



HAL
open science

Vers une ontologie pour les communautés de développement de logiciel libre.

Isabelle Mirbel

► **To cite this version:**

Isabelle Mirbel. Vers une ontologie pour les communautés de développement de logiciel libre.. 19es Journées Francophones d'Ingénierie des Connaissances (IC 2008), Jun 2008, Nancy, France. pp.73-85. hal-00416689

HAL Id: hal-00416689

<https://hal.science/hal-00416689>

Submitted on 14 Sep 2009

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.

Vers une ontologie pour les communautés de développement de logiciel libre

Isabelle Mirbel

Équipe EDELWEISS, INRIA Sophia Antipolis, Sophia Antipolis - France

Équipe KEWI, Pôle GLC, Laboratoire I3S, Sophia Antipolis - France

isabelle.mirbel@unice.fr

Type de communication : Recherche

Thèmes : IC et génie logiciel, Web sémantique

Résumé : Différentes approches du développement logiciel se sont succédées et cohabitent. Le développement de logiciel libre peut être vu comme un cas particulier de développement distribué ayant une structure de projet plus volatile, sans organisation clairement définie, nécessitant une participation à long terme et une vision commune des participants. La nature dynamique et libre de ce type de projet, qui repose essentiellement sur l'utilisation du Web en matière d'échange et de coordination des activités de développement, pose de nouveaux défis pour le partage de la connaissance.

Dans ce contexte, nous proposons d'exploiter les modèles et les techniques du Web sémantique pour expliciter les connaissances contenues dans les ressources d'une communauté de développement de logiciel libre. Dans cet article, nous présentons plus précisément les bases d'une ontologie dédiée à ces communautés. Lors de ce travail, notre objectif a été double. Nous avons d'une part cherché à définir le vocabulaire conceptuel mobilisé pour la représentation des connaissances dans les communautés de développement de logiciel libre vues comme des communautés de pratique. D'autre part, nous avons également cherché à réutiliser les ontologies du domaine du développement de logiciel libre proposées dans la littérature en les intégrant à notre travail.

Mots-clés : Ontologie, logiciel libre, communauté de pratique, web sémantique

1 Introduction

Différentes approches du développement logiciel, répondant notamment aux évolutions du contexte de développement, se sont succédées et cohabitent. Elles ont dû, entre autre, s'adapter à la taille en constante augmentation des équipes de développement et au fait que ces dernières deviennent parfois géographiquement distribuées. De nouvelles façons de développer des logiciels ont également émergé. C'est le cas du développement de logiciel libre ou *Free/Libre Open Source Software* (FLOSS¹) qui nous intéresse plus

¹http://fr.wikipedia.org/wiki/Free/Libre_Open_Source_Software

particulièrement dans le cadre de notre travail.

Selon (Ankolekar & al., 2006), les communautés qui développent des FLOSS sont généralement créées à partir d'un logiciel ou d'un langage développé par un individu ou une entreprise dont le code source est ensuite mis dans le domaine libre. L'activité des membres de la communauté, qui se constitue autour de ce logiciel ou de ce langage devenu libre, consiste alors à maintenir et à faire évoluer le code de départ. Linux, Mozilla, Apache, OpenOffice.org ou MySQL sont des exemples très connus de FLOSS. Les membres d'une communauté de développement de FLOSS travaillent dans des lieux géographiquement distincts, ne se rencontrent que rarement et coordonnent leurs activités en utilisant essentiellement les technologies du Web (mails, forums, listes de discussion, plate-forme de travail coopératif). Selon (Barcellini, 2005), l'organisation du développement est fonction de la taille du projet. Pour les projets de petite taille, il n'y a pas d'attribution des tâches et chacun prend en charge la tâche qu'il souhaite. Il n'y a pas non plus de planification du projet. Pour les projets de taille plus importante, les tâches sont réparties au sein des membres du projet. Selon (Ntioudis & al., 2006), le développement de FLOSS peut être vu comme un cas particulier de développement distribué ayant une structure de projet plus volatile, sans organisation clairement définie pour l'ensemble de ses membres, nécessitant une participation à long terme et une vision commune des participants. La nature dynamique et libre de ce type de projet pose de nouveaux défis pour le partage de la connaissance.

Selon (Ntioudis & al., 2006), la principale forme de connaissance échangée par les membres d'une équipe de développement est tacite et repose sur la communication directe entre collègues. De tels échanges sont possibles pour des développements réalisés par des équipes de petite taille et dont tous les membres se trouvent dans un même lieu géographique. Ils deviennent difficiles et insuffisants lorsque l'équipe atteint une taille trop importante ou lorsque le développement est distribué sur plusieurs lieux géographiques. Les solutions qui ont alors été proposées consistent à figer les interfaces et à standardiser les outils et les processus pour minimiser le besoin d'échange de connaissances tacites entre les membres du projet. Mais ces solutions ont également des limites. Toujours selon (Ntioudis & al., 2006), des études ont montré l'intérêt d'outils tels que les wikis (communication asynchrone) et les *instant messaging* (communication synchrone) dans de tels contextes.

