A Context Management Framework based on Wisdom of Crowds for Social Awareness applications

Adrien Joly

N° d'ordre 2010ISAL0081 Année 2010

Thèse

A Context Management Framework based on Wisdom of Crowds for Social Awareness applications

Présentée devant
L'Institut National des Sciences Appliquées de Lyon

Pour obtenir
Le grade de docteur

Formation doctorale : Informatique
École doctorale : InfoMaths

Par
Adrien Joly

Soutenance donnée le 14 octobre 2010 devant la Commission d'examen

Jury MM.

Directeur	P. MARET	Professeur (Université Saint-Etienne)
Encadrant	J. DAIGREMONT	Chef d'unité (Alcatel-Lucent Bell Labs Fr.)
Rapporteur	A. K. DEY	Professeur associé (Carnegie Mellon, USA)
Rapporteur	E. SOULIER	Maître de conférences, HDR (UTT)
Examinateur	F. LAFOREST	Maître de conférences, HDR (INSA de Lyon)
Examinateur	B. AMANN	Professeur (LIP6)
Président	H. MARTIN	Professeur (LIG)

Laboratoire de recherche : LIRIS
Contrat CIFRE avec Alcatel-Lucent Bell Labs France

N° d'ordre 2010ISAL0081 Année 2010

Thèse

Un cadre de gestion de contextes fondé sur l'intelligence collective pour améliorer l'efficacité des applications de communication sociale

Présentée devant
L'Institut National des Sciences Appliquées de Lyon

Pour obtenir
Le grade de docteur

Formation doctorale : Informatique
École doctorale : InfoMaths

Par
Adrien Joly

Soutenance donnée le 14 octobre 2010 devant la Commission d'examen

Jury MM.

Directeur	P. MARET	Professeur (Université Saint-Etienne)
Encadrant	J. DAIGREMONT	Chef d'unité (Alcatel-Lucent Bell Labs Fr.)
Rapporteur	A. K. DEY	Professeur associé (Carnegie Mellon, USA)
Rapporteur	E. SOULIER	Maître de conférences, HDR (UTT)
Examinateur	F. LAFOREST	Maître de conférences, HDR (INSA de Lyon)
Examinateur	B. AMANN	Professeur (LIP6)
Président	H. MARTIN	Professeur (LIG)

Laboratoire de recherche : LIRIS
Contrat CIFRE avec Alcatel-Lucent Bell Labs France

INSA Direction de la Recherche - Ecoles Doctorales – Quadriennal 2007-2010

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://sakura.cpe.fr/ED206** M. Jean Marc LANCELIN Insa : R. GOURDON	M. Jean Marc LANCELIN Université Claude Bernard Lyon 1 Bât CPE 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72.43 13 95 Fax : lancelin@hikari.cpe.fr
E.E.A.	**ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE** **http://www.insa-lyon.fr/eea** M. Alain NICOLAS Insa : C. PLOSSU ede2a@insa-lyon.fr Secrétariat : M. LABOUNE AM. 64.43 – Fax : 64.54	M. Alain NICOLAS Ecole Centrale de Lyon Bâtiment H9 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60 97 Fax : 04 78 43 37 17 eea@ec-lyon.fr Secrétariat : M.C. HAVGOUDOUKIAN
E2M2	**EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION** **http://biomserv.univ-lyon1.fr/E2M2** M. Jean-Pierre FLANDROIS Insa : H. CHARLES	M. Jean-Pierre FLANDROIS CNRS UMR 5558 Université Claude Bernard Lyon 1 Bât G. Mendel 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cédex Tél : 04.26 23 59 50 Fax 04 26 23 59 49 06 07 53 89 13 e2m2@biomserv.univ-lyon1.fr
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTE** Sec : Safia Boudjema M. Didier REVEL Insa : M. LAGARDE	M. Didier REVEL Hôpital Cardiologique de Lyon Bâtiment Central 28 Avenue Doyen Lépine 69500 BRON Tél : 04.72.68 49 09 Fax :04 72 35 49 16 Didier.revel@creatis.uni-lyon1.fr
INFOMATHS	**INFORMATIQUE ET MATHEMATIQUES** http://infomaths.univ-lyon1.fr M. Alain MILLE	M. Alain MILLE Université Claude Bernard Lyon 1 LIRIS - INFOMATHS Bâtiment Nautibus 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72. 44 82 94 Fax 04 72 43 13 10 infomaths@bat710.univ-lyon1.fr - alain.mille@liris.cnrs.fr
Matériaux	**MATERIAUX DE LYON** M. Jean Marc PELLETIER Secrétariat : C. BERNAVON 83.85	M. Jean Marc PELLETIER INSA de Lyon MATEIS Bâtiment Blaise Pascal 7 avenue Jean Capelle 69621 VILLEURBANNE Cédex Tél : 04.72.43 83 18 Fax 04 72 43 85 28 Jean-marc.Pelletier@insa-lyon.fr
MEGA	**MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE** M. Jean Louis GUYADER Secrétariat : M. LABOUNE PM : 71.70 –Fax : 87.12	M. Jean Louis GUYADER INSA de Lyon Laboratoire de Vibrations et Acoustique Bâtiment Antoine de Saint Exupéry 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél :04.72.18.71.70 Fax : 04 72 43 72 37 mega@lva.insa-lyon.fr
ScSo	**ScSo*** M. OBADIA Lionel Insa : J.Y. TOUSSAINT	M. OBADIA Lionel Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Tél : 04.78.77.23.88 Fax : 04.37.28.04.48 Lionel.Obadia@univ-lyon2.fr

*ScSo : Histoire, Geographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Un cadre de gestion de contextes fondé sur l'intelligence collective pour améliorer l'efficacité des applications de communication sociale

Résumé

A l'heure où les sites de réseaux sociaux transforment les usages sur le Web, les échanges entre personnes deviennent de plus en plus faciles, ludiques et riches. Le partage en temps réel de nouvelles, d'humeurs, et autres contenus (personnels ou personnellement sélectionnés) permet de tisser, de maintenir et de renforcer des liens sociaux entre personnes à des échelles encore inédites. Cependant, la quantité sans cesse croissante d'information circulant sur ces réseaux, souvent en temps réel, motive une régulation des signaux (ici appelées "interactions médiatisées"), de manière à réduire le temps nécessaire pour suivre ses réseaux sociaux, et modérer les interruptions induites, non favorables à une bonne productivité sur le traitement de tâches demandant une attention continue.

Dans le cadre de cette thèse, nous avons développé un système de filtrage et de recommandation de ces signaux qui repose sur la similarité contextuelle entre utilisateurs, producteurs et consommateurs de ces signaux, pour évaluer leur pertinence. Notre approche consiste à agréger et interpréter les données de contexte sur les terminaux des utilisateurs, sous forme de mots-clés pondérés (tags), avant qu'elles ne puissent être exploitées par le serveur de recommandation, à la demande de l'utilisateur. Dans ce mémoire, nous présenterons un état de l'art couvrant la gestion de données contextuelles, les réseaux sociaux et leurs pratiques actuelles sur internet, et des techniques de recherche d'information. Ensuite, nous proposerons une formalisation de notre problématique de filtrage contextuel, l'implémentation d'une application de réseautage social d'entreprise, et nous discuterons les résultats expérimentaux obtenus auprès d'utilisateurs.

Mots-Clés: recherche d'informations, modélisation de données de contexte, systèmes de recommandation, réseaux sociaux, surcharge informationelle

A Context Management Framework based on Wisdom of Crowds for Social Awareness applications

Abstract

At a time when social networking sites revolutionize the usages on the Web, it has become rich, easy, and fun to share private or professional content. Sharing personal information in real-time (such as news, moods, etc...), supports the maintenance of social ties at a high scale. However, the information overload which emerged from the growing quantity of signals exchanged on these services, often in real-time, motivates a regulation of these signals (called "mediated interactions"), in order to reduce the temporal cost for maintaining social networks, and implied interruptions, which have a negative impact on productivity on tasks that require long-lasting attention.

In the frame of this thesis, we have developed a filtering and recommendation system that relies on contextual similarity between users that produce and consume social signals, as relevance criteria. In our approach, contextual information is aggregated and interpreted on users' terminal(s), before being submitted on-demand to a server in the form of a set of weighted tags. In this thesis, we present a broad state of the art on context-awareness, social networks and information retrieval, we propose a formalization of our filtering problem, and we implement and evaluate its application for enterprise social networking.

Keywords: information retrieval, context-awareness, recommender systems, social networks, information overload

Acknowledgements

When I decided to start a PhD thesis, I was very motivated for working on my own research project during three years, but I also knew it was not going to be easy. Indeed, it has been a very enriching experience, but I could not have achieved it without support from my supervisors, colleagues, friends and family.

First, I would like to thank my research supervisor, Pierre Maret, for having trusted my insights and research directions, giving wise advice and confidence when I needed it, and providing several opportunities to present my work. As my co-supervisor (and former-) at Alcatel-Lucent Bell Labs France, I am thankful to Johann Daigremont and Fabien Bataille, for giving me the resources, responsibilities and autonomy I needed to carry out my thesis in the best conditions. It was a pleasure to supervise Johann Stan and Chafik Bachatene, who collaborated to my research during their internship at Alcatel-Lucent Bell Labs France.

Throughout my PhD thesis, I have had the chance to meet great people and new friends, including fellow PhD students Julien Subercaze, Romain Vuillemot, François Guern, and Dominique Decotter, who provided valuable references from various disciplines (including sociology and cognitive psychology), believed in my research, and always shared honest feedback about it. I'm thankful to my colleagues from Alcatel-Lucent worldwide, especially Dimitre Kostadinov, Jerôme Picault, Florian Rodary, Peter Schott, Christophe Scicluna and Makram Bouzid for their participation to the evaluation of my prototype, and for their help at various phases of my thesis. I would also like to express my sincere gratitude to the committee members of my PhD defense, and especially to Anind K. Dey and Eddie Soulier for the positive and enriching feedback they provided through their review of my thesis.

This thesis, and most of my scientific publications, were written in English. This would have not been a reasonable objective without some serious proofreading, which was kindly proposed by several friends, including Jenna Campbell, Thibaud Hulin, Alain Leufroy, Hayley Nilsen, Laure Pavlovic, Greg Payne and Peter Schott.

Finally, I would like to dedicate this work to my family. They have always supported me to pursue my vocation — to become a successful computer scientist — and my passion for performing music and travels. Thank you.

This work has been partially funded by the French Ministry of Higher Education and Research, in the frame of a CIFRE (*Conventions Industrielles de Formation par la REcherche*) convention between Alcatel-Lucent and INSA-Lyon. It was partly conducted as part of the ITEA2 EASY Interactions project, partially funded by the French Minister of the Economy, Industry and Employment.

Un cadre de gestion de contextes fondé sur l'intelligence collective pour améliorer l'efficacité des applications de communication sociale

Adrien Joly *†, Pierre Maret ‡ and Johann Daigremont †

* Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
† Alcatel-Lucent Bell Labs France, Site de Villarceaux, F-91620 Nozay, France
‡ Université de Lyon, Laboratoire Hubert Curien, UMR CNRS 5516, F-42000 Saint-Etienne, France

***Résumé*—A l'heure où les sites de réseaux sociaux transforment les usages sur le Web, les échanges entre personnes deviennent de plus en plus faciles, ludiques et riches. Le partage en temps réel de nouvelles, d'humeurs, et autres contenus (personnels ou personnellement sélectionnés) permet de tisser, de maintenir et de renforcer des liens sociaux entre personnes à des échelles encore inédites. Cependant, la quantité sans cesse croissante d'information circulant sur ces réseaux, souvent en temps réel, motive une régulation des signaux (ici appelées "interactions médiatisées"), de manière à réduire le temps nécessaire pour suivre ses réseaux sociaux, et modérer les interruptions induites, non favorables à une bonne productivité sur le traitement de tâches demandant une attention continue.**

Dans le cadre de cette thèse, nous avons développé un système de filtrage et de recommandation de ces signaux qui repose sur la similarité contextuelle entre utilisateurs, producteurs et consommateurs de ces signaux, pour évaluer leur pertinence. Notre approche consiste à agréger et interpréter les données de contexte sur les terminaux des utilisateurs, sous forme de mots-clés pondérés (tags), avant qu'elles ne puissent être exploitées par le serveur de recommandation, à la demande de l'utilisateur. Dans cette synthèse, nous présenterons un état de l'art couvrant la gestion de données contextuelles, les réseaux sociaux et leurs pratiques actuelles sur internet, et des techniques de recherche d'information. Ensuite, nous proposerons une formalisation de notre problématique de filtrage contextuel, l'implémentation d'une application de réseautage social d'entreprise, et nous discuterons les résultats expérimentaux obtenus auprès d'utilisateurs.

I. Introduction

A. Contexte

Cette thèse se situe à la rencontre de trois thématiques de recherche : la gestion de données de contexte utilisateur pour adapter les applications logicielles, l'évaluation de pertinence pour la recommandation et le filtrage d'informations, et l'étude des réseaux sociaux sur Internet.

La problématique de gestion de données de contexte (***context awareness***) a été proposée par Schilit et col. [1] en 1994. Elle s'inscrit dans le cadre de systèmes informatiques répartis, dits "ubiquitaires" (*ubiquitous computing* [2]), tels que définis par Weiser en 1991, permettant alors aux logiciels de s'adapter au contexte de leurs utilisateurs (ex. leur localisation). Alors que le concept d'*intelligence ambiante* [3] consiste à tirer parti de ces données de contexte pour permettre aux logiciels d'être pro-actifs envers les buts et habitudes de leur utilisateurs, le concept de *conscientisation ambiante* (*ambient awareness* [4]) préfère exploiter ces connaissances pour faciliter la communication et la collaboration entre personnes, en les rendant respectivement conscientes de leurs activités et contextes.

Avec l'avènement des téléphones intelligents et des réseaux sans-fils, cette vision se réalise : l'ordinateur classique, équipé d'un écran, d'un clavier et d'une souris, est progressivement remplacé par toutes sortes d'appareils mobiles autonomes et inter-connectés, procurant alors de nouvelles interfaces pouvant être utilisées en mobilité. Plus récemment encore, des milliers d'applications logicielles sensibles à la localisation de leur utilisateur sont utilisées en masse, rendant accessibles au grand public des premières implémentations d'intelligence et de conscientisation ambiante.

Parallèlement, de nouveaux usages se sont développés sur le Web, parmi lesquels le *réseautage social* – la communication et l'échange de données sur des ***réseaux sociaux*** – et l'annotation de ressources. Le premier permet aux personnes de maintenir et d'enrichir leur communication avec leurs réseaux professionnels, amicaux, voire familiaux, et d'étendre ces réseaux en partageant des contenus égo-centrés. Le second usage croissant, l'annotation de ressources, permet l'émergence d'une ***intelligence collective***. En effet, en apportant de modestes contributions, comme l'annotation de pages web grâce à des mots-clés couramment appelés *tags*, des millions d'utilisateurs font naitre une base de connaissances collaborative, permettant alors de mieux catégoriser, indexer, chercher et recommander l'information disponible en masse sur Internet.

Grâce aux appareils capables de se repérer dans le monde réel et aux informations fournies par des millions d'internautes, le Web peut désormais être vu comme une sur-couche digitale/informationnelle du monde réel. C'est d'ailleurs ainsi que sont nées les applications de *réalité augmentée*, permettant aux utilisateurs d'obtenir des informations d'Internet à propos de leur localisation actuelle, sans avoir à formuler la moindre requête. Dans cet écosystème de *surcharge informationnelle*

(***information overload***) où l'information croit plus vite qu'elle ne peut être consommée, et où les utilisateurs sont prêts à être interrompus à tout moment pour recevoir de nouvelles informations émises par leurs réseaux sociaux, le nouveau challenge est d'attirer l'attention des personnes sur les informations qu'elles trouveront les plus pertinentes.

B. Problématique

L'objectif général de cette thèse est d'exploiter des connaissances sur le contexte d'utilisateurs de réseaux sociaux afin de réduire la quantité d'informations émises par ces réseaux, en sélectionnant les plus pertinentes.

En particulier, nous étudions l'hypothèse selon laquelle *une information est pertinente pour un utilisateur, si le contexte dans lequel son auteur l'a partagé est similaire au sien*. Inspirés par la définition de Dey [5], nous qualifions de *contexte*, toute information décrivant l'activité courante d'un utilisateur, et l'environnement dans lequel il l'exerce : le lieu, les personnes, objets et activités environnantes. Vérifier cette hypothèse rendrait possible la création d'outils de sélection automatique d'informations pertinentes, et permettrait donc une utilisation plus efficace de réseaux sociaux, en réduisant le temps nécessaire pour le maintenir et la fréquence des notifications reçues en temps-réel. Dans un cadre professionnel, une telle réduction de charge pourrait révéler le potentiel des réseaux sociaux d'entreprise, en permettant une augmentation de la productivité et des opportunités de collaboration et d'échange entre professionnels, qu'ils travaillent ou non dans la même entreprise.

Afin de valider cette hypothèse, nous proposons :

- d'identifier des capteurs permettant de recueillir diverses données de contexte sur les utilisateurs,
- de formaliser un modèle de représentation de données de contexte, et un modèle de pertinence basé sur leur similarité,
- et d'évaluer l'application d'un tel système dans un environnement réaliste, auprès d'utilisateurs réels.

C. Plan

Dans une première partie d'état de l'art, nous allons commencer en section II par étudier les fonctionnalités, usages et impacts cognitifs des réseaux sociaux et systèmes de micro-blogging, suivie par une section sur les techniques de gestion d'informations dans les systèmes collaboratifs, puis par une section sur l'extraction et la modélisation de données de contexte. En section V, nous élaborerons notre cadre de gestion de contextes, son exploitation dans une application sociale d'entreprise en section VI, puis l'évaluation de notre système en section VII. Enfin, nous conclurons cette étude (section VIII), avant de proposer des pistes de recherche (section IX).

II. Réseaux sociaux : fonctionnalités, usages, et impact cognitif

Après les forums, les messageries instantanées, les blogs, entre autres moyens de communication et d'expression sur Internet, les réseaux sociaux permettent le maintien de communautés larges, réconciliant la gestion de la représentation publique de soi, d'espaces de discussions, et l'échange de messages privés. Ils permettent ainsi d'enrichir les liens entre professionnels, entre amis, voire en famille, et de faciliter la créations de nouveaux liens. Dans cette partie, nous allons commencer par décrire les fonctionnalités et usages de différents réseaux sociaux parmi les plus populaires sur Internet, puis nous allons étudier les impacts cognitifs auxquels sont potentiellement exposés les utilisateurs de médias sociaux où l'information est diffusée en temps réel.

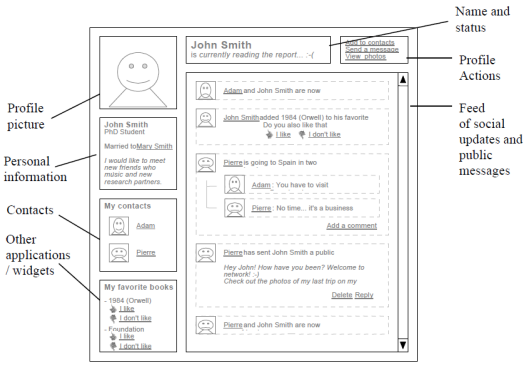

FIGURE 1. **Un profil utilisateur sur un site de réseau social**

A. Fonctionnalités de base

Sur tout site de réseau social, chaque membre est encouragé à renseigner et maintenir une page personnelle appelée ***profil***, tel qu'illustré en Figure 1. Identifié par son nom (ou un pseudonyme) ainsi qu'une photo (ou un avatar), chaque membre se présente à son réseau en partageant quelques éléments : une biographie concise, quelques contenus qu'il apprécie et permettent de mieux appréhender sa personnalité, ainsi que la liste de ses contacts. Les contacts d'un membre, élément central de tout réseau social, sont d'autres membres qui ont une relation privilégiée avec ce membre, en termes de visibilité et/ou de fonctionnalités interactionnelles. Selon la nature du réseau social, un contact peut être un ami, un collègue, un fan, un client, ou tout autre personne intéressée par l'actualité de l'autre.

1) Facebook: Sur le réseau social le plus populaire et actif sur Internet, un contact est généralement un proche, ou une personne déjà rencontrée avec qui l'on souhaite pouvoir se retrouver ou maintenir une correspondance informelle [6]. La connexion entre deux contacts est alors explicite, mutuelle et diffusée : l'un propose à l'autre, qui peut alors accepter ou refuser la relation d'amitié sur le réseau, et les autres membres de leurs réseaux seront informés de l'éventuelle connexion. Une fois connectés, deux contacts peuvent avoir accès à certaines informations et fonctionnalités potentiellement limitées aux membres du réseau de ceux-ci.

2) Twitter: Sur ce site de micro-blogging (une variante de réseau social), les contacts sont non mutuels et chaque membre a donc deux listes de contacts : les personnes suivies

(following), et les suiveurs (followers). Sur ce site, la plupart des profils sont publics, mais seules les personnes suivies par un membre peuvent lui envoyer des messages privés.

B. Fonctionnalités interactionnelles : messages et flux sociaux

Comme nous l'avons expliqué plus haut, l'appartenance d'un membre au réseau d'un autre membre donne accès à des fonctionnalités interactionnelles supplémentaires. Ces fonctionnalités, propres aux sites de réseaux sociaux, et spécifiques à chacun d'eux, consistent généralement à être en mesure de diffuser un message dans un réseau, et d'accéder aux messages des autres. Ces messages, que nous pourrions qualifier d'interactions médiatisées, et que nous appelons ***social updates*** dans la thèse, peuvent être de diverses natures. Cependant, ils partagent généralement des modalités communes de diffusion, et des fonctionnalités communes d'interaction. Concernant la diffusion, un message est généralement diffusé par un membre à l'attention de son propre réseau de contacts, mais il peut parfois être destiné à un contact particulier ainsi qu'au réseau de ce contact.

1) Diffusion à son propre réseau: Cette première modalité permet à chaque membre de partager des actualités, son humeur, ses activités ou autres contenus personnels (ou personnellement sélectionnés). Par ce biais, il donne alors l'opportunités à ses contacts de réagir en commentant ce message, de s'engager dans l'activité sociale qu'il représente éventuellement, de relayer ce message à son propre réseau de contacts, ou de consommer passivement cette information sans réagir publiquement, un peu comme en parcourant un journal quotidien d'actualités.

2) Diffusion ciblée: Cette deuxième modalité, quant à elle, est moins égo-centrée : elle consiste à partager un message avec une personne particulière, et donc d'attirer son attention de manière plus personnelle. Afin de vulgariser cette distinction avec la première modalité, Facebook emploie la métaphore du "mur". Le profil de chaque membre est constitué d'un mur, sur lequel ses contacts peuvent laisser des messages, lesquels seront visibles aux autres contacts de ce membre, ainsi qu'aux contacts de celui qui laisse le message. Sur les réseaux professionnels, comme LinkedIn, laisser un message sur le profil d'un contact consiste en fait à le valoriser en vantant ses qualités, à des fins de recommandation auprès d'employeurs/collaborateurs potentiels. Sur Twitter, cette modalité est rendue possible par la mention explicite du/des membre(s) concernés dans le corps même du message, en préfixant son/leur nom d'utilisateur par une arobase, tel qu'illustré en Figure 2.

Dans les deux cas, la visibilité des messages au réseau de leur auteur et de leur destinataire éventuel permet d'initier des discussions entre les membres de ces réseaux, et de les diffuser de manière virale via les contacts de second degré. Sur Twitter, ces usages ont émergé par l'entente collective des utilisateurs sur des syntaxes particulières : commenter le message d'un membre revient à diffuser un nouveau message mentionnant le nom d'utilisateur de ce membre, préfixé du diminutif "RE" (de "réponse"). Avant d'être adopté en tant que fonctionnalité

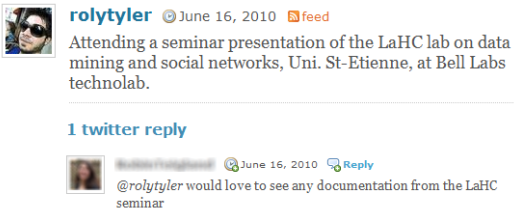

FIGURE 2. **Un message social et une réponse mentionnant l'auteur, sur Twitter**

nativement supportée par le site, le relai d'un message suit le même principe, sauf que le diminutif "RT" (de "re-tweet") est utilisé comme préfixe.

Afin de suivre les nouvelles de son réseau social sans avoir à consulter le profil de tous ses contacts, les derniers messages (et autres interactions) qu'ils ont émis sont énumérés dans un flux appelé ***newsfeed***. Chaque membre peut alors se tenir au courant de l'actualité de ses contacts, et des discussion qui s'y rattachent, en parcourant cette liste. Contrairement à l'échange de messages en mode "push" où les messages des contacts s'accumulent dans une boite de réception en attendant d'être consommés (ex. le courrier électronique), la consultation des messages sociaux se fait en mode "pull", et il n'y a aucune attente de consommation de chaque message de la part de leurs auteurs [7]. Alors que seules les personnes volontairement attentives à ce flux prendront connaissance des messages les plus récents, il reste possible de consulter les messages plus anciens en remontant dans la chronologie du flux, ou de ceux de leurs auteurs (leur profil).

C. Nature et contenu des messages sociaux

Comme nous l'avons introduit plus haut, les messages diffusés par les membres à leurs réseaux permettent de partager des actualités, des humeurs, des activités ou autres contenus personnels ou personnellement sélectionnés. Nous allons commencer par proposer catégorisation non exhaustive de la nature de ces messages :

1) Mises à jour de profil: Nombreux réseaux sociaux diffusent de manière automatique les changements qu'un membre a effectué sur son profil. De cette manière, lorsqu'un membre met à jour son statut marital (par exemple) sur son profil Facebook, ce changement de statut sera annoncé automatiquement à son réseau, comme si le membre avait manuellement rédigé un message en cette occasion, permettant alors à ses contacts de le féliciter ou de compatir, selon le cas. Cette fonctionnalité prend tout son sens dans un réseau social professionnel comme LinkedIn, où tout changement de situation professionnelle d'un membre (ex. changement d'employeur, de poste ou de projet) sera notifié automatiquement aux membres de son réseau. Grâce à ces notifications, les contacts peuvent éventuellement proposer de nouvelles opportunités pertinentes de collaboration.

2) Nouvelles: Sur les sites de micro-blogging et la plupart des réseaux sociaux, les membres sont invités à partager régulièrement des nouvelles, et autres courts billets d'humeur.

Pionnier dans cette nouvelle pratique de communication, Twitter proposait à ses débuts de répondre publiquement à la question "Que faites-vous ?", en 140 caractères seulement. Aujourd'hui, suite à l'appropriation de l'outil par ses usagers, la question est devenue "Que se passe-t-il ?", alors que Facebook demande "Qu'avez-vous à l'esprit ?" à ses utilisateurs. Par ce biais, les membres maintenant un profil personnel (ex. sur Facebook) partagent leur humeur du moment, racontent des blagues, partagent leurs bonnes (ou mauvaises) nouvelles, annoncent leurs activités présentes et à venir, ou encore rapportent un évènement ou situation dont ils sont témoins [8], [9]. Sur leur profil professionnel, les membres préfèreront diffuser des actualités en rapport avec leurs activités, et faire connaître leurs réalisations. Dans les deux cas, l'intention est double : se faire valoir, et susciter une réaction des autres contacts du réseaux.

3) Partage de contenus: Proche de la diffusion de nouvelles, en termes d'intention, certains réseaux sociaux permettent à leurs membres de partager des contenus plus riches que des messages textuels : vidéos, liens vers des pages web etc... Parfois dénués de commentaires, ces contenus peuvent être partagés à des fins d'information, mais aussi à des fins d'expression de ses gouts personnels. Dans certains cas, l'utilisateur peut être l'auteur du contenu, ex. une photo, une vidéo qu'il a prise lui-même.

4) Questions: Enfin, les réseaux sociaux sont un excellent médium pour poser des questions, demander des conseils et suggestions de manière plus ou moins publique. En effet, outre l'intérêt pour les personnes de s'entraider au sein d'un réseau, le fait de connaitre la personne proposant une réponse permet de mieux juger de sa pertinence, et de profiter d'un avis plus subjectif et personnalisé [10].

D. Autres interactions médiatisées et usages

Outre l'échange de messages et contenus saisis et diffusés explicitement par les utilisateurs de réseaux sociaux, des fonctionnalités interactionnelles plus spécifiques sont couramment proposées, voici quelques exemples :

1) L'annotation sociale de contenus: Le partage de photos a longtemps été une fonctionnalité phare des réseaux sociaux personnels. En partageant un album de photos sur son profil, un membre permet aux contacts de son réseau de consulter et commenter ces photos. Il est désormais possible (voire encouragé) d'associer les visages apparaissant sur ces photos avec le profil social de la personne correspondante, la photo concernée est alors automatiquement diffusée au réseau de cette personne. Pratique pour s'échanger des souvenirs entre amis, cette fonctionnalité suscite encore aujourd'hui de nombreuses controverses, pour des raisons évidente d'atteinte à la vie privée. Cette fonctionnalité n'en reste pas moins puissante quand elle est utilisée de manière respectueuse et responsable. Sur Twitter, il est possible de mentionner un autre utilisateur, mais ce contenu ne sera pas directement visible sur le profil de ce dernier.

2) Groupes: Certains réseaux sociaux permettent la création de sous-réseaux (ou communautés) rassemblant des personnes qui n'étaient pas forcément connectées entre elles. A l'image des profils personnels, ces groupes sont des espaces d'échange et de discussion autour d'une thématique plus ou moins spécifique. En rejoignant un groupe, un membre se rend visible aux autres membres, sans avoir besoin de les inclure à son réseau de contacts personnels. Il peut alors partager des contenus et interagir avec eux de manière cloisonnée, sachant qu'un groupe peut être caché et nécessiter un contrôle d'entrée (ou une invitation) par un modérateur.

3) Évènements: Présentant des fonctionnalités similaires à celles des groupes, les évènements sont des espaces d'annonce et d'interactions sociales autour d'un événement qui va avoir lieu prochainement, ou a eu lieu dernièrement. Outre la spécification de l'objet, de la date, du lieu, et des invités, la page d'un évènement est propice à sa préparation collaborative puis au partage de contenus autour de celui-ci (ex. diffusion de photos et vidéos prises à cette occasion). Sur Twitter, pour être associé à un évènement (ou un groupe), un message doit contenir un mot-clé (appelé ***hashtag***) représentant cet évènement de manière unique. Ainsi, suivre et participer aux discussions liées à un évènement ne nécessite pas d'intégrer explicitement un groupe de personnes, il suffit de rechercher les messages contenant le mot-clé correspondant, et de contribuer en incluant ce mot-clé dans ses propres messages.

4) Applications sociales: Initiée par Facebook en 2007, la transformation de sites de réseaux sociaux en plateformes a permis le développement d'applications tierces pouvant tirer parti du graphe social des utilisateurs. De nouvelles fonctionnalités interactionnelles sont alors apparues, telles que des jeux de société, des quizz, et des applications de recommandation sociale (ex. films, livres) basées sur les gouts de ses propres contacts. Tierces pour la plupart, ces applications sont le plus souvent financées par la diffusion de publicités et tentent donc d'inciter leurs utilisateurs à y faire participer un maximum de contacts. Pour cela, ces applications peuvent diffuser des messages au nom de leur utilisateur, de manière à inciter ses contacts à l'y rejoindre.

E. Réseaux sociaux mobiles

Avec la croissance des appareils mobiles supportant la technologie sans fil à courte portée *Bluetooth*, de nouvelles pratiques de communication de proximité sont nées. La tendance consistait alors à envoyer des messages OBEX à d'autres personnes alentours, à conditions que leur connexion Bluetooth soit activée et visible. Dans le but de faire des rencontres, certains personnalisaient l'identifiant de leur appareil tel qu'il était vu par les autres possesseurs d'appareils Bluetooth, afin d'y placer des informations personnelles comme leur sexe, leur age, etc... Se sont alors développé des applications de messagerie de proximité tirant parti de ces techniques (Nokia Sensor, Serendipity, Mobiluck), avant que des applications déclinées des sites de réseaux sociaux présentés ici ne puissent tirer parti des téléphones portables ayant un accès à Internet. De par sa simplicité, et la possibilité de diffuser des messages par SMS, Twitter devient l'un des réseaux sociaux mobiles les plus rejoints. Se développe alors un nouvel usage : la

diffusion de messages sociaux en mobilité, transformant le témoin d'évènements en journaliste, grâce à l'instantanéité de l'information produite. Aujourd'hui, la plupart des sites de réseaux sociaux proposent une ou plusieurs applications pour téléphones mobiles, et tirent parti des possibilités de géolocalisation. En particulier, sont apparus de nouveaux réseaux sociaux mobiles invitant leur utilisateurs à diffuser leur position actuelle à leur contacts, leur permettant ainsi de nouvelles opportunités de rencontre, de recommandations sociales de lieux, voire d'accès à certaines promotions commerciales : Dodgeball, Foursquare, Gowalla, Plyce, etc...

F. Réseaux sociaux et vie privée

Les réseaux sociaux ont provoqué de nombreux bouleversements en peu de temps, compromettent parfois des normes sociales comme le maintien de la vie privée. En guise d'exemple, suite à l'apparition du newsfeed sur Facebook, des milliers d'utilisateurs trouvant cette fonctionnalité trop intrusive ont menacé de se désinscrire du site, mais la popularité du site a en fait explosé au delà des espérances de son fondateur. Barkhuus et Dey ont observé que les utilisateurs sont prêts à fournir des données personnelles potentiellement sensibles, à condition que la satisfaction d'usage du système/service qui les exploite soit supérieure au coût que représenterait un contrôle manuel de ces données [11], ce qui s'applique parfaitement aux réseaux sociaux vu les statistiques d'usage sans cesse croissantes qu'ils affichent.

G. Micro-blogging temps-réel : usage et impacts

A mi-chemin entre blogs et réseaux sociaux, les systèmes de micro-blogging permettent la diffusion instantanée de messages courts à des millions de lecteurs. Dans cette partie, nous présentons les résultats d'un sondage que nous avons mené auprès de 256 utilisateurs actifs de solutions de micro-blogging, afin d'observer leurs usages, et d'étudier l'impact cognitif potentiellement induit.

1) Réception de nombreux messages: Nous avons observé qu'un tiers des participants suit de 100 à 250 flux, et que près d'un deuxième tiers suit plus de 250 flux, ce qui dépasse le nombre de personnes avec qui l'on est humainement capable de maintenir une relation sociale stable selon Dunbar [12]. 75% des participants pratiquent le micro-blogging à des fins professionnelles, 69% pour suivre l'actualité, et 64% pour profiter des opportunités que peuvent leur apporter ce réseau de contacts.

2) Intérêts à long terme: Nous avons mesuré un sentiment d'utilité des messages reçus pour les intérêt à long terme plus élevé que pour les intérêt immédiats (ex. messages utiles pour l'activité courante). Nous déduisons de cette observation que la réception de messages en temps-réel est rarement utile, comparée au cout cognitif qu'induit la lecture immédiate de messages n'ayant pas de rapport avec la tâche exécutée par l'utilisateur.

3) Un potentiel de distraction élevé: Plus d'un tiers des participants sont en écoute permanente de nouveaux messages, en gardant leur flux visibles à l'écran, voire en étant notifié instantanément par des pop-ups. De plus, 62% des participants avouent lire la majorité des messages qu'ils reçoivent. La distraction occasionnée par ces notifications peut réduire fortement leur performance sur leur tache principale [13], d'autant plus que rares sont les messages ayant un rapport direct avec cette tache. Il est important que les utilisateurs puissent contrôler leur degré d'interruptibilité en fonction du contexte [14].

4) Un besoin de filtrage: Alors que 78% des participants n'utilise aucun mécanisme de filtrage des messages qu'ils reçoivent, plus de la moitié d'entre eux souhaiteraient bénéficier d'un tel mécanisme. Outre les besoins reconnus de filtrage par type de message (entre actualités, messages professionnels, privés, etc...), 36% des participants aimeraient que les messages qu'ils reçoivent soient filtrés en fonction de leur activité courante, ce qui n'est pas directement faisable par les systèmes de micro-blogging tels qu'ils sont actuellement implémentés. Relativement faible, cette dernière proportion peut-être expliquée par la peur de rater un message intéressant, ou par manque de confiance envers l'intelligence du mécanisme de filtrage employé, tel que l'ont observé Iqbal & Horvitz. Cependant un tel mécanisme sachant exploiter des données sur leur contexte d'activité pourrait aider à réduire la distraction causée par des notifications trop fréquentes de messages non pertinents avec la tache courante [15].

H. Pistes d'amélioration et conclusion

Dans cette première section d'état de l'art, nous avons présenté les fonctionnalités proposées par la plupart des réseaux sociaux actuels, les usages constatés et étudié leur impact cognitif. De plus en plus utilisés par des solutions destinées aux entreprises, les mécanismes de diffusion et d'interaction apportés par les réseaux sociaux montrent un potentiel important pour améliorer la diffusion de connaissances et la collaboration entre professionnels, y compris au sein d'une même entreprise. Toutefois, le temps nécessaire pour maintenir manuellement un réseau, ainsi que les distractions occasionnées par la réception instantanée de trop nombreux messages sont un frein à leur adoption, par crainte d'une baisse de productivité.

Sur la base du sondage que nous avons mené, et de références en psychologie cognitive, nous avons identifié une piste d'amélioration de ces outils qui permettrait de profiter des flux de messages sociaux, tout en réduisant la perte de temps occasionnée par la lecture de messages trop souvent peu en rapport avec l'activité courante des utilisateurs. Elle consiste à développer un mécanisme de filtrage qui exploiterait des connaissances sur le contexte des utilisateurs pour ne notifier que les messages les plus pertinents.

III. Gestion d'informations dans les systèmes collaboratifs

Dans la section précédente, nous avons étudié l'intérêt des réseaux sociaux pour maintenir et enrichir des connexions entre personnes à des échelles supérieures à nos capacités cognitives naturelles. Dans cette section, nous allons étudier

cette problématique telle qu'elle est traitée dans le travaux de recherche sur la collaboration informatisée (CSCW), et identifier des techniques permettant de réduire la surcharge informationnelle causée par ces réseaux.

