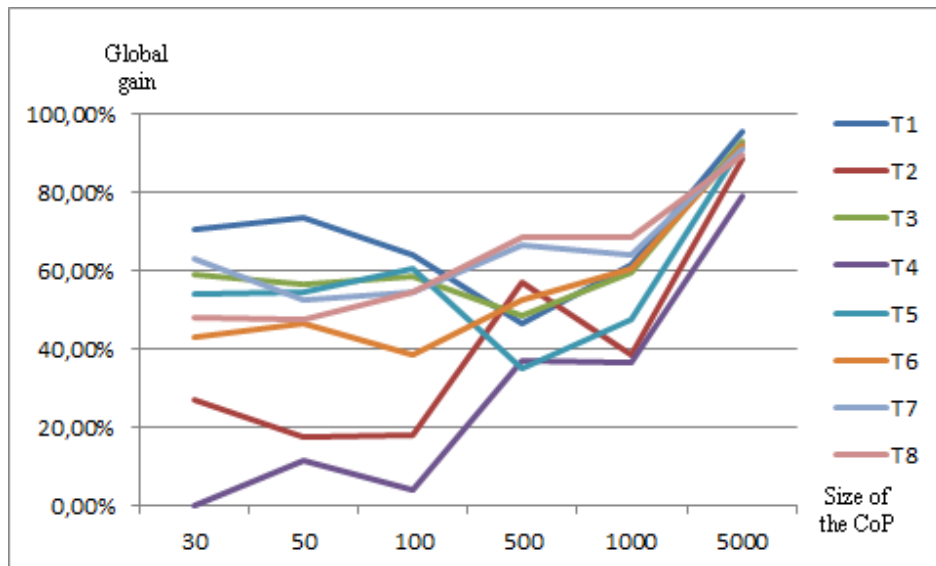


Figure 67. Variation du gain global pour différents types de difficultés et différentes tailles de la CoP

Un tableau détaillant la figure 67 est présenté en annexe 4, ainsi que des tableaux détaillant les variations du gain de filtrage et du gain de tri.

6. Conclusion

Dans ce chapitre nous avons exposé notre approche de sélection des membres à qui envoyer une demande d'aide lorsqu'un problème est publié. Notre problématique, étant considérée comme problème de formation de groupes, nous avons tout d'abord présenté les travaux existant dans le domaine de formation des groupes dans les systèmes d'apprentissage collaboratif. Après, nous avons détaillé notre approche de sélection des membres. Enfin, nous avons présenté une validation expérimentale de cette approche qui donne des résultats satisfaisants.

Conclusion

1. Bilan

Dans le cadre de ces travaux de recherche, nous nous sommes intéressés à la conception et au développement d'un système informatique communautaire mobile et adaptatif, que nous avons appelé NICOLAT. Ce dernier vise à supporter une Communauté de Pratique (CoP) d'enseignants usagers de la méthode pédagogique MAETIC, et qui rencontrent des difficultés dans l'usage de celle-ci, en classe. La résolution de ces difficultés représente l'apprentissage par la pratique offert à travers la CoP.

Nous nous sommes basés sur un ensemble de constats lors de l'expérimentation et la mise en place des dispositifs pédagogiques MAETIC par les enseignants, membres de la CoP, afin d'énoncer les différentes questions de recherche de ce travail et auxquelles nous avons réussi à répondre à travers NICOLAT.

Ainsi, nos principales contributions se résument dans les points suivants :

- La mise en place du noyau communautaire du système NICOLAT. Ce dernier est sous forme d'un réseau social respectant les particularités de la CoP MAETIC. Il permet aux membres de poster les difficultés et participer à leurs résolutions à travers des fils de discussions.
- La mise en place d'une couche de résolution de difficultés (RD) ayant pour objectif d'aider l'enseignant à résoudre sa difficulté par exploitation de l'historique des difficultés résolues dans le passé dans des contextes similaires. Ceci pour minimiser le nombre de demandes d'aide répétitives, et qui peuvent être considérées comme inutiles, sur le noyau communautaire. Le cycle du raisonnement RàPC a été utilisé pour guider ce processus. L'expérimentation a bien montré la pertinence de la modélisation du cas et de la mesure de similarité proposées.
- La mise en place d'une couche d'adaptation des interactions permettant de supporter les interactions des membres dans les outils d'interaction qu'ils préfèrent ou avec lesquels ils sont familiarisés le plus. L'adaptation supportée par cette couche concerne la transformation de la présentation du contenu échangé lors des interactions, depuis et vers une représentation interne du noyau communautaire. Les outils supportés sont des outils sociaux (Facebook et messagerie électronique) ou des outils mobiles (SMS et application Android). L'objectif visé par cette adaptation était, d'une part, d'augmenter la motivation des membres à utiliser le système et minimiser ainsi le nombre d'abandons. D'autre part, de permettre l'accès au système en cas de mobilité et minimiser ainsi le temps de réponse. Les résultats de l'expérimentation que nous

avons menée ont montré la satisfaction des membres de la CoP en termes de facilité d'usage et de motivation à répondre immédiatement aux demandes d'aide via les outils d'interaction qui correspondent à leurs préférences d'usage.

- La mise en place d'une approche de sélection dynamique des membres qui peuvent contribuer positivement à résoudre une difficulté à qui faire aboutir les demandes d'aide. L'objectif est toujours de limiter les démotivations des membres de la CoP (principal frein à l'apprentissage via la CoP) en leur permettant de recevoir une réponse, supposée pertinente, car issue des membres sélectionnés comme ayant les compétences nécessaires pour la résoudre. La validation expérimentale de cette approche a montré notamment que nos algorithmes proposés permettent d'obtenir un gain important en termes de temps d'exécution.

2. Perspectives à court terme

Nos contributions à travers la conception et le développement du système NICOLAT ouvrent les pistes de recherche suivantes :

- Pour la couche RD, nous proposons d'améliorer l'approche de similarité textuelle que nous avons utilisée en prenant en compte l'aspect sémantique des mots les plus utilisées par les membres de la communauté et qui peuvent améliorer les résultats de la recherche.
- Pour le support des outils d'interaction, une application iOS permettant aux utilisateurs des terminaux iOS d'interagir avec NICOLAT reste à développer. Aussi, vu que les blogs utilisés par les enseignants qui déploient MAETIC sont souvent des blogs WordPress, une perspective intéressante serait de développer un plugin WordPress permettant à l'enseignant d'interagir avec NICOLAT via son propre blog.
- Pour l'approche de sélection des membres, nous proposons deux pistes d'amélioration:
 - L'analyse des sentiments est un domaine qui s'intéresse à l'analyse automatique des opinions exprimés dans des textes ([Liu 2011](#)). Différents travaux se sont intéressés à l'analyse de sentiments dans les réseaux sociaux ([Paltoglou 2014](#)) ([Rosenthal, Nakov et al. 2015](#)) ([Cao, Lu et al. 2015](#)). Les travaux réalisés dans ce cadre peuvent être exploités dans NICOLAT pour mesurer la pertinence des commentaires qui ont aidé à résoudre une difficulté de manière automatique et sans intervention explicite de la part de l'enseignant.
 - Les valeurs des seuils de contributions dans notre approche sont définies par les experts et sont fixes par type de difficulté. En réalité, les seuils à considérer pour sélectionner un membre à contacter peuvent changer. En effet, à travers ses contributions fréquentes dans des situations différentes, il peut acquérir

l'habilité à répondre à des difficultés d'un type donné même s'il a un nombre de contributions plus petit que le seuil fixé par les experts. Notre proposition est d'utiliser des mesures statistiques ou des techniques de « machine learning », pour estimer la valeur du seuil à adopter pour chaque membre en fonction de son activité dans la CoP.

3. Perspectives à long terme

Sur le long terme, nous proposons de rendre le système NICOLAT générique et capable de fonctionner pour n'importe quelle Communauté de Pratique dans laquelle le principal moyen de partage des connaissances est la résolution communautaire de problèmes. Nous souhaitons aussi rendre les différents composants indépendants pour qu'il soit facile de les intégrer dans d'autres systèmes.