Dans les développements de FLOSS, la coordination et les échanges entre les différents contributeurs se font uniquement à l'aide d'outils informatiques via le Web. Des connaissances non structurées (i.e. implicites) sont échangées à l'aide de ces outils.

Les modèles et techniques du Web sémantique sont un moyen de rendre explicites des connaissances présentes sur le Web. Ces approches reposent sur : (i) la définition d'ontologie fournissant le vocabulaire conceptuel mobilisé par les représentations de connaissances nécessaires aux scénarii d'usage, (ii) l'annotation des ressources contenant des connaissances à expliciter (ex : documents) ou impliquées dans les usages de ces connaissances (ex : utilisateurs, équipes, ressources matérielles, etc.) et (iii) la proposition d'outils, comme des moteurs de recherche, permettant d'exploiter les annotations à l'aide des ontologies.

Nous donnons ici deux exemples de scénario d'usage au travers desquels nous souhaitons montrer plus précisément ce qui peut être amélioré dans l'échange et le partage de

connaissances entre les membres d'une telle communauté grâce à une approche basée sur les modèles et les techniques du Web sémantique.

Un nouveau membre, qui rencontre une difficulté lors de l'utilisation du FLOSS, par exemple, cherche des solutions sur les forums de discussion de la communauté. Il va vouloir évaluer la pertinence des informations données en fonction de l'expertise réelle des auteurs des messages. L'annotation des ressources de la communauté à l'aide de l'ontologie que nous proposons permet d'explicitier et de conserver les rôles explicites et implicites des membres, les activités auxquelles ils participent (développement, discussion, etc.) et les ressources qu'ils ont contribué à créer ou à modifier. Il devient alors possible de raisonner sur ces annotations pour déduire des profils de membre qui sont fonction de leur réelle implication au sein de la communauté et d'indiquer cette connaissance aux nouveaux membres recherchant de l'information sur les forums de discussion.

Un membre de la communauté qui décide de contribuer au développement du FLOSS en participant à la résolution d'un bug commence par chercher à savoir si d'autres membres de la communauté travaillent déjà sur le bug en question ou sur des aspects connexes. Il va également chercher à savoir où ces membres en sont de leurs contributions, s'il peut s'intégrer éventuellement dans le groupe de travail constitué autour du bug à résoudre. Dans ce scénario également, l'annotation des ressources de la communauté à l'aide de l'ontologie que nous proposons permet d'explicitier les activités de développement auxquelles prennent part les membres de la communauté et ainsi d'être en mesure de faire émerger des groupes de membres travaillant ensemble à la résolution d'un bug.

Des travaux exploitant les modèles et techniques du Web sémantique dans le cadre du développement de FLOSS ont déjà été proposés dans la littérature.

A. Ankolekar a imaginé un outil, Drhuv (Ankolekar, 2005; Ankolekar & al., 2006), qui utilise les modèles et techniques du Web sémantique pour aider à la résolution de bugs dans les communautés de développement de FLOSS. Drhuv propose une interface sémantique avancée qui exploite les liens entre les ressources de la communauté en fonction des termes mis en évidence dans les messages postés durant le processus de résolution de bug. Pour chaque message concernant un bug, des recommandations en terme de personne, de code source, d'autres bugs et de discussions associées sont faites. Quatre ontologies ont été proposées dans le cadre de ce travail : une ontologie pour le code (décrivant les concepts de modules, paquetages, fonctions, variables, etc.), une ontologie pour les bugs (décrivant les concepts de comptes-rendus de bug, de *patches*, etc.), une ontologie pour les interactions (décrivant les concepts de fichiers, de messages, etc.) et une ontologie de la communauté (décrivant les concepts de communauté, ressources, personnes, etc.). Ces ontologies sont disponibles sur le Web ².

G. Simmons propose une ontologie (en cours de construction) pour les communautés de développement de FLOSS (Simmons & Dillon, 2006) afin de permettre le développement de portails sémantiques dédiés à ces communautés. L'ontologie proposée, OSDO (*Open source Development Ontology*), est orientée tâches et outils spécifiques à l'activité de développement logiciel. Elle semble également destinée à des communautés très structurées dont les acteurs assurent des rôles et effectuent des tâches

²<http://www.cs.cmu.edu/~anupriya/{bugs,code,interactions,community}>.

bien établis.

Nous verrons dans cet article comment les concepts de ces deux ontologies s'articulent avec ceux de celle que nous proposons. Dans notre approche, nous nous intéressons plus spécifiquement à l'aspect communautaire de ces projets de développement.

La suite de l'article est structurée comme suit. Dans la section 2, nous définissons ce que nous entendons par communauté de pratique et nous expliquons pourquoi les communautés de développement de FLOSS peuvent être considérées comme des communautés de pratique. Dans la section 3, nous décrivons succinctement O'COP, une ontologie générique pour les communautés de pratique développée dans le cadre du projet PALETTE, qui nous a servi de point de départ pour construire notre propre ontologie. Dans la section 4, nous présentons une partie du contenu de notre ontologie pour les communautés de développement de FLOSS. Enfin, dans la section 5 nous concluons et donnons les perspectives de ce travail.