A. Systèmes de collaboration informatisée

Initialement défini par Dourish & Bellotti dans le cadre de travaux sur des outils de collaboration informatisés (CSCW), le concept d'***awareness*** (parfois traduit en français sous le terme de conscientisation) consiste à procurer une perception des activités d'autres personnes de manière à mieux contextualiser sa propre activité. A condition que les collaborateurs acceptent de partager entre eux des informations sur leurs taches et d'annoter celles des autres, il est possible d'augmenter leur productivité en réduisant les taches redondantes [16]. Nous remarquons que le concept de newsfeed, tel que décrit dans la section précédente sur les réseaux sociaux, est en fait une application informelle de ce principe.

B. Techniques de filtrage et recommandation

Notre objectif est de permettre une telle conscientisation à grande échelle, tout en réduisant de manière logicielle la charge induite par la lecture des informations partagées par les autres utilisateurs. Pour cela, deux types de mécanismes peuvent être mis en œuvre :

1) Filtrage basé sur le contenu: Ce type de filtrage permet de sélectionner des éléments pertinents en fonction de l'adéquation entre les caractéristiques de ces éléments et des critères attendus par l'utilisateur. De nombreux travaux utilisent cette technique pour sélectionner des documents pertinents en fonction de caractéristiques attendues par l'utilisateur. En 1997, Balabanovic décrit un système de recommandation de pages web où leur contenu est indexé par extraction statistique de mots-clés, et où le profil de l'utilisateur s'enrichit progressivement par notation manuelle de pertinence des recommandations [17]. Afin de réduire l'intervention manuelle de l'utilisateur, ce profil peut être généré dynamiquement en fonction ou de la caractérisation des documents actuellement ouverts par l'utilisateur [18]. Il est possible d'extraire des informations riches sur l'activité de l'utilisateur à partir des logiciels qu'il utilise [19]. Dans le cadre d'une expérimentation portant sur une application similaire, de meilleurs résultats de pertinence sont obtenus avec la méthode WordSieve d'extraction de mots-clés, qu'avec TF-IDF, classiquement utilisé dans les systèmes de recommandation basés sur le contenu [20]. Enrichie d'un mécanisme d'apprentissage, cette méthode prend en compte l'historique de l'utilisateur. Une technique aux avantages similaires est appliquée à des flux de messages agrégés à partir de blogs et réseaux sociaux sur l'application My6sense [1]. Outre la classification de contenus web, l'agrégation de données contextuelles à partir de plusieurs sources de données a été explorée dans le but de rappeler des notes à son auteur au moment le plus opportun [21].

2) Filtrage collaboratif: Contrairement aux techniques de filtrage basées sur le contenu, le filtrage collaboratif s'appuie seulement sur les liens entre des personnes et des éléments qui peuvent être recommandés à d'autres personnes. Le filtrage élément-élément consiste à trouver des utilisateurs liés avec un maximum d'éléments en commun, de manière à leur recommander les éléments qui leur manque. De la même façon, il est possible de recommander des personnes [22]. Si l'on considère que les signets web (bookmarks) d'une personne caractérise ses centres d'intérêts, il devient alors possible de recommander des personnes pertinentes par centre d'intérêt. Pour cela, Agosto regroupe les signets d'une personne par catégorie [23], en exploitant la taxonomie hiérarchique de l'annuaire Open Directory Project [2]. S'inspirant des principes sociologiques de l'Actor-Network Theory, Delalonde décrit un système de mise en relation basé sur les traces d'activités des utilisateurs [24]. Dans son approche, les utilisateurs caractérisent eux-même leurs données à l'aide de mots-clés. Toutefois, le vocabulaire prédéfini de 250 mots-clés s'est finalement montré trop contraignant auprès des utilisateurs invités à évaluer le système. Avec l'essor du web participatif, les tags – ces mots-clés associés librement par des milliers d'utilisateurs sur des contenus web – sont devenus un moyen puissant de caractériser les documents et les personnes dans le cadre de systèmes de recherche d'information [25], et de recommandation [26]. Cependant, sachant que seule une minorité de documents sont annotés sur des services de partage de signets comme Delicious [3], il n'est pas raisonnable de se contenter de cette seule source de méta-données [27].

Afin de bénéficier des avantages des deux méthodes de filtrage – la pertinence élevée du filtrage collaboratif à la généricité de l'analyse de contenu – Pazzani évoque la possibilité de les hybrider pour concevoir des *systèmes de recommandation collaboratifs via le contenu* [28].

C. Modèles de contenu et techniques d'extraction

Il existe deux courants majoritaires pour analyser et indexer du contenu de manière à pouvoir le recommander : le premier se base sur des mots en langage naturel – qu'ils soient ou pas tirés du contenu –, et le second consiste à identifier des entités sémantiques représentées dans le contenu.

1) Modèles basés sur les mots-clés: Plus haut, nous avons vu que plusieurs projets adoptent la méthode TF-IDF. Elle consiste à représenter un document sous forme de vecteur de mots-clés pondérés, en fonction de leur fréquence d'apparition dans le contenu, et de leur fréquence inverse d'apparition dans l'ensemble des documents considérés [29]. Afin d'améliorer les résultats de fréquence, il est aussi courant de réduire ces mots sous leur forme la plus simple, en retirant les pluriels et autres suffixes d'accords, ainsi qu'en réduisant les verbes conjugués sous leur forme infinitive. Certains algorithmes combinent de telles techniques statistiques et linguistiques à des techniques d'apprentissage permettant d'associer de mots

1. http ://www.my6sense.com/

2. http ://www.dmoz.org/
3. http ://delicious.com/

à des termes plus généraux décrits dans une taxonomie, c'est le cas du module logiciel KEA [4]. Cependant il est aussi possible d'obtenir des mots-clés descriptifs à partir d'annotations associées au contenu, sachant que cette pratique d'annotation est très courante sur Internet [30]. Malgré le manque de structure et l'ambiguïté sémantique des tags fournis par les internautes (ex. grâce à des sites comme Delicious), leur quantité permet de décrire les documents de manière très large et selon différents points de vue, et même de décrire des entités et événements du monde réel.

2) Modèles sémantiques: Basés sur la définition de concepts dans des ontologies, les modèles sémantiques explicitent le sens de manière formelle, de manière à ce que des systèmes automatiques puissent raisonner sur des bases de connaissances. Alors que de telles bases de connaissances sémantiques sont constituées en traduisant des référentiels sûrs (ex. DBpedia [5], une traduction sémantique de Wikipedia [6], et the WordNet lexical database [7]), plusieurs services permettent de bénéficier d'une telle traduction automatique en fournissant du contenu en langage naturel : Semantic Proxy [8], tagthe.net [9] et AlchemyAPI [10], pour les retourner sous forme de triplets au format RDF. La sémantique peut aussi émerger à partir d'un vecteur de mots-clés pondérés (ex. tags) [31]. Dans tous les cas, l'utilisation de modèles sémantiques impose la création et la maintenance d'ontologies à jour, et couvrant tous les sens nécessaires, sans quoi certaines annotations peuvent ne pas être représentées.

D. Pistes d'optimisation

Les systèmes basés sur des modèles vectoriels de mots-clés nécessitent souvent un espace de représentation de dimension élevée, ce qui réduit les performances des calculs de similarités, critiques dans les systèmes de filtrage et recommandation. Pour réduire cette dimension, il est possible de recourir à des techniques de statistiques incrémentales au lieu de TF-IDF [32], d'unifier les mots ayant une sémantique proche [33][34]. Cette dernière technique permet effectivement d'améliorer la performance d'un système de recommandation basé sur les tags [35], mais nécessite une phase d'apprentissage.

E. Conclusion

Dans cette section, nous avons introduit les fondements des systèmes de collaboration informatisés, identifié des techniques de filtrage permettant d'automatiser la sélection de documents et de personnes pertinentes dans de tels systèmes, et comparé plusieurs modèles de représentation de données. En particulier, le filtrage collaboratif via le contenu semble être la meilleure méthode pour augmenter le ratio signal-bruit dans le système que nous développons. De manière

4. http ://www.nzdl.org.org/Kea/
5. http ://dbpedia.org/
6. http ://www.wikipedia.org/
7. http ://wordnet.princeton.edu/
8. http ://semanticproxy.com/
9. http ://tagthe.net/
10. http ://www.alchemyapi.com/

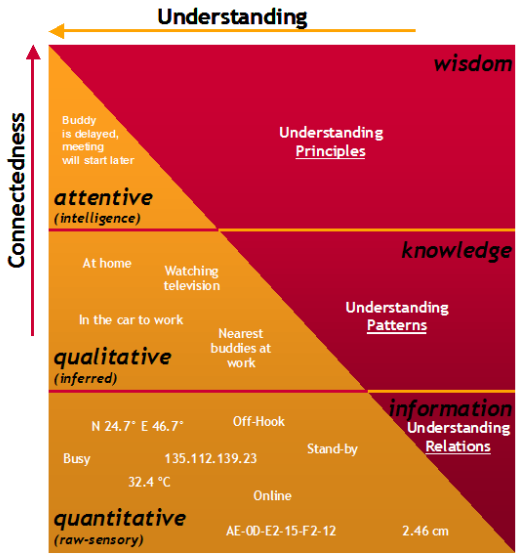

FIGURE 3. **Un message social et une réponse mentionnant l'auteur, sur Twitter**

à pouvoir profiter de l'annotation participative (tags) et de l'émergence spontanée de sens sur les réseaux sociaux (ex. hashtags, sur Twitter), un modèle vectoriel de données est le plus approprié pour décrire des activités et des contextes utilisateur de manière ouverte.

IV. Extraction et modélisation de données de contexte

Afin d'augmenter le ratio signal-bruit dans les applications de communication médiatisée, le contexte des utilisateurs a été identifié comme un bon critère de filtrage. Dans cette section, nous étudions l'exploitation de données de contexte utilisateur au sein d'applications informatiques, puis nous définissons plus en détails trois types de capteurs de contexte : physique, virtuel et social.

A. Systèmes de gestion de contexte

Les systèmes de gestion de contexte permettent de collecter et d'interpréter (cf Figure 3) de manière centralisée des données sur le contexte de l'utilisateur, afin que plusieurs applications puissent adapter leur fonctionnement en accord avec sa situation et les contraintes qui y sont éventuellement associées. Depuis son introduction par Schilit en 1994 [1], cette discipline de recherche en gestion d'informations a beaucoup évolué, et a accouché de nombreuses problématiques que nous allons aborder ici.

1) Premiers systèmes de gestion de contexte: Les premières applications sensibles au contexte étaient implémentées de manière à supporter certains types spécifiques de capteurs seulement. Malgré la similarité fonctionnelle de la plupart capteurs – en particulier, les capteurs de positionnement, – il n'y avait aucune réutilisation du code source permettant d'exploiter ces capteurs, ni moyen de fédérer les informations produites et induites au bénéfice de plusieurs applications. En 2000, Dey conçut alors le Context Toolkit, un

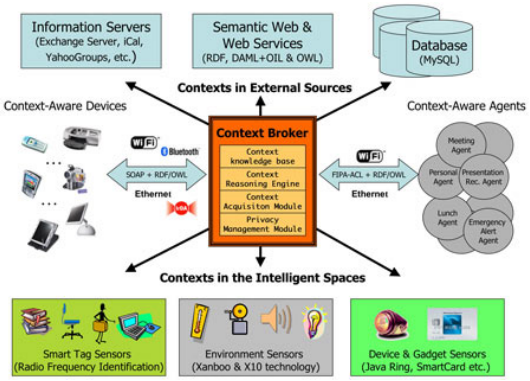

FIGURE 4. **Le système CoBrA**

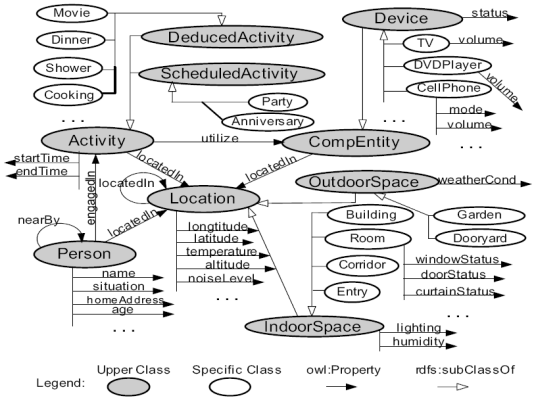

FIGURE 5. **Ontologie de contexte CONON, utilisée dans SOCAM**

cadre logiciel modulaire séparant les modules de gestion de contexte (capture, interprétation et stockage) des applications, afin de pouvoir les réutiliser dans d'autres applications, et de simplifier le développement de ces dernières [36]. Gu proposa ensuite les fonctionnalités attendues d'un système de gestion de contexte : acquisition de données de contexte à partir de capteurs variés (physiques et virtuels), interprétation de données contextuelles, distribution robuste et efficace de données de contexte aux applications qui l'ont demandé, et mise à disposition de modèles de programmation pour aider le développement d'applications [37].

2) Gestion sémantique du contexte: Les ontologies sont alors devenues la représentation de données la plus courante dans les systèmes de gestion de contexte [38][39], bénéficiant des capacités d'inférence et de raisonnement apportées par les langages de logiques de description (ex. OWL-DL) et de règles (ex. F-Logic) [40]. Se développe alors le système CoBrA (pour Context Broker Architecture) supportant de nombreux capteurs [41] (cf Figure 4), mais dont les données sémantiques doivent subir une traduction pour que les règles d'inférences puissent être appliquées. Néanmoins, l'ontologie SOUPA sur laquelle repose CoBrA, et liée à d'autres ontologies pré-existantes, a été adoptée par d'autres systèmes de gestion de contexte. L'un de ces systèmes, MOGATU ajoute la composante BDI permettant d'orienter le raisonnement en fonction de souhaits et buts de l'utilisateur [42].

3) Incertitude et qualité de contexte: Avec le système GAIA, le raisonnement probabiliste et en logique floue sont expérimentés dans le but de mieux s'adapter aux contextes incertains [43]. Dans SOCAM [37] (dont l'ontologie de contexte CONON est représentée en Figure 5) et KMF [44], de nouvelles propriétés permettent d'expliciter le degré de confiance, la fraicheur et l'origine de chaque donnée de contexte, pouvant alors être capturée, définie, agrégée, ou déduite. Malgré la nécessité de prendre en compte des contextes incertains ou flous, une évaluation met en évidence le fait que la gestion sémantique du contexte pose un sérieux problème de complexité algorithmique rendant difficile le passage à l'échelle [45].

4) Passage à l'échelle: En réponse aux problèmes de complexité algorithmique induits par l'utilisation massive de modèles sémantiques pour gérer les données de contexte, Tan propose d'exécuter les tâches de raisonnement dès la réception de nouvelles données de contexte, au lieu de les traiter à la demande [46]. D'autres systèmes, quant à eux, proposent une solution hybride répartissant de manière optimale les données de contexte entre une base relationnelle et une base sémantique, de manière à réduire la dimension de la base de connaissances, et donc la complexité des processus de raisonnement [47][48].

5) Maintien de la vie privée: Sachant que les systèmes de gestion de contexte ont pour vocation la collecte de données de contexte sur leurs utilisateurs, des mesures doivent être prises pour garantir à ceux-ci que ces données privées ne seront pas compromises. Alors que Context Toolkit se contente de sécuriser l'accès aux données par capteur [36], la plupart de ses successeurs ont prévu des règles de permissions plus complexes pour modérer la circulation de données privées. Cependant, il est difficile d'imaginer un utilisateur final personnaliser ces règles dans un système où les applications ne sont pas définies à l'avance. L'utilisateur est en fait contraint à faire confiance aux règles telles qu'elles ont été définies dans ces systèmes, sachant que les données de contextes lui sont cachées et donc non contrôlables, tel que le rapporte Dourish [49]. Ce dernier dénonce la perte de contrôle imposée aux utilisateurs, conséquence de la volonté de rendre ces systèmes invisibles.

6) Exploitation pragmatique du contexte: Depuis 2008, l'accès à la géolocalisation se généralise grâce aux interfaces logicielles fournies par Google (via leur interface web, et sur leur système d'exploitation Android, pour téléphones et autres plateformes mobiles), Yahoo ! (Fire Eagle) et Apple (sur l'iPhone). De nombreuses applications exploitant ces données sont alors adoptées par des millions d'utilisateurs, sans pour autant reposer sur des systèmes de gestion de contexte complexes tels que ceux présentés plus haut. Le site Pachube [11], quant à lui, s'inspire de ces systèmes pour proposer une plateforme collaborative de gestion de contexte basée sur un modèle de données et de règles simplifié. Il permet

11. http ://www.pachube.com/

ainsi aux utilisateurs de soumettre des flux de données issues de leurs capteurs, et d'exploiter ces données (possiblement combinées avec d'autres données, selon la pratique du mash-up de services web) pour actionner des appareils ou autres services web. Dénuées de sémantique et pourtant adoptées massivement par le grand public, toutes ces interfaces valident la thèse de Bolchini selon laquelle la volonté de créer des systèmes génériques – en utilisant notamment des modèles sémantiques – n'est pas une solution pragmatique pour proposer des applications utiles et utilisables [50].

7) Conclusion et pistes: Nous avons vu dans cette section que, séduits par l'interopérabilité et les capacités de raisonnement permises par les modèles sémantiques, la plupart des systèmes de gestion de contexte présentés s'avèrent trop complexes pour permettre le passage à l'échelle. De plus, la volonté de rendre ces systèmes transparents pour l'utilisateur implique une inquiétude quant à l'exploitation des données privées qui y transitent. L'apparition d'interfaces web et mobiles de géolocalisation – voire de partage de données de capteur au sens plus large – ont finalement triomphé auprès du grand public, par leur simplicité. Dans un web où les flux de données se multiplient et se combinent au gré des internautes, le nouveau challenge n'est pas de créer des systèmes de gestion de contexte génériques entre capteurs et applications, mais de savoir extraire et exploiter les données de contexte produites par des millions de personnes au travers de leurs flux sociaux.

B. Capteurs physiques : géolocalisation

Répartis dans notre environnement ou embarqués dans nos appareils mobiles, les capteurs physiques nous permettent d'identifier et de mesurer certaines de caractéristiques de notre environnement telles que la position géographique, la température, la luminosité ou l'ambiance sonore. Dans cette section, nous étudions en particulier l'utilisation de capteurs variés pour déterminer la localisation géographique d'une entité.

1) Modélisation d'une position géographique: Elle peut être représentée selon des repères physiques ou symboliques. Le modèle physique consiste à référencer une position à l'aide de sa latitude, longitude et altitude, afin de la situer de manière absolue sur le globe terrestre. Une position se représente alors sous forme de trois valeurs décimales, et éventuellement d'un rayon de précision. Le modèle symbolique, quant à lui, s'appuie sur des relations structurelles de nature politique (ex. pays, ville), architecturelle (ex. bâtiment, étage), ou subjective (ex. chez moi). La représentation symbolique est plus libre : utilisation de taxonomies, ontologies, voire de mots en langage naturel. Le passage d'une représentation à une autre est assuré par de nombreux services web dont Google Geocoding API [12].

2) Localisation par satellite: Le satellite est l'un des moyens de localisation les plus précis, du moment que le signal entre les satellites et le terminal n'est pas perturbé par des obstacles. Alors que le système de positionnement satellitaire GPS est couramment utilisé dans les appareils de navigation grand public pour automobilistes, et même dans les téléphones portables, cette contrainte rend leur utilisation impossible ou très difficile à l'intérieur des bâtiments. Un récepteur GPS fournit la position sous forme physique absolue (latitude, longitude et altitude), ainsi que d'autres valeurs telles que l'heure universelle, la vitesse, la direction ainsi que des indicateurs de précision.

3) Localisation par antennes: Plusieurs technologies de connexion sans fils (ex. Wi-Fi, Bluetooth, NFC) peuvent permettre à un récepteur de se localiser par triangulation, à condition qu'il soit à portée d'une ou plusieurs antennes dont l'emplacement géographique est connu. Sur les téléphones mobiles GSM, en quasi-permanence connectés à une ou plusieurs antennes, ce mécanisme de localisation a l'avantage de consommer moins de batterie que le GPS, mais montre une précision plus faible.

4) Localisation collaborative: Sachant que certains terminaux mobiles sont équipés de plusieurs interfaces sans-fil permettant la géolocalisation, il devient possible de créer collaborativement un référentiel d'antennes localisées. En effet, toute personne proche d'une antenne GSM ou Wi-Fi, et géolocalisée précisément par satellite, peut aider à déterminer la position géographique de cette antenne. Ainsi, d'autres personnes ne bénéficiant pas d'une localisation précise (ex. par satellite), peuvent tout de même bénéficier d'une localisation approximative, celle des antennes auxquelles elles sont connectées. Le produit VirtualGPS de Navizon [13], ainsi que ContextWatcher [51] exploitent cette technique.

C. Capteurs virtuels

Dans le cadre d'une activité informatisée, des données de contexte sur cette activité de l'utilisateur peuvent être extraites à partir des logiciels qu'il utilise [19].

1) Le contenu ou sujet de documents web: Les pages web en cours de visualisation peuvent être analysées par une extension du navigateur, ou par la capture du flux réseau induit par son chargement (ex. via un proxy). Des services web comme Delicious [14] permettent d'accéder à des métadonnées descriptives fournies par leur utilisateurs sur ces pages web. Par ailleurs, certains événements (ex. la soumission d'une réponse sur un forum) peuvent être interceptés après avoir souscrit à un flux RSS correspondant.

2) Le contenu ou sujet d'un document local: De manière similaire, le contenu d'un document consulté ou édité localement peut fournir de précieux indices concernant le contexte d'activité de l'utilisateur. Pour accéder à ce contenu, certaines applications comme Microsoft Office proposent des interfaces logicielles pouvant être exploitées [18]. Sinon, il reste possible d'identifier quels fichiers sont actuellement ouverts sur le système, et d'extraire dans un processus séparé le contenu de ces fichiers.

12. http ://code.google.com/apis/maps/documentation/geocoding/

13. http ://www.navizon.com/

14. http ://delicious.com/

3) Détection de problèmes: Les difficultés que rencontrent éventuellement un développeur peuvent-être identifiées à partir de plateformes comme Eclipse ou Microsoft Visual Studio. Par exemple, des tentatives fréquentes de compilation peuvent être rapportées de manière automatique pour proposer l'aide d'un collègue au développeur [52].

4) Mesure de présence: La présence et le niveau de stress d'un utilisateur peut être estimé à partir de l'analyse des mouvements de souris, des saisies au clavier, du l'activation du mode veille et des fermetures/ouvertures de session.

D. Capteurs sociaux

Grâce à son ouverture aux contenus et annotations des internautes, que ce soit sur des ressources web ou des entités du monde réel, le web est un environnement collaboratif dans lequel il devient possible de capter des données de contexte. Malgré leur qualité inégale, ces contributions, lorsqu'elles sont en quantité suffisante, permettent de décrire des ressources de manière plus humaine que les méthodes automatique, en profitant de plusieurs points de vue.

1) Référentiels collaboratifs: Lancé en 2001, Wikipedia est l'un des premiers sites web participatifs, invitant tous les internautes à constituer ensemble une encyclopédie en partant de rien. Cette encyclopédie en ligne peut-être aujourd'hui utilisée comme référentiel de connaissances sémantiques, grâce à DBpedia [15]. Alors que la proportion de contributeurs par rapport aux visiteurs de ce genre de site est très faible – de l'ordre de 1% – la pratique d'annotation participative (***tagging***) est devenue plus populaire, grâce au site de partage de signets web Delicious [16]. En effet, la saisie libre de mots-clés sur un signet est plus simple, plus rapide, plus utile et plus personnelle que la contribution à un article de Wikipedia. Ces annotations ainsi apportées collaborativement permettent de refléter de manière subjective la nature, la fonction, la catégorie et le contexte des ressources correspondantes [53][54].

2) Annotation d'entités du monde réel: Avec la popularité croissante de sites de documentation et de recommandation, de plus en plus d'entités du monde réel sont représentées et annotées sur le web : personnes, lieux, évènements, objets... Par exemple, il est possible pour une machine de reconnaitre visuellement une personne, un lieu, voire d'identifier l'émergence d'un événement grâce au partage de photos géolocalisées et annotées par les utilisateurs du service Flickr [55]. Les lieux d'intérêts (ex. sites touristiques) sont décrits, commentés, critiqués par les visiteurs sur des sites (et applications mobiles) comme Yelp [17], Qype [18] et DisMoiOu [19]. Les évènements sont explicitement annoncés et décrits sur des sites comme MySpace [20], Facebook [21], Taweet [22], voire les agenda partagés comme Google Agenda [23]. Les activités planifiées, telles que les prochains voyages, sont annoncées sur des sites comme Plancast [24], Tripit [25] et Dopplr [26]. Il est même possible pour une machine de reconnaitre une musique écoutée par l'utilisateur, grâce aux services Musicbrainz [27] et Shazam [28]. La plupart de ces services proposent des flux RSS permettant de suivre les activités de leurs utilisateurs, et/ou des interfaces logicielles permettant d'accéder aux données de leur référentiel.

3) Le capteur internaute: Grâce à la pratique croissante du micro-blogging (ex. Twitter [29]) et à leur ouverture (en terme d'accès aux informations), il est devenu aisé d'obtenir en temps-réel des données récentes sur des entités et sujets variés. En particulier, 41% des messages diffusés sur Twitter reflètent l'activité actuelle de leur auteur [9], et ce nombre atteint 51% en mobilité. Grâce à la facilité de diffusion à partir des téléphones mobiles, les utilisateurs sont de plus en plus tentés d'écrire à propos de ce qu'ils sont en train de vivre, les faits desquels ils sont témoins. Depuis qu'il est possible d'ajouter à chaque message la localisation géographique depuis laquelle il a été diffusé, ce genre de messages géolocalisés permet d'identifier des évènements, des situations locales, mieux que ne le feraient un capteur physique. L'internaute social devient alors un capteur à part entière.

E. Conclusions de l'état de l'art

Dans cet état de l'art en trois parties, nous avons observé que le suivi de personnes et de sujets sur des systèmes collaboratifs et sociaux était une activité couteuse en temps et en attention. Outre le temps nécessaire à la diffusion de messages, le nombre de message à lire augmente proportionnellement au nombre de personnes et sujets suivis, et la distraction occasionnée par les notifications intempestives de nouveaux messages réduisent l'attention de l'utilisateur sur son activité principale. Le besoin d'un support logiciel pour aider à réduire ces couts a été confirmé par les utilisateurs.

Nous avons vu que la connaissance du contexte d'un utilisateur est primordiale pour évaluer la pertinence de messages qui vont potentiellement l'interrompre. Sachant que, dans le cadre d'une activité informatisée, des données de contexte peuvent être tirées des documents et applications logicielles utilisées par l'utilisateur, une caractérisation du contenu associé est importante. C'est pourquoi nous avons décidé d'appliquer une méthode de filtrage collaboratif via le contenu, sur les flux de messages afin de ne notifier que les plus pertinents. Une représentation sous forme de mots-clés en langage naturel (tags) a été retenue, de manière à pouvoir profiter de plus nombreuses sources de données contextuelles, trop peu souvent décrites sous forme sémantique, et sachant qu'il est possible de faire émerger la sémantique à partir de jeux de mots-clés.

15. http ://dbpedia.org/
16. http ://delicious.com/
17. http ://www.yelp.com/
18. http ://www.qype.com/
19. http ://www.dismoiou.fr/
20. http ://www.myspace.com/
21. http ://www.facebook.com/
22. http ://taweet.com/
23. http ://www.google.com/calendar/
24. http ://plancast.com/
25. http ://www.tripit.com/
26. http ://www.dopplr.com/
27. http ://musicbrainz.org/
28. http ://www.shazam.com/
29. http ://www.twitter.com/

Enfin, dans notre analyse des systèmes de gestion de contexte, nous avons confirmé que la gestion de contexte sous forme sémantique pose des problèmes de passage à l'échelle. De plus, les systèmes de gestion de contexte trop génériques et pro-actifs font perdre le contrôle des utilisateurs sur l'exploitation de leurs propres données privées (ex. localisation), impliquant un rejet de leur part. Trois types de capteurs de contexte ont été définis :

- les capteurs physiques extraient de l'information échantillonnée à partir d'un environnement physique, ex. localisation, température ;
- les capteurs virtuels synthétisent des données contextuelles à partir d'activités informatisées, ex. lecture/rédaction de documents ;
- les capteurs sociaux agrègent des données qui ont émergé collaborativement sur Internet, à partir de service web participatifs, ex. tags apportés par les utilisateurs de sites de partage de signets, de contenus, et de micro-blogging géolocalisé.

V. Élaboration du cadre de gestion de contextes

Après avoir constaté l'amplification du problème de surcharge d'information sur les réseaux sociaux, la difficulté de caractériser le sujet et l'intention de l'utilisateur derrière chaque message, comparé plusieurs techniques de filtrage d'informations, et étudié comment exploiter des données de contexte sur les utilisateurs, nous allons élaborer un cadre permettant d'exploiter ces contextes pour filtrer les messages sociaux.

Dans cette section, nous allons définir le cadre conceptuel et logiciel de gestion de contextes. Dans la section suivante, nous développerons une application sociale basée sur ce cadre.

A. Problématique

Afin de réduire le cout lié à la distraction de l'utilisateur causée par la notification instantanée de messages sociaux, nous proposons de notifier seulement les messages jugés comme étant les plus pertinents par rapport à son contexte d'activité actuel. Nous allons donc modéliser un système de filtrage de données dans lequel les éléments sont des messages sociaux, et la critère de sélection sera basé sur la similarité de contextes entre chaque producteur et consommateur de ces messages.

1) Hypothèse: Un message est pertinent pour un utilisateur si le contexte dans lequel il sera reçu est similaire au contexte de l'utilisateur qui l'a émis, au moment où il l'a émis.

2) Questions:

- Comment modéliser le contexte d'activité d'utilisateurs, de manière à pouvoir mesurer la pertinence de messages sociaux ?
- De quels capteurs peut-on extraire des données de contexte utiles en guise de critère de filtrage ?
- Sur quel types de messages sociaux le filtrage est-il le plus efficace ?
- Comment maintenir la vie privée des utilisateurs ?

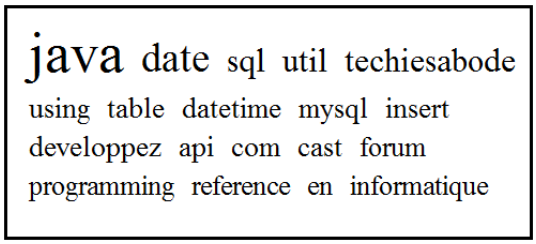

FIGURE 6. **Un nuage de tags représentant un contexte d'activité**

3) Contraintes:

- La qualité émerge de la quantité : En plus des capteurs physiques classiques, nous avons identifié les capteurs virtuels et sociaux comme des sources de données de contexte pertinentes pour notre système de filtrage. Or, contrairement aux capteurs physiques, ces derniers peuvent difficilement produire des données sémantiques, en particulier lorsque les données sont saisies par des humains. Nous proposons donc d'exploiter un format de représentation commun à tous ces types de capteurs.
- L'utilisateur doit être maitre de ses informations : Comme pour les messages envoyés à un réseau social, les données de contexte d'un utilisateur sont de nature privée, et la décision d'exploiter ou non ces données doit donc rester entre ses mains. Pour cela, nous proposons d'agréger les données de contexte sur l'équipement propre de l'utilisateur, sous forme compréhensible et manipulable, et de ne soumettre aucune donnée de contexte avant que l'utilisateur n'aie validé cet envoi.

B. Approche

En accord avec les contraintes ci-dessus, nous proposons de représenter les données de contexte sous forme de mots pondérés en langage naturel. En effet, toutes les données de capteurs peuvent être représentées selon cette forme, et elle est directement visible et simplement modifiable par un utilisateur novice. Un contexte pourra alors être représenté à l'utilisateur sous forme d'un nuage de tags (cf fig. 6).

Les poids – représentés en jouant sur la taille des mots, tels qu'affichés aux utilisateurs – représenteront alors l'importance et la confiance accordés aux mots pour définir le contexte.

C. Modélisation du contexte

A partir des capteurs étudiés dans la section précédente, nous définissons trois dimensions de contexte : activité, environnement et intentions de l'utilisateur. A défaut de définir des structures de données spécifiques, nous proposons ici une liste non exhaustive de caractéristiques qui pourraient intégrer un nuage de tags contextuel. En effet, la spécification de contextes et de règles prédéfinies restreindrait les cas d'usages possibles.

1) Activité de l'utilisateur: Les informations décrivant l'activité actuelle de l'utilisateur sont déterminantes pour mesurer l'impact de la réception de messages sociaux sur cette activité. Outre l'extraction de telles informations à partir de messages sociaux, voici quelques exemples :

– Le sujet et mots-clés de contenus en cours de consultation ou rédaction peuvent être extraits à partir des pages web et documents actuellement ouverts par l'utilisateur, à l'aide de capteurs virtuels (analyse de documents en local) ou sociaux (description participative de contenus publiés, ex. delicious).
– L'identification de contenus multimédia en cours de consultation (musique, vidéo) peut être assurée de manière similaire.
– La situation sociale (ex. en réunion, en discussion) peut être prévue grâce à des capteurs virtuels (calendrier de l'utilisateur, caractérisation d'un signal de microphone embarqué) ou sociaux (calendrier partagé).
– Une situation de déplacement peut être inférée à l'aide de capteurs virtuels (calendrier de l'utilisateur), sociaux (calendrier partagé), et/ou physique (mesure de vitesse et direction de déplacement).

2) Environnement de l'utilisateur: La connaissance de données sur la localisation, les personnes et les évènements environnants peut inciter les interlocuteurs potentiels à initier une communication, ou adapter leur discours :

– La localisation peut être extraite à partir de capteurs physiques (ex. GPS, antennes), représentée de manière textuelle à l'aide de services de geocoding, les lieux d'intérêt environnants peuvent être extraits depuis des référentiels web (ex. yelp, dismoiou), et les tags récupérés sur des sites de géotagging (ex. flickr).
– Les évènements en cours peuvent être extraits à partir de calendriers publics, mais aussi identifiés à partir de capteurs sociaux : contenus géolocalisés et partagés en temps réel (ex. photos, messages sociaux).
– Les personnes proches peuvent être identifiée par des capteurs physiques (ex. bluetooth) ou sociaux (contenus partagés par les deux personnes à un moment et position proches).
– La musique ambiante peut être reconnue par des services comme shazam.
– D'autres caractéristiques (bruit, foule) peuvent être inférées à partir de capteurs physiques (microphone, thermomètres, capteurs de présence), ou sociaux (quantité de messages sociaux au même moment et même localisation).

3) Intentions de l'utilisateur: Nous avons identifié l'intérêt des utilisateurs pour lire des messages sociaux en rapport avec leurs intentions et buts à plus long terme. En effet, les messages sociaux en rapport avec ces intentions ont valeur de conseil ou de recommandation dignes de confiance, car issues du réseau social de l'utilisateur. Actuellement, certaines de ces intentions peuvent être extraites :

– Prochaines destinations : à partir de calendriers, de services web de publication de déplacements (ex. tripit, dopplr), voire d'une combinaison de capteurs physiques et sociaux (reconnaitre un trajet en avion ou train en fonction de trajectoires partagées).
– Les gouts, préférences et souhaits d'une personne sont parfois manifestés explicitement via des messages sociaux, qui peuvent alors être analysés.

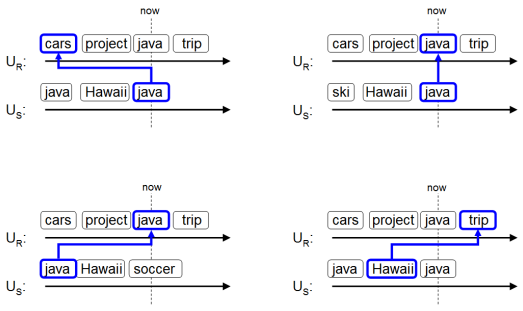

FIGURE 7. **Critères de pertinence en fonction du temps**

D. Modèle formel et algèbre

Conformément aux travaux de recherche d'information classiques [29], nous représentons un nuage de tags contextuel (assimilable à un profil instantané de l'utilisateur) sous forme d'un vecteur de termes pondérés : $v(d) = [w(t_1, d), w(t_2, d), ..., w(t_N, d)]$, pour chaque tag t_N, et d représentant une source de contexte. Les opérateurs suivants sont alors définis pour permettre la manipulation de ces vecteurs :

– L'opérateur d'*addition (+)* entre deux vecteurs retourne un vecteur contenant l'ensemble des termes, dont le poids est la somme des poids de chacun de ces termes.
– L'opérateur $\|v\|$ consiste à normaliser les poids d'un vecteur v, de manière à ce que leur somme fasse 1.
– L'opérateur d'*agrégation* est une addition normalisée de plusieurs vecteurs normalisés :

$$aggr(V) = \left\| \sum_{t=1}^{M} \|v_t\| \right\|$$

– La *fonction de similarité* est basée sur la mesure cosinus. Entre deux vecteurs R et S :

$$sim(R, S) = \frac{R \cdot S}{\|R\|\|S\|}$$

Elle retourne un score de similarité décimal entre 0 et 1, 1 lorsque les vecteurs sont identiques.

E. Mesure de pertinence

A partir des opportunités de filtrage identifiées plus haut, nous définissons un modèle de pertinence entre l'émetteur U_S et un receveur potentiel U_R d'un message social, selon quatre critères d'appariement temporels tels que représentés en Figure 7 :

1) Recommandation basée sur les intérêts passés: (en haut à gauche) Les intérêts du receveur sont conservés dans un profil agrégeant les tags contextuels passés, de manière à pouvoir recommander des messages en rapport avec ceux-ci.