Nous proposons que la généricité de NICOLAT soit assurée comme suit :

- Nous proposons de présenter le noyau communautaire comme une distribution Drupal. Notons qu'une distribution Drupal est un Drupal avec des modules et des thèmes préconfigurés que l'on utilise pour des installations d'un nouveau site. Avant l'utilisation de ce noyau par une CoP, il suffit de définir des attributs constituant le contexte de l'utilisateur et celui du problème la nouvelle CoP, suite à notre description de la généricité du noyau dans la section 5 du chapitre 2.
- Nous proposons de présenter la couche RP sous forme d'un module Drupal configurable. Ainsi, il sera possible d'identifier les attributs à exploiter par ce module dans la nouvelle CoP, leurs poids et de personnaliser le processus de la validation d'une solution et sa structure comme indiqué dans la section 6 du chapitre 2. Des liens entre les mots répandus du domaine cible pourront être définis afin de prendre en considération l'aspect sémantique de ces mots pour assurer l'efficacité du processus de recherche.
- Les composants et les applications permettant à l'utilisateur d'interagir via son outil d'interaction préféré seront aussi personnalisables. Dans ce cadre, l'application Facebook sera personnalisable de la même manière que le noyau communautaire. Cependant, les composants permettant d'interagir via le SMS et l'Email ainsi que l'application Android ne pourront être personnalisables qu'en utilisant des fichiers XML qui vont contenir les différents attributs utilisés par le système ainsi que les informations techniques permettant de communiquer avec le noyau.
- Pour l'approche de la sélection des membres, il sera possible de spécifier les règles sur lesquelles se basera la sélection statique ainsi que de définir les attributs sur lesquels se basera la sélection dynamique et leurs natures (relative/absolue).

Bibliographie

- Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1), 39-59.
- Alfonseca, E., Carro, R. M., Martín, E., Ortigosa, A., & Paredes, P. (2006). The impact of learning styles on student grouping for collaborative learning: a case study. *User modeling and user-adapted interaction*, 16(3-4), 377-401.
- Alhalabi, M.-F. (2009). *Modélisation et Développement d'Applications avec Comportement Adaptable*. (Thèse de doctorat), INSA de Lyon, France.
- Azouaou, F., & Desmoulins, C. (2006). *Using and modeling context with ontology in e-learning: the case of teacher's personal annotation*. Paper presented at the International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL), pp. 1-12, Dublin, Ireland.
- Bauer, J. (2003). *Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic*. (Thèse de doctorat), Université technique de Berlin, Allemagne.
- Baziz, M. (2005). *Indexation conceptuelle guidée par ontologie pour la recherche d'information*. (Thèse de doctorat), Université de Toulouse 3, France.
- Bergmann, R. (2003). *Developing industrial case-based reasoning applications: The INRECA methodology* (Vol. 1612): Springer Science & Business Media.
- Boticario, J. G., & Santos, O. C. (2007). An Open IMS-Based User Modelling Approach for Developing Adaptive Learning Management Systems. *Journal of Interactive Media in Education*, 19.
- Bourhis, A., Dubé, L., & Jacob, R. (2004). La contribution de la gestion des connaissances à la gestion de la relève. *Gestion*, 29(3), 73-81.
- Brauer, S., & Schmidt, T. C. (2012). *Group formation in elearning-enabled online social networks*. Paper presented at the 15th International Conference on Interactive Collaborative Learning, pp. 1-8, Villach, Austria.
- Brown, P. J. (1995). The stick-e document: a framework for creating context-aware applications. *ELECTRONIC PUBLISHING-CHICHESTER-*, 8, 259-272.
- Brown, P. J., Bovey, J. D., & Chen, X. (1997). Context-aware applications: from the laboratory to the marketplace. *Personal Communications, IEEE*, 4(5), 58-64.
- Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User modeling and user-adapted interaction*, 6(2-3), 87-129.
- Brusilovsky, P. (2001). User Modeling and User-Adapted Interaction. *Adaptive Hypermedia*, 11(1-2), 87-110.
- Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education (IJAIED)*, 13, 159-172.
- Burgos, D. (2008). *Extension of the IMS Learning Design Specification based on Adaptation and Integration of Units of Learning*. (Thèse de doctorat), Université Carlos III de Madrid, Espagne.

- Caballé, S., Daradoumis, T., & Khafa, F. (2007). Efficient embedding of information and knowledge into CSCL applications. *Technologies for E-Learning and Digital Entertainment* (pp. 548-559): Springer.
- Cao, N., Lu, L., Lin, Y.-R., Wang, F., & Wen, Z. (2015). SocialHelix: visual analysis of sentiment divergence in social media. *Journal of Visualization*, 18(2), 221-235.
- Chaari, T. (2007). *Adaptation d'applications pervasives dans des environnements multi-contextes*. (Thèse de doctorat), INSA de Lyon, France.
- Charlier, B., Boukottaya, A., Daele, A., Deschryver, N., El Helou, S., & Naudet, Y. (2007). *Designing services for CoPs: first results of the PALETTE project*. Paper presented at the TEL-CoPs' 07: 2nd International Workshop on Building Technology Enhanced Learning Solutions for Communities of Practice, pp. 76-86, Crete, Greece.
- Charlier, B., Nizet, J., & Van Dam, D. (2005). *Voyage au pays de la formation des adultes: dynamiques identitaires et trajectoires sociales*: Editions L'Harmattan.
- Chen, H., Finin, T., & Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(03), 197-207.
- Chen, W. (2006). Supporting teachers' intervention in collaborative knowledge building. *Journal of Network and Computer Applications*, 29(2), 200-215.
- Coutaz, J., Crowley, J. L., Dobson, S., & Garlan, D. (2005). Context is key. *Communications of the ACM*, 48(3), 49-53.
- de Vrij, N., & Helms, R. (2006). *Application of a Community of Practice to improve knowledge sharing in offshoring relations*. Paper presented at the 17th International Workshop on Database and Expert Systems Applications (DEXA'06), pp. 161-165, Krakow, Poland.
- Devèze, B., & Fouquin, M. (2004). *Case-Based Reasoning*. Ecole de l'ingénieur EPTA, France. Retrieved from <http://devezeb.free.fr/downloads/ecrits/cbr.pdf>
- Dey, A. K., & Abowd, G. D. (1999). *Towards a better understanding of context and context-awareness*. Paper presented at the Handheld and ubiquitous computing, pp. 304-307, Karlsruhe, Germany.
- Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction*, 16(2), 97-166.
- Dieng, R., Corby, O., Gandon, F., Giboin, A., Golebiowska, J., Matta, N., & Ribière, M. (2001). *Méthodes et outils pour la gestion des connaissances: une approche pluridisciplinaire du " Knowledge Management"* (Vol. 2): Dunod.
- Dillenbourg, P., Poirier, C., & Carles, L. (2003). Communautés virtuelles d'apprentissage: e-jargon ou nouveau paradigme. A. Taurisson et A. Sentini. *Pédagogies. Net. Montréal, Presses*, 11-47.
- Dubé, L., & Bourhis, A. (2003). The impact of structural characteristics on the launching of intentionally formed virtual communities of practice. *Cahier du GReSI*, 3(9), 1-20.
- Duque, R., Gómez-Pérez, D., Nieto-Reyes, A., & Bravo, C. (2015). Analyzing collaboration and interaction in learning environments to form learner groups. *Computers in Human Behavior*, 47, 42-49.

- Fahy, P., & Clarke, S. (2004). *CASS—a middleware for mobile context-aware applications*. Paper presented at the MobiSys Workshop on Context Awareness, pp. 304-308, Boston, USA.
- Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering education*, 78(7), 674-681.
- Freedman, R., Ali, S. S., & McRoy, S. (2000). Links: what is an intelligent tutoring system?. *intelligence*, 11(3), 15-16.
- Fuchs, B., Lieber, J., Mille, A., & Napoli, A. (2006). *Une première formalisation de la phase d'élaboration du raisonnement à partir de cas*. Paper presented at the 14ième atelier du raisonnement à partir de cas, Besançon, France.
- Garrot, E. (2008). *Plate-forme support à l'Interconnexion de Communautés de Pratique. Application au tutorat avec TE-Cap*. (Thèse de doctorat), INSA de Lyon, France.
- Göth, C., & Schwabe, G. (2008). *Designing tasks for engaging mobile learning*. Paper presented at the MLearn2008 conference, pp. 152-159, Telford, UK.
- Gu, T., Pung, H. K., & Zhang, D. Q. (2004). *A middleware for building context-aware mobile services*. Paper presented at the Vehicular Technology Conference, pp. 2656-2660.
- Hoareau, C., & Satoh, I. (2009). Modeling and processing information for context-aware computing: A survey. *New Generation Computing*, 27(3), 177-196.
- Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., & Retschitzegger, W. (2003). *Context-awareness on mobile devices-the hydrogen approach*. Paper presented at the 36th Annual Hawaii International Conference on System Sciences, p. 10, Big Island, HI, USA.
- Hull, R., Neaves, P., & Bedford-Roberts, J. (1997). *Towards situated computing*. Paper presented at the First International Symposium on Wearable Computers, pp. 146-153, Cambridge, MA, USA.
- Indulska, J., Robinson, R., Rakotonirainy, A., & Henriksen, K. (2003). *Experiences in using cc/pp in context-aware systems*. Paper presented at the 4th International Conference on Mobile Data Management, pp. 247-261, Melbourne, Australia.
- Jaczynski, M. (1998). *Modèle et plate-forme à objets pour l'indexation des cas par situations comportementales: application à l'assistance à la navigation sur le web*. (Thèse de doctorat), Université de Nice, France.
- Jagadish, D. (2014). *Grouping in collaborative e-learning environment based on interaction among students*. Paper presented at the International Conference on Recent Trends in Information Technology (ICRTIT), pp. 1-5, Chennai, India.
- Keegan, D. (2007). *Mobile learning: a practical guide*. Leonardo da Vinci programme of the European Commission. Retrieved from http://www.ericsson.com/thecompany/company_facts/businesses/programs/incorporating-mobile-learning-into-mainstream-education
- Kolodner, J., & Leake, D. (1996). A tutorial introduction to case-based reasoning. *Case-based Reasoning: Experiences, Lessons and Future Directions* (pp. 31-65): AAAI Press/MIT Press.