2 Communautés de développement de logiciel libre et communautés de pratique

Selon (Barcellini, 2005), les membres d'une communauté de développement de logiciel libre constituent une communauté en ligne (au sens de (Souza & Preece, 2004)), parce qu'il s'agit de personnes qui se connectent ensemble à Internet dans le but précis de développer un FLOSS et que ce groupe de personnes est gouverné par des normes et principes propres (le fait qu'un logiciel doit être libre par exemple). Selon (Ankolekar & al., 2006), deux types de communautés professionnelles en ligne sont à distinguer : les communautés orientées discussion et les communautés orientées produit. Les premières se limitent à un espace de discussion alors que les secondes proposent un espace de discussion, un espace de documentation et un espace d'implantation. Les communautés de développement de FLOSS sont des communautés orientées produit. Toujours selon (Barcellini, 2005), les communautés de développement de FLOSS peuvent être vues comme des communautés épistémiques car l'objectif de leurs membres est de produire et de construire des connaissances sur le FLOSS. Il ne s'agit pas seulement pour eux d'augmenter leurs compétences individuelles mais aussi d'atteindre un objectif de production de connaissances. Ces communautés sont régulées sur la base d'une évaluation par les pairs qui assure l'évolution et la pérennité de la communauté. Enfin, elles peuvent également être considérées comme des communautés méritocratiques car elles sont parfois extrêmement hiérarchisées pour les projets de grande taille.

De notre point de vue, les communautés de développement de FLOSS peuvent aussi être considérées comme des communautés de pratique (CoP). Selon (Wenger *et al.*, 2002), une communauté de pratique est constituée d'un groupe d'individus partageant un intérêt, des problèmes ou une passion pour un sujet donné. Le but d'une CoP est de permettre à ses membres de coopérer et d'échanger des savoirs pour créer une valeur collective utile à chacun. La communauté leur permet de partager des ressources communes (savoirs, expériences, documents) et de collaborer dans un processus d'apprentissage collectif.

Les technologies du Web ont favorisé l'émergence de CoP virtuelles. Selon (Zarb,

2006), les deux spécificités d'une CoP virtuelle sont d'exister en dehors de toute organisation particulière et, du fait de cette indépendance et de la dispersion géographique de ses membres, de reposer sur l'utilisation du Web. Les communautés de développement de FLOSS entrent dans cette catégorie de CoP.

Tirer le meilleur parti des modèles et des technologies du Web sur lesquelles sont basées les CoP virtuelles, nécessite de proposer des services correspondant aux spécificités du domaine de connaissance de la CoP. Dans ce contexte, le but de notre travail est d'exploiter les modèles et techniques du Web sémantique pour rendre explicites des connaissances contenues dans les ressources d'une communauté de développement de FLOSS. Pour cela nous proposons une ontologie dédiée à ces communautés. Elle permettra l'annotation des ressources de la communauté et la conception d'outils dédiés comme par exemple des outils collaboratifs de résolution de problème, des outils de recherche, de dissémination, de visualisation ou d'évaluation des connaissances.

Une ontologie dédiée à la modélisation d'une CoP, O'CoP, a été proposée dans le cadre du projet européen PALETTE³. Cette ontologie a constitué le point de départ de notre travail. Dans la section qui suit nous la présentons succinctement.

3 O'CoP, une ontologie pour les communautés de pratique

Le projet PALETTE s'intéresse aux services et aux outils, basés sur une approche Web sémantique, susceptibles de faciliter et de renforcer l'apprentissage individuel et organisationnel au sein de CoP. Dans ce contexte, une ontologie dédiée à la modélisation des CoP, O'CoP, est en cours de construction (Dieng-Kuntz & al., 2007). Elle vise à rassembler les concepts nécessaires à la description d'une CoP (ses acteurs ainsi que leurs compétences, ses ressources, ses activités, etc.) pour permettre l'annotation des ressources de la CoP en fonction de cette ontologie.

Trois niveaux sont distingués dans la structure de l'ontologie O'CoP (Tifous *et al.*, 2007) : (i) L'ontologie de haut niveau inclut les concepts nécessaires à la représentation des modèles génériques (Vidou *et al.*, 2006) qui ont servi de guide pour analyser le corpus constitué dans le cadre du projet PALETTE⁴ dans le but de construire les autres niveaux de l'ontologie. (ii) Le niveau intermédiaire inclut les concepts communs à toutes les CoP. (iii) Le niveau spécifique inclut les concepts spécifiques à chaque CoP.

L'ontologie de haut niveau, présentée dans (Vidou *et al.*, 2006) et le niveau intermédiaire détaillé dans (Tifous *et al.*, 2007; Dieng-Kuntz & al., 2007) ont été le point de départ de notre travail. La figure 1 rassemble les concepts principaux du niveau intermédiaire d'O'CoP.

Dans la suite de cet article, nous montrons comment nous avons étendu les concepts principaux du niveau intermédiaire de l'ontologie générique O'CoP afin de représenter les concepts pertinents pour annoter les ressources dans une communauté de développement de FLOSS.