2) Filtrage basé sur les intérêts actuels: (en haut à droite) Les messages récents sont filtrés en fonction de la similarité entre le contexte du receveur et de l'émetteur, créant alors des opportunités peu disruptives de collaboration immédiate.

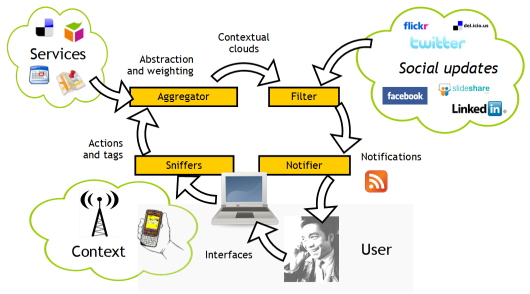

FIGURE 8. **Architecture du système, et boucle d'exécution**

3) Recommandation basée sur les intérêts actuels: (en bas à gauche) Des messages plus anciens sont recommandés lorsque le contexte de l'émetteur est similaire au contexte courant du receveur, permettant alors de proposer des solutions pertinentes (et un moyen de contacter leur auteur) par rapport à son contexte d'activité.

4) Recommandation basée sur les intérêts futurs: (en bas à droite) Si les intérêts futurs du receveur sont connus, des messages plus ou moins récents peuvent être recommandés par ordre de pertinence.

Dans le cadre de cette thèse, nous prenons en compte les contextes courants et passés, ce qui permet de réaliser les trois premiers critères. Nous définissons donc la fonction de pertinence suivante :

$$rel(S, R, P, \alpha) = sim(\|S\|, \|\alpha \cdot R + (1 - \alpha) \cdot P\|)$$

Pour chaque message social, la pertinence sera calculée entre l'émetteur dont le contexte courant est représenté par S, et le receveur dont le contexte courant est représenté par R. Les intérêts passés du receveur sont représentés par P, un vecteur agrégeant les tags des contextes précédents.

La variable décimale α, définie dans l'intervalle $[0; 1]$ permet au receveur d'ajuster le poids de son contexte courant par rapport à celui de ses intérêts passés, en fonction de la nature des recommandations qu'il souhaite recevoir (cas 1 et 2). Une autre variable peut être proposée au receveur pour ajuster l'âge minimal ou maximal des messages qui lui seront proposés (cas 2 et 3), influant alors sur les valeurs de S qui seront évaluées par la fonction de pertinence.

F. Conception du système

Conformément aux principes de programmation modulaire nécessaires pour permettre le développement séparé de sources de contexte et d'applications, nous avons conçu une architecture logicielle basée sur quatre modules, tels que représentés en Figure 8 :

*1) Sources de contexte (**sniffers**):* En charge de capturer des données brutes de contexte et de capturer les actions de l'utilisateur, chaque source de contexte doit fournir des nuages de tags contextuels (sous forme de vecteurs de termes pondérés) dès que le contexte de l'utilisateur a subi un changement, que ce changement soit provoqué par une action explicite de l'utilisateur (ex. ouverture d'un nouveau document) ou pas (ex. changement de température ambiante).

*2) Agrégateur de contexte (**aggregator**):* Ce module logiciel en exécution permanente sur le terminal de l'utilisateur reçoit les mises à jour de contexte provenant de chaque source afin de les agréger dans un nuage de tags contextuel représentant le contexte actuel de l'utilisateur, tel qu'il sera soumis au système de filtrage, à sa demande.

*3) Système de filtrage (**filter**):* Exécuté sur un serveur commun à tous les utilisateurs d'une même communauté, ce système reçoit et conserve les nuages de tags contextuels émis par chaque utilisateur, ainsi que les messages sociaux correspondants, de manière à leur proposer en retour une sélection de messages sociaux pertinents, grâce à la fonction de mesure de pertinence.

*4) Interface de notification (**notifier**):* Les messages sociaux proposés par le système de filtrage peuvent être transmis à l'utilisateur au travers de diverses interfaces : espace dédié sur un écran d'ordinateur, notification sur un terminal mobile etc... Ces interfaces doivent fournir une fonction permettant de contacter l'auteur d'un message, afin de permettre une éventuelle collaboration.

G. Structures de données

Le système de filtrage repose sur trois types de structures de données :

- Le nuage de tags contextuel, composé d'un vecteur de termes pondérés, et de fonctions permettant la normalisation de ce vecteur.
- La mise à jour de contexte, une structure associant un nuage de tags contextuel à la source de contexte qui l'a émise, ainsi qu'à l'heure précise à laquelle il a été émis. Lorsqu'un message social a été émis dans ce contexte, il est référencé dans cette structure.
- La recommandation de message social, une structure associant un message social, son émetteur, à un score de pertinence calculé spécifiquement pour un receveur donné en fonction de la similarité de son contexte avec celle de l'émetteur.

Les classes correspondantes sont décrites précisément dans le manuscrit de thèse.

H. Interfaces logicielles

Le module d'agrégation doit implémenter la méthode $onEvent()$ de l'interface $IEventListener$ afin de traiter les mises à jour de contexte émises par les sources de contexte. Dans notre implémentation de référence, ce module est matérialisé par un serveur HTTP exposant un point d'entrée $/event/$ auquel la structure de mise à jour de contexte doit être soumise, après avoir été sérialisée au format JSON. Une classe est fournie pour faciliter cette soumission vers notre serveur.

Responsable de la gestion de contextes au sein du serveur de filtrage, la classe $ContextManager$ propose trois types de souscriptions asynchrones :

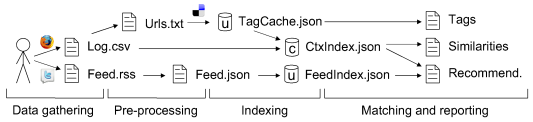

FIGURE 9. **Chaîne de recommandation pas à pas**

- $subscribeToEvents()$ permet de signaler les mises à jour de contexte aux classes implémentant l'interface $IEventListener$.
- $subscribeToMatch()$ permet de signaler les scores de similarités calculés entre les contextes de chaque utilisateur, à la suite d'une mise à jour, aux classes implémentant l'interface $IContextMatchHandler$.
- $subscribeToContext()$ permet de signaler les changements de poids observés pour chaque tag issu d'une mise à jour récente de contexte, aux classes implémentant l'interface $IContextListener$.

Dans notre implémentation de référence, la classe $DwrGuiDispatcher$ implémente l'interface $IContextMatchHandler$, de manière à afficher en temps réel les recommandations de messages sociaux sur page $Updates.html$ (interface de notification), via la technologie DWR.

I. Chaîne de recommandation pas à pas

Afin de permettre l'analyse expérimentale et reproductible de chaque étape du processus de filtrage, nous décrivons ici les modules logiciels indépendants à exécuter et les fichiers intermédiaires produits et exploités par ceux-ci, comme illustré en Figure 9.

1) Enregistrement des actions de l'utilisateur: Pendant la période de collecte des données, les activités de l'utilisateur sont enregistrées de manière transparente dans un fichier Log au format CSV. Dans le cadre de l'évaluation de notre système, les actions d'ouverture, de fermeture, et de changement de page web sont capturées à partir d'un navigateur web, en guise d'activités, et les URLs correspondantes sont listées dans un fichier texte.

2) Extraction et archivage des méta-données: Afin de constituer un contexte, les activités de l'utilisateurs sont analysées, interprétées par des algorithmes et services web d'extraction de tags. Dans le cadre de l'évaluation de notre système, ceux-ci produisent des tags à partir de la localisation, du contenu, et des descriptions participatives associées à chaque URL (celles listées dans le fichier texte généré à l'étape précédente). Les vecteurs de tags ainsi produits pour chaque combinaison d'URL et d'analyseur sont alors stockés dans un cache au format JSON.

3) Indexation des contextes: Sachant que les utilisateurs n'ont pas été invités à soumettre explicitement leur contexte pendant la période de collecte des données, il convient de simuler de telles soumissions de manière régulière et synchrone (pour tous les utilisateurs concernés) sur un axe temporel commun. En respect de la fréquence d'indexation fixée (ex. 10 minutes), chaque contexte est construit par agrégation des vecteurs de tags correspondant à toutes les URLs ayant été concernées par une action de l'utilisateur pendant la même tranche temporelle. Un fichier JSON est alors généré pour chaque utilisateur, listant ces nuages de tags contextuels ainsi constitués pour chaque tranche temporelle.

4) Indexation des messages sociaux: De manière similaire à l'indexation des contextes, les messages sociaux émis par les utilisateurs pendant la période de collecte des données (*feeds* récupérés au format RSS) sont indexés sur l'axe temporel commun, pour produire un fichier JSON par utilisateur. A ce stade, il est alors possible de connaitre le contexte des utilisateurs lors de l'émission d'un message social, sachant que les contextes et les messages sociaux sont indexés sur le même axe temporel.

5) Génération de rapports: Parmi les sept scripts développés :

- $recommender$ génère un rapport listant les scores de pertinence entre chaque message social et chaque contexte, à partir de fichiers d'index de contextes et de messages sociaux spécifiés.
- $genSurvey$ génère un formulaire HTML personnalisé pour chaque utilisateur, à partir de leurs fichiers d'index. Pour chaque utilisateur, cinq contextes sont sélectionnés et représentés sous forme de nuages de tags. Pour chacun de ces contextes, trois messages sociaux émis par d'autres utilisateurs sont proposés, ayant différents scores de pertinence : élevé, moyen et faible. L'utilisateur est alors invité à noter la pertinence perçue de chaque message par rapport au contexte correspondant.
- $genCtxFeedReport$ génère un formulaire HTML personnalisé à partir des fichiers d'index d'un utilisateur donné. Pour chaque message social émis par l'utilisateur, le nuage de tags contextuel de la période correspondante est représenté, et l'utilisateur est invité à évaluer la pertinence perçue entre ce message et son contexte.

J. Conclusion

Dans cette section, nous avons spécifié un cadre conceptuel permettant de valider notre hypothèse sur le filtrage contextuel de messages sociaux. En accord avec les contraintes identifiées (ex. exploitation d'annotations participatives, contrôle des utilisateurs), nous avons conçu un modèle de données de contexte basé sur des tags, puis quatre critères de sélection de messages pertinents. Enfin nous avons décrit l'implémentation de ce cadre conceptuel.

VI. APPLICATION À L'ENTREPRISE

Dans la section précédente, nous avons décrit un cadre de gestion de contexte basé sur des tags, ainsi qu'une méthodologie permettant de filtrer des messages sociaux en fonction de similarités de contextes. Dans cette section, nous proposons une application sociale d'entreprise reposant sur ce cadre, appelée '*Enterprise Contextual Notifier*'. Cette application exploite la navigation web des utilisateurs pour constituer leur contexte d'activité.

A. Étude de cas

Prenons le cas d'une entreprise dans laquelle des milliers d'employés travaillent sur des ordinateurs individuels. Répartis sur plusieurs sites et plusieurs pays, peu d'employés se connaissent et communiquent entre eux. La majorité des communications entre équipes repose sur la coordination et le transfert d'information opérés par des directeurs à différents niveaux hiérarchiques.

Nous proposons de créer un réseau social informatisé permettant à chaque employé de partager ses activités, et de consulter celles des autres, du moment qu'elles sont pertinentes avec les siennes. Le temps de maintenance de cet outil doit être négligeable, et les interruptions de taches doivent rester exceptionnelles. Cet outil doit permettre aux employés de partager leurs idées, activités en cours, intentions et questions ; et de les aider à obtenir des réponses de collègues pertinents, afin de promouvoir une communication et collaboration transversale, outre les barrières hiérarchiques.

B. Navigateur web en guise de capteur de contexte

Sachant que les employés travaillent sur des terminaux informatiques individuels, nous pouvons tirer des données de contexte à partir des logiciels qu'ils utilisent. En supposant que ces employés sont susceptibles de rechercher sur le web des informations en relation (au moins partiellement) avec leur activité en cours, nous allons considérer leur navigateur web comme un capteur de contexte. En application du cadre de gestion de contexte décrit dans la section précédente, nous définissons alors six moyens d'extraire des tags à partir de pages web :

1) Méta-données de la page: La fonction *Metadata* compte le nombre d'occurrences de chaque terme t en appliquant les coefficients α, β, et γ, selon la localisation de ce terme dans les méta-données de la page d :

$$w_1(t,d) = \alpha * |t \in T_d| + \beta * |t \in K_d| + \gamma * |t \in D_d|$$

où $|t \in T_d|$ est le nombre d'occurrences du terme t dans l'élément titre de la page d, $|t \in K_d|$ dans la liste de mots-clés, et $|t \in D_d|$ dans son texte de description. Dans le cadre de l'évaluation présentée plus bas, les paramètres suivants ont été utilisés : $\alpha = 50$, $\beta = 10$, et $\gamma = 1$.

2) Mots-clés de recherche: La fonction *SearchQuery* compte le nombre d'occurrences de chaque terme t dans une requête de recherche Q_d dont la page d présente les résultats :

$$w_2(t,d) = |t \in Q_d|$$

Cette fonction reconnait les requêtes formulées au moteur de recherche Google [30].

3) Nom de domaine: La fonction *DomainNames* considère en tant que termes les noms de domaine et de sous-domaine N_d de l'URL de la page d :

$$w_3(t,d) = |t \in N_d|$$

30. http ://www.google.com/

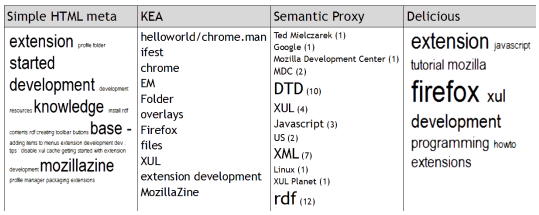

FIGURE 10. **Tags extraits à l'aide de quatre fonctions**

4) Annotation collaborative: La fonction *SocialBookmarks* compte le nombre de personnes qui ont publiquement annoté la page d en utilisant le tag t :

$$w_4(t,d) = \sum_{p \in P} tag(p,d,t)$$

où chaque personne p appartient au groupe P d'utilisateurs du service d'annotation, et où $tag(p,d,t)$ vaut 1 si la personne p a annoté la page d en utilisant le tag t, sinon 0. Cette fonction repose sur l'exécution d'un service web exposé par le site d'annotation Delicious [31].

5) Analyse sémantique: La fonction *SemanticAnalyzer* compte le nombre d'occurrences d'entités définies sémantiquement et représentées textuellement par le terme t, parmi celles identifiées dans la page d :

$$w_5(t,d) = |t \in R_d|$$

$$R_d = [\forall e \in E_d, repr(e)]$$

où $repr(e)$ est la représentation textuelle de l'entité sémantique e, R_d est la représentation textuelle des entités sémantiques E_d trouvées dans la page d. Cette fonction repose sur l'exécution du service web SemanticProxy [32].

6) Synthèse de mots clés: La fonction *KeyphraseExtractor* compte le nombre d'occurrences de chaque terme t de K_d, un ensemble de mots clés extraits du contenu de la page d :

$$w_6(t,d) = |t \in K_d|$$

Cette fonction utilise le module KEA, Keyphrase Extraction Algorithm [56].

C. Tags extraits à partir de pages web

Sur la Figure 10, nous comparons la nature des tags extraits à partir d'une même page à l'aide de quatre fonctions :

- La fonction Metadata basée sur le code HTML met en avant les mots contenus dans le titre de la page, ce qui est représentatif du contenu de la page, dans le cas présenté ici.
- La fonction KeyphraseExtractor basée sur KEA renvoie des mots pertinents (ex. $firefox$, xul), assez descriptifs (ex. '*extension development*'), mais parfois trop spécifiques (e.g. $ifest$, EM, $files$).
- La fonction SemanticAnalyser basée sur Semantic Proxy identifie des termes précis : RDF, DTD, XML et XUL

31. http ://delicious.com/
32. http ://semanticproxy.opencalais.com/

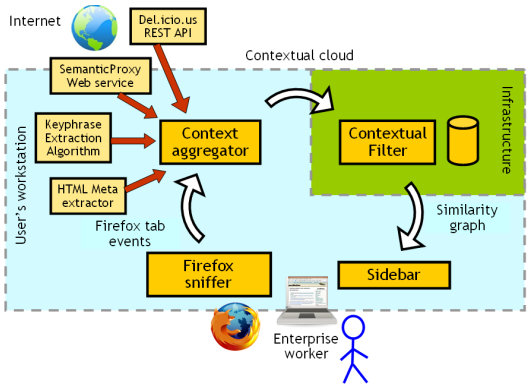

FIGURE 11. **Architecture et cycle de filtrage de ECN**

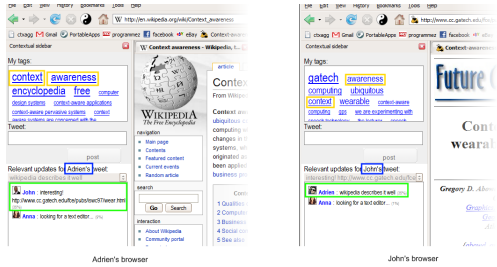

FIGURE 12. **Interface de ECN, vue par deux utilisateurs**

sont en effet des noms de technologies employées dans le contenu de la page. Des mots clés de poids plus faible reflètent moins le sujet de la page.

– La fonction SocialBookmarks basée sur Delicious renvoie des mots-clés moins nombreux mais plus descriptifs, de poids plus variés, et une forte représentativité de mots assez généraux mais très pertinents : $firefox$, $extension$, $development$. Même les mots de poids plus faible représentent bien le sujet et l'utilité de la page : $programming$, $tutorial$, xul.

Les tags issus de la fonction SocialBookmarks semblent particulièrement adaptés pour représenter le contexte d'un utilisateur basé sur sa navigation web. Néanmoins, les autres fonctions doivent être utilisées aussi, afin de couvrir les pages non annotées sur Delicious, et de renforcer le poids des mots-clés apparaissant dans les résultats de plusieurs fonctions.

D. Implémentation de l'application

L'application ECN a été implémentée en respectant une architecture modulaire, représentée en Figure 11, et s'appuyant sur le protocole HTTP pour communiquer entre elles :

1) Firefox Sniffer: L'unique capteur de contexte de ECN est une extension pour le navigateur web *Firefox*[33], développée en Javascript pour intercepter et transmettre à l'agrégateur les évènements d'ouverture, de fermeture et de changement de pages web.

2) Context Aggregator: Exécuté sur le poste de chaque utilisateur, l'agrégateur est un serveur léger développé en Java responsable de maintenir le nuage de tags contextuel de l'utilisateur à partir des actions et mises à jour de contexte soumises par les capteurs, de le rendre visible à l'utilisateur, et de le soumettre au serveur de filtrage, à sa demande. Dans notre implémentation, les six fonctions d'extraction et de pondération de tags (telles que définies plus haut) sont exécutées par l'agrégateur, à partir des URLs fournies par l'extension Firefox.

3) Contextual Filter: Le serveur de filtrage est un serveur HTTP en Java responsable de recueillir les messages sociaux (à partir de flux RSS, ex. Twitter) et les nuages de tags contextuels de chaque utilisateur, afin d'évaluer leur similarité, et de recommander les messages sociaux aux utilisateurs pour lesquels ils sont pertinents.

4) Side-bar: L'interface utilisateur d'ECN est générée en HTML par l'agrégateur, et affichée sous forme de barre latérale dans Firefox. Elle représente le nuage de tags contextuel de l'utilisateur, évoluant dynamiquement au gré de la navigation web de l'utilisateur. Un champ (appelé *tweet*) permet l'envoi d'un message social auquel sera attaché le nuage de tags contextuel. En dessous sont énumérés les messages sociaux recommandés par le serveur de filtrage, classés par ordre de pertinence décroissante. Sur la Figure 12, les messages sociaux des deux utilisateurs représentés, Adrien et John, leur sont mutuellement recommandés avec un score de pertinence de 35% car leur contexte de navigation est similaire.

E. Conclusion

Dans cette section, nous avons proposé une application sociale basée sur le cadre conceptuel décrit dans la section précédente. Cette application de recommandation contextuelle de messages sociaux repose sur 5 capteurs virtuels (Metadata, SemanticAnalyzer, SearchQuery, DomainNames, KeyphraseExtractor) et 1 capteur social (SocialBookmarks), tous basés sur la navigation web de l'utilisateur. Dans la section suivante, nous allons utiliser cette application pour évaluer le cadre conceptuel, dans le but de valider notre hypothèse.

VII. ÉVALUATION DU SYSTÈME

Sachant que l'évaluation d'un système de mise en relation entre personnes doit être centrée sur les personnes et leurs objectifs propres [57], nous avons décidé d'obtenir des données de volontaires sans leur imposer de contraintes, et de recueillir leur satisfaction concernant des recommandations de messages sociaux en fonctions de leurs propres contextes. Dans cette section, nous décrivons le l'expérimentation qui a été menée et présentons les résultats de quatre analyses :

– Qualité des nuages de tags contextuels perçue par les utilisateurs
– Pertinence de messages sociaux recommandés en fonction du contexte
– Pertinence de ses propres messages sociaux, par rapport à leur contexte d'émission
– Impact de l'origine des tags sur les scores de similarité contextuelle

33. http ://www.firefox.com/

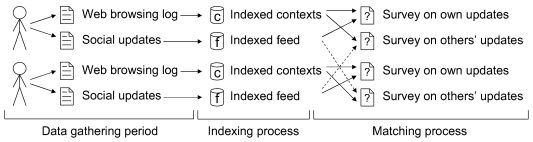

FIGURE 13. **Protocole expérimental**

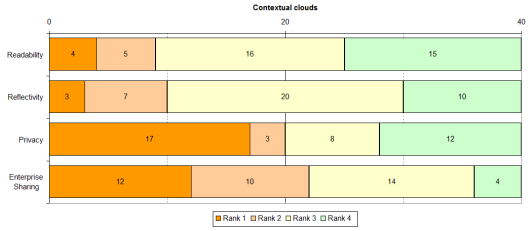

FIGURE 14. **Représentativité et volonté de partage**

A. Cadre expérimental

Afin de vérifier notre hypothèse sur la pertinence de messages sociaux contextuellement recommandés, nous avons enregistré près de deux semaines de navigation web et de messages sociaux fournis par huit employés volontaires, exécuté nos algorithmes pour obtenir 1846 nuages de tags contextuels, et demandé aux volontaires de noter la pertinence perçue de résultats sélectionnés.

Sachant que le système n'a pas été évalué sous sa forme interactive, les participants ne voyaient pas leur nuage de tags contextuel, et ne pouvaient donc pas décider du partage de leur contexte au moment où ils le souhaitaient. Nous avons alors choisi de considérer des nuages de tags contextuels générés par tranches de 10 minutes de navigation, et d'indexer les messages sociaux émis par les participants en associant chacun d'eux au contexte de la tranche correspondante. Dans le cas où le contexte d'une de ces tranches est vide, c'est le contexte de la tranche précédente qui y est associé.

Chaque nuage de tags contextuel ainsi indexé a été nettoyé, de manière à ce que chaque tag ne contienne qu'un seul mot, et soit dépourvu de ponctuation et autres caractères spéciaux. De plus, les mots non descriptifs (ex. déterminants, pronoms, et autres mots de liaison) sont retirés, et seuls les 20 tags les plus pondérés sont conservés, avant de normaliser les vecteurs de tags ainsi nettoyés. Tel qu'illustré en Figure 6, un nuage de tags contextuel contient finalement plusieurs types de mots, potentiellement de langues variées, des combinaisons de mots et des acronymes.

Nous avons ensuite appliqué la fonction de similarité à toutes les combinaisons de contextes, afin de produire une matrice de pertinence entre tous les utilisateur, et de générer deux types de questionnaires personnalisés (cf Figure 13) :

- Le premier propose à chaque participant une sélection de 5 contextes propres, afin d'évaluer leur représentation, et la qualité perçue de 3 recommandations (dont une jugée pertinente par la fonction de similarité contextuelle) par contexte,
- Le second propose à chaque participant la liste de ses propres messages sociaux associés au contexte de leur envoi, l'invitant ainsi à noter la pertinence de chaque message avec son contexte d'envoi.

B. Description des données de contexte

Avant de rapporter les résultats de l'évaluation menée auprès des participants, nous fournissons quelques caractéristiques observées pendant l'agrégation des données de contexte. Sur un total de 14625 URLs uniques accédées par les participants :

- SemanticProxy a fourni des tags pour 5292 URLs,
- des tags ont été extraits à partir des noms de sous-domaine de 3979 URLs,
- des méta-données ont été extraites du code source HTML de 2898 URLs,
- les tags donnés par les utilisateurs de Delicious ont pu être exploités pour 1395 URLs,
- et des termes de recherche ont été récupérés à partir de 1050 URLs Google.

C. Représentativité et volonté de partage

Afin d'évaluer la représentativité et la volonté de partage des nuages de tags contextuels, nous avons demandé à chaque participant de noter, pour cinq de leurs contextes et sur une échelle de 1 à 4, leur avis sur les quatre critères suivants :

- Lisibilité : *Ce nuage contient-il des mots en anglais et/ou en français ?* Réponses proposées : 1 - aucun, 2 - quelques mots seulement, 3 - plusieurs, or 4 - tous sont des mots bien orthographiés.
- Représentativité : *Ces mots reflètent-ils bien votre activité ?* Réponses proposées : 1 - pas du tout, 2 - vaguement, 3 - assez bien, or 4 - complètement.
- Confidentialité : *Avec qui seriez-vous prêt à partager ce nuage de tags contextuel ?* Réponses proposées : 1 - personne, 2 - à une personne spécifique, 3 - à un cercle de personnes choisi (amis, famille, collègues), or 4 - à n'importe qui.
- Partage dans l'entreprise : *Quel degré d'exploitation de vos contextes seriez-vous prêt à accepter en entreprise ?* Réponses proposées : 1 - aucun, 2 - je les garderai pour moi, 3 - prêt à les partager à condition que ça m'aide, or 4 - je suis prêt à la diffuser en temps réel et de manière transparente.

Tel que représenté en Figure 14, la lisibilité observée est de 3,05 (68,33%) en moyenne, et les notes individuelles sont assez homogènes. La représentativité moyenne est de 2,93 (64,17%). Certaines notes plus prononcées mettent en évidence l'impact du contexte sur ce type de représentation, et la qualité variable des tags extraits à partir des pages web. Ces résultats confirment que la visualisation de contexte sous forme de nuages de tags ne pose pas de problème aux utilisateurs.

Avec une note moyenne de 2,38 (45,83%), le rapport à la confidentialité révèle des résultats hétérogènes parmi les participants, en fonction des contextes. Avec une note moyenne de 2,25 (41,67%), la volonté de partage au sein de

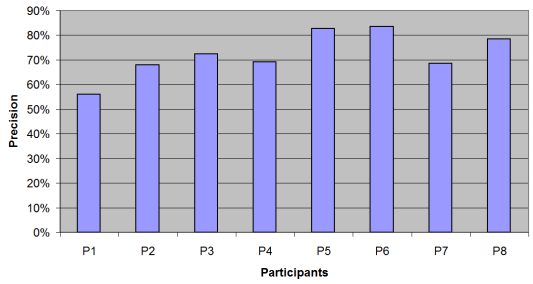

FIGURE 15. **Qualité des recommandations par participant**

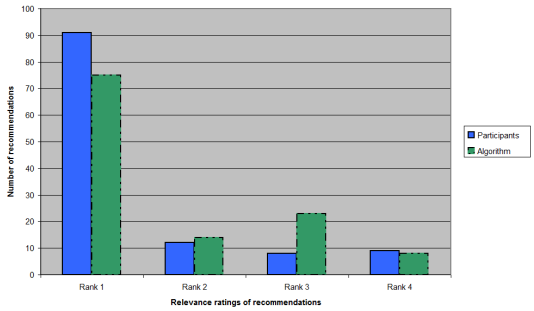

FIGURE 16. **Distribution comparative des notes de pertinence**

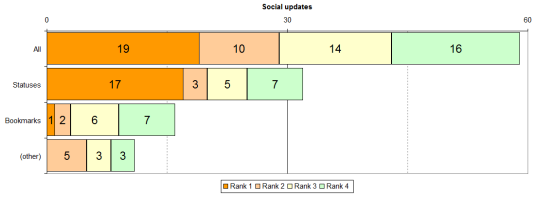

FIGURE 17. **Distribution des notes de pertinence par type de message social**

l'entreprise n'est pas pour autant corrélée avec les résultats précédents : la majorité des participants souhaitent conserver ces contextes pour eux-seuls. Deux explications sont proposées : tous les participants ne sont pas familiers avec les pratiques de réseautage social en entreprise, et certains tags contextuels n'étaient pas en rapport avec le travail, sachant que dans notre cadre expérimental il ne leur était pas possible de modifier/retirer des tags.

D. Pertinence contextuelle des messages recommandés

A présent, notre objectif est d'observer une corrélation entre les scores de pertinence estimés par le système et ceux perçus par les participants. Pour cela, certains messages recommandés aux participants étaient volontairement peu pertinents. Nous définissons alors une fonction de mesure de performance basée sur MAE (Mean Absolute Error) :

$$accuracy = 1 - \sum_{q=1}^{Q} |sim(C_q, U_q) - rating(C_q, U_q)|$$

Pour chaque recommandation q, $sim(C_q, U_q)$ est le score de pertinence entre le contexte de l'utilisateur U_q et celui de l'émetteur du message C_q. Alors que $rating(C_q, U_q)$ est la note de pertinence correspondante donnée par le participant. Ces scores, compris dans l'intervalle $[0; 1]$, sont représentés sous forme de pourcentages. Données dans l'intervalle $[1; 4]$, les notes de participants sont converties à l'aide de la fonction suivante :

$$rating = \frac{(grade - 1)}{3}$$

Comme nous le voyons sur la Figure 15, la performance varie entre 56% et 84%, avec une moyenne de 72%.

Sur la Figure 16, nous observons que la distribution des notes de pertinences sur les 120 messages sociaux recommandés aux utilisateurs est proche de celles estimées par le système. Les notes moyennes-hautes (surtout la note 3) sont légèrement surestimées par le système, alors que des notes basses (en particulier la note 1) ont été données plus fréquemment par les participants.

Nous observons que 63% des scores de pertinence attendus ont une note de 1, et que 19% ont une note de 3. Cette répartition montre que les contextes similaires étaient peu fréquents dans notre expérimentation. En augmentant le nombre de participants, plus de contextes similaires auraient pu être trouvés, améliorant ainsi la moyenne des scores.

E. Lien des messages avec leur contexte d'envoi

Afin de montrer que les nuages de tags contextuels sont de bons profils pour recommander des messages sociaux, nous avons demandé aux participants de noter la pertinence de leurs propres messages (ex. statuts twitter et notifications de signets delicious) par rapport au contexte de leur envoi.

Sur les 59 messages sociaux émis par les participants pendant l'expérimentation, le score de pertinence moyen noté par leurs auteurs est de 50,3%. La Figure 17 montre que 54% de ces messages sont des statuts Twitter, et 29% sont des notifications de signets Delicious.

Nous observons que le taux de pertinence moyen varie en fonction de la nature de ces messages : 71% pour les notifications de signets Delicious, contre 38% seulement pour les statuts Twitter. Il est naturel qu'un signet de page web soit plus similaires à son contexte, car les annotations préexistantes de cette page font partie du contexte au moment de la création du signet. Concernant les statuts, il a été prouvé que seuls 41% des statuts diffusés sur Twitter sont en rapport avec l'activité courante de son auteur [9], ce qui explique le score de pertinence correspondant que nous avons observé.

F. Capteurs virtuels et sociaux, étude comparative

Les nuages de tags contextuels générés dans le cadre de cette évaluation viennent de deux types de capteurs :

- quatre capteurs virtuels : Metadata (Méta-données HTML), DomainNames (noms de domaines), et SearchQuery (mots-clés de recherche Google) ;
- et un capteur social : SocialBookmarks (tags donnés par les utilisateurs de Delicious).

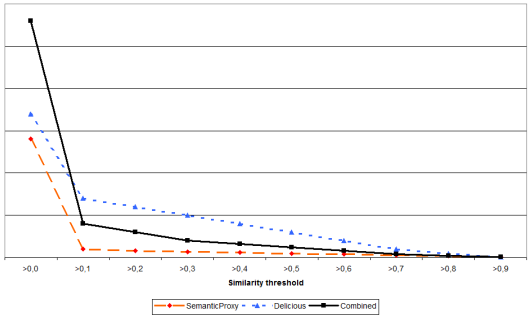

FIGURE 18. **Distribution comparative de contextes similaires, par seuil de pertinence**

Afin de comparer l'efficacité des capteurs virtuels et sociaux pour recommander des messages sociaux contextuellement, nous étudions la distribution des scores de similarité entre des contextes à base de tags issus de SemanticProxy (en tant que capteur virtuel) et de Delicious (en tant que capteur social).

La Figure 18 représente la distribution du nombre de combinaisons de contexte dont la similarité dépasse le seuil de pertinence, affiché en abscisse. Trois courbes de distribution sont comparées : la première représente les contextes dont les tags sont issus de SemanticProxy, la deuxième représente les contextes dont les tags sont issus de Delicious, et la troisième représente les contextes agrégeant ces deux sources de tags.

Nous observons que le nombre de contextes similaires issus du capteur virtuel SemanticProxy est inférieur à ceux issus du capteur social Delicious, alors que les tags ont été extraits à partir de plus d'URLs grâce à SemanticProxy que Delicious. De plus, 31% des similarités induites par des tags issus de Delicious ont un score de pertinence supérieur à 0,1 (10%), contre 5% seulement pour SemanticProxy. Cet avantage de Delicious s'explique par une grande popularité de tags descriptifs plus généraux, alors que SemanticProxy tire des tags plus spécifiques à partir du contenu, et donc plus rares. L'agrégation de tags issus de capteurs virtuels et sociaux implique un score de pertinence supérieur à 0,1 dans 8% des cas, mais globalement inférieur à ceux obtenus avec Delicious utilisé seul.

Alors que notre capteur social basé sur Delicious augmente les scores de pertinence contextuelle entre utilisateurs par rapport aux capteur virtuels, leur agrégation est nécessaire pour garantir une couverture suffisante des pages web ouvertes par les utilisateurs, ainsi que pour inclure des tags plus spécifiques permettant un granularité plus fine des recommandations.

G. Conclusions

Dans cette section, nous avons évalué l'application de notre cadre conceptuel de filtrage contextuel de messages sociaux, dans un environnement d'entreprise informatisée. Afin de valider notre hypothèse, nous avons réalisé une expérimentation et observé les résultats suivants :

- Nous obtenons une performance de 72% entre les scores prévus par la fonction de recommandation contextuelle de messages sociaux, et les notes données par les participants, ce qui est significatif pour un système de recommandation.
- En revanche, les participants ont rapporté un lien de pertinence moyen (50%) entre leurs messages sociaux et leurs propres contextes correspondants. Plus faible que prévu, ce résultat est dû au fait que seulement 41% des messages sociaux issus de Twitter sont en rapport avec l'activité courante de son auteur, sachant que ce type de message représente 54% des messages produits pendant la période d'expérimentation.
- Concernant l'efficacité des capteurs sociaux par rapport aux capteurs virtuels plus classiques (basés sur le contenu), nous avons apprécié la capacité des tags émergeant sur Delicious à trouver plus de similarités entre contextes. Notre approche présente un judicieux compromis entre cette capacité, la granularité des tags issus des capteurs virtuels, et la couverture des capteurs virtuels exploitant le contenu (ex. SemanticProxy).

VIII. CONCLUSION

Dans cette thèse, nous avons proposé une approche de filtrage de messages sociaux (ex. status Twitter, notifications Delicious) utilisant la similarité contextuelle entre utilisateurs comme critère de pertinence. Pour cela, nous avons mené un état de l'art des systèmes de réseaux sociaux, des systèmes de filtrage et de recommandation, et des systèmes de gestion de contexte. Afin de valider notre hypothèse en répondant aux limites identifiées dans les travaux précédents, nous avons spécifié un cadre de gestion de données contextuelles et une méthode d'évaluation de pertinence dans lesquelles le contexte est partiellement extrait de sources de données collaboratives, et manipulé visuellement par l'utilisateur. Nous avons alors appliqué ce cadre en implémentant un système de recommandation contextuelle de messages sociaux pour grandes entreprises, où le contexte est extrait à partir de la navigation web des utilisateurs, et évalué notre approche grâce à ce système.

Notre état de l'art et les résultats de notre évaluation prouvent l'utilité de notre approche :

- Les données de contexte peuvent émerger par des pratiques d'annotation collaborative sur Internet, considérant alors l'humain comme un capteur, à partir du moment où il contribue en masse suffisante.
- Considérés comme données caractéristiques dans notre système de filtrage collaboratif via le contenu, les tags extraits de capteurs sociaux offrent des opportunités de collaboration élevées et pertinentes, grâce à la bonne représentativité des annotations humaines fournies en langage naturel. Cependant, il est nécessaire d'agréger des tags venant d'autres capteurs, sachant qu'encore trop peu de ressources sont annotées de cette manière sur Delicious.
- La recommandation contextuelle de messages sociaux est prometteuse (72% de performance), à condition que ces messages soient en rapport avec le contexte actuel de

leurs auteurs, ce qui est actuellement vrai pour seulement 41% des messages échangés sur Twitter.

Les contributions suivantes sont proposées :

- Un large état de l'art, notamment une étude d'usage des systèmes de réseaux sociaux et de micro-blogging, et une analyse critique sur les systèmes de gestion de contexte.
- Les résultats d'un sondage concernant l'usage et les besoins d'utilisateurs de services de micro-blogging, et une analyse de l'impact cognitif des notifications instantanées induites.
- Un système de gestion de contexte basé sur des tags au lieu d'ontologies.
- Le modèle sous-jacent de données de contexte, ainsi que le modèle de pertinence utilisé pour piloter la recommandation de messages sociaux.
- Six capteurs (cinq virtuels et un social) permettant d'extraire des données de contexte à partir de la navigation web de l'utilisateur.
- Une application sociale pour grandes entreprises, permettant de promouvoir des opportunités nouvelles de collaboration et de communication de manière productive, par l'analyse d'activités informatisées.
- Et une méthode, ainsi que les outils informatiques associés, pour évaluer les systèmes de recommandation basés sur des données sociales.