- Krogseter, M., Oppermann, R., & Thomas, C. (1994). A user interface integrating adaptability and adaptivity. *Adaptive user support: Ergonomic design of manually and automatically adaptable software* (pp. 97-125).
- Lamontagne, L., & Lapalme, G. (2002). Raisonnement à base de cas textuels: Etat de l'art et perspectives. *Revue d'intelligence artificielle*, 16(3), 339-366.
- Langelier, L. (2005). *Guide de mise en place et d'animation de communautés de pratique intentionnelles: travailler, apprendre et collaborer en réseau*. Montréal: CEFRIO. Retrieved from http://www.cefrio.qc.ca/media/uploader/travailler_apprendre_collaborer.pdf
- Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*: Cambridge university press.
- Lecllet, D., & Talon, B. (2008). *La méthode pédagogique MAETIC*: Libro Veritas.
- Lewis, D. D. (1992). *Representation and learning in information retrieval*. (Thèse de doctorat), Université du Massachusetts, USA.
- Lilleorg, A., Tammets, K., Evert, T., & Ley, T. (2014). *Knowledge sharing tools, practices and barriers in transnational clusters*. Paper presented at the 14th International Conference on Knowledge Technologies and Data-driven Business, p. 8, Graz, Austria.
- Lipponen, L. (2002). *Exploring foundations for computer-supported collaborative learning*. Paper presented at the The Computer-supported Collaborative Learning Conference (CSCL '02), pp. 72-81, Boulder, CO, USA.
- Liu, B. (2011). Opinion mining and sentiment analysis. *Web Data Mining* (pp. 459-526): Springer.
- Magnisalis, I., Demetriadis, S., & Karakostas, A. (2011). Adaptive and intelligent systems for collaborative learning support: a review of the field. *IEEE Transactions on Learning Technologies*, 4(1), 5-20.
- Martín, E., & Paredes, P. (2004). *Using Learning Styles for Dynamic Group Formation in Adaptive Collaborative Hypermedia Systems*. Paper presented at the 4th International Conference on Web Engineering (ICWE 2004), pp. 188-198, Munich, Germany.
- Miao, Y. (2005). *Enabling learning designers to model dynamic learning processes*. Paper presented at the 5th IEEE International Conference on Advanced Learning Technologies, pp. 399-401, Kaohsiung, Taiwan.
- Mille, A. (2006). *Traces Based Reasoning (TBR) Definition, illustration and echoes with story telling*. (5205), Citeseer. Retrieved from <http://liris.cnrs.fr/Documents/Liris-2282.pdf>
- Nguyen, C. P. (2010). *Conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario*. (Thèse de doctorat), Télécom Bretagne, France.
- Nkambou, R., Bourdeau, J., & Mizoguchi, R. (2010). Introduction: what are intelligent tutoring systems, and why this book?. *Advances in Intelligent Tutoring Systems* (pp. 1-12): Springer.

- Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4(4), 251-277.
- O'Malley, C., Vavoula, G., Glew, J., Taylor, J., Sharples, M., Lefrere, P., . . . Waycott, J. (2005). *Guidelines for learning/teaching/tutoring in a mobile environment*. Mobilelearn project deliverable. Retrieved from <https://hal.archives-ouvertes.fr/hal-00696244/document>
- OEAS. (2012). *Development of Virtual Communities of Practice to Support Programmatic Efforts within University Systems*. Office of Educational Access and Success (OEAS), University System of Georgia. Retrieved from http://www.usg.edu/educational_access/documents/OEAS_12-01_Communities_of_Practice.pdf
- Oguz, F. (2007). *An exploration of the diffusion of a new technology from communities of practice perspective: Web services technologies in digital libraries*: University of North Texas.
- Oppermann, R., & Rasher, R. (1997). Adaptability and adaptivity in learning systems. *Knowledge transfer*, 2, 173-179.
- Ouraiba, E. A. (2012). *Scénarisation pédagogique pour des EIAH ouverts: Une approche dirigée par les modèles et spécifique au domaine métier*. (Thèse de doctorat), Université du Maine, France.
- Paasivaara, M., & Lassenius, C. (2014). Communities of practice in a large distributed agile software development organization—Case Ericsson. *Information and Software Technology*, 56(12), 1556-1577.
- Paltoglou, G. (2014). Sentiment analysis in social media. *Online Collective Action* (pp. 3-17): Springer.
- Papadimitriou, A., Grigoriadou, M., & Gyftodimos, G. (2012). MATHEMA: A Learner-controlled Adaptive Educational Hypermedia System. *Journal of Information Technology and Application in Education*, 1(2), 47-73.
- Paredes, P., & Rodriguez, P. (2006). *The application of learning styles in both individual and collaborative learning*. Paper presented at the Sixth International Conference on Advanced Learning Technologies pp. 1141-1142, Kerkrade, The Netherlands.
- Park, Y. (2011). A pedagogical framework for mobile learning: Categorizing educational applications of mobile technologies into four types. *The International Review of Research in Open and Distributed Learning*, 12(2), 78-102.
- Pascoe, J., Ryan, N. S., & Morse, D. R. (1998). *Human computer giraffe interaction: HCI in the field*. Paper presented at the Workshop on Human Computer Interaction with Mobile Devices, Glasgow, Scotland.
- Peers, J. (2007). *Learning through collaboration: An investigation of communities of practice in Streamkeepers groups*. (Thèse de Master), Université de Guelph, Canada.
- Pernin, J.-P. (2003). Objets pédagogiques: unités d'apprentissage, activités ou ressources. *Revue" Sciences et Techniques Educatives", Hors série*, 179-210.
- Persistent. (2012). Choosing the Right Collaboration Platform for Your Organization. Retrieved September 2015, from http://www.persistent.com/sites/default/files/wp_Collab_Platform_Comprision.pdf