³<http://palette.ercim.org>

⁴Ce corpus est constitué de transcriptions d'entretiens et de synthèses réalisés avec les médiateurs des CoP hébergées dans le projet PALETTE

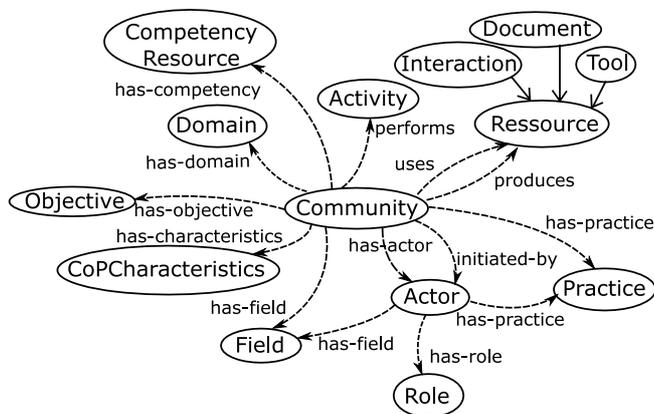


FIG. 1 – Concepts principaux de l’ontologie générique O’COP

4 Une ontologie pour le développement de FLOSS

Lors de la construction de notre ontologie pour les communautés de développement de FLOSS, notre objectif a été double. Nous avons d’une part cherché à représenter les concepts pertinents pour l’annotation des ressources d’une telle communauté. D’autre part, nous avons également cherché à réutiliser les approches de la littérature dans notre travail en explicitant les liens unissant notre ontologie avec celle proposée par A. Ankolekar (Ankolekar, 2005; Ankolekar & al., 2006) et celle en cours de construction par G. Simmons (Simmons & Dillon, 2006) comme nous le verrons plus précisément dans la suite de cette section.

De façon générale, le travail réalisé par A. Ankolekar est dédié à la résolution de bug. Les quatre ontologies qu’elle a proposées sont dédiées à l’annotation de ressources liées à la résolution de bugs. Nous nous sommes donc appuyés sur son travail pour définir le vocabulaire nécessaire à la description des ressources liées aux bugs (compte-rendus, discussions, messages, etc.) et également pour définir les concepts nécessaires à l’annotation de code source (fichier, paquetage, variable, etc.). Mais nous n’avons pas retenu son approche pour la partie description de la communauté car elle est relativement succincte et ne rassemble que des concepts dédiés à la résolution de bug (en terme d’acteurs et de rôles par exemple).

Nous nous sommes appuyés sur la proposition de G. Simmons pour enrichir notre description des rôles, des activités et des outils de la communauté car ces aspects sont particulièrement bien détaillés dans l’ontologie OSDO. Notre ontologie se démarque de celle proposée par G. Simmons par le fait qu’elle met en avant l’aspect communautaire alors qu’OSDO est vouée au développement de portails sémantiques. Cela se traduit par le fait que dans OSDO, les acteurs, les rôles implicites, les activités et les rôles liés à l’animation de la communauté sont peu ou pas pris en considération.

D’un point de vue méthodologique, nous avons suivi une approche descendante pour construire notre ontologie. Nous nous sommes basés sur les concepts principaux d’O’C-

OP et nous les avons spécialisés pour répondre aux spécificités des connaissances manipulées dans les communautés de développement de FLOSS. Pour organiser les concepts de notre ontologie, nous nous sommes appuyés sur les principes différentiels (Bachimont, 2000) tout en acceptant différents axes sémantiques pour spécialiser un concept (Kassel *et al.*, 2000). Nous nous sommes également attachés à définir les rôles en fonction des activités auxquelles leur existence est liée.

Pour mener à bien notre travail de réutilisation des ontologies existantes, nous avons séparé les concepts qui constituent le coeur de notre ontologie des concepts partie prenante dans les liaisons avec les ontologies existantes dans des fichiers différents. Nous avons proposé un fichier de liaison pour chacune des ontologies que nous avons souhaité réutiliser, de façon à être relativement indépendant de ces ontologies dont la pérennité n'est pas forcément systématiquement assurée (seule OCOP sera hébergée par un serveur de nom permanent à l'INRIA). L'intégration a été facile avec les ontologies bien documentées par ailleurs (Dhruv (Ankolekar, 2005) et O'COP (Dieng-Kuntz & al., 2007)). En revanche, la compréhension des éléments de OSDO, dont seul un fichier OWL est disponible, a été plus difficile et donc plus limitée. Afin de remédier à ce problème de documentation et permettre une bonne compréhension de l'ontologie que nous proposons, nous prévoyons d'utiliser ECCO, un éditeur contextuel et collaboratif d'ontologies (Ameur *et al.*, 2007; Tifous *et al.*, 2007), pour garder la trace des documents desquels les concepts ont été extraits.