IX. Pistes de recherche

Au cours de nos recherches, nous avons identifié les axes d'amélioration suivants :

- Afin d'augmenter les chances de recommander des messages pertinents, la qualité des nuages de tags contextuels doit être améliorée. Par exemple, des techniques de réduction dimensionnelle de vecteurs pourraient être employées pour éviter les tags sémantiquement proches ou redondants. Par ailleurs, il serait intéressant d'étudier plus profondément l'impact du temps sur le contexte, ex. la rémanence de tags précédents par décroissance progressive de poids.
- Sachant que seule la dimension de contexte d'activité a été évaluée dans cette thèse, la méthodologie et les outils correspondants pourraient être exploités pour explorer la recommandation de messages sociaux basée sur des données de contexte plus larges, combinant physique et social (ex. localisation, messages sociaux environnants).
- Afin de simplifier le contrôle de leurs contextes aux utilisateurs, de nouvelles techniques de visualisation, d'exploration et de manipulation des nuages de tags contextuels sont à développer.

Références

[1] B. N. Schilit, N. Adams, and R. Want, "Context-Aware computing applications," in *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, USA, 1994.

[2] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, 1991.

[3] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J. C. Burgelma, "Scenarios for ambient intelligence in 2010 (ISTAG 2001 final report)," ISTAG, Tech. Rep., 2001.

[4] E. Mynatt, A. Adler, M. Ito, and V. O'Day, "Design for network communities," in *CHI '97 : Proceedings of the SIGCHI conference on Human factors in computing systems*. Atlanta, Georgia, United States : ACM, 1997, pp. 217, 210.

[5] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of Context-Aware applications," *Human-Computer Interaction*, vol. 16, no. 2, 3 & 4, pp. 97–166, 2001.

[6] N. B. Ellison, C. Steinfield, and C. Lampe, "The benefits of facebook friends : Social capital and college students' use of online social network sites," *Journal of Computer-Mediated Communication*, vol. 12, no. 4, 2007.

[7] I. Erickson, "The translucence of twitter," *EPIC 2008, Ethnographic Praxis in Industry Conference*, p. 58, 2008.

[8] E. Mischaud, "Twitter : Expressions of the whole self," MSc Dissertation, London School of Economics and Political Science, 2007.

[9] M. Naaman, J. Boase, and C. Lai, "Is it really about me ? : message content in social awareness streams," in *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. Savannah, Georgia, USA : ACM, 2010, pp. 189–192.

[10] M. R. Morris, J. Teevan, and K. Panovich, "What do people ask their social networks, and why ? : a survey study of status message q&a behavior," in *Proceedings of the 28th international conference on Human factors in computing systems*. Atlanta, Georgia, USA : ACM, 2010, pp. 1739–1748.

[11] L. Barkhuus and A. K. Dey, "Is context-aware computing taking control away from the user ? three levels of interactivity examined," *UbiComp 2003 : Ubiquitous Computing*, vol. 2864/2003, pp. 149–156, 2003.

[12] R. I. M. Dunbar, "Co-evolution of neocortex size, group size and language in humans," *Behavioral and brain sciences*, vol. 16, no. 4, pp. 681–735, 1993.

[13] C. Roda, L. Ach, B. Morel, T. Nabeth, A. A. Angehrn, P. Rudman, M. Zajicek, D. Kingma, I. Molenaar, and T. Vanhala, "AtGentive deliverable d1.2, state of the art report," IST AtGentive, Tech. Rep. D1.2, May 2006.

[14] Y. Miyata and D. A. Norman, "Psychological issues in support of multiple activities," *User centered system design*, pp. 265–284, 1986.

[15] S. Iqbal and E. Horvitz, "Notifications and awareness : a field study of alert usage and preferences," in *CSCW '10 : Proceedings of the 2010 ACM conference on Computer supported cooperative work*. Savannah, Georgia, USA : ACM, 2010, pp. 30, 27.

[16] P. Dourish and V. Bellotti, "Awareness and coordination in shared workspaces," *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pp. 107 – 114, 1992.

[17] M. Balabanovic, "An adaptive web page recommendation service," *Proceedings of the first International Conference on Autonomous Agents*, 1997.

[18] J. Budzik, X. Fu, and K. J. Hammond, "Facilitating opportunistic communication by tracking the documents people use," in *CSCW 2000 Workshop on Awareness and the WWW. Retrieved January*, vol. 30, 2000, p. 2005.

[19] A. N. Dragunov, T. G. Dietterich, K. Johnsrude, M. McLaughlin, L. Li, and J. L. Herlocker, "TaskTracer : a desktop environment to support multi-tasking knowledge workers," in *Proceedings of the 10th international conference on Intelligent user interfaces*. San Diego, California, USA : ACM, 2005, pp. 75–82.

[20] T. Bauer and D. B. Leake, "Real time user context modeling for information retrieval agents," in *Proceedings of the tenth international conference on Information and knowledge management*. Atlanta, Georgia, USA : ACM, 2001, pp. 568–570.

[21] M. V. Kleek, D. R. Karger, and mc schraefel, "Watching through the web : Building personal activity and Context-Aware interfaces using web activity streams," in *Proceedings of the Workshop on Understanding the User - Logging and Interpreting User Interactions in Information Search and Retrieval (UIIR-2009)*, Boston, USA, 2009.

[22] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[23] L. Agosto, "Optimisation d'un réseau social d'échange d'Information par recommandation de mise en relation," Ph.D. dissertation, Université de Savoie, 2005.

[24] C. Delalonde, "Mise en relation et coopération dans les equipes distribuées de R&D. l'application de l'Actor-Network theory dans la recherche de connaissances," Thèse de Doctorat, Université de Technologie de Troyes, 2007.

[25] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme, "Information retrieval in folksonomies : Search and ranking," in *The Semantic Web : Research and Applications*. Springer, 2006, pp. 411–426.

[26] S. Niwa, T. Doi, and S. Honiden, "Web page recommender system based on folksonomy mining for itng'06 submissions," in *Information Technology : New Generations, 2006. ITNG 2006. Third International Conference on*, 2006, pp. 388–393.

[27] K. Bielenberg and M. Zacher, "Groups in social software : Utilizing tagging to integrate individual contexts for social navigation," Digital Media. Bremen, Germeny, University Bremen. Master of Science in Digital Media, Tech. Rep., 2005.

[28] M. J. Pazzani, "A framework for collaborative, Content-Based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, no. 5-6, pp. 393–408, 1999.

[29] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[30] R. Sinha, "A cognitive analysis of tagging," 2005. [Online]. Available : http ://rashmisinha.com/2005/09/27/a-cognitive-analysis-of-tagging/

[31] M. Tesconi, F. Ronzano, A. Marchetti, and S. Minutoli, "Semantify del. icio. us : automatically turn your tags into senses," in *Social Data on the Web, workshop at the 7th International Semantic Web Conference*, Karlsruhe, Germany, 2008.

[32] K. Ciesielski, M. A. Klopotek, and S. T. Wierzchon, "Histogram-Based dimensionality reduction of term vector space," in *Proceedings of the 6th International Conference on Computer Information Systems and Industrial Management Applications*. IEEE Computer Society, 2007, pp. 103–108.

[33] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. Berkeley, California, United States : ACM, 1999, pp. 50–57.

[34] G. Tsatsaronis and V. Panagiotopoulou, "A generalized vector space model for text retrieval based on semantic relatedness," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics : Student Research Workshop*. Athens, Greece : Association for Computational Linguistics, 2009, pp. 70–78.

[35] R. Wetzker, W. Umbrath, and A. Said, "A hybrid approach to item recommendation in folksonomies," in *Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*. Barcelona, Spain : ACM, 2009, pp. 25–29.

[36] A. K. Dey and G. D. Abowd, "The context toolkit : Aiding the development of Context-Aware applications," *Workshop on Software Engineering for Wearable and Pervasive Computing*, 2000.

[37] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, "An ontology-based context model in intelligent environments," *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, vol. 2004, 2004.

[38] T. Strang and C. Linnhoff-Popien, "A context modeling survey," *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp*, pp. 34–41, 2004.

[39] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

[40] T. Strang, C. Linnhoff-Popien, and K. Frank, "CoOL : a context ontology language to enable contextual interoperability," *Distributed Applications and Interoperable Systems : 4th Ifip Wg6. 1 International Conference, Dais 2003, Paris, France, November 17-21, 2003, Proceedings*, 2003.

[41] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakraborty, "Intelligent agents meet the semantic web in smart spaces," *IEEE Internet Computing*, vol. 8, no. 6, pp. 69–79, 2004.

[42] F. Perich, S. Avancha, D. Chakraborty, A. Joshi, and Y. Yesha, *Profile Driven Data Management for Pervasive Environments*. Springer, 2005.

[43] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell, "Reasoning about uncertain contexts in pervasive computing environments," *Pervasive Computing, IEEE*, vol. 3, no. 2, pp. 62–70, 2004.

[44] SPICE, "SPICE d4.1 : Ontology definition of user profiles, knowledge information and services," 2006. [Online]. Available : http ://www.ist-spice.org/documents/D4.1-final.pdf

[45] X. Wang, D. Zhang, T. Gu, and H. Pung, "Ontology based context modeling and reasoning using OWL," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, 2004, pp. 18–22.

[46] J. G. Tan, D. Zhang, X. Wang, and H. S. Cheng, "Enhancing semantic spaces with Event-Driven context interpretation," *Pervasive Computing : Third International Conference, Pervasive 2005, Munich, Germany, May 8-13, 2005, Proceedings*, 2005.

[47] D. Ejigu, M. Scuturici, and L. Brunie, "An Ontology-Based approach to context modeling and reasoning in pervasive computing," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, 2007, pp. 14–19.

[48] X. Lin, S. Li, J. Xu, W. Shi, and Q. Gao, "An efficient context modeling and reasoning system in pervasive environment : Using absolute and relative context filtering technology," 2005.

[49] P. Dourish, *Where the action is : the foundations of embodied interaction*. The MIT Press, 2004.

[50] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca, "A data-oriented survey of context models," *SIGMOD Rec.*, vol. 36, no. 4, pp. 19–26, 2007.

[51] J. Koolwaaij, A. Tarlano, M. Luther, P. Nurmi, B. Mrohs, A. Battestini, and R. Vaidya, "Context Watcher-Sharing context information in everyday life," in *Proceedings of the IASTED conference on Web Technologies, Applications and Services (WTAS). IASTED*, Calgary, Canada, Jul. 2006.

[52] J. Carter and P. Dewan, "Automatically identifying that distributed programmers are stuck," in *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*. IEEE Computer Society, 2009, p. 12.

[53] O. Ertzscheid, "Indexation sociale et folksonomies : le monde comme catalogue," Montpellier, France, May 2008. [Online]. Available : http ://www.slideshare.net/olivier/oe-abes-mai2008

[54] S. Golder and B. A. Huberman, "The structure of collaborative tagging systems," *Arxiv preprint cs.DL/0508082*, 2005.

[55] T. Rattenbury, N. Good, and M. Naaman, "Towards automatic extraction of event and place semantics from flickr tags," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. Amsterdam, The Netherlands : ACM, 2007, pp. 103–110.

[56] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, "KEA : practical automatic keyphrase extraction," in *Proceedings of the fourth ACM conference on Digital libraries*. ACM, 1999, p. 255.

[57] L. Terveen and D. W. McDonald, "Social matching : A framework and research agenda," *ACM Trans. Comput.-Hum. Interact.*, vol. 12, no. 3, pp. 401–434, 2005.

Contents

List of Figures **ix**

List of Tables **xi**

Glossary **xiii**

1 Introduction and motivation **1**

1.1 Introduction 1

1.2 Definitions, Context and Motivation 2

1.2.1 What is relevance, and how to measure it? 2

1.2.2 What is context, and how can it help? 3

1.2.3 How to gather necessary contextual information? 3

1.2.4 Vision: an ambient awareness scenario 4

1.3 Towards Ambient Awareness 6

1.3.1 Ubiquitous Computing 7

1.3.2 Ambient Intelligence 7

1.3.3 Emergence of crowd-sourced information on the Web 8

1.4 Dissertation plan 9

2 Social Networking Systems and their cognitive impact **13**

2.1 Social Networking Systems 14

2.1.1 Meaning of relationships 14

2.1.2 User profile enhancement 16

2.1.3 New social interaction modalities 18

2.1.4 Feeds of social updates 20

2.1.5 Pointless babble or awareness? 22

2.1.6 Content of status updates 23
2.1.7 Different types of social updates 24
2.1.8 Many opportunities to communicate 24
2.1.9 Mobile Social Networking 26
2.1.10 What about privacy? 27
2.1.11 Some usage statistics 27
2.1.12 Conclusion 29
2.2 Usage and impact of real-time microblogging 29
2.2.1 Results of the survey 29
2.2.2 Filtering mechanisms: challenges, solutions and opportunities . . 35
2.2.3 Discussion 36
2.3 Social notifications: cognitive impact and opportunities for improvement 36
2.4 Conclusion 38

3 From Awareness Systems to Information management techniques 39
3.1 Awareness and social matching systems 40
3.1.1 Definitions 40
3.1.2 Foundations and some applications 40
3.2 Recommender systems and information filtering techniques 43
3.2.1 Content-based filtering 43
3.2.2 Social matching and collaborative filtering 44
3.2.3 Discussion 45
3.3 Content models and feature extraction techniques 46
3.3.1 Keyword/phrase and tag-based models 46
3.3.2 Semantic extraction 47
3.3.3 Discussion 48
3.4 Improving performance and scalability 49
3.4.1 Dimensionality reduction of term vector space 49
3.4.2 Tag clustering techniques 49
3.5 Conclusion 50

4 Extracting and modeling contextual information **51**

4.1 What is context-awareness? . . . 53

4.2 Context management systems . . . 54

4.2.1 From context-aware applications to context management systems 54

4.2.2 Semantic context management systems . . . 55

4.2.3 Uncertainty and quality of context . . . 59

4.2.4 Popularization of context-aware applications . . . 63

4.2.5 Privacy concerns . . . 64

4.2.6 A problem of scalability . . . 65

4.2.7 Discussion . . . 66

4.3 Physical sensors: Extracting and modeling geographical contexts . . . 67

4.3.1 Absolute geographic location . . . 67

4.3.2 Beacon-based location . . . 68

4.3.3 Movement-based relative coordinates . . . 73

4.3.4 Political regions and logical spaces . . . 73

4.4 Virtual sensors: context from computer-based activities . . . 74

4.5 Social sensors: context from personal streams and crowd-sourced data . 75

4.5.1 Human messages as sensor data . . . 75

4.5.2 Web repositories of user-generated content . . . 76

4.5.3 Extracting context by mashing up web streams and services . . . 78

4.6 Conclusion . . . 79

5 A Context Management Framework for Social Awareness **81**

5.1 Aims . . . 81

5.2 Problem definition and scope . . . 82

5.2.1 Hypothesis . . . 82

5.2.2 Constraints and choices . . . 83

5.2.3 Approach: a tag-based context model . . . 84

5.2.4 Methodology . . . 85

5.3 Interaction model . . . 85

5.3.1 General interaction flow . . . 85

5.3.2 Computer-based interaction . . . 86

5.3.3 Nomadic interaction . . . 86

5.4 Contextual model 87
5.4.1 Contextual dimensions and information sources 87
5.4.2 Formalization: a tag-based contextual algebra 90
5.5 Social matching scheme 92
5.5.1 The temporal dimension of social matching 92
5.5.2 General relevance function 94
5.6 Design and implementation of the framework 95
5.6.1 Data flow 95
5.6.2 Main structures 96
5.6.3 Using the framework for real-time matching 100
5.6.4 Using the framework to generate batch matching reports from logs 103
5.7 Conclusion 108

6 A Social Awareness Application and its evaluation 109
6.1 Case study 109
6.2 Application of the approach 110
6.2.1 Enterprise context model 110
6.2.2 Web-browsing-based context sensors 110
6.2.3 A comparison of tags gathered from virtual and social sensors . . 112
6.2.4 Software implementation 114
6.3 Context-based relevance of social updates, an evaluation 118
6.3.1 Evaluation requirements 118
6.3.2 Experimentation plan 118
6.3.3 Experimental setup and recommendation process 120
6.3.4 Results 121
6.3.5 Discussion 125
6.4 Matching contexts from virtual and social sensor tags, a comparative analysis 126
6.4.1 Description of the data 126
6.4.2 Analysis of results 127
6.4.3 Discussion 131
6.5 Conclusion 132

7 Conclusion **135**

7.1 Research summary 135

7.1.1 Research context 135

7.1.2 Design, application and evaluation of a context management framework for social awareness 137

7.2 Contributions 138

7.3 Findings 139

7.4 Recommendations and best practices 139

7.5 Opportunities for improvement 141

References **143**

A Introduction to ontologies and the Semantic Web **153**

B Samples of personalized surveys **155**

C Survey on usage of real-time microblogging **157**

List of Figures

1.1 Vision: an ambient awareness scenario 5
1.2 From first modems to ambient awareness systems 8

2.1 Social interactions in a Social Networking Site 15
2.2 Typical user profile in Social Networking Sites 16
2.3 The "Movies" application on Facebook 17
2.4 An opportunistic contact on Twitter 21
2.5 Two sample social updates 24
2.6 Simplified model of SNS communication 25
2.7 Distribution of respondents among number of followed feeds 30
2.8 Uses of microblogging 30
2.9 Usefulness of microblog updates 31
2.10 Notification modalities for real-time updates 32
2.11 Attention given to notified updates 32
2.12 Probability of consulting content attached to updates 33
2.13 Use of filtering 34
2.14 Filtering criteria most expected from respondents 34

4.1 Contextual information: from raw sensor data to ambient intelligence . 52
4.2 Overview of CoBrA 56
4.3 The SOUPA ontology 57
4.4 Partial definition of the CONON ontology extended with the home domain 60
4.5 Creating a rule-based service using SPICE's End User Studio 62
4.6 HCoM: Hybrid Context Management and reasoning system 65
4.7 Beacon-based positioning 69

LIST OF FIGURES

4.8 Comparison of three GSM-based positioning methods 70

5.1 A sample contextual tag cloud 84
5.2 Interaction flow 86
5.3 Different matching modalities 93
5.4 Overview of the contextual recommendation loop 95
5.5 Hierarchy of classes encapsulating contextual clouds 97
5.6 Hierarchy of classes encapsulating contextualized events 99
5.7 Hierarchy of classes encapsulating recommendations 100
5.8 Batch sequence, from logs to reports 103

6.1 A sample contextual tag cloud, in an enterprise environment 110
6.2 Tag clouds extracted using four methods 113
6.3 Flow of the contextual aggregation and recommendation process 114
6.4 Screenshot of the *side-bar* user interface in Firefox 117
6.5 Experimentation plan 119
6.6 Distribution of ratings on contextual clouds 122
6.7 Average precision values for each participant 123
6.8 Comparative distribution of recommendation ratings 124
6.9 Comparative distribution of relevance ratings of social updates 125
6.10 Tags from SemanticProxy: Cumulative distribution of context matches . 128
6.11 Tags from Delicious: Cumulative distribution of context matches 129
6.12 Tags from all sensors: Cumulative distribution of context matches . . . 129
6.13 Number of matching contexts using only tags from SemanticProxy . . . 130
6.14 Number of matching contexts using only tags from Delicious 130
6.15 Number of matching contexts using tags from all sensors 131

B.1 A social update from a participant's personalized survey 155
B.2 Another social update from a participant's personalized survey 155
B.3 A sample context from a participant's personalized survey 156

List of Tables

2.1 Typical interaction modalities on Social Networking Sites 19

6.1 Types of browsing events handled from Firefox 115
6.2 Specification of parameters for events sent by the Firefox sniffer to the Context Aggregator 116

Glossary

API Application Programming Interface, an external interface exposing some methods of a program, in order for other programs to invoke those methods

Context Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. (Dey, 2000)

Context-Awareness The ability, for software components, to adapt to contextual information

HTTP Hypertext Transfer Protocol, the protocol that is commonly used to request resources on the web (e.g. pages), and to invoke web services

Interface A specified set of method prototypes that can be implemented by a program for allowing external invocation of those methods

JSON JavaScript Object Notation, a lightweight and hierarchical data-interchange format

Microblogging A social networking system in which people regularly share concise status updates to their community

Physical sensor A device that can extract contextual information from its physical environment

REST Representational State Transfer, an basic abstraction of HTTP that allows one to invoke actions on identified state-less resources, usually returning outputs in standard markup languages such as XML or JSON

RESTful A RESTful web service is a simple web service implemented using HTTP and the principles of REST

RSS feed A dynamic list of recent updates that can be requested regularly from a web server

Sensor A device (or software component) that can extract contextual information from its environment

Social bookmark A link to a resource (e.g. web page) that is intentionally stored, and possibly shared, by an identified individual on a social bookmarking system, on which individuals can attach tags

Social networking system (SNS) A software-based system in which people are represented by a personal profile, and can communicate by emitting social updates to their community

Social sensor A mechanism to interpret/enrich contextual information using crowd-sourced information

Social update A piece of content that is shared on a social network by an identified individual, and that can be viewed, notified, relayed, or commented by other people of the network

Status update A social update which consists of a short text about (or related to) its author

Tag A descriptive keyword entered by a human individual

Tag cloud A weighted set of tags which can be represented as a bag of words of different sizes, according to their respective weights

Virtual sensor A software component that can extract contextual information from other software components

Web services A script that can be invoked by requesting a web server using HTTP, can possibly return some output, given some parameters

Wisdom of Crowds Data and knowledge emerging from collective contributions, on collaborative spaces

Chapter 1

Introduction and motivation

1.1 Introduction

On Social Networking Sites (such as Facebook[1], Twitter[2], LinkedIn[3]) and other collaboration software, people maintain and create new social ties by sharing personal (but not necessarily private) *social updates* (e.g. *status updates*) regularly to their community. A *social update* is a short message sent to a group of people (e.g. a community of colleagues, friends, *followers*). It can consist of a single sentence (a news, a question), include a picture, a comment on a picture, a hyperlink, to share their current thoughts, activities, intentions and needs.

Contrarily to emails, social updates are informal, not aimed at a specific list of recipients, and are not expected to be acknowledged. Instead, community members can go through the list of short social updates of the people or topics (e.g. *hashtags* on twitter) they follow, to get a quick feeling of *awareness* about people and subjects they care about. However, as the number of people and subjects being followed increases, the time required to get through to the social updates they emit also increases, causing a loss of productivity. Additionally, as social updates are broadcast (and thus, potentially consumed) in real-time, they can create frequent interruptions that can reduce people's ability to focus on a demanding task, especially when the social update is not relevant for this task (because it would induce a costly cognitive disruption).

[1] http://www.facebook.com/
[2] http://www.twitter.com/
[3] http://www.linkedin.com/

In response to this emerging problem of *information overload*, we model, develop and evaluate a context management framework for ranking *social updates* by relevance according to real-time similarities in-between users' contexts. We also demonstrate how information emerging on crowd-sourced web sites can be leveraged as '*social sensors*' to generate and enhance underlying contextual information.

1.2 Definitions, Context and Motivation

1.2.1 What is relevance, and how to measure it?

According to Hjrland & Christensen (2002), '*something (A) is relevant to a task (T) if it increases the likelihood of accomplishing the goal (G), which is implied by T.*'

As humans, we perceive (and receive) a lot of information, either when communicating or not. As our senses make us aware of our environment, we are surrounded by signals. How many of those signals are '*relevant*' to our current task? Few, as most signals do not increase the likelihood of accomplishing the goal aimed by this task. However, several signals can help to achieve other goals. E.g. if you have to finish the redaction of a report before noon, the coffee machine that you hear, the photos that you see on your desk, and the fan which cools you down are not relevant for your task, by they are part of your context, and can help you achieve other goals: having a break to socialize with your colleagues, remembering to pick up your children from school on time, and prevent being sick by wearing warmer clothes.

The problem is to allow a certain degree and scope of awareness, for ensuring a sufficient productivity on ongoing task(s), while keeping up with other goals. A part of this problem is handled naturally: cognitive processes drive attention (i.e. in-depth mental processing) selectively on relevant signals perceived through our senses (James *et al.*, 1981; Posner, 1980), but one's surrounding environment can still produce disruptions that interrupt him in focusing on ongoing tasks, and thus reducing his productivity. Moreover, in the frame of a computer-supported interaction, additional signals are brought in enormous quantities (e.g. notifications from instant messaging programs, incoming electronic mail, and real-time social updates), imply even more opportunities of being interrupted in our tasks. In that frame, computational support is to be provided to help users cope with information overload. A relevance function is thus to be defined and executed to filter abusive interruptions caused by irrelevant

electronic signals (e.g. chats, emails and social updates that are not useful/important for carrying on the current task).

1.2.2 What is context, and how can it help?

'*Context*' was defined by Dey (2000) as '*information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*'. Based on this definition, Dey also defines '*context-awareness*' systems: '*A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.*'.

From these definitions, we identify that contextual information can be used as a criterion for evaluating the relevance of information (and more specifically in our case, social updates) on the user's task. We support the hypothesis that: by sharing clues about his current context, one can implicitly solicit relevant information and communication, without having to formalize a request (e.g. make a phone call, send an email, ask a specific person). In the aim of supporting one user in being notified of the most relevant signals, the challenges are to:

- identify what is his/her task,
- determine which contextual entities are relevant to the execution of this task,
- and how to evaluate the relevance of social updates for that user, knowing his/her current task and context, and other users'.

1.2.3 How to gather necessary contextual information?

In most context-aware applications, researchers have been relying either on physical sensors to extract contextual information, or on specific software sensors depending on the kind of computer-based information being manipulated by the user. In both cases, the nature and the representation/format of contextual cues was determined specifically for each sensor, and for each context-aware application. E.g. an application requiring temperature information would necessarily have to rely on a sensor that samples temperature information from the environment. In order to ensure interoperability

between heterogeneous context producers (i.e. sensors) and consumers (e.g. context-aware applications), common and extensible representation mark-ups were proposed by researchers, e.g. ontologies.

As '*tagging*' becomes a common practice on the Internet, rich contextual information can also emerge from human-generated content. Indeed, descriptive tags have been collaboratively given by people to many kinds of resources and entities: social bookmarking sites, reviews of places, events, people etc... Thus, such tags associated to documents that a user is manipulating, and to his surrounding entities (e.g. places, people, events...), can represent meaningful information about his current context. However, contextual information emerging in such ways is fuzzier (in terms of certainty and ambiguity), more error-prone and harder to classify than information sampled by sensors. Indeed, most human-generated content is published on the web in natural language, in various locales, with implicit semantics and a certain amount of misspellings, subjective and wrong information. It is possible to classify such information using ontologies, by inferring semantics from large amounts of content. However, we support that user-generated content (and social updates in particular) often contain features (e.g. words) that are not yet described in ontologies, and would sometimes take longer to describe than the time during which this feature is being used. This phenomenon is particularly true on real-time social networks, in which users create and relay trending words (also called '*buzz-words)*' that have a very contextual and temporary meaning. Such features are very important for real-time awareness, and users cannot wait until a semantic description has emerged (i.e. in an ontology) before being made aware of them.

For those reasons, we developed a novel context management scheme based on tags, instead of traditional ontologies.

1.2.4 Vision: an ambient awareness scenario

In this section, we present an usage scenario to illustrate our vision of ambient awareness, relying on the contribution of the thesis.

As shown on Figure 1.1, Christine's context is synthesized and represented as a tag cloud containing:

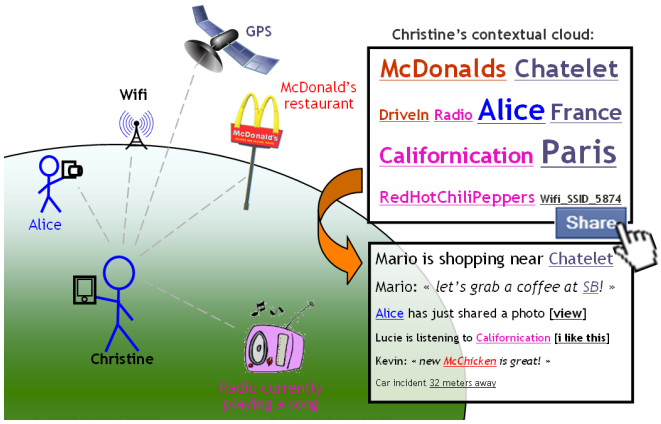

Figure 1.1: Vision: an ambient awareness scenario -

- *Paris*, *Chatelet* and *France*, which qualifies her current location (respectively, the name of the city, the place and the country), as inferred from GPS and Wi-Fi signals (including its identifier $Wifi_SSID_5874$, also presented as a tag) received by her cell phone.

- *McDonalds* qualifies a near-by restaurant that could be identified by a wireless signal (e.g. Wi-Fi hotspot), or a geo-coded directory/database. *DriveIn* is also included in the context, as a capability of this restaurant.

- One of Christine's friend, *Alice*, is part of the tag cloud because she was detected in short range.

- A *radio* has also been detected as currently playing the song *Californication* by the *RedHotChiliPeppers*. This information can be emitted wirelessly by the radio itself, or published through a web service in real-time.

In this illustrative tag cloud, the size of tags is arbitrary. However, the size of a tag is to be computed from:

- the quality of this tag: the degree of confidence concerning the relevance of this tag in the current context,

- the quantity of this tag: the cumulative weights of this tag as it is extracted from several contextual and social sources.

As Christine observes her contextual tag cloud on the screen of her mobile phone, she decides to share it with her friends, so that they can know what she is doing. One second later, the screen displays a list of social updates that are relevant to her current context:

- She discovers that $Mario$, a friend of hers, is also $currentlyshoppingnearChatelet$, and proposes to $grabacoffee$ near-by.
- *Alice*, who is also near-by, has just shared a photo.
- Another friend, *Lucie*, is currently listening to the same song.
- $Kevin$ posted a status message to recommend a burger he just ate at $McDonalds$.
- And, she is notified that a car incident happened a $fewmetersaway$ from her.

This scenario illustrates the benefits of ambient awareness, an implicit, pervasive and quick mean of communication and relevant information, relying on contextual information and social networking systems.

1.3 Towards Ambient Awareness

The contributions of this thesis are positioned in the overlap of several research domains: context-awareness (as a branch of ambient intelligence), knowledge management, and information filtering. Before reviewing the corresponding research background in the following sections, we propose an introduction to general concepts and visions, such as ubiquitous computing, ambient intelligence and social systems, to explain the context of our contribution.

Up to our knowledge, Mynatt *et al.* (1997) from Xerox PARC and Stanford University were the first researchers to use the phrase '*Ambient Awareness*', as we consider it in this thesis. In their paper '*Design for network communities*', they present peripheral and ambient awareness as natural kinds of social interaction that must be taken in account in network communities. Indeed, ambient awareness is the human capability to sense events occurring, moods of people and subjects being discussed in the environment. A person can stay aware without having to actually be involved in a mutual conversation with anyone else, just observation of ambient signals is needed.

In this section, we study the translation of ambient awareness from a natural human environment to computer-supported '*smart*' environments. This translation is empowered by the visions and technologies of ubiquitous computing, ambient intelligence and the social web.

1.3.1 Ubiquitous Computing

Proposed by Weiser (1991) from Xerox PARC, the vision of ubiquitous computing consists of accessing and manipulating digital information through several specific objects, instead of having to rely on computers with keyboards, mice and screens. In his foundational paper '*The Computer for the 21st century*', he promotes a few mobile collaboration devices being used opportunistically at PARC, communicating through wireless networks. The early implementation of this vision demonstrated that high distribution of simple chips embedded in objects is more useful than one omnipotent computer, and successfully predicted the current ubiquity of wireless networks and autonomous chips and sensors embedded in contemporary devices and appliances.

Whereas Weiser's vision has become true with the emergence of wireless networks, smartphones, and other autonomous internet-connected objects, his dream about interoperability between these objects has not been completely reached yet.

1.3.2 Ambient Intelligence

Based on Weiser's vision, '*ambient intelligence*' was firstly coined on 1998 by Eli Zelkha and Brian Epstein from Palo Alto Ventures, as the title for a series of workshops commissioned by Philips towards more fragmented entertainment solutions, especially intended for people's living rooms. Compared to Weiser's vision, the concept of '*ambient intelligence*' was focusing on the pro-activity of a common intelligence federating people's appliances. In order to fulfill one's needs without expecting explicit human intervention, such system had to become aware of and adaptive to him (i.e. his identity and preferences) and to his environment.

Building upon ubiquitous computing, Schilit *et al.* (1994) proposed the concept of '*context awareness*', to enable sensing, recognizing and acting upon the context of ubiquitous objects (and of the users who carry them on) in the real world. In 2001, this concept was integrated to ambient intelligence, which motivated the launch of the sixth framework (FP6) in Information, Society and Technology (IST) of the European

Commission, following the advice of the Information Society and Technology Advisory Group (ISTAG, Ducatel *et al.* (2001)).

Today, ambient intelligence has become a major research domain which gave birth to numerous scientific articles, conferences and journals. Recent off-the-shelf technologies such as QR-codes (i.e. 2D bar-codes), beacon-based positioning (e.g. GPS), RFID and other NFC technologies (Want, 2008) have become practical bridges for researchers and end-users to explore the frontiers between the real world and the digital/virtual worlds.

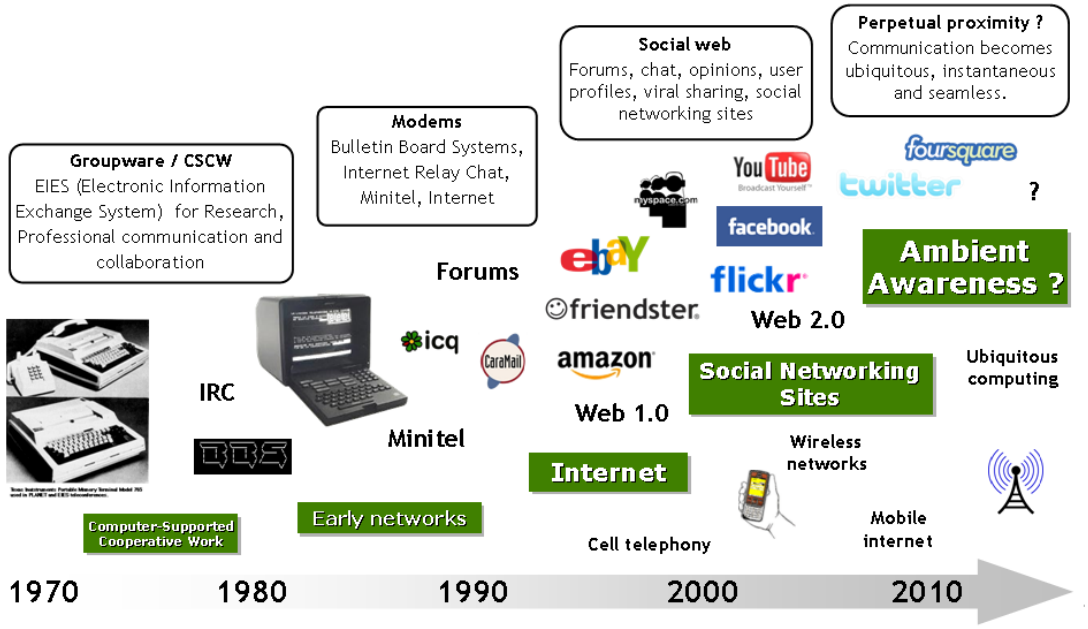

Figure 1.2: From first modems to ambient awareness systems - This timeline illustrates the history of communication technologies towards ambient awareness

1.3.3 Emergence of crowd-sourced information on the Web

Another evolution which emerged with Web 2.0 is '*crowd-sourcing*': the creation of value from Internet users, through sharing of user-generated content, and collaborative annotation of content. In particular, the '*tagging*' practice consists of annotating content by typing words that subjectively reflect the apparent nature, function, category and context of a piece of content for each annotator (Ertzscheid, 2008; Golder & Huberman, 2005). These '*tags*' can then be leveraged to categorize, index, search, and recommend content to users so that they can easily find it.

We have seen that, thanks to their numerous contributions on web platforms, Internet users (the '*crowd*') generate a growing amount of information. The quality of information provided freely by users is highly heterogeneous, and often low when taken individually. However, the enormous quantity of contributions turn humans into '*social sensors*', providing emerging information about content and entities from the real world, with their rich variety of locales and points of view.

1.4 Dissertation plan

In this last section, we depict the sequence of research activities that have been undertaken from the problem statement introduced in this first chapter, towards the contributions discussed in chapter seven.

In the first part of the thesis (chapters two to four), we have carried out a state of the art in the several domains that are relevant to address the theoretical and technical challenges towards Ambient Social Awareness applications we envisioned, and to position our contribution to existing work:

- In chapter two, we introduce web-based social networking systems, describe their common and specific characteristics and functions, and analyze their usage through the enumeration of salient trends and major evolutions that we have observed since their appearance on the Internet. All along this study, we identify functional lacks, and opportunities to potentially improve the social experience of their users. Then, we study the cognitive impacts of the social communication behaviors which emerged with these new communication tools and platforms. In particular, we further analyze how users of real-time micro-blogging services (e.g. Twitter) consume social updates, based on the results of a survey we carried out. In that chapter, we conclude that some automated support is to be provided to users for improving their communication experience while regulating negative cognitive impacts (i.e. preventing productivity loss by reducing interruptions). In particular, we identify that contextual information about users can be leveraged in that matter.

- In chapter three, we review several research efforts on improving social awareness, communication and collaboration using computer-based techniques. Whereas

some of them focus on human-computer interaction design, we identify that information filtering techniques and recommender systems can also be considered to improve and regulate information streams between people. In particular, the relevance of incoming content for a person can be evaluated against personal subjects of interests and needs, either explicit (e.g. queries on a search engine) or implicit (e.g. profiling). According to our specific approach, we review several techniques for extracting and representing information from user-manipulated content, in order to gather a dynamic user profile representing his interests during his ever-changing activities. Then, we compare relevance estimation techniques based on similarity functions, and identify existing techniques to improve the performance (and thus, reduce complexity) of relevance estimation.