- Phobun, P., & Vicheanpanya, J. (2010). Adaptive intelligent tutoring systems for e-learning systems. *Procedia-Social and Behavioral Sciences*, 2(2), 4064-4069.
- Popovici, D. (2012). *Gestion du contexte pour des applications mobiles dédiées aux transports*. (Thèse de doctorat), Université de Valenciennes, France.
- Quinn, C. (2000). mLearning: Mobile, wireless, in-your-pocket learning. Retrieved September 2015, from <http://www.linezine.com/2.1/features/cqmmwiyp.htm>
- Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S. M., Ritter, A., & Stoyanov, V. (2015). *Semeval-2015 task 10: Sentiment analysis in twitter*. Paper presented at the 9th International Workshop on Semantic Evaluation, SemEval, Denver, Colorado, USA.
- Sabo, M. (2015). *Drupal Commons in Practice: Social Business and Social Network*. (Thèse de Master), Université de Masaryk, Tchègue.
- Salton, G. (1971). *The SMART retrieval system—experiments in automatic document processing*: Prentice-Hall.
- Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5), 513-523.
- Sancho, P., Fernandez, R. F., & Manjon, B. (2008). *NUCLEO: Adaptive computer supported collaborative learning in a role game based scenario*. Paper presented at the Eighth IEEE International Conference on Advanced Learning Technologies (ICALT'08). pp. 671-675, Santander, Cantabria.
- Sawant, R., Singhal, A., Nigam, P., & Shah, U. (2011). *ClassroomWiki: a collaborative Wiki for institutional use*. Paper presented at the The International Conference & Workshop on Emerging Trends in Technology, p. 1359, Mumbai, Maharashtra, India.
- Schilit, B., Adams, N., & Want, R. (1994). *Context-aware computing applications*. Paper presented at the First Workshop on Mobile Computing Systems and Application (WMCSA 1994) pp. 85-90, Santa Cruz, CA, USA.
- Schilit, B. N., & Theimer, M. M. (1994). Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5), 22-32.
- Shannon, C. E. (1948). *A mathematical theory of communication*: American Telephone and Telegraph Company.
- Shute, V. J., & Regian, J. (1993). Principles for evaluating intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, 4, 2-3.
- Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. *Cambridge handbook of the learning sciences*, 2006, 409-426.
- Stars, B. (2012). *Knowledge Sharing Tools and Practices*. BSR Stars. Retrieved from <http://www.bsrstars.se/wp-content/uploads/2012/10/Report-Knowledge-sharing-tools.pdf>
- Stenuit Hautdidier, F. (2006). *Créer et animer des communautés de pratique: Préconisations pour une entreprise de formation et de conseil*. (Thèse de doctorat), Institut national des techniques de la documentation, Paris, France.

- Strang, T., & Linnhoff-Popien, C. (2004). *A context modeling survey*. Paper presented at the First International Workshop on Advanced Context Modelling, Reasoning And Management, Nottingham, England.
- Taylor, S. J., & Bogdan, R. (1984). *Introduction to qualitative research methods: The search for meaning*: Wiley.
- Topousis, D. E., Dennehy, C. J., & Lebsack, K. L. (2012). Nasa's Experiences Enabling the Capture and Sharing of Technical Expertise Through Communities of Practice. *Acta Astronautica*, 81(2), 499-511.
- Tourtoglou, K., & Virvou, M. (2010). Simulated Annealing in Finding Optimum Groups of Learners of UML. *Intelligent Interactive Multimedia Systems and Services* (pp. 147-156): Springer.
- Towle, B., & Halm, M. (2005). Designing adaptive learning environments with learning design. *Learning design* (pp. 215-226): Springer.
- Tricot, A. (2007). L'expérimentation et la démarche scientifique. *Les Cahiers de l'Acedle*(4), 85-92.
- Tricot, A., & Lafontaine, J. (2002). Une méthode pour évaluer conjointement l'utilisation un outil multimédia et l'apprentissage réalisé avec celui-ci. *Le français dans le Monde*, 41-52.
- Truong, B. A., Lee, Y.-K., & Lee, S.-Y. (2005). *Modeling and reasoning about uncertainty in context-aware systems*. Paper presented at the Conference on e-Business Engineering (ICEBE 2005) pp. 102-109, Beijing, China.
- Vaast, E. (2002). *Les communautés de pratique sont-elles pertinentes?* Paper presented at the Conférence de l'Association Internationale de Management Stratégique (AIMS 2012), Paris, France.
- Valk, J.-H., Rashid, A. T., & Elder, L. (2010). Using mobile phones to improve educational outcomes: An analysis of evidence from Asia. *The International Review of Research in Open and Distributed Learning*, 11(1), 117-140.
- Van Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P., & Koper, R. (2007). Authoring a full life cycle model in standards-based, adaptive e-learning. *Journal of Educational Technology & Society*, 9(1), 72-83.
- Vermunt, J. D. (1992). Learning styles and directed learning processes in higher education: towards a process-oriented instruction in independent thinking: Lisse: Swets and Zeitlinger.
- Viana, W. (2010). *Mobilité et sensibilité au contexte pour la gestion de documents multimédias personnels: CoMMedia*. (Thèse de doctorat), Université de Grenoble, France.
- Wenger, E. (1998). Communities of practice: Learning, meaning, and identity. *Cambridge UP*.
- Wenger, E. (2005). *La théorie des communautés de pratique*: Presses Université Laval.
- Wenger, E., McDermott, R. A., & Snyder, W. (2002). *Cultivating communities of practice: A guide to managing knowledge*: Harvard Business Press.

- Xiaohang, W. (2003). *The Context Gateway: A Pervasive Computing Infrastructure for Context Aware Service*. School of Computing, National University of Singapore & Context-Aware Department., Institute for Infocomm Research. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.8304&rep=rep1&type=pdf>
- Yin, C. (2010). *Samcco: un Système d'Apprentissage Mobile Contextuel et Collaboratif dans des Situations Professionnelles*. (Thèse de doctorat), Ecole Centrale de Lyon, France.
- Zarraonandia, T., Fernández, C., & Doderó, J. M. (2006). *A late modelling approach for the definition of computer-supported learning process*. Paper presented at the Workshop on Adaptive Learning and Learning Design, Dublin, Ireland.

Publications et Communications

Belmeskine, R., Begdouri, A., & Groux-Lecllet, D. Favoriser les interactions entre enseignants d'une Communauté de Pratique via des dispositifs mobiles : Expérience dans une FST au Maroc. Article accepté pour publication dans le proceedings de la conférence TICE'2014. Béziers, France, 18 – 20 Novembre 2014.

Belmeskine, R., Lecllet-Groux, D., & Begdouri, A. Toward Maximizing Access Knowledge in Learning: Adaptation of Interactions in a CoP Support System. In *Open Learning and Teaching in Educational Communities* (pp. 550-551). Springer International Publishing. EC-TEL, Graz, Austria, 16-19 September 2014.

Belmeskine, R. Le système iNformatIque COmmunautaire mobiLe et AdapTatif : Le système NICOLAT. Les cinquièmes rencontres jeunes chercheurs en EIAH (RJC EIAH 2014), La Rochelle, France, 16 – 17 Juin 2014.

Belmeskine, R., Begdouri, A., & Groux-Lecllet, D. (2013). Resolution of difficulties approach for a Community of Practice members: design, implementation and experiment. *International Journal Of Research In Education Methodology*, 4(1), 409-422.

Belmeskine, R., Begdouri, A., & Lecllet, D. Architecture for the "Adaptation of Interactions" layer in the CCE-MAETIC (Community Computing Environment support of the MAETIC community of Practice). In *International Conference on Engineering Education and Research (ICEER 2013)*, Marrakesh, Morocco, 1 – 5 Juillet 2013.

Belmeskine, R., Lecllet, D., & Begdouri, A. Environnement Informatique Communautaire support d'une CoP pour l'aide à la résolution de difficultés. In *8ème Colloque Technologies de l'Information et de la Communication pour l'Enseignement (TICE 2012)*, Pages 31-43, Lyon, France, 11 – 13 Décembre 2012.

Belmeskine, R., Begdouri, A., & Lecllet, D. Community Computer Environment supports a COP, CBR approach to solve difficulties. In *IEEE International Colloquium on Information Science and Technology (CIST'2012)*, Fez, Morocco, 22-24 Octobre 2012.

Questionnaire – Le système NICOLAT

Merci de prendre le temps de répondre à ce questionnaire réalisé par :

D. Lecllet, A. Begdouri et R. Belmeskine

Quel est votre statut en tant qu'enseignant ?

Quelle est la discipline enseignée ?

Quel est le domaine d'enseignement ?

Quel est votre âge ?

Sexe ? F M

A-Utilisation de Facebook

Utilisez-vous Facebook ? Oui Non

Si oui,

Consultez-vous votre compte Facebook ?

Rarement Occasionnellement Fréquemment Souvent

Avez-vous l'habitude de manipuler des applications Facebook ? Oui Non

Est-ce que vous trouvez que notre application Facebook est facile à manipuler ?

Oui Non

Est-ce que vous trouvez que l'utilisation de l'application Facebook au lieu de l'EIC répond plus à vos habitudes ? Oui Non

Si vous recevez une notification Facebook d'une demande d'aide à la résolution d'une difficulté MAETIC lors de votre connexion sur Facebook, Pensez-vous que vous pouvez répondre immédiatement ? Oui Non

Utilisez-vous la version mobile de Facebook ? Oui Non

Si oui,

Si vous recevez une notification Facebook d'une demande d'aide à la résolution d'une difficulté MAETIC sur votre dispositif mobile, Pensez-vous que vous pouvez répondre immédiatement ? Oui Non

B-Utilisation de l'Email

Utilisez-vous l'email ? Oui Non

Si oui,

Quel service de messagerie utilisez-vous ?