Notre ontologie rassemble actuellement 46 classes et 8 relations qui constituent son coeur. Nous avons également défini des liaisons avec 4 ontologies existantes. 10 classes participent aux liaisons avec Dhruv, 38 à celles avec OSDO et 46 à celles avec O'COP. 2 propriétés ont été définies avec les classes de Dhruv et 4 avec celles d'O'COP. Notre ontologie est écrite en RDFS et OWL DL. Pour des raisons de place nous ne montrerons pas ici l'ensemble de l'ontologie qui réutilise également des concepts de l'ontologie proposée dans le cadre du projet SIOC⁵. Elle est disponible sur le Web⁶. Dans cet article, nous nous concentrerons sur les concepts décrivant les rôles et les activités de la communauté et donnerons quelques directions sur la mise en oeuvre de cette ontologie.

4.1 Les rôles

Dans une communauté de développement de FLOSS, nous distinguons des rôles explicites et des rôles implicites. Les rôles explicites sont ceux affichés au sein de la communauté (le rôle de développeur, d'utilisateur, etc.). Les rôles implicites sont le reflet de l'implication des acteurs dans la vie de la communauté (la participation aux discussions, etc.). Parmi les rôles implicites, nous distinguons les initiateurs de discussion et les animateurs de discussion. Les premiers initient souvent des discussions dans les forums de la communauté, les seconds participent activement aux discussions initiées sur les forums de la communauté.

Les différents rôles explicites joués par les acteurs d'une communauté de développement de FLOSS, dont certains ont été évoqués précédemment, sont présentés dans la figure 2.

⁵<http://sioc-project.org/>

⁶<http://ns.inria.fr/oflossc/>

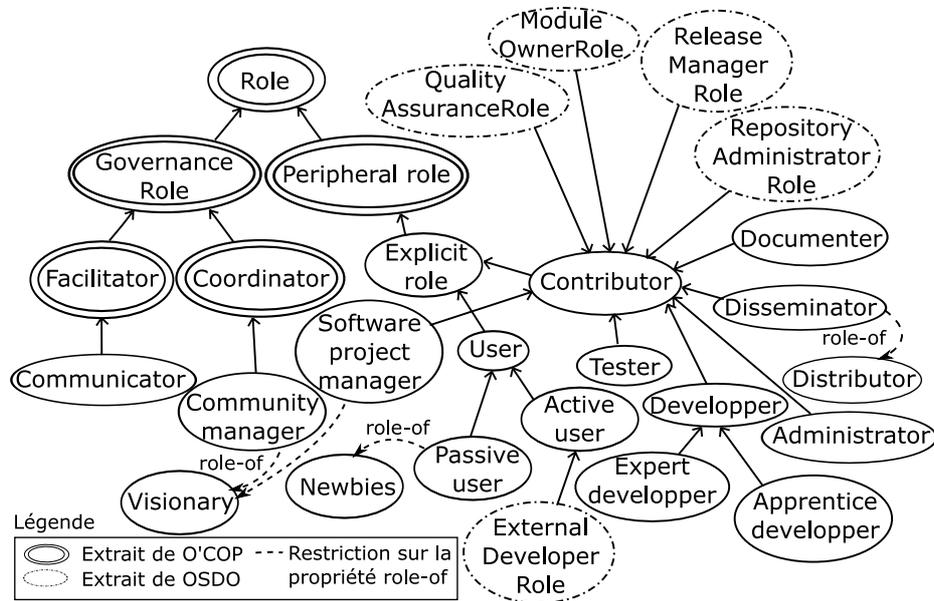


FIG. 2 – Les rôles

Dans le niveau intermédiaire d’O’COP, le concept de rôle est spécialisé afin de distinguer les rôles de gouvernance, ayant pour but d’aider les acteurs de la communauté à interagir et à partager leurs connaissances, des rôles dits périphériques joués par les acteurs qui construisent et utilisent la connaissance de la communauté. Contrairement à ce que le terme «périphérique» pourrait laisser penser, ces rôles ne sont pas secondaires. Ils rassemblent la majorité des rôles joués par les acteurs de la communauté et sont spécifiques au domaine d’activité (donc décrits à l’aide de peu de concepts dans le niveau intermédiaire d’O’COP). Parmi les rôles de gouvernance, qui ne sont pas directement liés au domaine d’activité de la CoP et donc plus facile à modéliser de façon générique (et ainsi décrits à l’aide de plus de concepts que les rôles périphériques dans le niveau intermédiaire d’O’COP) sont distingués les rôles de facilitateurs des rôles de coordinateurs. Le rôle de facilitateur consiste à oeuvrer pour faciliter les interactions entre les acteurs. Le rôle de coordinateur consiste à organiser et coordonner les activités et les événements de la CoP.

Nous avons spécialisé le concept de facilitateur en celui de communicateur (Rahtz, 2005) qui se distingue de son concept père par l’activité spécifique d’encouragement et de félicitation à laquelle il est associé comme nous le verrons dans la suite de cette section. Le responsable de la communauté (*Community manager*) coordonne les activités de la communauté. Nous l’avons donc représenté comme une spécialisation du concept de coordinateur.