- In chapter four, we explore research literature on '*context-aware*' computing, as we have identified user context as a good source of information for evaluating relevance of socially-mediated information and communication. After selecting a definition of '*context*', we review several research efforts on context-aware software and the underlying sensor infrastructures, and discuss the issues related to these applications: modeling contextual information and its uncertainty, privacy concerns for their users, and scalability problems. Three kinds of sensors are considered: (i) physical sensors transmit information that is sampled from a physical environment (e.g. geographical positioning of users and other entities), (ii) virtual sensors synthesize contextual information from computer-based activities, and (iii) social sensors which can provide emergent contextual information from crowd-sourced content by combining various web services.

In the second part of the thesis, we design, implement and evaluate a context management framework, and its ambient social awareness application:

- In chapter five, we define the main problem of the thesis, relying on the lacks and possibilities identified in the three chapters of the previous part. Firstly, we demonstrate the importance of contextual knowledge about people for evaluating the relevance of social communication signals (e.g. microblog/status updates, shared content, inter-personal messages), and the impact of relevance on induced cognitive disruption. In order to regulate the cost of interruptions potentially caused by these signals on people's attention, we propose to emphasize to each

user the signals that are the most relevant to him, according to contextual cues about his current task and environment. Following the enumeration of several constraints that are required to maximize the usability of context-based social awareness applications, we design a context information model, an algebra for aggregating contexts and evaluating contextual relevance, and the information flow of the underlying context management software.

- In chapter six, we apply our theoretical framework to develop a social awareness application for improving communication and collaboration in high-scale computer-based environments (e.g. a company, a research lab). After proposing a motivating case study, we develop a set of software mechanisms for extracting contextual information from computer-based activities, by specializing the functions of the theoretical framework (as defined in the previous chapter). Then, we describe the interaction flow of the application, and the implementation of each software module that it contains.

- In chapter seven, we analyze and compare the quality of tags gathered from virtual and social sensors. Then, we evaluate the performance of our theoretical framework (as defined in chapter five) and some parts of its software application (as defined in chapter six) for estimating the relevance of social updates in an experimental setup.

To conclude the thesis, in chapter eight, we discuss our contribution and the implied findings and limitations. Future work, recommendations and perspectives are then proposed.

Chapter 2

Social Networking Systems and their cognitive impact

After forums, blogs, instant messaging platforms and other Internet-based communication media, Social Networking Systems (SNS) have become a new communication paradigm on the Internet, enabling large friends communities to keep in touch, improving collaboration between colleagues, and facilitating new contacts.

A SNS is a virtual community in which users maintain a personal web page (called '*profile*' or '*space*') that allows members of his/her community to know more about the interests and latest activities of its owner, and to interact with him/her in many ways. The goal of each SNS is different, but in most cases, possible social interactions involve people that consented to connect as '*contacts*' or '*friends*' on the system.

In this chapter, we analyze the evolution, distinctive functionalities, and usage of social networking systems: social networking sites, microblogging systems and mobile social software. Then, we present and discuss the results of a survey we conducted about the usage of microblogging users concerning real-time notifications and their expectations concerning filtering of social feeds. Finally, we further study those observations from a cognitive psychology point of view, and identify some opportunities for improvement towards preventing information overload.

2.1 Social Networking Systems

Social networking sites are web-based platforms that rely on manually shared content and status messages for triggering communication. Amongst the most popular informal SNS, '*Facebook*' [1] helps friends keeping in touch by providing each of them with a personal and dynamic news feed that contains updates about the content they shared and the comments they wrote, as further explained in this section. The microblogging site '*Twitter*' [2] proposes a simpler approach: invite its users to regularly share concise and public status updates about their activities, current interests and opinions, for their followers.

In this section, we compare the features of several of the most popular Social Networking Systems against regular communication mediums.

2.1.1 Meaning of relationships

Like popular instant messengers (e.g. Windows Live Messenger, Yahoo! Messenger, ICQ), SNS users have to invite their contacts on the platform to enable proper communication. On one of the first social networks, classmates.com (1995), your contacts were people you have actually been in school with. But, ten years later, the well known SNS Myspace [3] (2005), which was reportedly getting more page views than Google (Rosenbush, 2005), built its popularity on a weak meaning for "friendship" connections. Indeed, on this site, many "friends" (contacts) have actually never seen each other in real life. It is also usual to see a celebrity in one's friend list. Indeed, Myspace has become the best place for teenagers to boost their social ego, and for artists (and brands) to show off their content and get closer to their fans.

With Facebook, Myspace's biggest rival, the meaning of "friend" seemed more natural, as it was originally intended for students of Harvard College to keep in touch. With time, its population grew out of Harvard to seduce students and workers around the world. But still, it seems most people that connect with each other on the site already knew each other before in real life (Ellison *et al.*, 2007). Indeed, a study (Lampe *et al.*, 2006) proved that this kind of SNS is more used for "social searching" (i.e. look up already known people) than for "social browsing" (i.e. find strangers online). The

[1] Facebook: http://www.facebook.com/
[2] Twitter: http://www.twitter.com/
[3] Myspace: http://www.myspace.com/

same trend seems to apply on the popular professional SNS LinkedIn [1], although some people gather as many professional contacts as possible to increase their visibility to potential employers and/or collaborators.

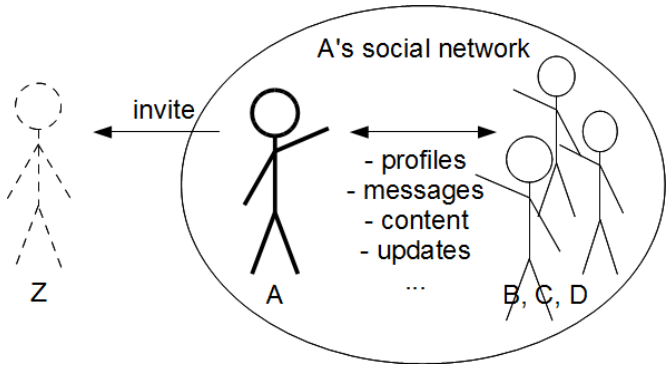

Figure 2.1: Social interactions in a Social Networking Site -

The notion of connection has a different meaning depending on the site, but the fact of being recognized as a contact is usually required to enable most kinds of social interactions provided by these sites. As depicted on Figure 2.1, a user (A) can benefit from these rich interactions with the contacts (B, C and D) of his social network, whereas the only possible action between this user (A) and a person (Z) who is not a contact of his social network is to invite him/her by email to join his social network. These interactions include: personalize one's own profile, writing a public message to a contact's profile, commenting publicly a contact's photos, being notified of what actions were undertaken by one's contacts on the site... Here, the "public" visibility is often restricted to the recipient's contacts, but this constraint is not common to all SNS.

In Twitter, which we will further describe below, the contacts are called "followers" who chose to be notified of one's last social updates in real-time, even though anyone can actually read them. Whereas connection as "contact" or "friend" requires a mutual approval on most SNS, twitter does not.

We have seen that SNS rely on connections between contacts, but the meaning of these relationships differ from one SNS to another. In every case, however, a connection is to be declared explicitly, by consulting a user profile, searching for a person or responding to an invitation.

[1] LinkedIn: http://www.linkedin.com/

2.1.2 User profile enhancement

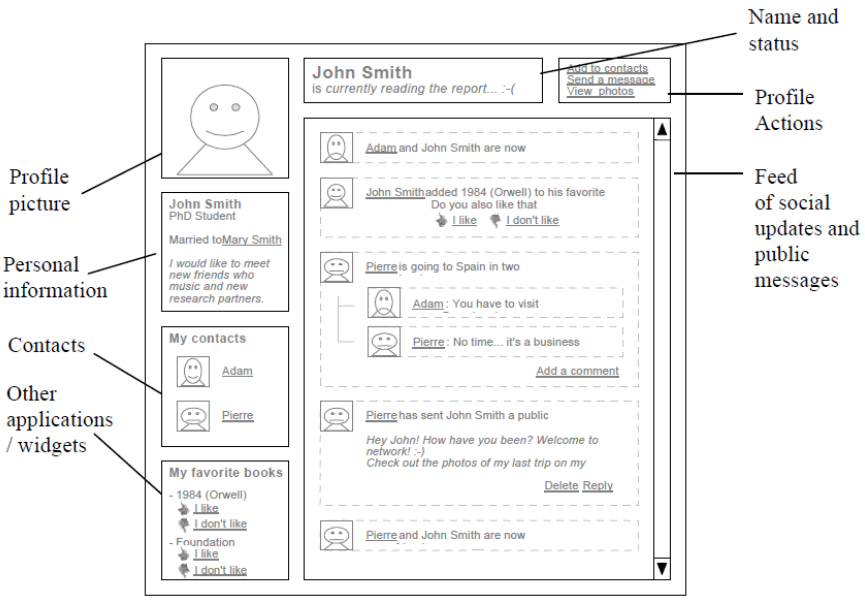

Figure 2.2: Typical user profile in Social Networking Sites -

On former SNS (i.e. classmates.com), user profiles were structured in a specific manner, usually by filling predefined fields. A major reason for the popularity of Myspace is the possibility for users to personalize deeply the appearance of their profile page with HTML and CSS code. With a little bit of hacking, anyone could feel more unique, not only with the content but also with the style of their profile, making it more attractive to get more friend requests, and thus appear to be more popular.

On most other SNS, users were invited to personalize their profile with pre-defined styles or controlled choice of colors, media and fonts. The biggest step in this domain was made by Facebook in 2007, when they opened their website as a platform for externally-developed add-on applications (Economist, 2007). These applications can appear as public widgets on the user's profile, leverage his/her community and integrate new social interactions in the facebook interface. For example, as depicted on Figure 2.3, a social application about users' favorite movies can display a widget on the user's profile in order to show off the user's favorite movies to the user's contacts, and to invite them to interact about these movies (e.g. ratings). This innovation made Facebook become more popular than Myspace for the first time in 2008 (Beky, 2008), showing

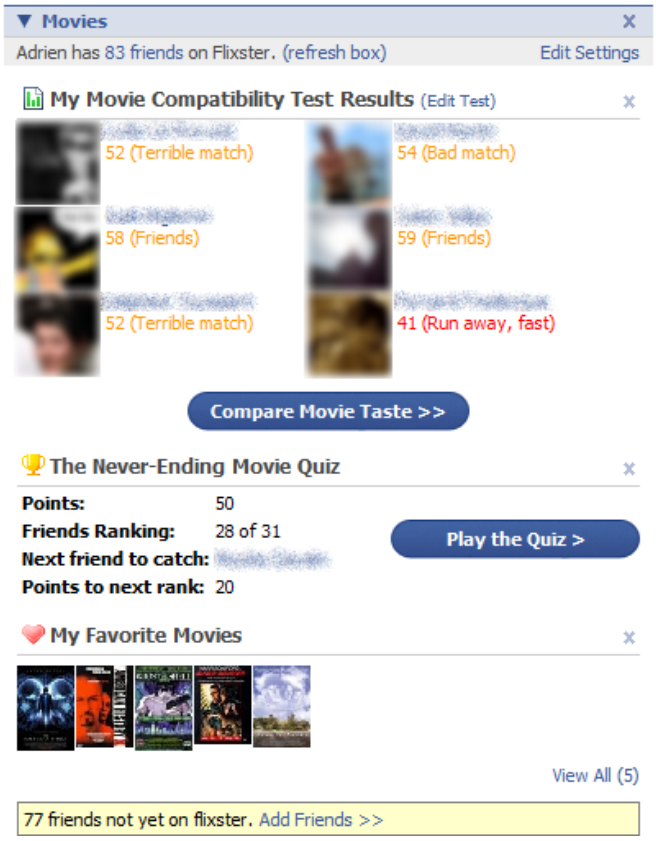

Figure 2.3: The "Movies" application on Facebook - ©2008 Flixster, Facebook. Used with permission.

that it is the diversity of applications (and thus, new social interaction opportunities) which attracts users on SNS.

Since this evolution, most competing SNS have turned into social applications platforms like Facebook, allowing third-party applications to leverage the user's community in order to enrich its social experience. This trend gave birth to several collaborative initiatives such as OpenSocial (OpenSocial & Group, 2009), which defines a common API to build interoperable applications on compliant social platforms, including LinkedIn. Note that Facebook has not joined this project, so they keep full control on their API. Besides the addition of content brought by third-party social applications on user profiles, we will discuss some of the new interactions modalities implied in the next paragraph.

2.1.3 New social interaction modalities

In this subsection, we present several interaction modalities listed in Table 2.1, which were introduced by SNS and their add-on applications.

The most common feature is public profile messaging. This feature was metaphorically called "wall" by Facebook. It allows to turn a simple message into a public announcement or display of affection, allowing friends to know what is happening in their community. This modality has been appropriated by users to create new rituals. On Myspace, the most respected ritual is to respond to a connection approval by sending a public message "thanks for the add" on the recipient's profile page. This is a good way to show the recipient's contacts that there is a new contact on board, advertise, in the case of an artist or a brand, and also to recommend the recipient to the people who consult his profile (e.g. for business networking).

SNS also allow users to advertise bulletins to their community. A bulletin (also called "updates" on Facebook) is an announcement or message that will be listed on every contact's bulletin listing. It is a good way for artists to announce new material, and for other people to spread good/bad news, general questions and even jokes. It can also be used as an incentive to gather attention and feedback from contacts.

One can also buy a "gift" for someone, which eventually appears as a small image on the recipient's profile, with the name of the sender. As silly as it may seem to pay for sending a small image to another internet user, it is actually seen by users

Interaction	Recipient(s)	Visibility/ notification	Intention(s)
Profile message	Contact/own profile	Public (all contacts)	- Introduction of a newly added user - Public display of interest/affection, or recommendation of the recipient (e.g. business) - Let the recipient's contacts know what's going on between them
Bulletin/ Posted item	Contact/own profile	Public (all contacts)	- Share interesting content with contacts - Announce an important event to all contacts - Request feedback from contacts
Gift	Contact	Public (all contacts)	- Public display of interest/affection, with more impact that a profile message, because gifts are usually not free
Events (invitation)	Contact	Public or Private	- Invite (some) contacts to an event - Enable communication between attending people (e.g. for arranging a common gift, adding contacts) - Share content related to the event (e.g. photos, videos, links)
Groups (invitation)	Contact	Public or Private	- Gather people around a same interest or cause to enable communication about it - Opportunity to add contacts
Poke	Any person	Private	- Say "hello, check out my profile" to someone probably just met in real life (less formal than a connection request) - Temporary inclusion of the recipient in the sender's contacts, allowing visibility of his/her profile and rich communication
Private message	Any person	Private	- Have private interpersonal discussions (no particular interest for social networking)

Table 2.1: Typical interaction modalities on Social Networking Sites

as a distinctive sign of affection, as this "gift" cannot be duplicated to other contacts without paying for it again.

Creating "events" and inviting contacts to them is a good way to announce this event with practical (and possibly targeted) information such as their location, time, and motivations to attend. Because people are invited to respond whether they are going to attend or not (or maybe), it gives the opportunity to communicate with attending people (e.g. for arranging a common gift because a party, or to keep in touch with new friends made during the party) and with indecisive people (i.e. to personally insist with good reasons to accept the invitation). An "event" can also gather and hold content related to this event (e.g. photos, videos, links...) to all participants.

A social network can hold several communities which are interested in specific subjects or causes. By joining a group, one has the opportunity to integrate conversations that are related to the subject of this group, possibly with people that are not part of his/her contacts. This can lead to connecting with some of these people.

Despite the strong meaning of "friend" given to Facebook contacts, one can "poke" another if he/she is interested in contacting that person without bothering for a "friendship" request. This "poke" is a notification that also allows the recipient to browse the profile of the sender without restriction, like if they were friends for a limited period.

Other interactions include social tagging of people, comparison and matching of user preferences (e.g. favorite movies, music and books) and quiz results, and games (e.g. poker). Of course, it is also possible to send private messages, but this communication modality shows no interest for social networking.

2.1.4 Feeds of social updates

In September 2006, Facebook added a '*news feed*' to their users' home page: a personalized list of the latests events which occurred in one's social network, including profile changes (including marital/relationship status, new photos) and public messages. That way, Facebook users no longer had to regularly check their friends' profiles to know what they have been up to. Instead, they just had to log into Facebook, and read social updates from their news feed, like they would read a newspaper. At first, this announcement scared many users for privacy reasons (Arrington, 2006). Whereas many predicted that this feature would kill Facebook, it actually catalyzed a massive increase in the site's growth. Since then, many sites have been integrating a news feed.

It allows one to have a social feeling of what is happening in the community, and it helps spreading information in a viral fashion.

With the integration of news feeds on SNS, Internet users discovered a new and addictive way of keeping in touch with other people: '*microblogging*'. Microblogging sites allowed them to share their news, mood, ideas, jokes and status through short messages, broadcasted to their communities, or to the whole web. As said by Thompson (2008), this '*ambient awareness*' is '*very much like being physically near someone and picking up on his mood through the little things he does - body language, sighs, stray comments – out of the corner of your eye.*' The pioneer in this domain is Twitter (Java *et al.*, 2007), launched in October 2006. On this website, users are invited to post public '*status updates*', by answering the question '*what are you doing?*' from time to time, without exceeding 140 characters. It is possible to send such updates from anywhere by SMS, and to be notified of updates sent by '*followed*' twitter users through a personalized feed but also by SMS, which turns this SNS into a simple mobile SNS that works with any mobile phone without Internet access. This microblogging concept has been adopted by users through various unexpected usages: to advertise websites and bulletins, share personal opinions and moods, ask for advice, enable fluid and transparent client-to-business communication, relay news and hazardous events they witnessed (Mischaud, 2007) (by '*re-tweeting*' the social update from their original author), and even advertising some other users to their followers on fridays (the '*#followfriday*' practice).

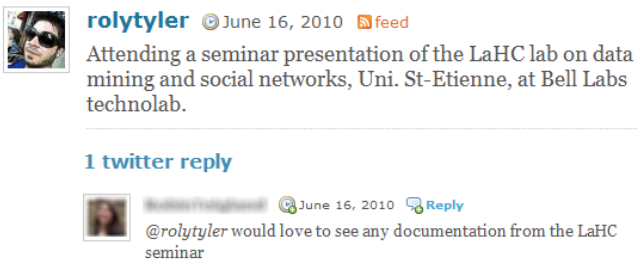

Figure 2.4: An opportunistic contact on Twitter - The update sent by '*rolytyler*' was found and replied by a stranger, who was probably looking for one of the keywords contained in '*rolytyler*"s update.

By default, Twitter status updates are public. Additionally to the possibility of fol-

lowing users, Twitter Search [1] allows to invoke dynamic queries (i.e. feed subscriptions) on the whole public Twitter feed, according to several criteria: presence of given words, author, time and geographical location of posting, etc... This functionality enables new kinds of opportunistic communication, in which people can be discovered for having included a specific keyword in one of their updates (see Figure 2.4), and frequency of keywords found in updates can be leveraged for statistics. Whereas Twitter is good tool to improve visibility, Erickson (2008) deplore that is it too poor for proper awareness, as it does not provide a feeling of accountability to its users. However, its growing usage, and its influence towards other microblogging services and SNS, makes Twitter a novel, efficient and popular way to communicate.

> *"It's like I can distantly read everyone's mind," Haley went on to say. "I love that. I feel like I'm getting to something raw about my friends. It's like I've got this heads-up display for them."*
> Testimony from a microblog user, (Thompson, 2008)

Thanks to social feeds, microblogging systems and other SNS can help to virtually increase the number of individuals with whom a stable inter-personal relationship can be maintained, known as the '*Dunbar number*' (Dunbar, 1993) which observed values can vary between 100 and 200 for humans. Indeed, a computer-supported social network can be used to overcome cognitive limits of one's '*social memory*', and therefore to be potentially able to keep in touch with more than 200 people at once. Overcoming this limit can expand people's ability to solve problems, by connecting with more '*weak ties*', those unfamiliar or remote acquaintances that are intimate enough to want to help out. Professional SNS such as LinkedIn rely on this assumption to help people finding professional opportunities thanks to their network of colleagues.

2.1.5 Pointless babble or awareness?

When Twitter started to become very popular, some criticism arose from skeptics, as its users would mostly broadcast '*pointless babble*' Boyd (2009). But Thompson (2008) explains that some microblog updates might not seem useful when taken individually, but can become interesting and inspiring when received in higher pace:

[1] Twitter Search: http://search.twitter.com/

> *This is the paradox of ambient awareness. Each little update – each individual bit of social information – is insignificant on its own, even supremely mundane. But taken together, over time, the little snippets coalesce into a surprisingly sophisticated portrait of your friends' and family members' lives, like thousands of dots making a pointillist painting. This was never before possible, because in the real world, no friend would bother to call you up and detail the sandwiches she was eating. The ambient information becomes like "a type of E.S.P.," as Haley described it to me, an invisible dimension floating over everyday life.*

This opinion is shared by Marc Davis, a chief scientist at Yahoo and former professor of information science at the University of California at Berkeley: '*No message is the single-most-important message. It's sort of like when you're sitting with someone and you look over and they smile at you. You're sitting here reading the paper, and you're doing your side-by-side thing, and you just sort of let people know you're aware of them.*'

2.1.6 Content of status updates

Naaman *et al.* (2010) identified two types of Twitter users: '*Informers*' are conversational users that mostly use status updates for consuming, sharing and relaying information (news etc...) without forgetting to mention and reply to their peers, whereas '*Meformers*' are more egocentric but however necessary to help users keep maintain relationships with strong and weak ties. Furthermore, four dominant categories of practices were observed: information sharing (22% of updates), opinions/complaints (25% of updates), statements and random thoughts (25% of updates), and '*me now*' (41% of updates) relating to the author's personal activity and context at the time of posting. This last category represent 51% of updates posted from mobile devices, and 37% from non-mobile applications.

In another study on 624 facebook and twitter users, Morris *et al.* (2010) reported that 50% of respondents had used status messages to ask a question, similar to Q&A sites. Most of them were expecting recommendations (29%) and opinions (22%). They prefer submitting those questions to SNS because they trust their contacts more than strangers, they expect more subjective answers, and also because some questions are hard to formulate on search engines and public Q&A sites that lack context. The intention of advertising current interests to their community was also cited as a motivation for prefering SNS-mediated questions.

2.1.7 Different types of social updates

Figure 2.5: Two sample social updates - A status update from Twitter, and a bookmark update from Delicious.

Following the microblogging phenomenon, many web sites and social applications have been producing updates, which we generalize under the name of '*social updates*'. The following kinds of social updates are identified:

- Status updates, as introduced above, are manual messages shared by users to their community only, or publicly on the Internet. Beyond providing personal statuses, users also post status updates for information sharing and for asking questions.
- Content sharing updates are less explicit and personal, they are generated automatically when a user contributes some content (e.g. a photo, a video, a blog post) in order to advertise this content to the community. Those updates usually mediate the publication of a user's past activity rather than a current one, but they add some visibility to him/her, and can trigger feedback from the community.
- Bookmarking updates immediately mediates a part of the author's context: one of the pages that he/she was browsing. Those updates can reflect current interests of the user, or improve the knowledge of user's preferences (i.e. for profiling).

2.1.8 Many opportunities to communicate

Even though spontaneous messages are the most simple and universal way of communicating, SNS also provide many other opportunities to communicate. Indeed, it is possible to comment (or reply to) almost every piece of information shared on a SNS, including photos, videos, bulletins and even quizz results for example. Like on blogs, comments are a great way to communicate with someone in a specific context in which we assume the recipient to be interested (i.e. because he/she is the one who posted the

piece of information that was commented). Of course, new comments are also considered as social updates, and thus they can be advertised on the feeds of every contact of the sender's community, according to the privacy preferences set by the users.

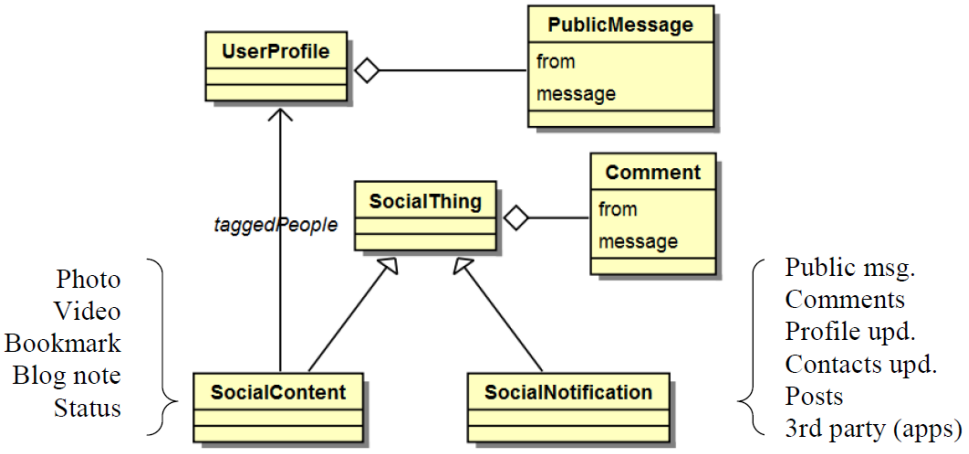

Figure 2.6: Simplified model of SNS communication -

It also means that a comment can lead to other comments from other people, giving birth to a discussion. On Facebook, comments can even be posted on status updates. As depicted on Figure 2.6, we can generalize SNS communications in a simple model in which the main conversational classes are PublicMessage and Comment. The user profile holds public messages, whereas social content and notifications hold comments. Both of them define a starting point for human communication. Whereas social content is manually entered by the user, notifications are generated by the platform and its applications from socially-relevant interactions. Social links are provided to the author of every message and comment (through the from attribute), and they can also be added on social content which is related to them (e.g. photos in which they appear) through the taggedPeople relation. This explicit link gives a good reason to notify people when content is posted about them, which leads to a good probability to get a response from them (e.g. to comment a photo on which they appear).

In his talk on the Impact of Social Models, and based on statistical studies of Twitter and Facebook (Facebook, 2010; Sysomos, 2009), Wroblewski (2009) observed that the contribution of users to social networks is driven by the attention of their connections, among other social model properties. The more connections, the more motivation to share. Indeed, the number of daily Twitter updates increases from 3 to 6 when followers

increased by 1000. Yahoo Answers showed a 480% increase in contributions when the number of relationships increased to 20.

2.1.9 Mobile Social Networking

The emergence of Bluetooth technology on most mobile phones led to new usages related to proximity, because of its short range coverage (usually around 10 meters). One of the earliest usages was called '*bluechat*' [1]; it consisted of sending OBEX messages to surrounding Bluetooth-enabled devices. Connecting with people in public places relies on the Bluetooth discovery function (implemented on all Bluetooth-compliant mobile phones) in order to look up devices in range. Users could change the public name/identifier of their Bluetooth mobile phone in order to advertise personal information such as sex, age and language. This usage led to the emergence of new programs allowing improved Bluetooth communication on mobile phones, for instant messaging (e.g. like IRC) and mobile social networking. Some of them were specialized for the purpose of dating (or, finding people to meet in the vicinity, with the ultimate target of dating), allowing participants to specify expectations of the people they would like to date: sex, age, interests, physical description... This was given the name of '*Bluedating*' [2]. Many bluedating programs appeared: Nokia Sensor, Serendipity, MobiLuck... Despite the benefits of considering the location (portion of one's context) in internet-based-like social platforms, none of these programs reached the critical amount of users to take off. There was no reason to join a dating network that nobody was using. One of the reasons of this failure was the lack of interoperability between these programs, preventing users of one program to communicate with users of another program. As none of these programs were raised as a standard, none became popular.

Bluetooth technology has proved to be a very interesting way to enable location/proximity-based services including communication and provision of local information, thanks to this short range coverage. Moreover it is widely supported by most mobile phones (and other devices) and is not a big battery consumer. However, the popularization of Internet connectivity on mobile phones encouraged a new breed of mobile social applications, called MoSoSo, Mobile Social Software (Lugano, 2007). Most applications are a mobile transposition of computer-based social networking systems. Most recent and

[1] Bluechat: http://en.wikipedia.org/wiki/Bluechat
[2] Bluedating: http://en.wikipedia.org/wiki/Bluedating

popular MoSoSo applications, including Foursquare [1] and Gowalla [2], invite their users to '*check-in*' to the places where they go. By doing that, a user's friends can decide to join him/her there, he/she can leave some tips about that place for other people, and he/she can be awarded different '*badges*' to show off on his/her social network profile, depending on his/her loyalty to the place, the discovery of a new area or other challenges. Considered as a sort of social game, this kind of application also implies links between the real world which the user explores, and the digital world which his/her friends can read about.

2.1.10 What about privacy?

In Barkhuus & Dey (2003), it was observed that '*users are willing to accept a large degree of autonomy from [context-aware and personalized] applications as long as the application's usefulness is greater than the cost of limited control*'. This behavior seems to apply well to most social applications. In 2010, Mark Zuckerberg, CEO of Facebook said that privacy was no longer a social norm. At a time when Facebook was re-designing its interface and communication workflows to a fashion quite similar to Twitter, this controversial phrase clearly shows the intention of inviting social network users to share more, to more people. Of course, Zuckerberg proved several times that he could orchestrate the behavior of Facebook's users to make them accept new usages. Indeed, users were happy to share more, because interacting on Facebook was worth it. This behavior proves that their privacy can be eroded, as long as the service brings sufficient value to every user.

2.1.11 Some usage statistics

Some selected statistics about Facebook (Facebook, 2010):

- More than 400 million active users:
- More than 35 million users update their status each day
- More than 60 million status updates posted each day
- More than 3 billion photos uploaded to the site each month

[1] Foursquare: http://foursquare.com/
[2] Gowalla: http://gowalla.com/

- More than 5 billion pieces of content (web links, news stories, blog posts, notes, photo albums, etc.) shared each week
- More than 3.5 million events created each month
- An average user has 130 friends on Facebook, spends more than 55 minutes per day on the site, writes 25 comments each month, is invited to 3 events per month, and is a member of 13 groups.
- Facebook Platform currently holds 500,000 active applications, more than 250 of them have more than one million monthly active users.
- There are more than 100 million active users currently accessing Facebook through their mobile devices, and they are twice more active on Facebook than non-mobile users.

Some selected statistics about Twitter (Berry, 2010; Weil, 2010):

- In February 2010, Twitter was handling 50 million tweets a day
- In average, it represents 600 tweets per second
- Twitter.com is visited by 180 million visitors every month
- 75% of Twitter's traffic comes from outsite their own site (e.g. mobile clients)

More general statistics on usage of social networking sites: News (2010)

- The average social-networking user around the world spent more than 5 and a half hours on their social networking sites, in December 2009.
- In the USA, 6 hours.
- In France, 4 hours.
- This duration increased by 82% from December 2008.

2.1.12 Conclusion

In this section, we have seen that SNS are powerful social communication platforms with several different communication modalities relying on the viral spreading of social content, interactions and updates. Nevertheless, the big enthusiasm generated by SNS has led to a productivity issue which made many companies restrict access to SNS. The paradox in that matter is that several initiatives are translating the SNS paradigm to enterprises in order to improve corporate knowledge management and sharing through participation of their employees. With intranet-based SNS, enterprise workers can benefit from immediate expertise and news rather than relying on classical Knowledge Management practices, which are costly in terms of redaction, and less personalized than talking directly with an expert. SNS can thus support decision making, by incorporating business intelligence from large amounts of individual contributions, assuming that overall participation is high.

SNS are a potential killer application for casual awareness, opportunistic communication (e.g. between businesses and customers), and for improving knowledge sharing, collaborations, productivity and fulfillment of professionals, but they are much too time consuming yet, since every piece of information to spread has to be entered manually. In the next section, we will further analyze how productivity can suffer from the use of social networking tools, and identify opportunities to improve them in this regard.

2.2 Usage and impact of real-time microblogging

In April 2010, we conducted a survey towards 256 users of real-time microblogging platforms, mostly Twitter users, in order to analyze the usage and impact of those platforms. In this section, we present the results of this survey, and discuss their explanation and possible usage improvements towards reducing information overload and interruptions.

2.2.1 Results of the survey

As seen on the distribution chart 2.7, most respondents (31%) follow between 100 and 250 people feeds. The second category of respondents (28%) follow more than 250 feeds, which is much higher than the Dunbar number (i.e. the maximum number of people one can naturally maintain a stable relationship with).

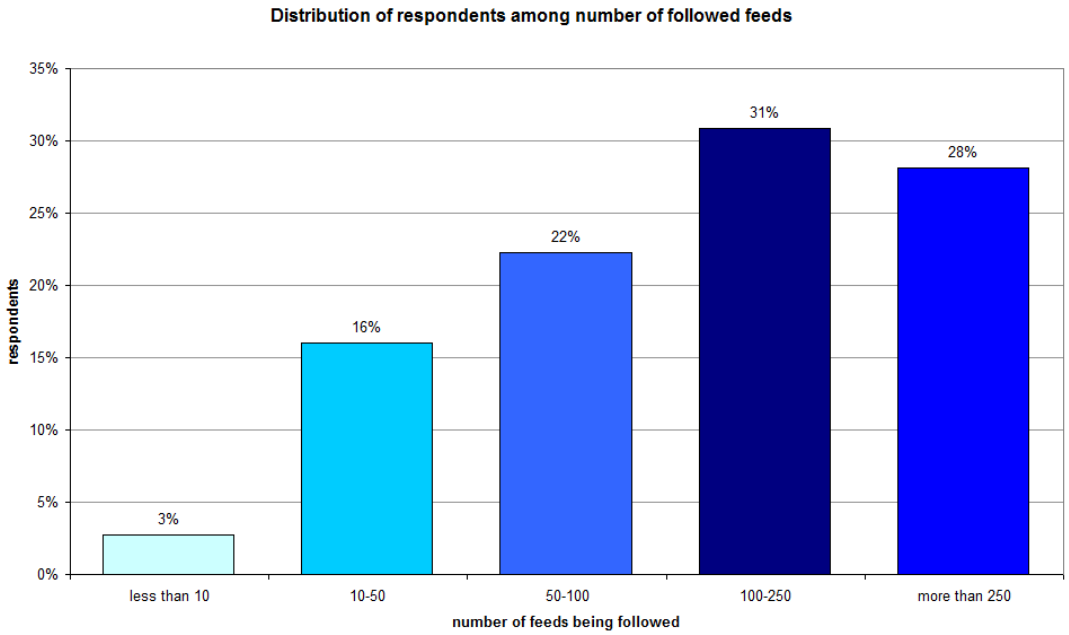

Figure 2.7: Distribution of respondents among number of followed feeds -

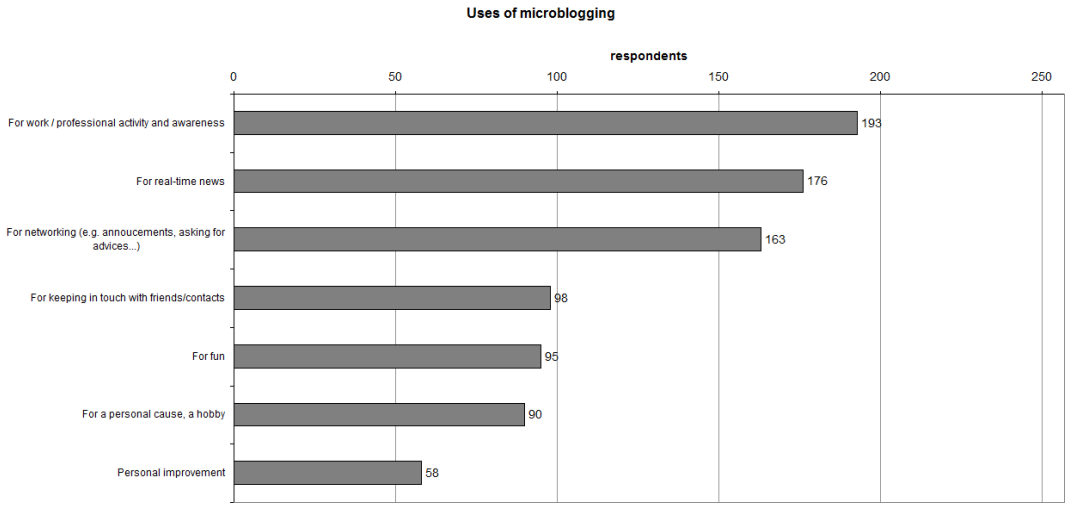

Figure 2.8: Uses of microblogging -

Figure 2.8 reveals that 75% of respondents use microblogging platforms for their professional activity, 69% use those for keeping up with news, and 64% leverage the '*six degrees of separation*' effect for networking (i.e. making announcements, or ask for advice from their contacts). Personal and fun usage of microblogs is lowly represented (23% to 38%), at least by the respondents of this survey.

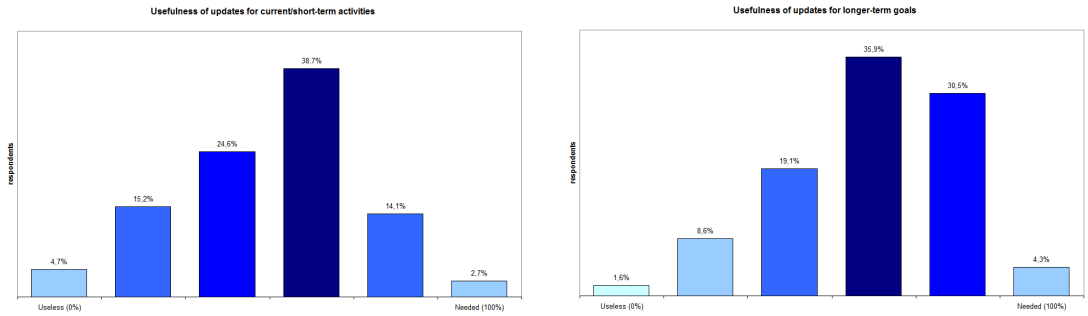

Figure 2.9: Usefulness of microblog updates - for current activities (left), and for longer-term goals (right)

On a 6-rank scale between 0 and 5, most respondents (between 35.9% and 38.7%) gave a medium-high rate of 3 to evaluate the usefulness they perceive over all the updates they read, as depicted on Figure 2.9. However, we observe that usefulness is rated slightly higher for longer-term goals (4 being the second most given rate, with 30.5%) than for current activities (2 being the second most given rate, with 24.6%). This can support the hypothesis that, in most cases, new real-time updates should rather not interrupt users in their current activities, as many updates show little usefulness for this activity.