Gmail Hotmail Yahoo Autre .

Consultez-vous votre boîte mail ?

Rarement Occasionnellement Fréquemment Souvent

Est-ce que vous trouvez que l'utilisation de l'email est facile ?

Oui Non

Est-ce que vous trouvez que l'utilisation de l'email au lieu de l'EIC répond plus à vos habitudes ? Oui Non

Si vous recevez un email d'une demande d'aide à la résolution d'une difficulté MAETIC lors de votre connexion à votre boîte mail, Pensez-vous que vous pouvez répondre immédiatement ? Oui Non

Consultez-vous votre boîte mail sur un dispositif mobile ? Oui Non

Si oui,

Si vous recevez un email d'une demande d'aide à la résolution d'une difficulté MAETIC sur votre dispositif mobile, Pensez-vous que vous pouvez répondre immédiatement ? Oui Non

C-Utilisation de l'application Android

Utilisez-vous un Smartphone ou une tablette Android? Oui Non

Si oui,

Avez-vous accès à internet ? Oui Non

Vous téléchargez de nouvelles applications ? Jamais Rarement
Occasionnellement Fréquemment Souvent

Est-ce que vous trouvez que cette application est facile à manipuler ?

Oui Non

Si vous recevez une notification de la part de NICOLAT sur votre dispositif, Pensez-vous que vous pouvez répondre immédiatement ?

Oui Non

D-Utilisation des SMS

Utilisez-vous le SMS ? Oui Non

Si oui,

Dans la moyenne, combien de SMS vous envoyez par jour ?

Moins de 5 Entre 5 et 10 Plus de 10

Si vous recevez un SMS de la part de NICOLAT, Pensez-vous que vous pouvez répondre immédiatement ? Oui Non

Annexe 2 : Profils des membres de la communauté

La communauté que nous avons élaborée se compose de 30 membres {U0,U2, ..., U29}. Le tableau 13 ci-dessous présente les profils pédagogiques de ces membres. Par rapport aux attributs compétence MAETIC, compétence GP, compétence Blog, la valeur 1 signifie que l'enseignant est novice, la valeur 2 signifie qu'il est intermédiaire et la valeur 3 signifie qu'il est expert.

Membre	Spécialité	Comp. MAETIC	Comp. GP	Comp. Blog
U0	Géologie	3	3	3
U1	Industriel	2	2	3
U2	Electronique	3	2	2
U3	Géologie	2	3	3
U4	Telecom	2	2	3
U5	Informatique	2	2	3
U6	Industriel	2	3	1
U7	Chimie	2	2	2
U8	Géologie	1	2	2
U9	Telecom	1	2	2
U10	Mécanique	1	2	1
U11	Informatique	1	1	2
U12	Industriel	1	1	2
U13	Mécanique	1	2	1
U14	Mécanique	1	2	1
U15	Electronique	1	1	1
U16	Math	1	1	1
U17	Math	1	1	1
U18	Mécanique	1	1	1
U19	Informatique	1	1	1
U20	Biologie	1	1	1
U21	Biologie	1	1	1
U22	Telecom	1	1	1
U23	Chimie	1	1	1
U24	Chimie	1	1	1
U25	Biologie	1	1	1
U26	Electronique	1	1	1
U27	Telecom	1	1	1
U28	Chimie	1	1	1
U29	Electronique	1	1	1

Tab 13. Profils pédagogique des membres de la communauté

Lors de la constitution des profils des membres de la communauté et dans l'objectif d'être au plus proche de la réalité, nous avons veillé sur la diversité de ces profils en termes de spécialités. Par rapport aux niveaux de compétences, nous avons veillé sur le fait qu'il doit y avoir plus d'enseignants novices que d'intermédiaires et plus d'intermédiaires que d'experts.

Cette communauté se compose de 3 (10%) enseignants qui enseignent la biologie, 4 (13,33%) enseignent la chimie, 4 (13,33%) enseignent l'électronique, 3 (10%) enseignent la géologie, 3 (10%) enseignent l'informatique, 4 (13,33%) enseignent la mécanique, 4 (13,33%) enseignent la télécommunication, 2 enseignent (6,66%) les maths et 3 enseignent (10%) le génie industriel. La figure 68 ci-dessous présente la répartition des spécialités des membres de la communauté.

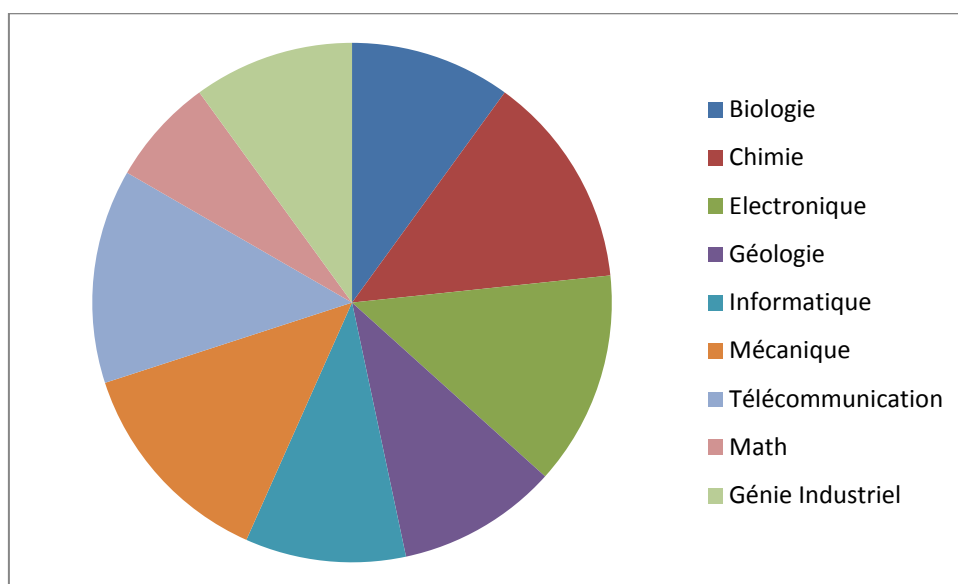


Figure 68. La répartition des spécialités des membres de la communauté

Ainsi, 22 (73,33%) enseignants sont novices par rapport à l'usage de MAETIC, 6 (20%) ont un niveau intermédiaire et 2 (6,66%) sont des experts. Par rapport à la maîtrise de la gestion de projet, 17 (56,66%) enseignants sont novices, 10 (33,33%) ont un niveau intermédiaire et 3 (10%) sont des experts. Enfin, 19 (63,33%) enseignants sont experts dans l'usage des blogs, 6 (20%) ont un niveau intermédiaire et 5 (16,66%) sont experts. La figure 69 ci-dessous présente la répartition des membres de la CoP par rapport aux différents niveaux liés à ces compétences.

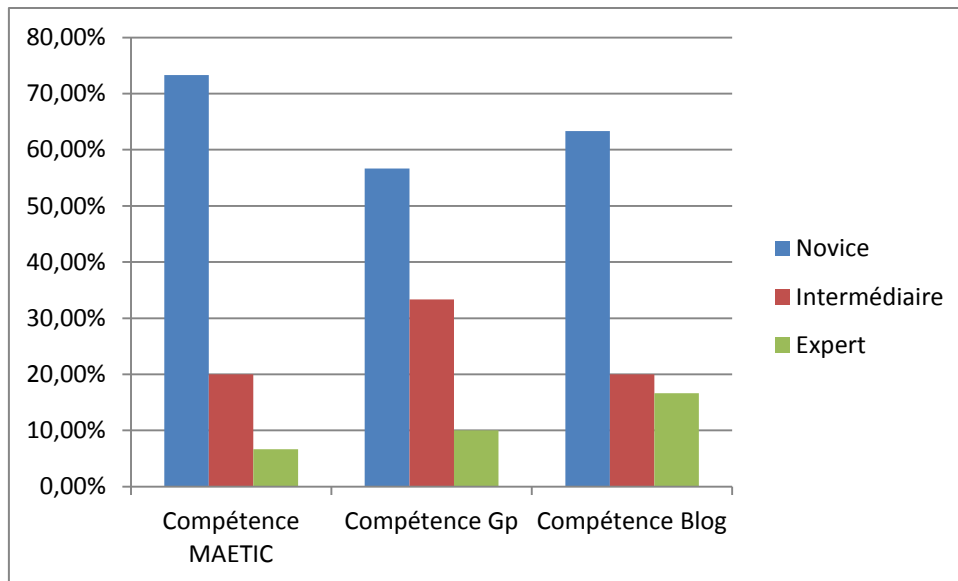


Figure 69. La répartition des niveaux de compétences des membres de la communauté

Annexe 3 : Traces des contributions des membres de la communauté

Pour pouvoir simuler les étapes de la sélection dynamique, il nous faut les traces de contributions des membres de la communauté liées aux difficultés de même type que celui de la difficulté cible. Dans ce cadre, nous procédons à générer des traces de contributions en partant d'une base qui se compose de 30 exemples de contextes de difficultés. Ces contextes sont présentés dans le tableau 14 ci-dessous.