Parmi les rôles périphériques spécifiques aux communautés de développement de FLOSS, nous avons distingué les utilisateurs de FLOSS de ceux qui contribuent au

développement ou à la dissémination du FLOSS. Les utilisateurs se distinguent des contributeurs par le fait qu'ils ne font qu'utiliser le FLOSS et discuter de ces bugs et de ces évolutions. Ils ne contribuent pas directement à la création ou l'amélioration ni des produits de la CoP (FLOSS, documentation) ni de son site Web. Parmi les utilisateurs nous distinguons :

- les utilisateurs actifs (Barcellini, 2005) qui participent à l'évolution du FLOSS en signalant des dysfonctionnements ou en émettant de nouvelles fonctionnalités mais qui ne contribuent pas eux-mêmes au développement et
- les utilisateurs passifs (Barcellini, 2005) qui se contentent d'utiliser le FLOSS uniquement.

Parmi les contributeurs, nous trouvons les développeurs, les administrateurs, un chef de projet (i.e. l'individu ou l'entreprise qui a créé le code de départ du FLOSS ou le successeur), les testeurs, les personnes en charge de la documentation et les distributeurs (Barcellini, 2005; Simmons & Dillon, 2006; Rahtz, 2005).

Les développeurs experts, en plus de développer le FLOSS, discutent des changements dans le code, aident les nouveaux membres et prennent les décisions à court terme (Rahtz, 2005). Les nouveaux développeurs ne font pas partie du groupe des développeurs experts. Ils s'occupent de corriger les bugs signalés dans le FLOSS. Les administrateurs ont plus de pouvoir que les autres développeurs en ce qui concerne la prise de décision et les modifications du code du FLOSS. Ils sont responsables des décisions à moyen terme. Ils participent également à la mise à jour de la documentation, surtout s'il n'y a pas de rôle dédié dans la communauté, et maintiennent le site Web du FLOSS (Barcellini, 2005). Ce sont généralement des personnes qui ont contribué à la communauté pendant une période de temps relativement longue et qui sont nommées par le chef de projet comme faisant partie du groupe des administrateurs (Rahtz, 2005). De même, c'est le chef de projet qui décide de l'intégration d'un membre de la communauté dans le groupe des développeurs experts, lequel a dû au préalable faire ses preuves (Barcellini, 2005).

Nous avons trouvé des concepts similaires aux nôtres dans l'ontologie de G. Simmons : *FounderRole* correspond à notre rôle de chef de projet, *GuardianRole* correspond au rôle d'administrateur, *InternalDeveloperRole* correspond au rôle de développeur. Les utilisateurs actifs y sont également distingués des utilisateurs passifs (*ActiveUserRole*, *PassiveUserRole*). G. Simmons suggère des rôles supplémentaires pour les contributeurs : *ReleaseManagerRole*, *RepositoryAdministratorRole*, *QualityAssuranceRole* et *ModuleOwnerRole*. Il propose également un rôle de développeur externe (*ExternalDeveloperRole*). Nous avons réutilisé ces rôles dans notre ontologie, comme cela est illustré dans la figure 2.

4.2 Les activités

Dans les modèles génériques proposés dans l'ontologie de haut niveau d'O'COP, quatre types d'activités ont été distingués : les activités de communication qui visent à transmettre de l'information, les activités d'interaction qui concernent l'échange et le partage de connaissances, la conception collaborative, la co-production, les activités de négociation et les activités d'apprentissage. Nous avons repris ce canevas pour

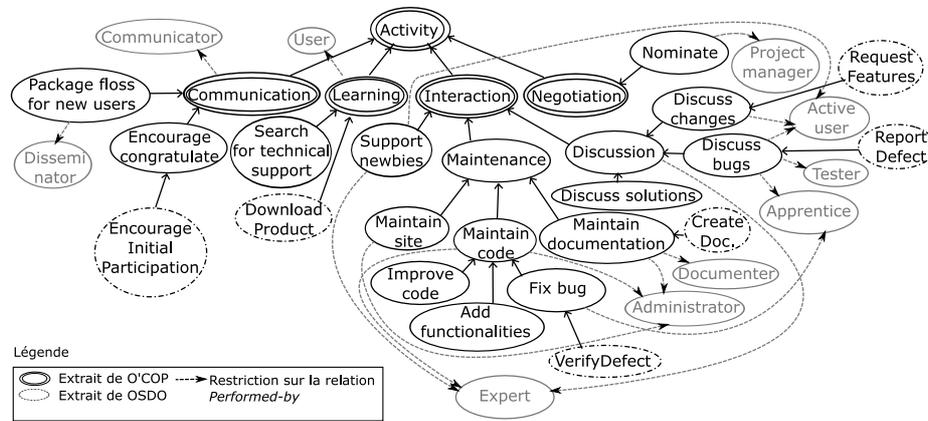


FIG. 3 – Les activités et les rôles qui leur sont liés

organiser les différents concepts correspondant aux activités d'une communauté de développement de FLOSS. Ces activités sont présentées dans la figure 3. Les rôles, qui ont été présentés précédemment sont également rappelés dans cette figure.

Parmi les activités de communication, nous distinguons l'activité qui consiste à encourager les acteurs de la communauté et l'activité qui consiste à préparer le FLOSS pour qu'il soit télé-chargé par les utilisateurs.