According to Figure 2.10, most respondents (35%) keep the last updates from their microblogging platform always visible on their screen. About as many respondents (33%) activated pop-up (also called '*toaster*') notifications on their screen, in order to keep up with the last updates in real-time. Fewer respondents chose to monitor the number of unread updates (20%), in order to decide when to read those messages, or to receive notifications on their mobile phone (e.g. SMS) (14%).

As seen on Figure 2.11, the amount of attention given to real-time update notifications varies among respondents who receive such notifications, whereas 18% of them do not receive such notifications. Most of them (62%) read most notified updates, depend-

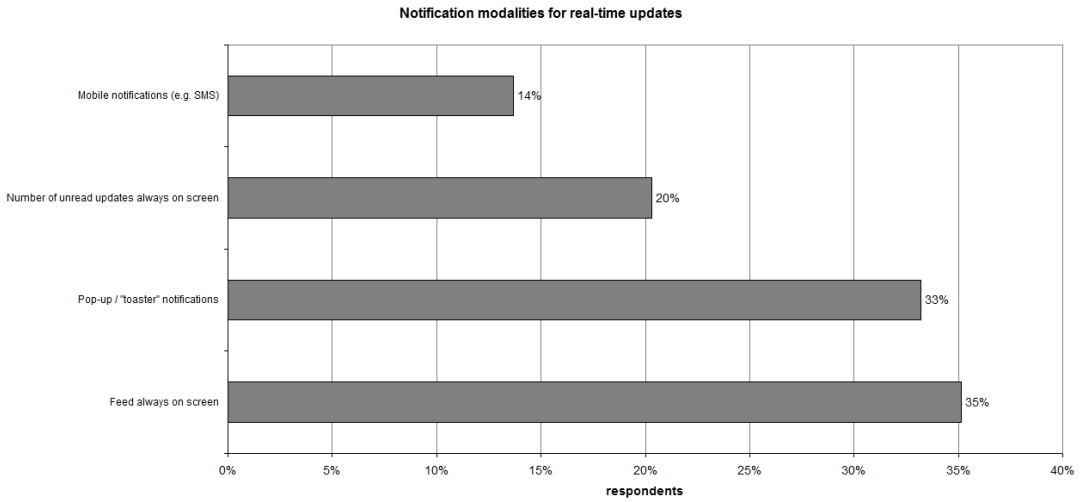

Figure 2.10: Notification modalities for real-time updates -

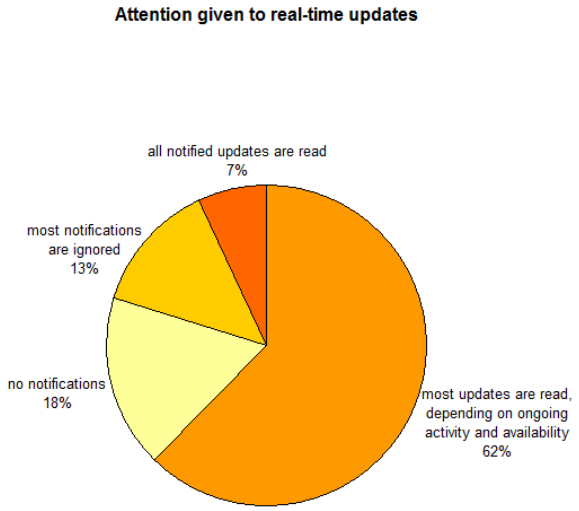

Figure 2.11: Attention given to notified updates -

ing on their ongoing activity and availability. Others either read all notified updates (7%), or simply ignore those (13%).

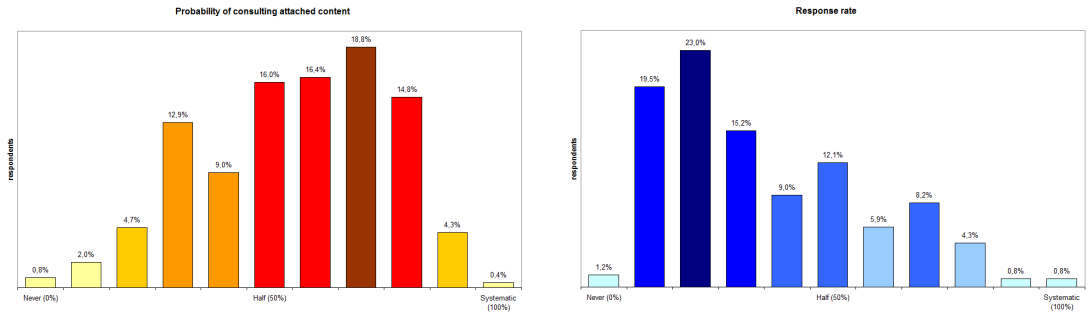

Figure 2.12: Probability of consulting content attached to updates - (left), and of responding to them (right)

From Figure 2.12, we can observe that microbloggers consume more content than they produce. Indeed, 66% of them consult the attached content of 50% to 80% updates they receive, whereas the same proportion of respondents actually respond by replying or relaying 10% to 50% updates.

In response to the large amount of social updates they receive, Figure 2.13 depicts that 78% of respondents don't use filtering mechanisms. The most number of others who do (8%) developed their own filtering mechanism, instead of leveraging existing services offering filters (5%) or trends (4%). 5% respondents, beyond not using filtering mechanisms, would rather receive more updates. This last category represents a close portion to respondents who follow less then 10 feeds, as depicted on Figure 2.7, which would explain their eagerness for more updates.

Although we have observed that most respondents don't use filtering mechanisms, Figure 2.14 reveals a need for filtering: 59% of them would like to filter by type of updates (e.g. news, professional, personal, fun, etc...) and 51% by subject. This need for filtering mechanisms is confirmed by other studies (Bernstein *et al.*, 2010). Relevance filtering is also expected by respondents, either according to their current interests (37%) or current activity (36%). Fewer of them would like to benefit from other filtering criteria on updates, such as the type of content (20%), geographical proximity (18%), or relevance with future activities and plans (16%).

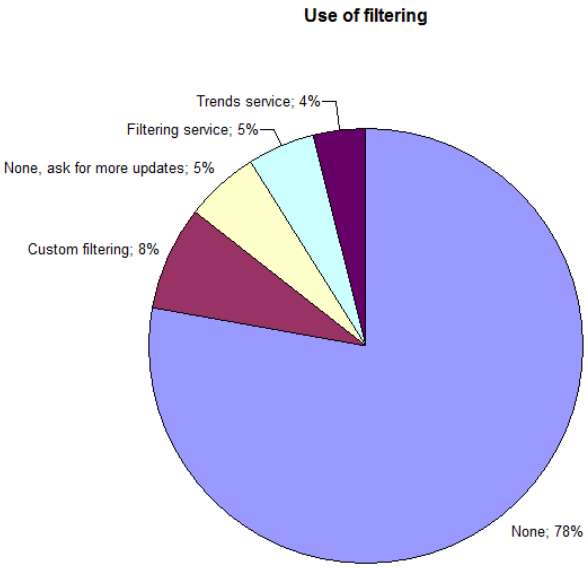

Figure 2.13: Use of filtering -

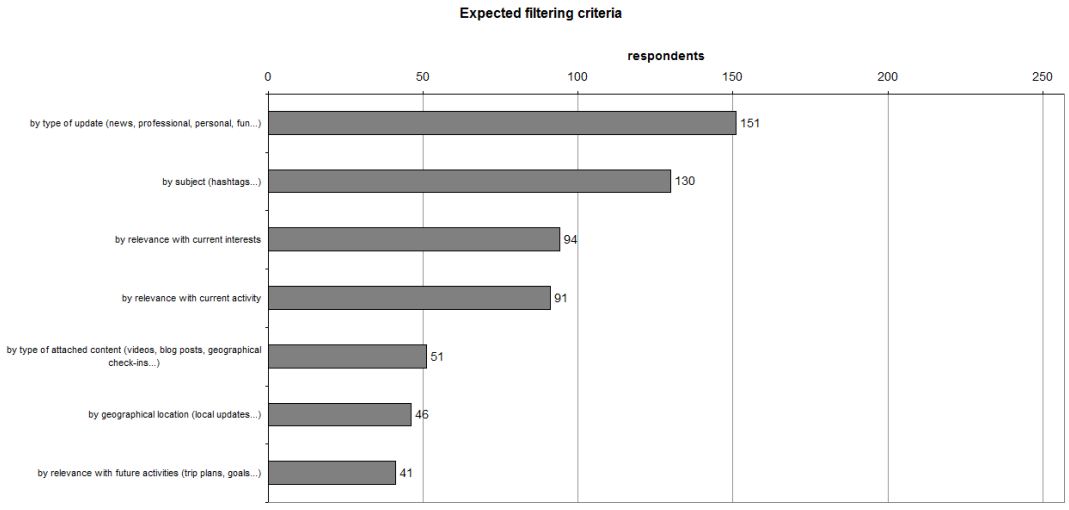

Figure 2.14: Filtering criteria most expected from respondents -

2.2.2 Filtering mechanisms: challenges, solutions and opportunities

On Figure 2.14, we have seen that the two most expected filtering mechanisms are by type of update, and by subject. This can be surprising, as it is already granted for microblogging users to chose whose feeds they follow, and to follow subjects of interest represented by corresponding terms and hashtags. However, these native features provided by current microblogging platforms are a partial solution to that problem: people do not always post updates of the same type (or at least, they do not have to), and it can be tedious for users to select the right terms and hashtags for following updates on a specific subject. Query enrichment and semantic search mechanisms could support them for this second need. Concerning the type of updates, a solution could be that people post updates through different feeds, depending on their type. They would thus have to expose and maintain separate professional, personal, and other specific feeds, and their followers would have to follow the feeds they are interested in, depending on the type of updates posted to those. Another solution would be to ask users to provide some metadata to describe the type of update they intend to post, as long as microblogging platforms would allow type-filtered subscriptions to their author's feed, instead of simply following his whole feed.

The two following most expected filtering mechanisms are more dynamic: they rely on relevance of updates with the consumer's current interests and activities. As a person's current interests can be profiled, updates can be ranked against this profile, in order to deliver the most relevant ones. Such filtering mechanism is proposed by the '*my6sense*' application [1], in which users' interests are profiled according to descriptive features (e.g. metadata, tags, or frequent words from the content) that are attached to the content they decide to read. Concerning the relevance with current activities, such profiling becomes more complex, as users can achieve their activities out of the scope of the application that delivers updates, e.g. on various computer applications, or even without the computer. In that case, some knowledge on the current activity of a user must be gathered by analyzing contextual cues acquired from several sensors: physical (e.g. user's location from a GPS device), virtual (e.g. name of the document being edited with word-processing software on the computer) and social (e.g. messages and

[1] my6sense: http://www.my6sense.com/

updates from other users, giving clues about the user's current activity and context). That challenge is addressed in the following chapters of this thesis.

2.2.3 Discussion

This survey reveals that users of real-time microblogging platforms (such as Twitter):

- usually consume many updates (with or without being notified in real-time), mostly for professional use, news and networking;
- mostly receive updates that are interesting in regard to their longer-term goals, more than to their current activities;
- consult the content attached to many updates, but do not respond to them often;
- and that they would benefit from filtering mechanisms.

In response to those observations, we proposed some opportunities for improvement:

- In order to allow users to filter updates by type more easily, we recommend that each user maintains separate accounts (with different followers) for each type of updates (e.g. personal messages, news on specific topics, etc...)
- New metadata could also be added to microblogging systems, so that authors could precise the type of each update they post.
- In response to the need for filtering by relevance with users' current interests and activities, we proposed to develop a system that can maintain a dynamic profile of users, based on physical, virtual and social sensors.

In the next section, we study the results of this survey and identify opportunities for improvement, from a cognitive point of view.

2.3 Social notifications: cognitive impact and opportunities for improvement

In the previous section we observed the usage and habits of microblogging users, and proposed technical solutions for user-identified needs (e.g. filtering). Here, we discuss this usage and its impact from a cognitive point of view, and identify additional opportunities for improvements.

On Figure 2.10, we observed that many microblogging users activated screen-based notifications (feed always visible, or pop-up notifications), in order to be aware of new updates in real-time. Those notifications can interrupt and distract users involved in a primary task (e.g. writing a report), as they can happen any time, and possibly at a frequent pace.

In some cases, interruptions are useful for this primary task, and can even facilitate task performance (Speier *et al.*, 2003). However, in (Roda *et al.*, 2006), it has been reported that interruptions can also '*increase the load on attention and memory (Gillie & Broadbent, 1989), may generate stress (Bailey* et al.*, 2001; Zijlstra* et al.*, 1999), and compromise the performance of the primary task (Franke* et al.*, 2002; McFarlane & Latorella, 2002; Nagata, 2003; Speier* et al.*, 2003), especially when the user is working on handheld devices in mobile environments (Nagata, 2003).*' According to Miyata & Norman (1986), consuming unexpected messages (in our case: social updates) can be both beneficial and disruptive. They recommended that users can control the trade-off, to be able to accept diverted or not, depending on the context.

In their study on usage of email notifications thrown by MS Outlook, Iqbal & Horvitz (2010) observed that, even through notifications can imply costly focus interruptions of a primary on-going task, the lack of awareness induced by the disablement of those is also costly. Indeed, many users who were forced to disable email notifications, opened MS Outlook more frequently to check for incoming messages, in order to prevent having to process long chunks of emails later.

From this study and the results of our survey, we can infer that some users are addicted to being notified of new information in real-time. We certainly believe that emails are still the preferred way to send formal and decisive messages that can have an impact on users' on-going tasks, especially in a professional context. This potential impact can explain the users' need for being notified of new emails in real-time, and for having to process long chunks of unread emails, whereas a microblogging user would not mind skipping hundreds of updates. However, we support that social updates can be important or useful for people's long-term decisions (e.g. career, goals; as seen on Figure 2.9), personal activities, and even synchronous group arrangements and announcements while in mobility (e.g. canceled flight, selection of a new time and place to meet; as usually done by exchanging SMS between only two people).

2.4 Conclusion

In this chapter, we analyzed the specificities and usage of social networking sites (SNS) and microblogging systems as communication systems, Facebook, Twitter and Myspace have been cited several times as examples. Then we studied recent usage trends concerning the attention given by microblogging users to real-time notifications, based on the results of a survey we conducted. We identified emerging needs for computer-supported filtering of social updates, either expressed by users through the survey, and by literature on cognitive psychology (Roda *et al.*, 2006). In particular, context has been identified as a promising criteria for reducing the frequency of social updates to notify Iqbal & Horvitz (2010).

Chapter 3

From Awareness Systems to Information management techniques

In the previous chapter, we studied the usage of popular social networking systems, allowing people to keep in touch, for maintaining professional, personal, affective links in higher scales than it is possible naturally or with previous communication tools.

In this chapter, we survey research efforts also aiming at (or enabling to) creating and maintaining social ties, but also to enhance more specific kinds of interactions, such as collaborative work. In particular, we focus on:

- '*awareness systems*', systems that allow people to be aware of the status of other people;
- and '*social matching systems*', systems that allow to discover relevant people, and their activities.

After presenting a selection of awareness and social matching systems, we will study several techniques that can enable such systems, including information filtering and extraction techniques, and underlying similarity measures that can be leveraged. Then, we will identify existing techniques to improve the performance and scalability of awareness systems.

3.1 Awareness and social matching systems

3.1.1 Definitions

Originally used by neurologists and psychologists (Sieb, 1990), the term of '*awareness*' has been adopted by computer scientists as '*an understanding of the activities of others, which provides a context for your own activity*' (Dourish & Bellotti, 1992), with the aim of supporting human awareness, communication and interaction with computer systems. More specifically, the Ambient Awareness project [1] defines awareness systems as '*Awareness Systems are computer mediated communication systems that help people maintain a peripheral awareness of others.*' According to Liechti (2000), Awareness Systems can be divided in four categories: group awareness, workspace awareness, contextual awareness and peripheral awareness. Those categories represent different goals, research disciplines and human interaction models.

Social matching systems, in the other hand, are defined by Terveen & McDonald (2005) as '*a type of recommender system that bring people together rather than recommend items to people. They offer great potential to increase social interaction and foster collaboration among users within organizational intranets and on the Internet as a whole.*'

In our case, we intend to enable awareness by filtering social information streams, assuming that users will accept to be notified only about updates that are the most relevant to their current activity and context. Notifications can be delivered to users through various modalities, including: dedicated computer screen areas, '*toaster*'/pop-up notifications on computer screens, mobile phones (e.g. vibration and screen), peripheral displays, and/or sounds. In this thesis, we focus on computer-mediated notifications, and intend to leverage context-based social matching techniques for identifying the most relevant updates to deliver.

3.1.2 Foundations and some applications

In 1992, Dourish & Bellotti (1992) proposed the '*shared feedback*' approach, consisting of tracking users' actions on shared documents, and allowing them to comment those, in order provide useful context information and avoid work duplication.

[1] Ambient Awareness project description: http://www.awareness.id.tue.nl/main.php

Based on the Strauss' Theory of Action, Fitzpatrick (1998) proposed the '*Locales*' framework for supporting design of collaborative and communication applications. This framework relies on three assumptions: (1) user's actions constantly evolve based on the actions of others, (2) interaction occurs within social worlds, and (3) the interactional needs of users are important.

Following Fitzpatrick's principles and recommendations on civic structures, individual views, interaction trajectory, mutuality, and embodied interaction from Dourish (2001) (defined as '*the creation, manipulation, and sharing of meaning through engaged interaction with artifacts*'), Amelung (2005) developed the '*Context-Aware Notification System*' ('*CANS*'). Extending the '*Sakai*' collaborative environment [1], CANS acts as a publish/subscribe system for delivering notifications on actions of other users of the environment. Notifications are selected adaptively to the user's previous and current activities, projects and preferences on the collaborative environment. The social context used for generating and dispatching notifications relies on knowledge and actions handled within the Sakai environment. The author recognizes that the Internet and the physical world are additional social contexts that are also relevant to users. For example, users might be interested to be notified when their colleague (or any other person of interest) has arrived or left the office, as this information would impact the communication and collaboration modalities with him.

Erickson & Kellogg (2000) experimented the social proxy as an abstract representation of activity in an enterprise communication space: '*babble*'. In these semiprivate asynchronous discussion spaces, users are aware of the level of participation of their colleagues on several ongoing entitled discussions, and can decide opportunistically to join one of those. Additionally, common context between people has been identified as an important factor to trigger new interactions (Vogiazou *et al.*, 2003) (e.g. sign of respect, conversation, or exchange).

As extensive usage of awareness systems can lead to '*information overload*', Damian *et al.* (2007) claims that updates and notifications should be filtered by relevance to users' tasks, in order to prevent users from ignoring too frequent updates potentially containing crucial information for the success of a project. Machine-interpretable contextual information can support awareness systems to identify to unplanned interactions (e.g. such as changes in the team members involved, which are very frequent in

[1] Sakai: http://sakaiproject.org/

collaborative development projects), and thus dynamically adapt the dispatching and presentation (e.g. notification modalities) of awareness information to users according to their current context (Liechti, 2000). Several awareness experiments have been carried out for collaborative bug tracking (Ellis *et al.*, 2007) and configuration management of collaborative software development projects (Biehl *et al.*, 2007; Sarma *et al.*, 2008), in order to promote communication, reduce overlapping work, and improve ability to detect and resolve conflicts. In terms of visual summarization of awareness, Lin *et al.* (2008) propose an interactive and dynamic visualization framework for summarizing social networking interactions, and applied it with the Flickr social photo sharing site.

Delalonde (2007) developed a social matching application based on the social foundations of Actor-Network Theory. By formulating a query, the user is proposed a list of relevant people, based on their activity traces. User profiles are built by aggregating weighted keywords from the content they contributed, using TF-IDF (Term Frequency - Inverse Document Frequency, (Salton & McGill, 1986)). The recommended people are ranked using a '*ContactRank*' measure which evaluation relies on the knowledge, relay, reputation and participation of users on keyword-based subjects. It is possible for users to collaborate through a wiki-based interface in which best solutions to a question are discussed by participants, similarly to Q&A systems (e.g. Yahoo!'s). The evaluation of this system was based on a vocabulary of 250 keywords, and revealed that the participation rate among users was too low to observe enough emerging collaborations. Two shortcomings were criticized by users: the keyword vocabulary was strict, preventing semantically close profiles to be leveraged, and user feedback mechanisms would allow users to improve the relevance of recommendations.

Furthermore, this study revealed that users will not necessary be motivated to use a new tool. One explanation can be that information retention gives some power to employees, and they have no clear reason of being altruistic. Indeed, this behavior can hardly be expected between colleagues who have no common history, culture, nor context, especially in an high-scale enterprise environment. Providing more interaction history and transparency to users might ease their involvement with the system.

With '*Panorama*', Vyas *et al.* (2007) proved that informal ties between co-workers are also essential to improve awareness, and thus better collaboration. In that case, computational support is not given to work-related content, but to casual content, so

that people decide to meet more naturally and eventually make each other aware of professional activities.

3.2 Recommender systems and information filtering techniques

Our problem is to rank the relevance for a user U of Social Updates from every other user X, according to the similarity of contextual information between U and X. This problem can find some solutions in the domain of Recommender Systems (Adomavicius & Tuzhilin, 2005; Terveen & McDonald, 2005), which splits into two categories: content-based and collaborative recommendation system. We now study some specificities and relevant applications for each category, as possible ways to model and solve our problem.

3.2.1 Content-based filtering

Content-based filtering and recommender systems rely on users' preferences, provided explicitly by users or learned from a history of actions, to rank the potential interest/relevancy of new items, according to those preferences (user's profile).

Up to our knowledge, the closest existing solution to our problem is a web-based service and mobile application called My6sense [1]. This software can filter entries from RSS feeds and social streams (thus including social updates from Social Networking Systems) according to the user's preferences. This content-based filtering solution relies on a profile which contains the user's favorite (or at least usual) subjects of interest, and which is continuously evolving by tracking entries that are viewed or annotated by the user. However, despite the evolving design of user profiles, the filtering is not adaptive to the user's current context.

Under the assumption that user profiles and context can be inferred from the content of documents that are currently browsed by the user in a computer-centered environment, it is possible to run term frequency algorithms such as TF-IDF (Salton & McGill, 1986) on currently opened documents in order to identify keywords that represent the subject of this current browsing/editing activity. Following this approach, the Fab recommender system (Balabanovic, 1997; Balabanovic & Shoham, 1997) recommends web pages which most representing keywords match the user's profile, using a cosine

[1] My6sense: http://www.my6sense.com/

similarity measure on 100-keyword lemmatized and normalized vectors. However, this method does not scale up when the number of users increase, because it is necessary to compute similarities between every user.

With WordSieve, Bauer & Leake (2001) outperform TF-IDF in terms of precision, but requires training data to tune several parameters. Dragunov *et al.* (2005) extended the extraction of contextual information to support additional desktop document manipulation applications such as the Microsoft Office suite. With their PRUNE framework, Kleek *et al.* (2009) model contextual information as entities (e.g. person, place or resource) and events, allowing extraction for heterogeneous sources like RSS/ATOM feeds, REST/JSON APIs, or even manual user entry. In the '*Notes that Float*' application, they employ this framework to attach contextual information to notes entered by the user, so that they can be ranked to increase their visibility adaptively to the relevance with the user's current content, which relies on its similarity with the context at the time these notes were added.

3.2.2 Social matching and collaborative filtering

Social matching systems consist of finding similar user profiles, in order to create communication opportunities. Collaborative filtering and recommender systems rely on social matching for finding similar users, and recommend items that are highly rated by one to a similar user who has not yet rated those items.

Budzik *et al.* (2000) provide collaboration opportunities by recommending people that are browsing similar documents, based on a TF-IDF content analysis, users' context being represented by weighted term vectors. These recommendations also include some information about people's current activities, as identified by a software module that tracks users' actions on their computer (e.g. chat sessions, running applications etc...).

Although its usage is more explicit and manual, social bookmarking is also a good practice for discovering new content and people. On the popular del.icio.us [1] social bookmarking website, and its enterprise clone DogEar (Millen *et al.*, 2006), users bookmark web pages that they like, can provide some '*tags*' to annotate them and find them easily, and share those annotated bookmarks on the web. Tags are vocabulary-free keywords that can be selected, entered or even created (e.g. acronyms, personal keywords)

[1] del.icio.us social bookmarking: http://del.icio.us/

by users, relating to the page's subject, nature, ownership, category refinement, characteristics, or for self reference, task organization Golder & Huberman (2005). The association of tags from several users to every web site implies the emergence of a crowd-sourced description of those web sites, and a classification based on tags (called folksonomy). Each user can thus be profiled by the set of tags that he/she has given to the most web pages, and those hyperlinked tags can be used as pivots to navigate from page to users, and inversely. The tag clouds emerging on web pages and users give promising collaborative recommendation applications (Hung *et al.*, 2008; Niwa *et al.*, 2006). Furthermore, semantics can also emerge from statistical analysis of those tag clouds (Specia & Motta, 2007; Tesconi *et al.*, 2008; Wetzker *et al.*, 2009).

Also relying on bookmarks, Agosto (2005) developed '*SoMeONe*', a collaboration tool that can recommend contacts (and their selection of web resources) to users by identifying communities of interest around topics, using collaborative filtering. Topics are extracted from the titles of bookmark categories stored by each user on SoMeONe. Their sociological studies led to the '*SocialRank*' algorithm which integrates social factors such as reputation, relevance, originality, redundancy and reactivity of contacts. However, the topics on which SoMeONe relies on, are statically defined by the Open Directory Project [1], which reduces the spectrum and granularity of topics that can possibly be represented. Moreover, the authors identified that more '*awareness*' was expected from the users of their system: by notifying new recommendations, providing explanations on why they were recommended, and allowing users to give more explicit feedback.

Bielenberg & Zacher (2005), in their '*groop.us*' prototype, have applied the same approach while relying on tags (folksonomies) attributed to web pages by users (on the delicious.com social bookmarking website) instead of hierarchical categories. After clustering contacts by similar tags into groups, they identified the necessity of studying the combination of tags from multiple social platforms (e.g. weblog entries) instead of relying too much on delicious.

3.2.3 Discussion

Similarly to Budzik *et al.* (2000)'s approach, we believe that opportunistic communication is key for awareness, and that the similarity of current working contexts (i.e. based

[1] http://www.dmoz.org/

on common concepts of documents being currently manipulated) is a great opportunity to propose such communication.

In response to our problem, collaborative filtering techniques seem to be more appropriate, as it consists of matching people with similar interests. However, in our case, interests are contextual, therefore they do not rely on items but on a dynamic set of features that can be extracted from an open range of information, including documents. Therefore, feature extraction techniques from content-based filtering are also to be leveraged, turning our system into a hybrid filtering system. This requirement fits into the category of the '*collaborative via content*' filtering paradigm (Pazzani, 1999).

3.3 Content models and feature extraction techniques

In the previous section, we observed that filtering and recommendation systems rely on finding similarities between user profiles and/or content. Finding similarities in such data structures is not a trivial task (Quesada, 2008), and can be achieved at different levels:

- at the keyword (or tag) level, weights can be compared using various similarity functions, including the classic cosine similarity, or the contrast model (Tversky, 1977);
- or by measuring the similarity of semantic concepts (e.g. defined in ontologies, see Appendix A), either identified from plain text, emerging from rich tag sets (e.g. co-occurrence analysis) (Cantador, 2008; Specia & Motta, 2007; Yi, 2008), or provided by users (Damme *et al.*, 2007; Tesconi *et al.*, 2008).

3.3.1 Keyword/phrase and tag-based models

Like in several projects surveyed in the previous section, CALVIN (Leake *et al.*, 2000) relies on TF-IDF to extracts the context of a user's task in the form of weighted vectors, by indexing the most frequent terms appearing in opened documents.

Higher abstraction terms can also emerge from textual content, e.g. concepts and/or topics possibly defined in specific taxonomies (e.g. thesaurus or vocabulary). KEA [1] is a Java-based open-source library that can extract such keywords (or keyphrases) after a

[1] Keyphrase Extraction Algorithm: http://www.nzdl.org/Kea/

training phase in which documents are associated to the expected keyphrases. However, KEA can also be used without training data, implying a reduced output expressivity, but providing the benefits of a more advanced keyword extraction than just counting keywords (e.g. stemming, stop-word removal, 4-feature-based extraction).

On crowd-sourced information spaces (e.g. web-based social bookmarking systems), despite some issues related to semantic ambiguity and variant spelling of terms combined in folksonomies (Marlow *et al.*, 2006; Mathes, 2004; Sinha, 2005), the increasing amount of tags given by Internet users to annotate digital resources (e.g. web pages tagged on delicious) naturally turned folksonomies into a good metadata representation and content classification scheme from which humans are able to recognize meaningful descriptions (Lamantia, 2008). Poor in quality (i.e. possibly ambiguous, structure-less, no common vocabulary) but rich in quantity and variety (e.g. openness to words from any language), tags have become good indexing features for filtering and recommender systems (Hotho *et al.*, 2006; Niwa *et al.*, 2006). Furthermore, with the growing use of twitter and geotagging applications in mobility, tags and natural words are now emerging from places, events and other real-world entities (Naaman & Nair, 2008), enabling those entities to be indirectly leveraged as features.

3.3.2 Semantic extraction

Relying on ontologies and subject-relation-object triples, the semantic representation of data prevents ambiguity of terms in natural language and enables powerful reasoning capabilities that can improve the performance of similarity measures in filtering and recommendation systems.

With the rise of the '*Semantic Web*', its enabling technologies (e.g. RDF, microformats, see Appendix A) and the semantic federation of data (e.g. LinkedData[1], which includes DBpedia[2], the semantic translation of Wikipedia[3] (Auer & Lehmann, 2007)), several web services have been developed to extract semantics from plain text content, including Semantic Proxy [4], tagthe.net [5] and AlchemyAPI [6]. Those services can be

[1]LinkedData: http://linkeddata.org/
[2]DBpedia: http://dbpedia.org/
[3]Wikipedia: http://www.wikipedia.org/
[4]Semantic Proxy: http://semanticproxy.com/
[5]tagthe.net: http://tagthe.net/
[6]AlchemyAPI: http://www.alchemyapi.com/

invoked by software developers to return a list of identified semantic instances from a given text, e.g. people, places, events, technologies.

Semantics can also be inferred from statistical analysis of crowd-sourced features (e.g. tags given by users on social bookmarking systems). Pera *et al.* (2009) built a word similarity matrix by identifying correlations in library keywords, in order to broaden the results of a search engine. By evaluating several term-based similarity measures against the Wordnet lexical database [1], Cattuto *et al.* (2008) observed that synonyms and spelling variants could be discovered among large tag sets (which are heterogeneous by nature in vocabularies and languages) using cosine similarity. Additionally, they observed that FolkRank (Hotho *et al.*, 2006) and co-occurence relatedness analysis performed better to identify hierarchies of concepts (hyponym and hypernym relations between tags).

At last, words (including tags) can be mapped to ontological concepts (Damme *et al.*, 2007; Tesconi *et al.*, 2008). However, this method requires human support, either from Internet users or from experts. In the first case, users are solicited for providing the semantics of the tags they want to use to annotate some content. With fact-tags (Amazon, 2008), users have to select the nature of their tags from a predefined list of categories and relations: subjective review of an item, characteristic of an item, comparison with another item. In Machine-tags (Cope, 2007), tags are to be prefixed by a predicate and a namespace referring to an ontology or a vocabulary, similarly to the RDF markup. In the second case, a mapping between words in natural language to the corresponding semantic representation are to be maintained by experts. Databases like Wikipedia or WordNet can be leveraged to generate a preliminary mapping (Szomszor *et al.*, 2008).

3.3.3 Discussion

In this section, we presented several models and techniques for finding similarities between user profiles and items, either semantic or term-based. We conclude that tags are good primary features to leverage, while keeping in mind that semantics can later emerge from their quantity without soliciting users.

[1] Wordnet: http://wordnet.princeton.edu/

3.4 Improving performance and scalability

In the previous sections, we have seen that the performance of vector-based matching systems can suffer from their high dimensionality: similarities are complicated to find, and their computation hardly scale when the number of users increase. In this section, we identify several techniques that could potentially be applied to improve the performance and the scalability of our awareness system, in the future.

3.4.1 Dimensionality reduction of term vector space

The performance of vector-based matching systems can be improved by reducing the dimensionality of the term vector space. In this regard, we have identified the possible use of incremental histogram-based statistics as an evolution of TF-IDF (Ciesielski *et al.*, 2007), and of leveraging semantic relatedness of terms as in the '*Generalized Vector Space Model*' (Tsatsaronis & Panagiotopoulou, 2009).

Latent Semantic Analysis can also reduce the dimensionality of term vectors, by removing noise and combining some of the dimensions towards a more conceptual classification (Hofmann, 1999). This technique (and its derivatives, including Latent Semantic Mapping (Bellegarda, 2007)) have been shown to improve the accuracy of a tag-based recommendation system (Wetzker *et al.*, 2009). Nevertheless, we note that this technique requires to learn from training data.

3.4.2 Tag clustering techniques

Hierarchical tag cloud clustering is proposed by Brooks & Montanez (2006) to cluster a set of tagged documents into a common/shared hierarchy of tags (based on tag co-occurrence). Emerging root nodes are general tags, whereas leafs are more specific tags. A similar approach was applied in the discerner and extender methods to transform concept maps into topic maps (Leake *et al.*, 2003).

In our case, co-occurrent leaf tags between two users' tag clouds would imply a more precise relevance measure and should thus be highly ranked than co-occurrent root tags. Additionally, users' contexts (partly extracted from a set of documents) are dynamically evolving over time, making this algorithm too costly to run every time a new contextual tag cloud is processed. Therefore, an on-line clustering algorithm should be privileged

to enable the scalability of the system. Growing Neural Gas (Fritzke, 1995) is a good candidate according to these requirements (Stan *et al.*, 2008).

3.5 Conclusion

In this chapter, we have presented several awareness and social matching systems, surveyed content-based and collaborative filtering techniques which we decided to combine into a hybrid filtering system, chose to focus on a keyword/tag-based model instead of a semantic model, and identified potential optimization techniques that we could adopt in the future for improving the performance of our system.

In the next chapter, we study how users' context can be modeled and transformed into features that can be leveraged in our context-based filtering system.

Chapter 4

Extracting and modeling contextual information

In this thesis, we support the hypothesis that, by sharing clues about their current context, one can implicitly solicit relevant information and communication opportunities, without having to make a phone call or to submit a query on the Internet.

In the previous chapters, we analyzed common user practices with social networking systems and studied the scientific techniques that can be leveraged for identifying relevant people and social updates, given a user profile.

In the aim of supporting the user in being notified of the most relevant signals, the challenges are to:

- identify what is the user's current task,
- determine which contextual entities are relevant to the execution of this task,
- and evaluate the relevance of social updates for a user, knowing his/her current task and context, and other users'.

In this chapter we will address the two first challenges listed above in order to build user profiles, by defining what we call '*context*', studying the state-of-the-art of context-aware applications, context models and context management systems, and identifying context sensors of three kinds: physical, virtual, and social.

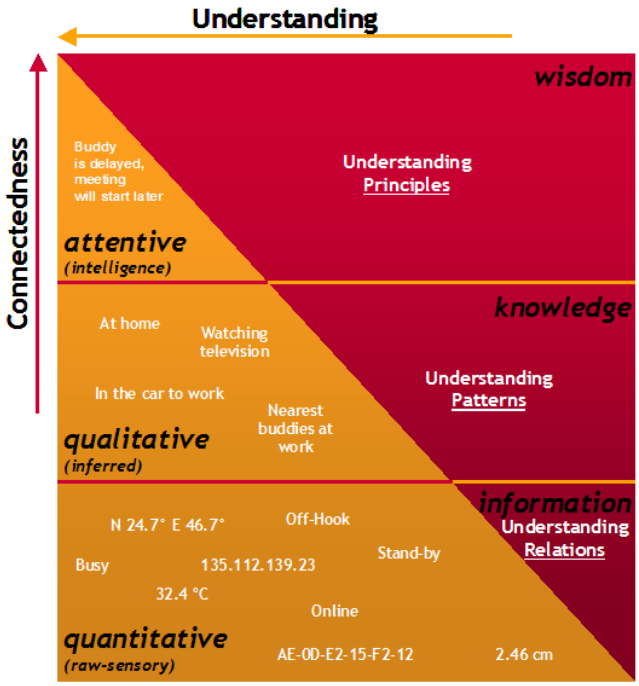

Figure 4.1: Contextual information: from raw sensor data to ambient intelligence - ©Bell Labs

4.1 What is context-awareness?

Context-Awareness is an area of Computer Science which deals with the adaptation of computing systems to the user's current context. Introduced by (Schilit *et al.*, 1994), the most recognized definition for the term '*context*' was proposed by (Dey, 2000): '*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*' By being '*aware*' of the user's context, a computing system can make deductions about the current situation of the user, from various hardware sensors such as GPS receiver (i.e. for positioning), surrounding Bluetooth equipments (and thus, surrounding people), and even software sources (called virtual sensors) like the user's agenda and the list of currently opened documents. As shown on Figure 4.1, by combining (i.e. aggregation of context data from several sensors) and inferring on this sensed information (i.e. interpretation, abstraction or enhancement of context), a meaningful knowledge can be derived, for instance position or activity like '*in a meeting*' or '*watching TV*' can be deduced.