Contexte
{Informatique, [21,30], 4, [31, 40], 1, Licence, 2, 2, 2}
{Informatique, [+31], 4, [20, 30], 1, Master, 3, 2, 3}
{Informatique, [+31], 4, [20, 30], 2, Master, 3, 2, 2}
{Informatique, [21, 30], 4, [20, 30], 1, Licence, 3, 3, 1}
{Chimie, [21, 30], 2, [41, 50], 1, DUT, 2, 2, 2}
{Industriel, [21, 30], 3, [+50], 1, DUT, 2, 2, 3}
{Industriel, [+31], 5, [+50], 3, Licence, 1, 2, 2}
{Industriel, [11, 20], 5, [+50], 2, Master, 1, 3, 3}
{Telecom, [+31], 3, [41, 50], 3, DUT, 1, 2, 2}
{Math, [11, 20], 1, [41, 50], 3, DUT, 1, 2, 3}
{Chimie, [21, 30], 1, [+50], 3, DUT, 3, 2, 3}
{Informatique, [11, 20], 4, [31, 40], 3, DUT, 1, 2, 1}
{Electronique, [+31], 3, [+50], 1, Licence, 2, 1, 1}
{Industriel, [+31], 1, [41, 50], 2, Master, 2, 2, 3}
{Informatique, [11, 20], 6, [20, 30], 1, Licence, 2, 1, 2}
{Biologie, [21, 30], 1, [20, 30], 2, DUT, 1, 3, 3}
{Telecom, [21, 30], 1, [41, 50], 3, Licence, 3, 2, 3}
{Géologie, [+31], 4, [41, 50], 3, DUT, 1, 3, 2}
{Telecom, [+31], 5, [20, 30], 1, DUT, 1, 2, 2}
{Informatique, [+31], 2, [20, 30], 3, Licence, 1, 1, 2}
{Géologie, [11, 20], 2, [41, 50], 3, Master, 2, 1, 3}
{Chimie, [11, 20], 3, [41, 50], 3, DUT, 3, 3, 1}
{Mécanique, [21, 30], 2, [+50], 1, Licence, 3, 3, 1}
{Mécanique, [+31], 3, [31, 40], 2, Master, 1, 3, 1}
{Géologie, [21, 30], 1, [31, 40], 3, DUT, 3, 3, 3}
{Telecom, [11, 20], 4, [31, 40], 3, Master, 2, 3, 1}
{Biologie, [+31], 3, [31, 40], 1, Master, 1, 3, 1}
{Math, [+31], 1, [31, 40], 3, DUT, 2, 1, 1}
{Electronique, [11, 20], 3, [20, 30], 2, Master, 2, 1, 1}
{Géologie, [+31], 1, [20, 30], 1, DUT, 2, 3, 3}

Tab 14. Base des contextes utilisés pour générer les traces

Ces contextes ont été générés aléatoirement. Cependant, nous avons modifié 10% d'eux pour qu'ils soient similaires au contexte de la difficulté cible. Avoir des traces des contextes similaires à celui de la difficulté cible est nécessaire pour pouvoir évaluer la sélection dynamique. Sinon, cette dernière n'ajoute rien et le système ne sollicitera que les membres sélectionnés en utilisant la sélection statique.

Ainsi, nous avons configuré les nombres de contributions des membres pour qu'ils soient proportionnels par rapport à leurs profils. L'objectif ici est d'être le plus proche d'une communauté réelle. La figure 70 ci-dessous présente la répartition des nombres de contributions des membres de la CoP par rapport aux difficultés de type 'Pannes réseau'.

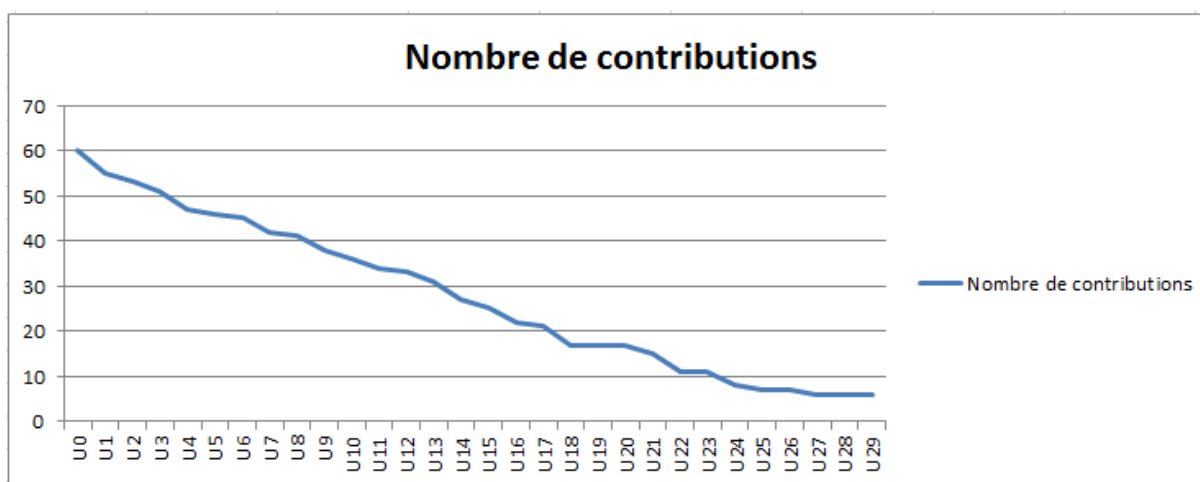


Figure 70. Nombres de contributions des membres de la communauté par rapport difficultés de type « Panne réseaux »

Le tableau 15 ci-dessous présente les traces de contributions des membres candidats liées aux difficultés de type 'Pannes réseau' ainsi que les indices de contributions correspondant à ces traces. Dans ce tableau nous présentons les traces en utilisant la représentation du contexte pertinent.

	Indices de contributions	traces de contributions			Indices de contributions	traces de contributions	
U0	<ul style="list-style-type: none"> {<u>11</u>,8,5,8,5,2,10,2,9}, {23,13,24}, {18,5,15,<u>14</u>,7,1}, {17,12,21,10}, {15,9,36}, {60,<u>20</u>,11}, {60,<u>34</u>,17}, {60,48,20}, {60,37,19}} 	1411:4	4311:3	U13	<ul style="list-style-type: none"> {<u>10</u>,5,0,2,6,1,4,3,0}, {8,9,14}, {8,4,6,<u>8</u>,3,2}, {12,5,8,6}, {15,4,12}, {31,<u>16</u>,7}, {31,<u>20</u>,9}, {31,27,12}, {31,21,12}} 	1411:3	4511:1
		1622:1	4511:4			1622:2	5312:2
		<u>1423:1</u>	5531:1			<u>1423:2</u>	5132:2
		<u>1422:1</u>	5312:1			1221:1	5521:2
		1221:1	5132:1			<u>1433:1</u>	6223:1
		<u>1433:2</u>	5521:2			<u>1433:1</u>	7313:2
		<u>1433:1</u>	6223:2			2232:1	7113:1
		2232:1	7313:6			2112:3	7212:1
		2113:2	7113:3			2411:1	8111:1
		2112:1	7212:1			4123:1	8331:2
2411:4	8111:2						
3332:4	9111:5						
				U14	{ <u>3</u> ,5,0,4,8,1,0,3,3},	1622:1	5531:2