L'activité d'apprentissage supportée par la communauté est celle de recherche d'information technique sur le FLOSS par les utilisateurs.

En matière d'interaction, nous retrouvons les activités déjà présentées lors de la description des rôles spécifiques à la communauté, à savoir les activités d'aide aux nouveaux membres, de maintenance et de discussion. En ce qui concerne les activités de maintenance, nous distinguons la maintenance du code du FLOSS, de la documentation et du site. En ce qui concerne la maintenance du code, nous distinguons l'ajout de nouvelles fonctionnalités, l'amélioration du code existant et la correction de bug. Les discussions portent sur les bugs détectés puis corrigés, les solutions de conception du FLOSS et les améliorations et nouvelles fonctionnalités du FLOSS.

Enfin, en ce qui concerne les activités de négociation, elles se réduisent ici à la nomination d'acteurs dans des groupes particuliers (le groupe des administrateurs et le groupe des développeurs experts).

L'ontologie OSDO rassemble des activités propres aux communautés de développement de FLOSS. Comme cela est illustré dans la figure 3, nous avons réutilisé certains concepts de cette ontologie dans la notre. Des activités liées à la maintenance du code (*MaintainRepository*, *DevelopInitialProject*, *ApproveCode*, *CommitCode*, *SubmitCode*, *PrepareBuild*, *ReleaseInitialProject*, *ManageProjet*, *Test*), plus spécifiques que celles que nous avons présentées, sont également proposées par G. Simmons. Nous les avons rattachées à notre ontologie en les considérant comme des spécialisations de l'activité de maintenance du code. Ces activités n'ont pas été représentées dans la figure 3 pour des raisons de lisibilité.

4.3 Vers une mise en oeuvre de l'ontologie

L'ontologie que nous venons de présenter a pour but de permettre d'annoter les ressources de la communauté qui pourront alors être exploitées par des outils idoines. La base de données et les fichiers de log (notamment ceux des outils de gestion des versions du code et ceux sur le télé-chargement du FLOSS) utilisés pour organiser et publier les ressources du site Web d'une communauté de développement de FLOSS ont des formats bien définis. Ils serviront de point de départ aux mécanismes d'extraction permettant de générer une partie de la base d'annotation, notamment en ce qui concerne les annotations sur les acteurs, les rôles explicites et les activités de maintenance (du site, du code et de la documentation). Dans le premier scénario que nous avons présenté en introduction de cet article, il s'agit de déduire des profils de membre qui sont fonction de leur réelle implication au sein de la communauté et d'indiquer cette connaissance aux nouveaux membres recherchant de l'information sur les forums de discussion. L'exploitation des logs du système de gestion de version pourrait ainsi permettre la création d'annotation décrivant plus précisément l'activité des membres de la communauté (par exemple le type des fichiers qui ont été modifiés par un membre de la communauté : code, documentation, site Web) rendant ensuite cette connaissance plus facilement accessible aux nouveaux membres par exemple à l'aide de requêtes SPARQL⁷ prédéfinies ou libres sur ces annotations à l'aide d'un moteur de recherche comme CORESE⁸.

En ce qui concerne l'analyse des messages postés dans les forums, nous comptons nous inspirer de SEMFAQ (Makni *et al.*, 2008). Cette approche a pour but d'alimenter une FAQ à l'aide d'un mécanisme automatique d'extraction de connaissances à partir de mails. Nous souhaitons nous en inspirer pour alimenter notre base d'annotation sur les différents types de message (bugs, évolutions, maintenance) et en déduire les acteurs impliqués dans les activités de discussion. Certains rôles implicites (utilisateur actif ou passif, initiateur ou animateur de discussion par exemple) pourront être déduits de l'analyse des entêtes des messages postés sur les forums de discussion. Dans le second scénario que nous avons présenté en introduction de cet article, il s'agit par exemple d'explicitier les activités de développement auxquelles prennent part les membres de la communauté et ainsi d'être en mesure de faire émerger des groupes de membres travaillant ensemble à la résolution d'un bug. L'exploitation des entêtes des messages pourrait permettre la création d'annotation décrivant plus précisément l'activité des membres de la communauté dans les forums de discussion et ainsi connaître les membres interagissant pour la résolution d'un bug. Cette connaissance pourrait être extraite du type de message et de la participation des membres à un même fil de discussion.

À terme, l'ontologie que nous proposons devra permettre d'améliorer le support informatique aux pratiques des membres d'une communauté de développement de FLOSS, tout particulièrement pour les nouveaux membres. L'explicitation de connaissances supplémentaires sur les rôles joués par les membres de la communauté, comme cela a été brièvement illustré dans les scénarii d'usage précédents, permettrait une meilleure

⁷<http://www.w3.org/TR/rdf-sparql-query/>

⁸<http://www.sop.inria.fr/teams/edelweiss/wiki/wakka.php?wiki=Corese>

compréhension des contributions de chacun dans le but de faciliter la participation des membres de la CoP.