Most research projects leverage the context of users of a system in order to adapt the interaction between the user and the system (Christopoulou, 2008), or for the system to predict users' goals and activities (Oliver *et al.*, 2002; Snoeck, 2007; Tapia *et al.*, 2004), and pro-actively make decisions intended to support the user in his future contexts (Nurmi *et al.*, 2005; Wang *et al.*, 2004a). Such applications include Intelligent Meeting Rooms (Chen *et al.*, 2004a; Leong *et al.*, 2005), Smart Homes (Gu *et al.*, 2004b), Personalized mobile advertisement (Zhdanova *et al.*, 2006) and electronic healthcare (Broens *et al.*, 2007). With the emergent popularity of augmented reality application on mobile devices, contextual information can also be displayed directly to users. Baldauf & Simon (2010) proposed to represent context in the form of '*ambient tag clouds*' which evolves while the user moves from a place to another, based on the geographical distance between his/her device and geo-referenced content.

In our case, we consider context as sampled information about people's environment and actions in time, and we propose to leverage this context to give relevant opportunities for users to communicate with each other. The aim is not to adapt human-system interaction, or to envision a pro-active system for user support, but to create relevant

links between people. Several context-aware social applications have been developed (Gaonkar *et al.*, 2008; Koolwaaij *et al.*, 2006), but the design of these applications was based on context-awareness (bottom-up approach), whereas we decided to rely on existing (and successful) SNS and their specific communication paradigm (top-down approach) in order to improve user acceptance.

4.2 Context management systems

4.2.1 From context-aware applications to context management systems

The first applications relying on context appeared in the 1990's, with the '*Active Badge*' location system (Want *et al.*, 1992) and the '*ParcTAB*' mobile computing system (Schilit *et al.*, 1993). At that time, context solely consisted of location information, applications were driven by a set of rules based on location changes, and relying on a specific sensor infrastructure.

After Dey *et al.* (2001) promoted Context Toolkit as a generic context management toolkit that could be leveraged by various context-aware applications as an intermediate re-usable software layer, researchers explored several ways to structure contextual information: key-value models, markup scheme models, graphical models, object-oriented models, logic-based models and ontology-based (semantic) models. Gu *et al.* (2004b) expressed the following requirements for context management systems: '*An appropriate infrastructure for context-aware systems should provide support for most of the tasks involved in dealing with contexts - acquiring context from various sources such as physical sensors, databases and agents; performing context interpretation; carrying out dissemination of context to interested parties in a distributed and timely fashion; and providing programming models for constructing of context-aware services.*'

Strang & Linnhoff-Popien (2004) concluded that ontology-based models (as introduced in Appendix A) were best suited for expressing context, according to six requirements towards the enablement of the ubiquitous computing vision: distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality and applicability to existing environments. At that point, most research on context-awareness was based on semantic context management systems (Baldauf *et al.*, 2007; Lassila & Khushraj, 2005).

4.2.2 Semantic context management systems

The use of ontologies (see Appendix A) to store and manipulate context have an impact on other aspects of the underlying system: context knowledge exchange, learning, user interactions, security (especially for privacy control) and applications. In this subsection we will review several semantic-based approaches for context management systems and identify the most critical lacks to overcome towards our goal.

One of the first semantic context modeling approaches was the Aspect-Scale-Context (ASC) model proposed by Strang *et al.* (2003). Compared to non-semantic models, ASC enabled contextual interoperability during service discovery and execution in a distributed system. Indeed, this model consists of three concepts:

- Aspects are measurable properties of an entity (e.g. the current temperature of a room)
- Scales are metrics used to express the measure of these properties (e.g. Celsius temperature)
- Context qualifies the measure itself by describing the sensor, the timestamp and quality data

Contexts can be converted from a scale to another using Operations, also described semantically, and can be mapped to an implemented service. This model has been implemented as the CoOL Context Ontology Language. The CoOL core ontology can be formulated in OWL-DL (Dean & Schreiber, 2004) and F-Logic (object-oriented). The CoOL integration is an extension of the core to inter-operate with web services. OntoBroker (Decker *et al.*, 1999) was chosen for semantic inference and reasoning, supporting F-Logic as knowledge representation and query language.

With EasyMeeting, Chen *et al.* (2004a) proposed a pragmatic application to demonstrate the benefits of their semantic context-aware system called CoBrA, for Context Broker Architecture. This application assists a speaker and its audience in a meeting situation by welcoming them in the room, dimming the lights, and displaying the presentation slides, either by vocal commands or automatically. The underlying prototype that they developed is a multi-agent system based on JADE [1] in which a broker maintains a shared context model for all computing entities by acquiring context knowledge

[1] Java Agent DEvelopment Framework: http://sharon.cselt.it/projects/jade/

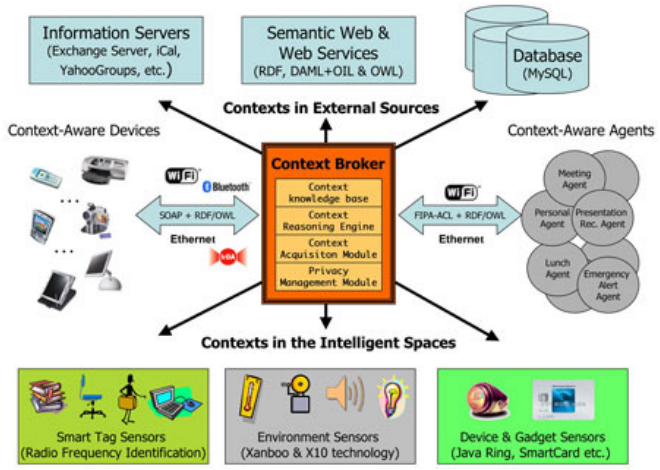

Figure 4.2: Overview of CoBrA - (Chen *et al.*, 2004a)

from various sensors and by reasoning on this knowledge to make decisions, as depicted on Figure 4.2. In the EasyMeeting application, this broker can deduce the list of expected participants and their role in the meeting by accessing their schedule, and can sense their actual presence when the bluetooth-enabled mobile phone declared in their profile is detected in the room. That way, the system can notify the speaker about their presence, decide to dim the lights and turn off the music when he arrives. These decisions are made possible by reasoning on the context knowledge using rules defined by the EasyMeeting application. The context knowledge is represented as RDF triples relying on the COBRA-ONT OWL ontology that includes vocabularies from the SOUPA ontology (Chen *et al.*, 2004b) covering time, space, policy, social networks, actions, location context, documents, and events, as depicted on Figure 4.3. Inferencing on the OWL ontology is handled by Jena's API [1] whereas the Jess rule-based engine [2] is used for domain-specific reasoning. The execution of rules (when results cannot be inferred from ontology axioms alone) uses the forward-chaining inference procedure of Jess to reason about contextual information. Note that, in this case, essential supporting facts must be extracted from RDF to JESS representation and the eventual results have to be injected in RDF to the knowledge base, which implies additional overhead in the process.

[1] Jena Semantic Web Framework: http://jena.sourceforge.net
[2] Jess rule engine: http://herzberg.ca.sandia.gov/

CoBrA's broker enforces privacy policies to define rules of behavior and restrict context communication. The enforcement of user-defined policies relies on the Rei role-based policy-reasoning engine (Kagal *et al.*, 2003) which does description logic inference over OWL. CoBrA also implements a meta-policy reasoning mechanism so that users can override some aspects of a global policy to define specific constraints at their desired level of granularity. However, they do not provide a tool for the user to express his/her privacy policy.

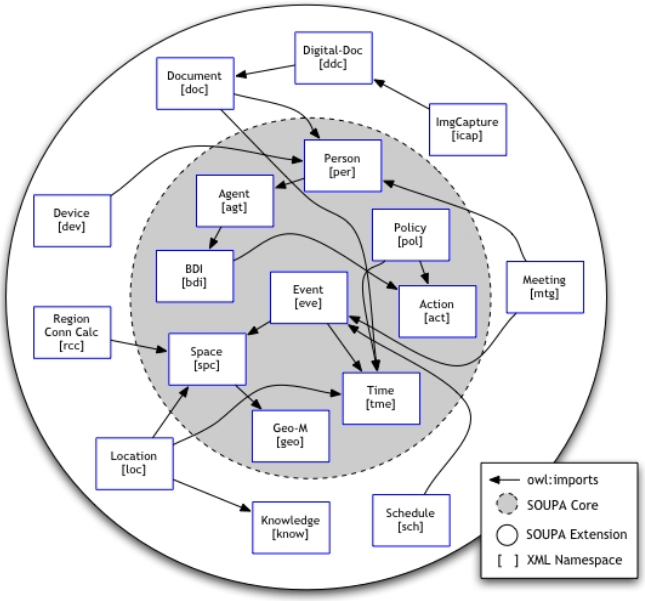

Figure 4.3: The SOUPA ontology - (Chen *et al.*, 2004b)

The SOUPA ontology proposed by Chen *et al.* (2004b) and used in CoBrA was a collaborative effort to build a generic context ontology for ubiquitous systems. Since 2003 it has been maintained by the '*Semantic Web in Ubiquitous Comp Special Interest Group*'. The design of this ontology is driven by use cases and relies on FOAF, DAML-Time, OpenCyc (symbolic) + OpenGIS (geospatial) spatial ontology, COBRA-ONT, MoGATU BDI (human beliefs, desires and intentions) and Rei policy ontology (rights, prohibitions, obligations, dispensations). SOUPA defines its own vocabulary, but most classes and properties are mapped to foreign ontology terms using the standard OWL ontology mapping constructs (*equivalentClass* and *equivalentProperty*), which allows interoperability. In the core ontology in which both computational entities and human users can be modeled as agents, the following extensions are added: meeting & schedule,

document & digital document, image capture and location (sensed location context of things).

Like CoBrA, MOGATU (Perich *et al.*, 2005) is a context-aware system based on the SOUPA ontology. However, this decentralized peer-to-peer multi-agent system implements several use cases covering automatic and adaptive itinerary computation based on real-time traffic knowledge, and commercial recommendation. In this approach, each device is a semi autonomous entity driven by the user's profile and context, relying on a contract-based transaction model. This entity is called InforMa and acts as a personal broker that handles exchanges with other peers. The user profile semantically defines his beliefs, desires and intentions, following the BDI model that is part of the SOUPA ontology. Beliefs are weighted facts depicting user's knowledge and preferences such as his schedule and food preferences, whereas desires express the user's goals. Intentions are defined as a set of intended tasks that can be inferred from desires or explicitly provided. However no clues are given by the authors about how these beliefs and intentions are defined by the user or the system, which let us assume that this is still a manual process yet to be enriched with profiling mechanisms and a graphical user interface to edit the profile. Moreover, this work being apparently focused on trusted peer-to-peer exchange of information according to the BDI user profile, details on the actual reasoning process on context knowledge are not given. InforMa is able to process queries that can possibly involve other peers and advertise information to these peers in vicinity, relying on graph search and caching techniques. However no details were given on how pro-activity is made possible. Another lack identified in the underlying BDI model is that the representation of pre-conditions and effects of intentions are left to the applications, but we have found no clues on how applications fill this issue. Facing an important cost of network transmissions in the exchange process, it seems that this research group is focusing on peer-to-peer networking optimization and trusted exchanges more than on the actual context management. However, they suggested that preparing purpose-driven queries in advance and caching intermediate query results could improve the performance of their system, which is an interesting approach that should be considered in distributed context-aware systems.

4.2.3 Uncertainty and quality of context

The CORBA-based GAIA platform proposed by Ranganathan *et al.* (2004) focuses on hybrid reasoning about uncertain context, relying on probabilistic logic, fuzzy logic and Bayesian networks. In their approach, context knowledge is expressed using predicates which classes and properties are defined in a DAML+OIL ontology (Horrocks, 2002). Predicates can be plugged directly into rules and other reasoning and learning mechanisms for handling uncertainty. This choice reduces the overhead of the CoBrA system relying on RDF triples. Rules are processed by the XSB engine [1], which is described as a kind of optimized Prolog that also supports HiLog, allowing unification on the predicate symbols themselves as well as on their arguments. HiLog's sound and complete proof procedure in first-order logic is needed to write rules about the probabilities of context.

GAIA's authentication mechanism demonstrates the usefulness of fuzzy/uncertain context reasoning. It allows users to authenticate with various means such as passwords, fingerprint sensor or bluetooth phone proximity. Each of these means have different levels of confidence, and some user roles may require that the user authenticates himself on two of them to cumulate their confidence level up to the required level.

Although GAIA proposes a common reasoning framework, application developers have to define the expected context inputs and specify the reasoning mechanism to be used by providing Prolog/HiLog rules (for probabilistic/fuzzy logic) or Bayesian networks. A graphical user interface is provided to help developers construct rules, whereas MSBN (Microsoft's Belief Network) can be used to create Bayesian nets. Although Bayesian networks are a powerful way to perform probabilistic sensor fusion and higher-level context derivation, they need to be trained. Moreover, inference with large networks (more than 50 nodes) becomes very costly in terms of processing and can result in scalability problems.

Based on previous works, Gu *et al.* (2004b) propose SOCAM (Service-Oriented Context-Aware Middleware), another OWL-based context-aware framework with the aim to address more general use cases by adding more qualitative information on acquired context. The classifiedAs property allows the categorization of context facts as

[1] XSB: http://xsb.sourceforge.net/

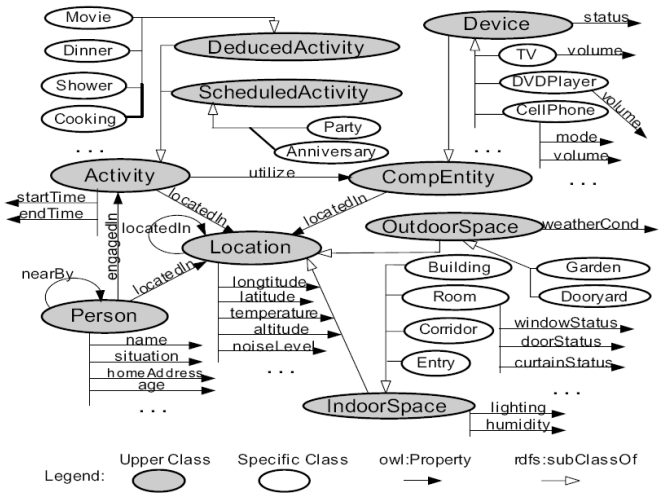

Figure 4.4: Partial definition of the CONON ontology extended with the home domain - (Wang *et al.*, 2004b)

Sensed, Defined, Aggregated or Deduced. The dependsOn property allows the justification of a deduced context based on other context facts. Another contribution is the possibility to qualify context information with parameters such as accuracy, resolution, certainty and freshness. The SOCAM framework was proven (Gu *et al.*, 2004a) to reason successfully on uncertain contexts using Bayesian Networks, but no performance results were given. The same group of authors have also carried out a performance experiment of the CONON ontology (Wang *et al.*, 2004b) depicted on Figure 4.4, which is the name that was given to SOCAM's context ontology. Their results show that the duration of the reasoning process exponentially increases with the number of RDF triples stored in the context knowledge base, which reveals that this approach is not scalable for a widespread context-aware system. Therefore two leads were proposed to increase performance:

- to perform static, complex reasoning tasks (e.g., description logic reasoning for checking inconsistencies) in an off-line manner.
- to separate context processing from context usage, so that context reasoning can be performed by resource-rich devices (such as a server) while the terminals can acquire high-level context from a centralized service, instead of performing excessive computation themselves.

Later works of that team were focused on the peer-to-peer architecture for context information systems.

Basing on the CONON ontology, Truong *et al.* (2005) proposed the PROWL language ('*Probabilistic annotated OWL*') to generalize fuzzy/probabilistic reasoning from applications to domains by mapping Bayesian Networks to ontology classes and properties. This approach must be experimented with various context-aware applications to prove its feasibility.

The FP6 IST project SPICE (Service Platform for Innovative Communication Environment) brought a fresh approach to ubiquitous system, considering them in a wider scope centered on semantic knowledge management for improved ubiquitous end-user services (SPICE, 2006, 2007). On its Knowledge Management Layer, SPICE proposes two different implementations of the context provisioning subsystem: the IMS Context Enabler (ICE) (Strohbach *et al.*, 2007) and the Knowledge Management Framework (KMF). In ICE, the SIP protocol (Session Initiation Protocol) is leveraged to control the parameters of the exchange sessions (e.g. data sets to communicate, update trigger, update frequency) and to flexibly adjust the communication path based on the changes in network structure and available context information. Both KMF and ICE rely on a shared ontology called the Mobile Ontology which is freely downloadable on the Internet [1], the most important difference being the interfaces: ICE uses SIP whereas KMF uses OWL over SOAP Web Services for exchanging context information. However, gateways are also provided so that context data can be converted from a format to the other. Therefore we will abstract these implementations and focus on the common knowledge model. Embracing the recommendations of the W3C, SPICE Mobile Ontology is defined in OWL and the context data is expressed in RDF. Inspired from the Dutch project Freeband Awareness, SPICE's Physical Space ontology has a finer granularity than any previous context ontology: it notably defines properties for connections between rooms and floors. Following the approach of the '*Doppelgnger User Modeling System*' (Orwant, 1995), SPICE's User Profile ontology supports domain-specific and conditional (situation-specific) submodels. In this approach, the profile contains subsets which are considered on certain conditions expressed with the form: Context Type, Operator, Value. This allows variations of the profile, depending on the user's context and/or the targeted application/service.

[1] SPICE Mobile Ontology: http://ontology.ist-spice.org/

The Knowledge Management Layer also contains a Knowledge Storage module, a Profile Manager, a Service and Knowledge Push and Notification module and three kinds of Reasoners: a Predictor, a Learner and a Recommender. The reasoners can request past knowledge directly from context sources or from an external knowledge storage source. Both feedback-based and observation-based learning are supported, generating *LearntRule* and *LearntRuleSet* instances in OWL. The results can be leveraged to propose Recommendations to the user. Experimental results on the use of different learning techniques are to be published. Another interesting contribution of SPICE in the context-awareness domain is the use of a *KnowledgeParameter* class that is used to qualify context information with values defining their probability, confidence, timestamp, temporal validity and accuracy. However we have not found any mechanism that is similar to the *dependsOn* property supported by SOCAM to justify high-level context with lower-level facts from which it was inferred.

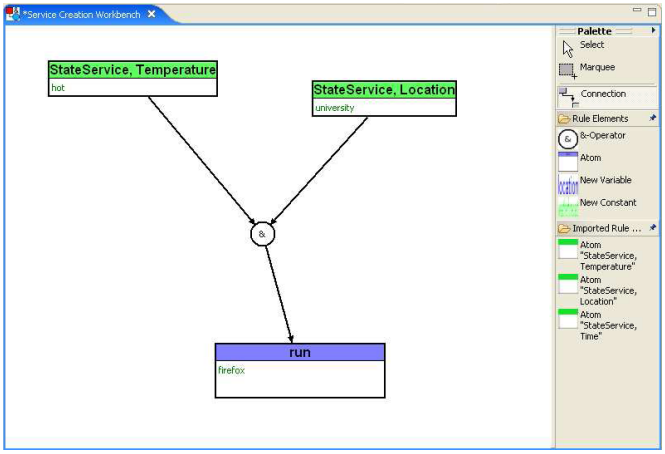

Figure 4.5: Creating a rule-based service using SPICE's End User Studio - (SPICE, 2007)

Another part of the SPICE project called the Distributed Communication Sphere (Kernchen *et al.*, 2007) allows dynamic discovery of users' surrounding devices, networks and services. This part includes components that leverage context knowledge to enable multimodal interaction, content delivery, data synchronization and dynamic widgets on terminals, requiring a lightweight rule engine to be deployed on every terminal. SPICE also provides the End User Studio, an Eclipse-based GUI (Graphical User Interface) shown on Figure 4.5 that allows end users to create custom trigger-action rules visually.

4.2.4 Popularization of context-aware applications

Since the middle of 2008, new open technologies have emerged on the Internet, providing location information to web pages (e.g. Google's Geolocation API [1], Yahoo! Fire Eagle [2]) and mobile applications running on smartphones (e.g. Apple iPhone [3], Android [4]) for free. Developing location-based application have thus become easy for developers, even beyond the industry and scientific communities. Contrarily to scientific efforts on context-awareness systems, numerous location-aware applications were brought to end-users (e.g. cell-based and GPS-based positioning in Google Maps mobile [5], Qype Radar [6], Layar [7]...) without relying on complex context models nor context management platforms. These applications have been very usage-specific, but also very functional, effective, and easy to use for a broad population.

Also in 2008, the Pachube website [8] proposed Internet users to share sensor data (e.g. temperature, humidity...) in real-time on their web platform, so that it could be leveraged by other users and applications. Today, Pachube can act as a simplified context management framework, allowing users to define rules that can trigger several kinds of actions depending on specified contextual conditions: notify people through their phone, notify applications or even activate devices and appliances. This platform can be seen as a first open and functional realization of the '*Internet of Things*' vision. Nevertheless, this platform does not rely on a semantic context model. Sensor data is handled as a neutral data stream, that are described with a name, a unit and tags for the sole sake of visual representation and classification of streams on the Pachube website.

Following an extensive review of context models proposed with context management systems from the scientific community, Bolchini *et al.* (2007) observes that the intended generality and expressivity of semantic models for any possible application actually reduce their usability. The growth of numerous specific web-based and mobile

[1] Google's Geolocation API: http://code.google.com/apis/gears/api_geolocation.html
[2] Yahoo! Fire Eagle: http://fireeagle.yahoo.net/
[3] iPhone: http://developer.apple.com/iphone/library/navigation/index.html#section=Frameworks
[4] Android Developers - Location and Maps: http://developer.android.com/guide/topics/location/
[5] Google Maps for mobile: http://www.google.com/mobile/maps/
[6] Qype Radar: http://www.qype.co.uk/go-mobile
[7] Augmented Reality Browser: Layar: http://www.layar.com/
[8] Pachube: http://www.pachube.com/

applications that can leverage location information without relying on context management platforms and models, support this conclusion.

4.2.5 Privacy concerns

As context management systems gather information about their users (e.g. their location), secure privacy enforcement mechanisms are to be implemented to avoid leaks of private information to unexpected parties (Lederer *et al.*, 2003).

The Context Toolkit from Dey (2000) features custom owner permissions to protect private sensor data by specifying access rights to authorized parties. Applications have authenticate to sensors, in order to access their data, through a public-key infrastructure. Most of its successors, such as CoBrA (Chen *et al.*, 2004a), Gaia (Ranganathan *et al.*, 2004) and SPICE (SPICE, 2006, 2007) rely on rule-based security policies for enforcing access rights to context information about users. In all cases, specific privacy-critical contexts have to be predicted to generate rules, which would be a tedious task for end users to execute.

According to the classification by Dourish (2004), the context-awareness frameworks we are reviewing here fall in the category of '*Architectures for Adaptation*' in which information structures are designed on '*predefined patterns*', without possibility for users to visualize nor manipulate their own context. Therefore, we can argue that the applications enabled by these systems require users to fit those predefined patterns, and to trust the system concerning the use of their private information, hoping without warranty that their privacy will not be violated.

The necessity for such rule-based policies, however, is implied by the seamlessness (i.e. no visibility of context, nor interaction expected with the context-aware application in this regard) that context-aware application intend to guarantee to their users. Dourish (2004) criticizes the lost of control (and thus, trust) from users, caused by this seamlessness: '*The flipside of the "seamless access" to which most mobile systems aspire is the systematic hiding of network heterogeneity; but it is precisely that heterogeneity that determines users' assessment of security.*'

Therefore we conclude that, in order for users to trust a context-based system, some control must be given back to them, even if this reduces seamlessness. We agree that context information should be structured, represented and integrated in a way that allows visualization, manipulation and appropriation by the users.

4.2.6 A problem of scalability

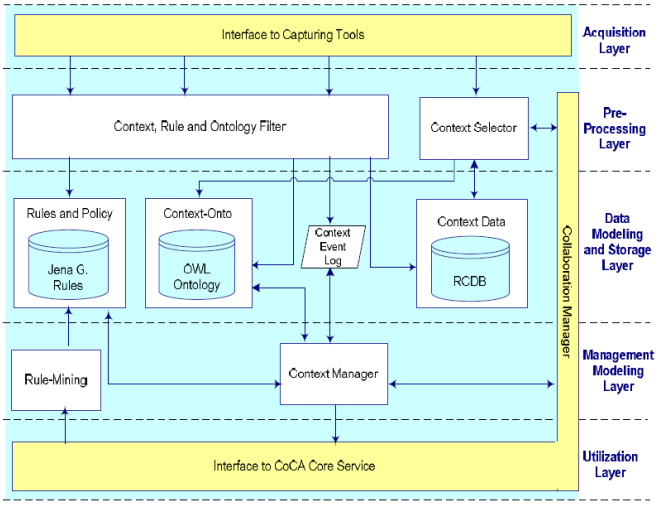

Figure 4.6: HCoM: Hybrid Context Management and reasoning system - (Ejigu *et al.*, 2007)

In previously reviewed semantic context-aware systems, we identified that the complexity of semantic inferencing and reasoning mechanisms imply a scalability problem. In response to this issue, Tan *et al.* (2005) propose to move from on-demand context reasoning to event-driven context interpretation so that reasoning on context data is processed as soon as it is received by the context management framework. Ejigu *et al.* (2007) proposed an hybrid context management and reasoning system (HCoM) which relies on a heuristic-based context selector to filter the context data to be stored in the semantic context base for reasoning, the rest being stored in a relational database, as depicted on Figure 4.6. They report that this approach is more scalable than pure semantic context-awareness systems when the number of static context instances increases. Lin *et al.* (2005) propose a similar approach but they filter context data according to their relevance to running applications instead of usage heuristics, in order to boost the reasoning performance. However, in their distributed system, the performance is reduced because of increased communication overheads. Moreover, it does not support uncertainty yet.

4.2.7 Discussion

In this section, we have reviewed several approaches addressing modeling, distribution and reasoning on contextual knowledge. Although semantic technologies have been shown as powerful enablers for empowering interoperable and generic context-awareness systems that can support heterogeneous sensors and applications, they also imply scalability, privacy and usability problems.

Concerning scalability, the complexity of semantic inference and reasoning mechanisms imply that required processing time grows exponentially with the amount of knowledge, which is a major issue towards the realization of context-aware applications. Hybrid context management approaches can reduce the amount of semantic computations, but as long as semantic reasoners with non-linear complexity are being used it is not possible to assert that these systems are scalable.

Concerning privacy protection, several security mechanisms have been implemented on context management systems for enforcing it. However, in the sake of seamlessness intended for context-awareness applications, these mechanisms rely on rule-based policies. As policies can differ for each user, they are expected to customize them, which is a complex task requiring them to anticipate privacy-sensitive situations before they occur. A trade-off is to be discussed between seamlessness and user intervention in privacy control mechanisms.

Concerning usability, it seems that, by intending to maximize genericity and extensibility, context-awareness researchers neglected important human implications: these systems and their applications could not satisfy end-users because they would bring too less effectiveness comparing to the efforts expected from these users (e.g. trust to a system sharing private information, difficulty of use). On the other hand, it has been observed that several context-awareness solutions (e.g. Location APIs, Pachube...) have been opened to the public on the Internet, giving birth to numerous pragmatic and usable end-user applications, without requiring a complex context model, nor a context management system. Several of these applications have met a very broad public, even if some people do not use them for privacy protection reasons. The trend is that each application (and service) has one specific purpose, and explicitly requires (or provides) specific context information. Users are free to combine several applications and services

(or create combined applications, '*mash-ups*') if they need to leverage more contextual information about them.

4.3 Physical sensors: Extracting and modeling geographical contexts

Geographical positioning has been one of the most considered category of contextual information. Over the years, technological advances enabled new ways to determine one's geographical location, each with specific quality and constraints.

Locations can be defined in several ways:

- Absolute geographic location: absolute coordinates of an object on Earth (latitude and longitude)
- beacon-based location: estimated position of an object based on its distance with at least one positioned beacon (e.g. a radio antenna including GSM, WIFI, Bluetooth or even RFID/NFC access points)
- movement-based relative location: estimated position of an object based on movement information, assuming that the position of the starting point is known, and that the actual position of the object is checked regularly
- semantic/structural/political location: this category does not address how to determine a location but what it means to users of the system. A location can have a political meaning (e.g. country names), a structural meaning (e.g. level in a building) or any other semantic (e.g. my grandfather's farm).

In this section, we present several methods to sense, model and aggregate contextual clues in order to leverage the resulting location information in context-aware software.

4.3.1 Absolute geographic location

The absolute geographic location of an object consists in a latitude, a longitude and possibly an altitude, which are represented by floating-point decimal values.

Such coordinates can be provided by GPS [1]. This system consists of a set of 24 satellites that orbit around the Earth, and that receivers can rely on to determine

[1] Global Positioning System

their position in real-time, when in range. The usual precision of most GPS devices is 10-20 meters, and 1 meter for D-GPS (Differential GPS). The main constraint for this technology is the quality of reception from the satellites. A GPS receiver requires a clear line of sight to the sky, and must be able to intercept signals from a minimum of 3 or 4 satellites simultaneously, in order to determine the current geographical coordinates in 2D or 3D, respectively. These constraints make it difficult to resolve indoor locations, because walls highly reduce the strength of signal from satellites. Additionally, GPS connectivity is costly in terms of energy, and can be slow to provide a first position when turned on. Nevertheless, this technology has become very usual, from driving navigation devices to most recent smartphones.

Geographical coordinates can be formatted using the Basic Geo Vocabulary ontology [1] proposed by the W3C Semantic Web Interest Group. This ontology defines a *point* class (extending the *SpatialThing* class) with the following properties:

- *lat* to store the latitude
- *long* to store the longitude
- *alt* to store the altitude

Note that GPS receivers also provide the following information: $gpsPositionDillution$, $gpsHorizontalDillution$, $gpsVerticalDillution$, $gpsSpeed$, $gpsUtcTime$, $gpsHeading$,

4.3.2 Beacon-based location

Whereas a GPS receiver can provide an absolute geographic position directly to applications, indirect alternatives also exist to determine a location based on other sensors. Because they are relying on fixed beacons, wireless networks can help. By beacon, we can mean: GSM cells (including Femtocell), fixed WIFI access points, fixed Bluetooth devices and fixed RFID/NFC readers. Mobile wireless devices can also be leveraged to determine a location, as long as their current absolute location is known by the system in real-time or near-real-time.

Note that, infrastructure-based location mechanisms, also relying on wireless network access points, can provide one's approximate position directly to location-aware

[1] Basic Geo Vocabulary ontology: http://www.w3.org/2003/01/geo/

services, like a GPS would, but this approach is thus equivalent to the previous *absolute geographic location*. In this part, we will focus on client-based location mechanisms relying on beacons.

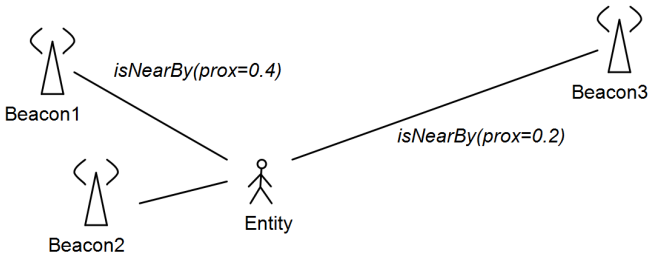

Figure 4.7: Beacon-based positioning -

In opposition to absolute geographic locations, and as depicted on Figure 4.7, the beacon-based location of an entity would be defined by one or more $isNearBy$ associations between $SpatialThing$ instances (including the entity which position is evaluate, and beacons in range). The geographical distance of those instances can be represented in a *proximity* attribute, or a *distance* attribute.

The *proximity* attribute can represent the signal strength between the entity's wireless network interface and each beacon. Such data can then be interpreted (e.g. by triangulation or fingerprinting algorithms) in order to compute the absolute position of the entity, assuming that the coordinates of the beacons are known.

In the case of a NFC/RFID, swiping a chip on a reader is equivalent to being in range of a wireless beacon, with a near-zero distance.

Cell-based location

In opposition to GPS positioning, cell-based (e.g. GSM) positioning is also possible indoors, as long as beacons are in range. Although this technology is less precise than GPS positioning, it is natively usable with cell phones without additional hardware and it's less power consuming.

Cell-based localization is the use of multi-lateration to determine the location of cell phones, usually with the intent to locate the user. Multi-lateration, also known as hyperbolic positioning, is the process of locating an object by accurately computing the time difference of arrival (TDOA) of a signal emitted from the object to three or more receivers. It also refers to the case of locating a receiver by measuring the time

difference of arrival (TDOA) of a signal transmitted from three or more synchronized transmitters. Multi-lateration should not be confused with tri-lateration, which uses absolute measurements of time-of-arrival from three or more sites.

Cellular phones can provide the identifier of the cell (antenna) on which the phone is connected. This information is useful to provide a rough but reliable location in most areas, as long as they are covered by the cell infrastructure.

Chen *et al.* (2006) experimented the use of a proposed client-based GSM positioning system in residential (26 cells/Km^2) and downtown (66 cells/Km^2) areas of Seattle. They utilized three different methods: the centroid algorithm, fingerprinting and Monte Carlo (with Gaussian processes signal models) to estimate their position and evaluate the error.

	Storage Required (for Seattle, compressed)	CPU Usage	Accuracy (Dense Towers)	Accuracy (Sparse Towers)	Required Density of Training Data	Requires Same-Device Training Set	Benefits from Cross-Provider Scanning	Tolerant of Phones Exposing Single Cell
Centroid	Low (44KB)	Low	232m	760m	Low	No	Yes	Yes
Finger-printing	High (188MB)	Med.	94m	277m	High	Yes	No	No
Gaussian Processes	Med. (80MB)	High	126m	196m	Med.	No	Yes	Yes

Figure 4.8: Comparison of three GSM-based positioning methods - (Chen *et al.*, 2006)

As shown on Figure 4.8, they realized that the fingerprinting method was the most accurate in downtown areas (dense towers), whereas Gaussian Processes were the most accurate in residential areas (sparse towers). Their experimental results show that existing GSM devices can achieve a positioning accuracy with a median error of 94 to 196 meters using cells from a single provider. Furthermore, they have identified an opportunity to significantly improve accuracy by scanning cells across all available providers, for a medium error of 65-134 meters, which is a factor of 3-4x of the published accuracy for WiFi beacon-based positioning projects.

WiFi-based location

Nowadays, technologies for wireless LAN are going through an implementation boom. Numerous wireless network providers are installing their systems in hotels, cafs, airports and other kind of buildings in which a facility of high speed Internet access is considered profitable.

In the area of tourism and cultural heritage, WiFi has been deployed experimentally in museums, archaeological excavations, hotels, fun fairs; obviously with the main aim of providing interesting services related to the user localization at any moment (e.g. providing more information about surrounding artefacts).

Wireless networks are widespread 75 meters inside buildings, and 300 meters outdoor, potentially attaining some kilometers by using specific antennas.

The WIFI connectivity of a device, similarly to cell infrastructures, can be used to locate it relatively to hot spots and other WIFI transmitters. This localization is made possible by triangulation or fingerprinting of signal strength with the surrounding WIFI devices:

- In the case of infrastructure-based coordinates triangulation, access points (APs) and their controllers detect visiting devices. The devices are seen with a certain signal strength by each AP; the APs also know their precise coordinates. Using the signal strength and coordinates of each AP, it is possible to determine the coordinates of the device. The advantage of this method is to map to any coordinates and locate any device. The drawback is to require complete control over the WiFi infrastructure. (Thornycroft, 2009)

- With device-based WiFi fingerprinting, the device itself learns to recognize sets of signal strengths from its surrounding APs, and to deduce positions where it has already been. The advantage of this method is to locate anywhere without controlling the WiFi environment. The drawback is to require intelligence on the device, and also a learning phase.

Bluetooth-based location

Bluetooth is an industrial specification for wireless personal area networks (PANs). Bluetooth provides a way to connect and exchange information between devices such as mobile phones, laptops, PCs, PDAs, printers, digital cameras, and video game consoles over a secure, globally unlicensed short-range radio frequency. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group[1].

[1]Bluetooth Special Interest Group: https://www.bluetooth.org/

The main use of Bluetooth is for short range (about 10 meters) connections between devices (e.g. PDA's, mobile phones and laptops). This technology is suitable for relatively static contexts, in which it is difficult to set a wireless network in a conventional way (e.g. inside cars).

Thanks to their unique MAC address, fixed Bluetooth devices can be used to '*tag*' (or check in) a place so that a visiting user can be located in that place.

Hybrid / Collaborative positioning

The Navizon [1] mobile service proposes a hybrid approach for positioning: it uses location knowledge of surrounding peers to locate devices when unable to use GPS localization. This knowledge can be extracted either by GPS, cell or WiFi positioning, depending of the capabilities of each participating device. This technology empowers a commercial product called '*Virtual GPS*' which returns approximate GPS coordinates using the surrounding Navizon peers, as if the host device was GPS-enabled. Note that Navizon also proposes location-based applications like GeoTags and local services. This technology is probably patented and no public research references have been found on this technology, but this hybrid approach is inspiring and demonstrates that peer-to-peer networks and collaborative platforms can be useful to enhance context data.

In the ContextWatcher software (Koolwaaij *et al.*, 2006), the Location Provider of IST project MobiLife can infer the location of a mobile user by GPS and/or GSM cell positioning information, and can use previously discovered mappings between GSM cells and GPS coordinates.

Since 2008, many web-based interfaces are openly provided to support application developers for leveraging location information inferred from several sensors (e.g. GSM/cell-based, GPS, IP routes), as introduced in the previous section: Yahoo! FireEagle, Google Location API, Android and iPhone APIs. Most of these APIs can adaptively rely on various positioning mechanisms, and they can represent the resulting location using various structures: approximate GPS coordinates, street-based address, or place name(s). In order to switch between those representations, geocoding (and reverse geocoding) services are also exposed on the web: the Google Geocoding API [2], the

[1] Navizon: http://www.navizon.com/

[2] Google Geocoding API: http://code.google.com/apis/maps/documentation/geocoding/

Geocoding API from Yahoo! Maps Web Services [1].