		3322:1 4432:1	9112:4		{7,6,14}, {11,1,6, <u>3</u> ,5,1}, {7,6,7,7}, {9,6,12}, {27, <u>16</u> ,10}, {27, <u>15</u> ,5}, {27,24,11}, {27,22,16}}	<u>1433:1</u> <u>1433:1</u> 2232:1 2113:1 2112:3 4432:1 4123:2 4311:1	5312:2 5132:1 5521:3 6331:1 8111:1 8331:2 9111:2 9112:1	
U1	{ <u>11</u> ,7,2,7,7,6,6,4,5}, {15,22,18}, {17,4,15, <u>15</u> ,3,1}, {8,22,12,13}, {22,9,24}, {55, <u>28</u> ,16}, {55, <u>35</u> ,13}, {55,49,23}, {55,34,23}}	1411:3 1622:1 <u>1423:1</u> <u>1422:5</u> <u>1433:1</u> 2232:1 2113:2 2112:2 2411:2 3332:1 3322:1 4432:3 4123:2 4311:1	4511:1 5531:2 5312:4 5132:1 6223:2 6331:4 7313:1 7113:4 7212:1 8111:1 8331:3 9111:3 9112:2		U15	{{ <u>5</u> ,3,1,3,5,2,5,0,1}, {8,8,9}, {7,5,3, <u>1</u> ,5,4}, {8,3,5,9}, {10,2,13}, {25, <u>15</u> ,3}, {25, <u>18</u> ,7}, {25,18,8}, {25,19,7}}	1622:4 1221:1 2113:1 2112:2 3332:1 4432:1 4123:1 4311:1	5531:1 5521:4 6223:2 7313:1 7113:2 7212:2 9112:1
U2	{{ <u>11</u> ,12,2,6,8,4,2,6,2}, {19,14,20}, {13,8,11, <u>11</u> ,7,3}, {14,17,14,8}, {13,16,24}, {53, <u>31</u> ,21}, {53, <u>27</u> ,8}, {53,41,24}, {53,38,22}}	1411:3 1622:3 <u>1423:1</u> <u>1422:1</u> 1221:1 <u>1433:2</u> 2232:5 2113:4 2112:1 2411:2 3332:1 3322:1 4432:2	4123:1 4311:2 4511:1 5531:4 5312:1 5132:1 5521:2 6331:4 7212:2 8111:4 8331:2 9111:1 9112:1		U16	{{ <u>7</u> ,2,3,1,1,0,4,0,4}, {4,9,9}, {9,2,4, <u>6</u> ,0,1}, {7,4,5,6}, {8,3,11}, {22, <u>11</u> ,4}, {22, <u>18</u> ,9}, {22,15,3}, {22,16,9}}	1622:1 <u>1423:1</u> <u>1422:1</u> 1221:1 <u>1433:1</u> <u>1433:2</u> 2113:1 2411:1	3332:1 3322:2 4123:1 5312:1 7113:3 7212:1 9111:2 9112:2
U3	{{ <u>11</u> ,6,3,9,9,4,1,4,4}, {12,17,22}, {14,6,10, <u>15</u> ,5,1}, {13,15,13,10}, {19,10,22}, {51, <u>29</u> ,16}, {51, <u>32</u> ,12}, {51,43,16}, {51,34,19}}	1411:3 1622:1 <u>1422:2</u> 1221:1 <u>1433:2</u> <u>1433:2</u> 2232:1 2113:2 2411:3 3332:2 3322:1 4432:3	4123:3 4511:3 5312:4 5132:3 5521:2 6223:3 6331:1 7212:1 8111:2 8331:2 9111:2 9112:2		U17	{{ <u>4</u> ,5,4,1,2,4,0,0,1}, {4,6,11}, {5,3,7, <u>5</u> ,1,0}, {5,6,4,6}, {9,5,7}, {21, <u>14</u> ,7}, {21, <u>15</u> ,7}, {21,15,11}, {21,9,7}}	1411:1 <u>1423:2</u> <u>1433:1</u> 2232:1 2113:2 2112:1 2411:1 3332:1	3322:3 4311:1 5531:1 5132:1 6223:2 6331:2 9112:1
U4	{{ <u>10</u> ,3,2,7,11,4,5,4,1}, {16,14,17}, {9,7,11, <u>8</u> ,10,2}, {13,11,9,14}, {16,11,20},	1411:3 1622:2 <u>1423:2</u> 1221:2 <u>1433:1</u>	5531:4 5312:2 5132:2 5521:3 6223:3		U18	{{ <u>3</u> ,3,2,1,1,1,4,0,2}, {8,4,5}, {4,3,5, <u>3</u> ,2,0}, {4,3,7,3}, {6,2,9}, {17, <u>8</u> ,4}, {17, <u>11</u> ,4}, {17,13,5}, {17,10,6}}	1411:1 <u>1423:1</u> <u>1422:1</u> 2232:2 2112:1 3332:1 3322:1	4511:1 5521:1 6331:1 7313:2 7113:1 7212:1 9111:2
					U19	{{ <u>3</u> ,2,3,3,2,1,1,0,2}, {5,3,9},	1221:2 4123:2	1622:1 5132:2

	{47, <u>28</u> ,14}, {47, <u>26</u> ,13}, {47,39,21}, {47,28,15}}	2232:1 2113:2 3332:1 3322:1 4432:2 4123:1 4311:1 4511:3	6331:1 7313:3 7113:1 7212:1 8111:2 8331:2 9112:1		{7,4,5, <u>0</u> ,0,1}, {6,2,8,1}, {4,5,8}, {17, <u>12</u> ,6}, {17, <u>12</u> ,2}, {17,9,2}, {17,12,7}}	3322:1 3332:2 6331:1 2112:1 4311:1	7212:1 9111:1 9112:1 2232:1
U5	{{ <u>12</u> ,5,2,8,8,4,3,5,2}, {15,16,18}, {11,5,12, <u>11</u> ,6,4}, {15,13,8,13}, {18,9,22}, {49, <u>32</u> ,15}, {49, <u>31</u> ,14}, {49,38,21}, {49,29,16}}	1411:2 1622:4 <u>1423:3</u> 1221:1 <u>1433:2</u> 2232:2 2113:1 2112:1 2411:1 3332:1 3322:1 4432:3 4123:3	4311:1 4511:1 5531:2 5312:3 5521:3 6223:2 6331:2 7313:1 7113:2 8111:2 8331:3 9112:2	U20	{{ <u>6</u> ,2,1,2,2,1,1,1,1}, {2,9,6}, {6,1,4, <u>6</u> ,0,0}, {7,4,4,2}, {7,6,4}, {17, <u>10</u> ,6}, {17, <u>14</u> ,7}, {17,16,4}, {17,15,9}}	<u>1422:2</u> <u>1433:1</u> <u>1433:3</u> 2113:1 2112:1 3332:1	4123:2 5312:2 6331:1 7212:1 8111:1 9111:1
U6	{{ <u>10</u> ,5,4,7,11,5,1,0,2}, {12,11,22}, {9,2,13, <u>14</u> ,7,0}, {13,10,9,13}, {15,16,14}, {45, <u>32</u> ,22}, {45, <u>29</u> ,12}, {45,40,18}, {45,31,19}}	<u>1423:2</u> <u>1422:2</u> <u>1433:2</u> <u>1433:4</u> 2232:1 2113:1 2112:2 2411:1 3332:2 3322:2 4432:3	4123:1 4311:2 4511:1 5531:4 5312:3 5132:2 5521:2 6223:1 6331:4 7113:1 9111:2	U21	{{ <u>6</u> ,1,0,1,2,1,1,2,1}, {4,6,5}, {2,2,5, <u>6</u> ,0,0}, {6,2,4,3}, {7,3,5}, {15, <u>8</u> ,5}, {15, <u>10</u> ,7}, {15,14,6}, {15,9,6}}	1411:1 <u>1423:2</u> <u>1433:1</u> <u>1433:2</u> 2232:1 4311:1	5312:2 6223:1 7313:1 8111:1 8331:1 9111:1
U7	{{ <u>9</u> ,6,5,6,4,2,6,1,3}, {12,10,20}, {9,10,13, <u>4</u> ,4,2}, {10,6,14,12}, {14,4,24}, {42, <u>22</u> ,8}, {42, <u>25</u> ,10}, {42,27,9}, {42,30,13}}	1411:1 1622:2 <u>1422:1</u> 1221:4 <u>1433:1</u> 2232:3 2113:2 2411:1 3332:1 3322:4 4311:4 4511:2	5531:1 5312:1 5132:1 5521:1 6223:2 7313:2 7113:3 7212:1 8331:1 9111:2 9112:1	U22	{{ <u>4</u> ,2,0,3,0,0,1,1,0}, {4,5,2}, {3,0,2, <u>6</u> ,0,0}, {1,8,2,0}, {4,0,7}, {11, <u>7</u> ,4}, {11, <u>9</u> ,5}, {11,11,6}, {11,6,4}}	1411:1 <u>1422:2</u> <u>1433:1</u> 2113:2 4432:2 4123:1 7313:1 8331:1	
U8	{{ <u>10</u> ,5,2,7,7,0,4,4,2}, {10,11,20},	1411:1 1622:3	4311:2 4511:2	U23	{{ <u>2</u> ,1,0,2,5,0,1,0,0}, {2,6,3}, {3,1,4, <u>2</u> ,1,0}, {1,3,3,4}, {4,1,6}, {11, <u>6</u> ,2}, {11, <u>9</u> ,3}, {11,10,3}, {11,9,6}}	<u>1422:1</u> 1221:1 2113:1 4432:1 4123:1 5312:3 5132:1 5521:1 7313:1	
				U24	{{ <u>1</u> ,0,0,1,2,1,0,2,1}, {1,3,4}, {4,2,1, <u>0</u> ,1,0},	1221:1 4123:1 5132:1	