5 Conclusion

Le travail que nous avons présenté dans cet article se situe dans le domaine de l'ingénierie des connaissances. Son but est de faciliter l'échange et le partage de connaissances dans une communauté de développement de FLOSS. Les caractéristiques intrinsèques à ce modèle de développement, fortement distribué, ayant une structure de projet très volatile, sans organisation clairement définie et dont la communication inter membre repose essentiellement sur l'utilisation du Web en font un domaine d'application privilégié.

Dans ce contexte, nous avons choisi une approche Web sémantique basée sur la proposition d'une ontologie permettant l'annotation des ressources de la communauté afin d'en améliorer leur exploitation à l'aide d'outils dédiés. Nous avons également choisi de considérer ce type de communauté sous l'angle des communautés de pratique afin de privilégier les aspects d'apprentissage par l'échange et le partage de connaissances.

Dans ce sens, nous avons étudié les concepts pertinents pour annoter les ressources d'une communauté de développement de FLOSS en nous basant sur O'COP, une ontologie générique pour les communautés de pratique.

Dans cet article, nous avons détaillé en particulier les concepts liés joués par les acteurs de la communauté et aux activités qu'ils y réalisent. Dans ce travail, nous nous sommes attachés à réutiliser les ontologies qui ont été proposées pour les communautés de développement de FLOSS (Dhruv et OSDO) et pour les communautés sur le Web (SIOC).

Dans sa version actuelle, notre ontologie ne couvre qu'une partie du domaine des communautés de développement de FLOSS vues sous l'angle des communautés de pratique. Des travaux sont en cours pour étendre l'ontologie, notamment en ce qui concerne la modélisation des pratiques de la communauté et de ses membres. Nous comptons également travailler à l'intégration de notre travail avec l'ontologie DOAP⁹. Nous travaillerons aussi sur les mécanismes d'annotation afin de pouvoir valider notre approche par les usages.

Remerciements

Je remercie Fabien Gandon et Alain Giboin pour leurs conseils et leurs commentaires sur cette ontologie.

Références

AMEUR Y., BENNIS M., CORBY O., DIENG KUNTZ R., DURVILLE P., FANKAM C., GIBOIN A., GRILHERES B., JEAN S., LUONG P., PIERRA G. & SARDET E.

⁹<http://usefulinc.com/ns/doap>

- (2007). *Outils et Services de gestion des ontologies*. Rapport interne WP 2/Task 2, Deliverable e-WoK HUB.
- ANKOLEKAR A. (2005). *Towards a semantic web of community, content and interactions*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh.
- ANKOLEKAR A. & AL. (2006). Supporting online problem-solving communities with the semantic web. In *15th international conference on World Wide Web*, p. 575–584.
- BACHIMONT B. (2000). *Ingénierie des connaissances, évolutions récentes et nouveaux défis*, chapter Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en Ingénierie des connaissances. Eyrolles.
- BARCELLINI F. (2005). Les discussions en ligne en conception de logiciels libres : analyse des traces d'un processus asynchrone de conception à distance. Master's thesis, Conservatoire National des Arts et Métiers.
- DIENG-KUNTZ R. & AL. (2007). *CoP-dependent ontologies*. Rapport interne D.KNO.02, Palette IST Project FP6-028038.
- KASSEL G., ABEL M.-H., BARRY C., BOULITREAU P., IRASTORZA C. & PERPETTE S. (2000). Construction et exploitation d'une ontologie pour la gestion des connaissances d'une équipe de recherche. In *Journées francophones d'Ingénierie des connaissances*.
- MAKNI B., KHELIF K., CHERFI H. & R. D.-K. (2008). Utilisation du web sémantique pour la gestion d'une liste de diffusion d'une cop. In *8ième journée francophone Extraction et Gestion des Connaissances*.
- NTIOUDIS S. & AL. (2006). *Report describing state-of-the-art KM in Software Engineering*. Rapport interne D1, TEAM IST Project 35111.
- RAHTZ S. (2005). What is an open source software community? <http://www.oss-watch.ac.uk/events/2005-07-04/index.pdf>.
- SIMMONS G. & DILLON T. (2006). Towards an ontology for open source software development. In *IFIP Working Group 2.13 Foundation on Open Source Software*, p. 65–75.
- SOUZA D. C. & PREECE J. (2004). A framework for analyzing and understanding online communities. *Interacting with Computers*, **16**(3), 579–610.
- TIFOUS A., EL GHALI A., DIENG-KUNTZ R., GIBOIN A., EVANGELOU C. & VIDOU G. (2007). An ontology for supporting communities of practice. In *International Conference On Knowledge Capture, K-CAP'07*, p. 39–46.
- VIDOU G., DIENG-KUNTZ R., EL GHALI A., EVANGELOU C., GIBOIN A., TIFOUS A. & JACQUEMART S. (2006). Towards an ontology for knowledge management in communities of practice. In *6th International Conference on Practical Aspects of Knowledge Management, PAKM 2006*, p. 303–314.
- WENGER E., MCDERMOTT R. & SNYDER W. (2002). *Cultivating communities of practice*. Harvard Business School Press.
- ZARB M. (2006). Modelling participation in virtual communities of practice. Master's thesis, Information Systems Department at the London School of Economics, London, UK.