	{10,5,9, <u>9</u> ,5,3}, {13,9,11,8}, {19,6,16}, {41, <u>23</u> ,11}, {41, <u>27</u> ,6}, {41,31,12}, {41,30,14}}	<u>1423:1</u> <u>1422:2</u> 1221:1 <u>1433:1</u> <u>1433:1</u> 2232:2 2112:2 2411:1 3322:2 4432:2 4123:1	5531:1 5312:2 5132:2 5521:2 7313:1 7113:1 7212:2 8111:2 8331:2 9112:2	{2,1,3,2}, {2,2,4}, {8, <u>6</u> ,2}, {8, <u>3</u> ,2}, {8,7,3}, {8,6,4}}	5521:1 6223:1 8111:1 8331:1 9111:1		
				U25	{{ <u>0</u> ,3,1,1,1,0,0,0,1}, {4,0,3}, {2,1,1, <u>2</u> ,1,0}, {2,2,2,1}, {1,2,4}, {7, <u>4</u> ,4}, {7, <u>5</u> ,0}, {7,4,4}, {7,4,3}}	2232:1 2112:1 2411:1 3332:1 4432:1 5531:1 9112:1	
U9	{{ <u>6</u> ,9,6,3,4,0,4,1,5}, {14,9,15}, {15,5,8, <u>5</u> ,4,1}, {11,10,10,7}, {11,7,20}, {38, <u>19</u> ,9}, {38, <u>27</u> ,7}, {38,24,13}, {38,25,16}}	1411:1 1622:1 <u>1422:2</u> 1221:2 2232:2 2113:3 2112:3 2411:1 3332:3 3322:3 4432:1	4123:1 4511:1 5531:2 5132:1 5521:1 7313:2 7113:1 7212:1 8111:1 9111:2 9112:3	U26	{{ <u>1</u> ,1,1,2,0,0,1,1,0}, {3,1,3}, {1,2,3, <u>0</u> ,1,0}, {4,0,3,0}, {1,2,4}, {7, <u>3</u> ,2}, {7, <u>3</u> ,1}, {7,4,2}, {7,5,2}}	1221:1 2232:1 3332:1 4311:1 4511:1 7313:1 8111:1	
U10	{{ <u>4</u> ,3,2,3,10,2,6,3,3}, {9,10,17}, {9,7,13, <u>1</u> ,6,0}, {7,4,13,12}, {13,6,17}, {36, <u>18</u> ,8}, {36, <u>18</u> ,7}, {36,30,13}, {36,26,15}}	<u>1423:1</u> 1221:3 2232:1 2113:1 2112:1 3322:2 4311:2 4511:1 5531:2 5312:3	5132:2 5521:3 6223:1 6331:1 7313:3 7113:1 7212:2 8111:1 8331:2 9111:3	U27	{{ <u>4</u> ,0,0,1,0,1,0,0,0}, {0,3,3}, {0,3,1, <u>2</u> ,0,0}, {3,1,1,1}, {3,0,3}, {6, <u>5</u> ,0}, {6, <u>3</u> ,2}, {6,4,2}, {6,4,0}}	<u>1423:1</u> <u>1422:1</u> 1221:2 4311:1 6223:1	
U11	{{10,1,3,2,3,4,4,3}, {10,11,13}, {9,7,6,8,2,2}, {12,9,7,6}, {10,13,11}, {34,21,12}, {34,20,7}, {34,25,12}, {34,21,9}}	1411:2 1622:2 <u>1422:1</u> 1221:2 <u>1433:3</u> 2411:1 3332:2 3322:1 4432:1 4123:1	5531:2 5132:1 6223:2 6331:2 7113:1 7212:3 8111:3 8331:1 9111:1 9112:2	U28	{{ <u>0</u> ,1,0,2,1,2,0,0,0}, {1,1,4}, {1,2,3, <u>0</u> ,0,0}, {0,1,4,1}, {1,2,3}, {6, <u>4</u> ,3}, {6, <u>3</u> ,1}, {6,5,2}, {6,4,2}}	2232:1 4311:2 5132:1 6223:1 6331:1	
U12	{{ <u>7</u> ,4,0,6,6,2,4,2,2}, {8,13,12}, {13,3, <u>9</u> ,4,3,1}, {7,7,12,7},	1411:1 1622:1 <u>1422:1</u> 1221:2	5312:2 5132:1 5521:2 6223:1	U29	{{ <u>2</u> ,0,1,1,0,0,0,0,2}, {4,0,2}, {2,0,1, <u>1</u> ,1,1}, {3,2,1,0},	1411:1 1622:1 3332:1 4511:1	

{8,5,20},	<u>1433:1</u>	6331:1	{2,1,3},	9111:1	
{33, <u>17</u> ,6},	<u>1433:1</u>	7313:3	{6, <u>2</u> ,1},	9112:1	
{33, <u>20</u> ,14},	2113:3	7113:1	{6, <u>3</u> ,0},		
{33,30,12},	2112:1	8111:1	{6,3,0},		
{33,26,17}}	4123:4	8331:1	{6,3,1}}		
	4311:2	9111:2			
	5531:1				

Tab 15. Traces et indices de contributions des membres par rapport aux difficultés de type « pannes réseau »

Annexe 4 : Variations des gains pour différents types de difficultés

	30	50	100	500	1000	5000
T1	44,44%	47,62%	34,43%	27,56%	38,57%	89,06%
T2	26,92%	12,50%	11,94%	38,78%	35,57%	54,91%
T3	25,93%	20,45%	26,98%	21,33%	52,00%	76,01%
T4	0,00%	11,32%	2,67%	27,75%	32,06%	48,98%
T5	30,77%	23,81%	23,46%	16,06%	28,41%	70,55%
T6	28,57%	31,71%	23,08%	24,23%	43,04%	65,53%
T7	29,63%	14,29%	25,00%	43,14%	42,95%	75,95%
T8	48,15%	31,82%	50,68%	65,02%	53,48%	85,12%

Tab 16. Variation du gain filtrage pour différents types de difficultés et différentes tailles de la CoP (+F-T / -F-T)

	30	50	100	500	1000	5000
T1	70,37%	73,81%	63,93%	46,46%	61,43%	95,76%
T2	26,92%	17,50%	17,91%	57,03%	38,59%	88,68%
T3	59,26%	56,82%	58,73%	48,67%	59,53%	93,03%
T4	0,00%	11,32%	4,00%	37,00%	36,51%	79,04%
T5	53,85%	54,76%	60,49%	35,04%	47,73%	92,33%
T6	42,86%	46,34%	38,46%	52,58%	60,63%	91,93%
T7	62,96%	52,38%	54,69%	66,67%	64,26%	90,98%
T8	48,15%	47,73%	54,79%	68,42%	68,67%	89,71%

Tab 17. Variation du gain global pour différents types de difficultés et différentes tailles de la CoP (+F+T / -F-T)

	30	50	100	500	1000	5000
T1	46,67%	50,00%	45,00%	26,09%	37,23%	61,22%
T2	0,00%	5,71%	6,78%	29,81%	4,69%	74,89%
T3	45,00%	45,71%	43,48%	34,75%	15,69%	70,95%
T4	0,00%	0,00%	1,37%	12,80%	6,54%	58,93%
T5	33,33%	40,62%	48,39%	22,61%	26,98%	73,98%
T6	20,00%	21,43%	20,00%	37,41%	30,88%	76,60%
T7	47,37%	44,44%	39,58%	41,38%	37,36%	62,50%
T8	0,00%	23,33%	8,33%	9,73%	32,65%	30,80%

Tab 18. Variation du gain tri pour différents types de difficultés et différentes tailles de la CoP (+F+T / -F-T)