4.3.3 Movement-based relative coordinates

In some cases, the position of an entity can be computed relatively from movement data. This data can be provided by inertial sensors / odometers or vision-based algorithms (Gay-Bellile *et al.*, 2010). This data alone can provide the relative position of an entity over time. A mono-dimensional sensor (e.g. providing speed/acceleration information) can be sufficient if the entity is moving on a known path (e.g. on a rail). If the relative position of an entity can be mapped at some points to its absolute position (e.g. provided by a GPS receiver), this entity can be located with absolute coordinates with a decreasing confidence (i.e. the relative error increases).

For systems that cannot directly provide absolute coordinates, virtual/relative coordinates are to be gathered from sensors, and the context-aware system can map the virtual coordinates to absolute coordinates whether an absolute location is given by the same entity from time to time. Similarly to the previously-defined *point* class, we could define a *relativePoint* class, with relative coordinates in three dimensions. Every instance of this class must be associated with an instance of a *checkPoint* class which defines the function to convert virtual coordinates into absolute coordinates, for a given entity.

4.3.4 Political regions and logical spaces

A location may be represented as a reference to a geographical object in an ontology. Many geographical ontologies exist [2] [3]. Additionally to GPS coordinates, they often include notions of country, region, city, places, and other points of interest.

It may be necessary to combine with ontologies defining finer concepts such as meeting rooms and offices, train stations, business plants, warehouses etc. This is notably a unifying concept introduced by the SPICE project [4] which has the advantage of covering all sorts of locations in an object-oriented way: the complexity of choosing

[1] Yahoo! Geocoding API: http://developer.yahoo.com/maps/rest/V1/geocode.html
[2] WSMO ontology: http://www.wsmo.org/2004/d3/d3.2/b2c/20041004/resources/loc.wsml.html
[3] Space ontology: http://pervasive.semanticweb.org/ont/2004/06/space
[4] SPICE core ontology: http://ontology.ist-spice.org/mobile-ontology/0/10/core/0/core.owl

the location description is removed to a *location* class, that may have any number of representations.

Flanagan (2006) shows that clustering methods can be used to recognize such locations from available positioning sources. Additionally, '*geo-tags*' can be attached to real-world places and objects, allowing people to find digital information (e.g. local history, art, or communities) on the Internet about them. These geo-tags can be printed in the form of QR-codes, e.g. as proposed by Sotokolan [1].

4.4 Virtual sensors: context from computer-based activities

Computer software and network communications can also provide valuable contextual information about users' computer-based activities (Dragunov *et al.*, 2005), such as:

- the subject of web-based documents (including web pages) currently being browsed can be either analyzed using APIs exposed by web browsers, or by capturing network transmissions, e.g. through a proxy. Some particular events (e.g. posting a message on a forum) can be intercepted by subscribing to corresponding RSS feeds. Several web services can be leveraged to extract additional or crowd-sourced descriptions about web-based documents, e.g. tags from Delicious social bookmarking service;

- the subject of a local document (or email) currently being manipulated can be analyzed from Microsoft Word (Budzik & Hammond, 2000; Budzik *et al.*, 2000), or similar applications which expose a sufficient API;

- the current status of a software development task can be captured from IDEs (Integrated Development Environments) such as Eclipse or Microsoft Visual Studio, e.g. frequent unsuccessful compilations can reveal a technical difficulty (Carter & Dewan, 2009);

- additionally, the level of availability of the user can be inferred by analyzing mouse movements, keyboard activity, title of active window, screen-saver mode or even

[1] Sotokolan: http://www.sotokolan.com/

log on/off events. OS-specific daemon(s) must be developed in order to capture such events.

4.5 Social sensors: context from personal streams and crowd-sourced data

The Internet has become a digital space in which rich information has been collaboratively contributed by millions of people, through the use of dedicated web sites and services. At the so-called '*Web 1.0*' era, the content of web sites was mostly institutional or personal. But with the rise of '*Web 2.0*', many web sites opened their content to their audience, allowing people to contribute their knowledge, point of views, advices, and even personal status information and activities. Despite the unequal quality of those contributions, useful knowledge can emerge from their enormous and ever-growing quantity.

In this section, we firstly present some of those streams and crow-sourced web sites, then we identify how contextual information about people can be extracted from them.

4.5.1 Human messages as sensor data

Contextual information about people can be extracted from their recent social updates, as they are shared through the social networking systems they use. Several microblogging systems, including Twitter [1] already provide the possibility for their users to share the geographical location of their status updates.

By knowing the location of social updates, contextual information about the author's locality can emerge, like if the author was a sensor. For example, a social update containing a message like '*The power has been cut for 30 minutes now, I'm going for a walk outside!*' (which can seem useless even though many messages of this kind are observed on personal microblogging systems), provides the information that there is a power outage at the address of its author (which GPS coordinates were attached to the social update). If other people also post similar social updates from the same neighborhood, it can give a clue about the actual scale of the power outage. The only other sensors that could be queried for being informed of this problem are part of the infrastructure of the electricity provider. By sharing such observations as geolocalized social

[1] Twitter: http://www.twitter.com/

updates, people can quickly know that others are experiencing the same problem, and "power outage" can be identified as contextual information about the corresponding location and about the people that are currently located there.

Alternatively, some other contextual clues might also be provided as meta-data of social updates (similarly to '*hashtags*'), or could be queried from semantic web databases available on the Internet (e.g. LinkedData) (Stankovic *et al.*, 2009).

4.5.2 Web repositories of user-generated content

Launched in 2001, Wikipedia [1] was one of first public crowd-sourced web site. This free encyclopedia has been allowing anyone to edit the content of any article. Whereas this openness has implied many disputes on pages related to controversial subjects (e.g. facts about presidential candidates just before election, about historical events, companies, etc...), it has grown to become a major and useful reference, covering many languages. This encyclopedia has been translated to a semantic database called DBpedia [2] since 2007, enabling its user-generated content to be machine-readable, so that computer programs (and mashups) can leverage knowledge facts by formulating precise queries.

Even though wikipedia has been opened to any voluntary contributions, contributors are still few, compared to the number of readers. Participating in social bookmarking sites, like Delicious [3], have become more popular, as the contribution process was quicker, simpler and more personal. After creating a (free) account on the site, users can immediately bookmark web pages that they want to keep, because they enjoy them, they want to be able to easily find them later, and they (often) want to share them with other people. In order to make bookmarked web pages more easy to find later, users are invited to annotate them with '*tags*', unconstrained words (in any language, without even spell checking) that subjectively reflect the apparent nature, function, category and context of those web pages (Ertzscheid, 2008; Golder & Huberman, 2005). Web pages bookmarked (and tagged) by several people are thus described by a '*tag cloud*', a displayed set of tags which size depend on the number of people who used each tag to describe this page. As any URL-located resource can be bookmarked on social bookmarking sites, these descriptions can apply on various types on entities

[1] Wikipedia: http://www.wikipedia.org/
[2] DBpedia: http://dbpedia.org/
[3] Delicious: http://www.delicious.com/

represented by those resources. For example, tags given to a page that presents a car, are most probably associated to the car, than to the page/site itself. Now that web pages exist for almost anything on earth (e.g. people, objects, places, events, etc...), social bookmarking is a promising paradigm for gathering crowd-sourced descriptions and classifications of virtual and real entities.

More specific repositories also exist to represent and describe real word entities, and discover their involvement with people's activities. Concerning music, Musicbrainz [1] and Shazam [2] can identify the name and interpreter of a song from a sampled audio (e.g. recorded with a microphone), and tags given by people to songs and artists are gathered on web sites like Last.fm [3], which also maintains a history of the last songs that users listened to. Image sharing web sites like Flickr [4] can be considered as social bookmarking applied to photographs, as it is possible to tag one's own and other people's photographs, including the time and geographical location where the picture was taken.

Additionally, real-world places are described, reviewed by people and geographically located on various web sites (and their mobile applications) such as Yelp [5], Qype [6] and Google Maps [7]. And some events are announced on sites like Myspace [8] (mainly for major events, festivals, concerts, and other artistic performances), Facebook [9] (for commercial and private events) and Taweet [10] (more generally). Public events can also be advertised through social calendars such as Google Calendar [11]. Future personal plans (e.g. upcoming trips) can be advertised by people on sites like Plancast [12], Tripit [13] and Dopplr [14]. Rattenbury *et al.* (2007) have proven that names of places and events

[1] Musicbrainz: http://musicbrainz.org/
[2] Shazam: http://www.shazam.com/
[3] Last.fm: http://www.last.fm/
[4] Flickr: http://www.flickr.com/
[5] Yelp: http://www.yelp.com/
[6] Qype: http://www.qype.com/
[7] Google Maps: http://maps.google.com/
[8] Myspace: http://www.myspace.com/
[9] Facebook: http://www.facebook.com/
[10] Taweet: http://taweet.com/
[11] Google Calendar: http://www.google.com/calendar/
[12] Plancast: http://plancast.com/
[13] Tripit: http://www.tripit.com/
[14] Dopplr: http://www.dopplr.com/

can also emerge by analyzing the frequency and temporal distribution of tags associated to geolocated pictures.

Most web sites cited above expose public feeds that one can subscribe for being aware of last updates, and/or APIs that allow computer programs to query information, given specific criteria (e.g. information about a place, a topic, at a given time range). Thousands of other APIs are referenced on sites like Programmable Web [1]. Also note that tags are not directly available on all the web sites cited above, but keywords can be identified from the user-generated content they feature. It is also possible that pages from those sites are tagged on Delicious.

4.5.3 Extracting context by mashing up web streams and services

We have presented several web-based repositories that can be leveraged to gather information about real-world entities and activities. However, they do not directly provide ways for extracting context information about a person. In this subsection, we propose an example use case illustrating how they can be combined and processed in this regard.

Scenario: Alice is walking around in her neighborhood. She can hear some loud music a few hundred meters away from her, so she decides to walk in that direction. A few minutes later, shes arrives in front of a stage where a rock band is currently performing. She enjoys their music but does not now the name of that band, although she would like to. She grabs her mobile phone to check her contextual cloud, it contains the name of the band: '*Nemesis*'! She is so glad of discovering that band that she decides to capture this contextual cloud, and shares it to her social network. A few seconds later, her mobile phone vibrates to attract her attention back to the screen. Bob, a friend of hers, posted a social update about that concert a few hours ago: '*mad about not being able to see Nemesis' concert tonight :'-(*'. She decides to capture a short video of the concert, with a personal message for Bob, and shares it to the social network, knowing that Bob will receive the video. The metadata of the video are automatically populated from her contextual cloud: time, place, name of the band. Additionally, she realizes that even the name of the song is included. A few seconds later, Bob sends her a SMS: '*Hey, I didn't know you were going to Nemesis' concert*

[1] Programmable Web: http://www.programmableweb.com/

tonight, I'm jealous! You should come visit me when I'm finished, you'll have to tell me more about their performance!'

Awareness aspects: Thanks to her contextual cloud, Alice discovered the name of the band, without asking for it. By sharing this contextual cloud, she was notified that a friend of hers would have liked to come to that concert. This notification triggered an opportunistic communication between Alice and Bob, which motivated them for meeting each other.

Technical solution:

- When Alice arrived in front of the stage, her contextual cloud aggregated tags from social updates that have been posted by surrounding users, and from the city's federated concert agenda. Because '*Nemesis*' was appearing on most of those social updates, and on the event planned at her location, the corresponding tag was highly weighted.

- By sharing her contextual cloud, a social update was generated and shared to her friends, through their social network, making Alice's context visible to Bob. The matching system identified a social update which Bob posted about that band, the '*Nemesis*' tag being also part of his context when he posted it.

- When she captured a video, a music recognition service was invoked to identify the name of the song, which was added to Alice's contextual cloud.

- The tags of her contextual cloud were transfered as metadata to the video hosting system, so that other people can easily find it.

4.6 Conclusion

In this chapter, we have presented a state of the art of context management systems, from which identified lacks and opportunities for improvement that motivated the adoption of some requirements towards the development of our own context management system. In the following sections, we described three types of sensors that we intend to leverage in our application: (i) physical sensors and their associated techniques which can provide the location of users, (ii) virtual sensors that can extract tags and events from computer programs being used, and (iii) social sensors to extract additional tags

from social streams and crowd-sourced repositories hosted on the Internet. As an example of how those repositories could be leveraged, we proposed a use case and its enabling techniques.

In the next chapter, we will define a context management framework leveraging those three types of sensors, towards the development of social awareness applications.

Chapter 5

A Context Management Framework for Social Awareness

In the previous chapters, we have reviewed the state-of-the-art of social networking systems, awareness systems and information filtering techniques, and context management frameworks. We identified a need for filtering social updates in order to reduce the frequency of interruptions, enabling techniques that can be leveraged, and relevant context extraction and modeling methods.

In this chapter, we introduce and develop our approach for modeling contextual information and filtering social updates against it.

5.1 Aims

In this thesis, we intend to enhance digital social networking practices by supporting users with the sharing and filtering of information in high-scale communities, including professional/collaborative environments.

Current social networking systems rely on subscription to specific people's social feeds. This characteristic implies that reading all the social updates from subscribed social feeds is costly, in terms of attention, as their number increase. Furthermore, some users like social updates from their subscribed feeds to be displayed in real time on their screens, but it has been proven that frequent interruptions (implied by the notification of new social updates) can reduce the user's focus and performance on his/her main task.

Users confirmed that some support needs to be provided for them to cope with the quantity of social updates they receive. Novel information filtering techniques are to be studied to address this issue.

By supporting users with adapted ranking and filtering mechanisms, it is possible to reduce the quantity of notified social updates by selecting the most relevant ones.

According to Vogiazou *et al.* (2003), it essential to understand, analyze and convey the '*context*' of a collaborative or communicative act, in order for this communication to make sense. From this analysis, we identify context as a promising criteria for filtering and ranking social updates. In order to rank and filter social updates according to contextual properties, we need to gather contextual information about users.

Therefore, the proposed contributions of this thesis are:

- to model representative contextual information about users and their activities,
- to develop a filtering model that leverages contextual information for measuring the relevance of users' social updates,
- and to evaluate this context-based filtering model in a social awareness application.

5.2 Problem definition and scope

5.2.1 Hypothesis

The hypothesis that motivates our research is proposed as follows:

> '*a social update shared by person U on a social networking system is relevant to another person X if the current context of X is similar to U's context at the time of sharing.*'

This hypothesis raises the following questions:

- How to model contextual information about users and their activities, towards efficient relevance-based filtering of social updates?
- Where to gather contextual information from?
- On which kind of social updates does this contextual filtering perform best, and why?

- How to ensure that the user can control the transmission of potentially private information about his/her context and activity?

In this section, we address those questions, by specifying constraints and choices, proposing our approach and a methodology to evaluate it.

5.2.2 Constraints and choices

Quality emerges from quantity

In chapter 4, we have surveyed several modeling and management schemes for contextual information. It is clear that the current technological ecosystem allows gathering rich (however uncertain and heterogeneous) contextual information that could be leveraged for filtering social updates. In addition to data generated by physical sensors, we identified that additional contextual information can be gathered from software-based sensors ('*virtual sensors*'), and from humans ('*social sensors*'), thanks to social streams and crowd-sourced data which emerges from web-based services. However, we also identified that a semantic model would reduce the spectrum of contextual information that can be leveraged, because such models are framed by domain ontologies defined by experts, and thus can not cover concepts and entities that have not been defined in yet existing ontologies.

In order to gather rich contextual information, we prefer to leverage the quantity (in spite of semantic expressivity) of human-entered information, over the quality (but rareness) of semantically-represented information. This choice implies more noise and ambiguity than semantic approaches, but we support that quality naturally emerges from quantity of imperfect data.

Tangibility for user-centric privacy control

Information exchanged on social networks is personal by nature, but also explicit: users write social updates from their own decision, and with content they intend to share. On the other hand, as contextual information about people is to be gathered automatically, it might include information that they are not willing to share. A social application which relies on such contextual information thus implies some potential privacy issues. Even if this context is not directly visible to other people, the idea of sending and storing it on a server, somewhere on the Internet, is somewhat scary. Indeed, it might

be possible for complete strangers to access this critical information, either by manipulating the servers legally or illegally (e.g. hacking). A technical anomaly could also cause a leak of information to unauthorized audiences.

Additionally, in their definition of '*Systems that Display Their Context*', Dourish (2004) recommend that a context-based system should '*allow users to explore the technical context in which their applications operate*', so that users can '*make continual determinations of the potential consequences of their actions and their opportunities to reconfigure or realign the technologies through which they are conducting their actions*'.

Whereas it would be tempting to automatically (and thus, transparently) gather as much contextual information from users as possible in order to optimize relevance rankings, we support that users must be able to control whatever contextual information going to be sent to the Internet. In order to be humanly understandable and editable, the amount of contextual information must be reduced, so that each person can easily and quickly control (and manipulate, if necessary) contextual information that is going to be sent to the Internet.

5.2.3 Approach: a tag-based context model

Following these two constraints, and similarly to the Library Mirror application (Koch, 2005), we decide to model and represent contextual information as '*tag clouds*': a limited set of words, with various sizes depicting their individual representativity (e.g. importance and/or confidence of each word for representing the user's context).

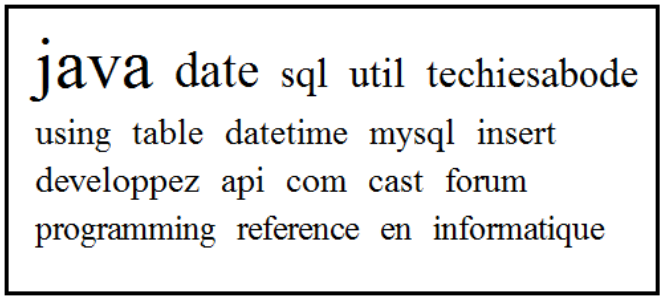

Figure 5.1: A sample contextual tag cloud -

With such a representation (as depicted on Figure 5.1), it is quick and easy for users to see what information has been gathered, modify some parts if necessary (e.g. remove some tags), and chose to share it, or not.

Additionally, the simplicity of this representation makes it possible to leverage numerous existing information sources, whatever the form (e.g. either semantic, structured or plain content), and for the user to add contextual information manually if desired or required.

5.2.4 Methodology

In order to validate this approach, we design the contextual and ranking models, then we apply this framework to address a specific case study in a closed contextual scope: a computer-based enterprise environment. Theoretical and software developments are thus twofold: (i) the common framework, and (ii) specific developments for the case study. The context model and ranking algorithms are evaluated by volunteers in the frame of an experiment. Limitations and future research issues are identified to improve the performance of the system.

Relying on a common framework, two prototypes are developed in parallel: (i) an '*on-line*' system that implements a interaction model responding to an '*enterprise awareness*' case study, in order to measure human acceptance and usability, and (ii) an '*off-line*' toolkit for evaluating the quality of relevance ranking algorithms, by generating personalized surveys from actual user data.

In this chapter, we present the common framework. The '*enterprise*' case study and application is presented and evaluated in the following chapter.

5.3 Interaction model

In this section, we specify the framework from the user's point of view.

5.3.1 General interaction flow

As depicted on Figure 5.2, the interaction flow is a loop which iterates from the *contextchange* event, which can happen at any point of this flow. Such change imply a *contextsynthesis* process which produces a *contextualtagcloud* that is displayed to the user. At that point, the user can decide to *edit* the contextual cloud before *sharing* it with the community. The *socialranking* process matches users with similar contexts, and returns a list of relevant *socialrecommendations* to the user, which he can *browse* or *contact*.

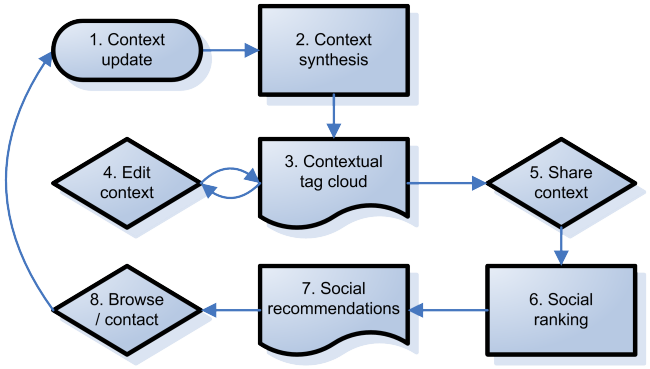

Figure 5.2: Interaction flow -

This flow complies with our previously specified requirements:

- The contextual information can be manipulated by the user, allowing him to add, adjust or remove contextual tags to better represent his context to the community without compromising privacy.
- No contextual information is ever shared before the user explicitly submits it. This also protects users from loosing privacy control.

5.3.2 Computer-based interaction

On a computer, the user benefits from a potentially large screen and a mouse as navigation interface. It is thus acceptable to allocate a reasonable dedicated portion of the screen for the graphical interface of this social recommendation system (e.g. displayed in a side-bar). In that case, the current contextual tag cloud and social recommendations can both be permanently displayed and dynamically updated, allowing the user to observe near-real-time evolution of the context as he interacts with his computer.

This set up allows a rich user experience. A dynamic display of the contextual tag cloud can make users want to share it frequently, and thus guaranteeing more numerous and recent social recommendations.

5.3.3 Nomadic interaction

As mobility in open environments imply frequent transitions through rich contexts, nomadic interaction is to be designed. In this case, one can assume that users are

carrying hand-held devices with compact screens and limited (and/or less reliable) interaction interfaces: a few buttons, a compact pointing solution, a touch-screen, motion recognition, and possibly speech recognition. Nomadic situations imply that users can not always benefit from a comfortable interaction experience: he can be disturbed by visual, acoustic, or mechanic interferences at any time.

With such constraints, and complying to modern smartphone operating systems, we propose that users can request the contextual tag cloud to be displayed by:

- pressing a specified button,
- touching a specific graphical icon,
- shaking the device with a specific motion,
- etc...

On some devices, it is also possible to allocate a permanent screen space that is visible as soon as the device is activated (e.g. a gadget in Android OS). Similarly to the '*Ambient Tag Cloud*' concept (Baldauf *et al.*, 2009), the contextual tag cloud will evolve dynamically as the user's context (including his location and surrounding people and other entities) evolve.

While using a hand-held device, we assume that the physical constraints are too restricted to manipulate contextual tags as easily as the computer-based version of the graphical interface proposed above. Thus we propose to allow removal of tags only. By sharing the contextual tag cloud, the screen would display relevant social recommendations in place of the contextual tag cloud. At that point, the user can browse those recommendations or switch back to the contextual tag cloud.

5.4 Contextual model

5.4.1 Contextual dimensions and information sources

Based on the sensors described in chapter 4, we consider several dimensions of contextual information that can be leveraged for evaluating the relevance of social updates.

User's activity

Information about on-going activities is essential for determining the potential relevance of a person and his/her social updates for another person. We propose some examples of tags, and how they can be extracted:

- *browsing*, *reading*: Such activities can be extracted using virtual and social sensors. Subjects and topics describing the content being read by the user can be provided by his/her web browser and/or other reading software and devices. Descriptions can be enriched by leveraging web-based repositories associating the content with tags, such as social bookmarking sites (e.g. delicious).

- *watching*, *listening*: The multimedia player(s) of the user can provide information about the content being played (e.g. a video, a song). If this content is being played from a web site, the description of this content might be extracted similarly to the previous example. In some cases, music logging services such as last.fm can also be queried.

- *meeting*, *attending*, *talking*: A planned meeting or event and its description can possibly be identified from the user's calendars (e.g. MS Outlook, Google Calendar), including events planned in social networks (e.g. Facebook, Taweet). In both cases, a speech analysis sensor can also infer the oral communication status of the user.

- *traveling*, *driving*: Such activities can be extracted from travel planning services (e.g. dopplr, tripit) in which the user has explicitly shared his/her travel plan. They can also be inferred by measuring the proximity of the user and a wireless beacon located the transportation means (e.g. the user's mobile phone is connected to a bluetooth equipment installed in a car), and a stable geographical movement (the transportation means can be inferred by the speed of movement and/or the correlation of its path with a roadmap). Clues about the destination and the reason of this trip can also be extracted, depending on the sensor.

All those examples can also be extracted from social updates relating to the user (e.g. '*@adrien, according to tomorrow's traffic previsions, you should take the national road instead of the freeway*' sent by a contact).

Surrounding environment, people and events

Additionally to the user's activity, his/her surrounding context (environment, people, unplanned events) can provide interesting attributes for finding relevant contacts and social updates. We propose a list of attributes and example tags:

- Current location, place: information about the user's current location (e.g. country, city, street address, venue) can easily be extracted by several physical sensors, supported with geocoding services (in order to transform geographical coordinates into a named place/address), and web pages describing the place (e.g. qype, yelp). Location-based tags can also be identified from geolocalized annotations (e.g. local photos from flickr)

- On-going events: assuming that unplanned events can be announced in real-time by user-generated content (e.g. status messages, pictures and videos shared with annotations), they can emerge from geolocalized social feeds.

- Environmental attributes (e.g. *crowded*, *noisy*) can be inferred from physical sensors such as the microphone of the user's mobile phone, but also from external sensors (e.g. surrounding microphones).

- Surrounding people (e.g. $Mike$, $JacquesChirac$) can be inferred from a recent mediated activity (e.g. a status update) of users that are in a close geographical range, but also from the short-range wireless connections (e.g. bluetooth) between the users' mobile phones.

- Broadcasted music (e.g. $RedHotChiliPeppers$) can be identified by music recognition services (e.g. shazam).

Future plans and intentions

We have identified that social update consumers enjoy discovering information related to longer-term plans and intentions. Figuring out users' goals can be complex, but some future plans and intentions can be extracted from users' calendars, as explained earlier.

Identifying such plans is important for our filtering model, as social networking systems are a good way for getting recommendations and advices from trusted contacts.

Those advices can possibly be relevant with the user's plans. For example, a friend's recommended venue is relevant if the user intends to go there sometime, especially if this venue is in a remote place which the user is not familiar with (e.g. another country).

Such plans can possibly be extracted from the user's calendar, or trips he/she advertised on trip planning services (e.g. tripit, dopplr).

Discussion

One can argue that the context should be structured (e.g. as an ontology) according to those dimensions and the identified attributes. We reply that those dimensions and attributes are not exhaustive, and that relying on a predefined structure would thus reduce the support for unexpected situations. We insist on the fact that, in our filtering approach, the semantics of contextual tags do not matter, as the relevance between people and their social updates relies on named entities, whatever their relation to the user. This intended lack of structure enables a wide combination of use cases, including:

- matching a user that is attending a concert of a band, with an other user reading a web page about that same band,
- recommending photos from a given place to someone who intends to travel in that region,
- notify users about to enter a metro station, that one of the metro lines is temporally stuck (e.g. as posted by someone else on that line),
- etc...

Modeling rules for every possible use case is not possible, therefore we believe that a structured context model is not needed.

5.4.2 Formalization: a tag-based contextual algebra

In order to represent and to manipulate tag-based contexts, we chose to rely on the well-known vector space model, as proposed by Salton & McGill (1986), and to define specific functions to attribute weights to every term. The weighting functions are

equivalent to *context interpreters* (Dey, 2000): they transform raw data (information from sensors) into higher-level information (tags), by abstraction.

Traditionally, a set of terms representing a document d is formalized as a vector of weights $v(d) = [w(t_1, d), w(t_2, d), ..., w(t_N, d)]$ attributed for each term $t = [t_1, t_2, ..., t_N \in \mathbb{R}]$. In our model, terms do not necessarily represent a document, they are human-readable attributes that can be extracted from physical, virtual and social sensors to represent some information about one's context. For each contextual sensor to leverage, a corresponding weighting function is to be implemented (or re-used), depending on the specific requirements of each application.

In the case of our computer-based awareness application developed in the next chapter, terms can be extracted from documents being currently viewed or manipulated, as part of the context of the user, but they can also be tags given by people to describe these documents.

In order to manipulate such weighted term vectors (also called contextual tag clouds), we propose the following algebra:

- The *addition operator (+)* yields a vector in which each term is weighted by the sum of the weights of that term in the provided vectors.

- The normalized form $\|v\|$ of a vector v conforms to $\sum_{t=t_1}^{N} \|v_t\| = 1$, with weight values $v_t \in \mathbb{R}$ in the range $[0, 1]$.

- The *aggregation operator* is the addition of given weight vectors, after their individual normalization. Thus, the aggregation operator applied to a set of vectors $V = [v_1, v_2, .., v_M]$ acts as the following function:

$$aggr(V) = \left\| \sum_{t=1}^{M} \|v_t\| \right\|$$

- The *similarity function* between normalized vectors, like in traditional vector-based models, relies on cosine similarity. Thus, the relevance of a vector R with another vector S is computed by:

$$sim(R, S) = \frac{R \cdot S}{\|R\|\|S\|}$$

  which returns a similarity score $r \in \mathbb{R}$ in the range $[0, 1]$, 1 being the maximum (i.e. contextual equality).

In the following section, we develop a social matching scheme and the corresponding relevance functions relying on this algebra.

5.5 Social matching scheme

5.5.1 The temporal dimension of social matching

In this section, we study the temporal dimension of social recommendations. According to the trends of usage of Social Networking Systems presented in Chapter 2, and Awareness systems presented in Chapter 3, we can imply the following statements:

- Most existing recommender (and filtering) systems leverage a history-based profile of recipients (i.e. people who consume content) to propose relevant content, possibly recent.
- Some systems recommend content that is relevant with the recipients' current activity and context.
- Current microblogging systems provide new social updates in real-time, without profiling nor filtering. Consumers can decide which feeds they want to follow: people and/or subjects (represented by hashtags, keywords, or queries).
- Microblog consumers do not necessarily read social updates in real-time, but many like to keep up with new updates (e.g. by keeping the feed, or the number of unread items visible on the screen).
- Microblog consumers mostly find social updates interesting and relevant more for longer-term objectives (planned activities, expected achievements, goals), than for their current activity.
- Filtering support is expected by microblog consumers.

From these statements, we identify the need for four social matching modalities, as depicted on Figure 5.3:

- **Profile-based filtering of new social updates**. Like in traditional filtering systems, history logs can be leveraged for building user interest profiles, in order to filter new social updates according to their similarity to those profiles. In our

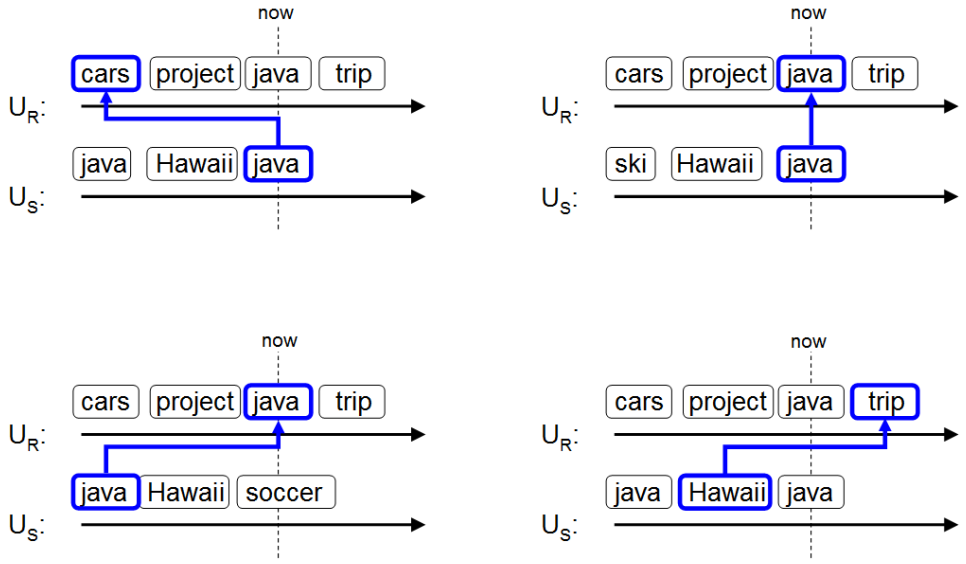

Figure 5.3: Different matching modalities - between a sender U_S and a potential recipient U_R, depending on their contextual time lines: profile-based filtering of new content (top-left), context-based recommendation of people (top-right), context-based recommendation of content (bottom-left), plan-based recommendation of content (bottom-right)

case, profiles can be a large set of weighted tags, containing many contextual clouds, over a wide period of time.

- **Context-based recommendation of people**. Additionally to reading social updates from selected people and subjects, new people can be recommended, accordingly to the user's current activity and context. This case is especially useful for getting punctual (and synchronous) support from relevant people on a specific issue, or for creating and enriching communities (of interest and practice), and it does not require users to produce social updates.

- **Context-based recommendation of social updates**. Social updates, even after some time, are a catchy way to promote, discuss or share opinions about a subject that is important to the person that sends them, providing a communication opportunity with this person. Previous social updates can be recommended to users whose current context is similar to the sender's at the time of posting, so that they can communicate asynchronously.

- **Plan-based recommendation of content**. Most microblog consumers think that social updates are usually more interesting for longer-term activities than for current ones. If users' goals, plans, and other future activities could be identified automatically, it would be possible to rank the relevancy of social updates according to those.

In this thesis, we are investigating context-based recommendation of social updates and people, and integrating profile-based information in this process.

5.5.2 General relevance function

Based on the algebra defined in the previous section, we propose a general relevance function that can be used to implement three of the four social matching modalities introduced above.

$$rel(S, R, P, \alpha) = sim(\|S\|, \|\alpha \cdot R + (1 - \alpha) \cdot P\|)$$

This function relies on:

- S is the current contextual cloud of the sender,
- R is the current contextual cloud of the recipient,
- P is the interests profile of the recipient (an aggregation of all user's contextual clouds),
- $\alpha \in \mathbb{R}[0, 1]$ is the contextual factor, to adjust the importance of the current context against the user's profile as criteria for evaluating the relevance,
- $sim(S, R)$ is the function that computes the cosine similarity between vectors S and R, as defined in the previous section.

In the following paragraphs, we explain how to use this general function to achieve three matching modalities. A major distinction between those modalities is the age of S, the contextual cloud of the sender.

In profile-based and context-based recommendation of social updates, relevance is to be estimated with the context of the sender S at the time of posting. The recipient's filtering criteria can be adjusted between his/her profile and current context,

by adjusting the value of the parameter α. With $\alpha = 1$, recommendations will be based on the recipient's current context. With $\alpha = 0$, they will be based on the recipient's profile (i.e. accumulation of previous contexts).

In order to achieve context-based recommendation of people, S must contain the current (or last) context of the sender, and α must be set to 1.

5.6 Design and implementation of the framework

In this section, we transform the conceptual framework defined above into a software framework, according to the object-oriented design and programming methodology. Similarly to the Context Toolkit proposed by Dey *et al.* (2001), we apply the '*separation of concerns*' paradigm by implementing some building blocks that can be leveraged as interfaces for developing context production modules on one side, and context-aware applications on the other side. That way, application developers can focus on the core functionality of their context-aware applications, while re-using context production modules possibly developed by other parties.

5.6.1 Data flow

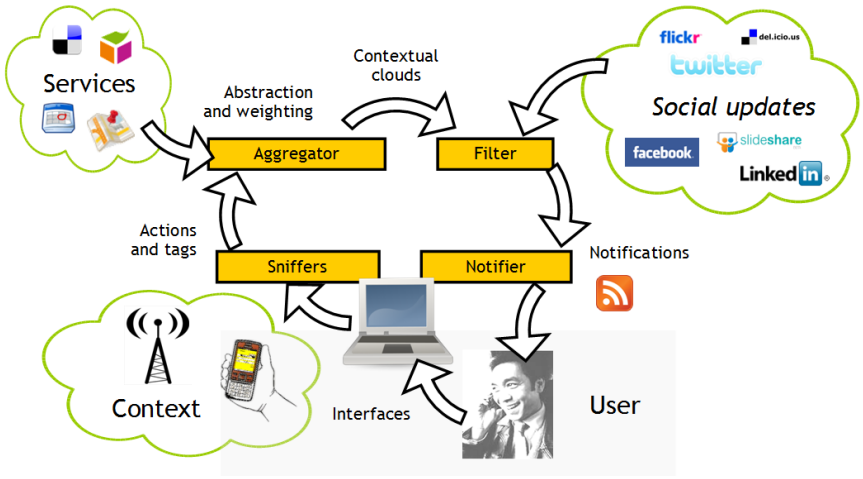

Figure 5.4: Overview of the contextual recommendation loop -

As depicted on Figure 5.4, contextual information is extracted virtual sensors (e.g. from user-manipulated content) and physical sensors by '*context sniffing*' modules. A

'*context aggregation*' module gathers contextual information from all these sniffers, and enriches it using social sensors in order to combine them in a '*contextual cloud*' that represents the user's current context. When submitting a '*social update*' (e.g. a '*tweet*'), this contextual cloud is submitted to the '*filtering system*' (i.e. a server running in the infrastructure) as a meta-description of the social update, so that the filtering system can match it with other users', using a context-based relevance function. This social update will then be '*notified*' to relevant people (e.g. users whose last contextual cloud is similar to the one submitted with the social update). That way, every user gets a dynamic (near real-time) list of recent social updates, sorted by decreasing relevance rank as their context evolves. Like with regular Social Networking Systems (e.g. Twitter), these short social updates can be quickly read by users to remain *aware* of relevant activities going on in their communities. They can also decide to reply to social updates or to call their sender.

5.6.2 Main structures

We are now presenting the classes that we have designed in Java, around the three following concepts: contextual clouds (data), social updates (events), and contextual matches (data+event).

Classes of contextual clouds

As explained earlier in this chapter, contextual clouds are formalized as term vectors weighted by real numbers, which can be normalized. Besides adjusting the weights, normalization of contextual clouds implies additional processing steps, such as: removal of stop-words and other excluded terms, and selection of most weighted tags. Normalized contextual clouds are thus finished products that are not meant to be modified, but instead to be exchanged between context producers (sniffers, aggregators, repositories) and consumers (applications, aggregators, repositories). Before this normalization, an intermediate structure is to be defined for context sniffers and aggregators to progressively synthesize and manipulate contextual clouds.

For those reasons, we have designed two classes $CtxCloudFloat$ and $CtxCloudNorm$, respectively encapsulating an intermediate structure and a finished product. The common structure and functionality of those classes is encapsulated in $CtxCloudFloatBase$, defining a term-to-weight hash map to store the weighted tag vector and exposing

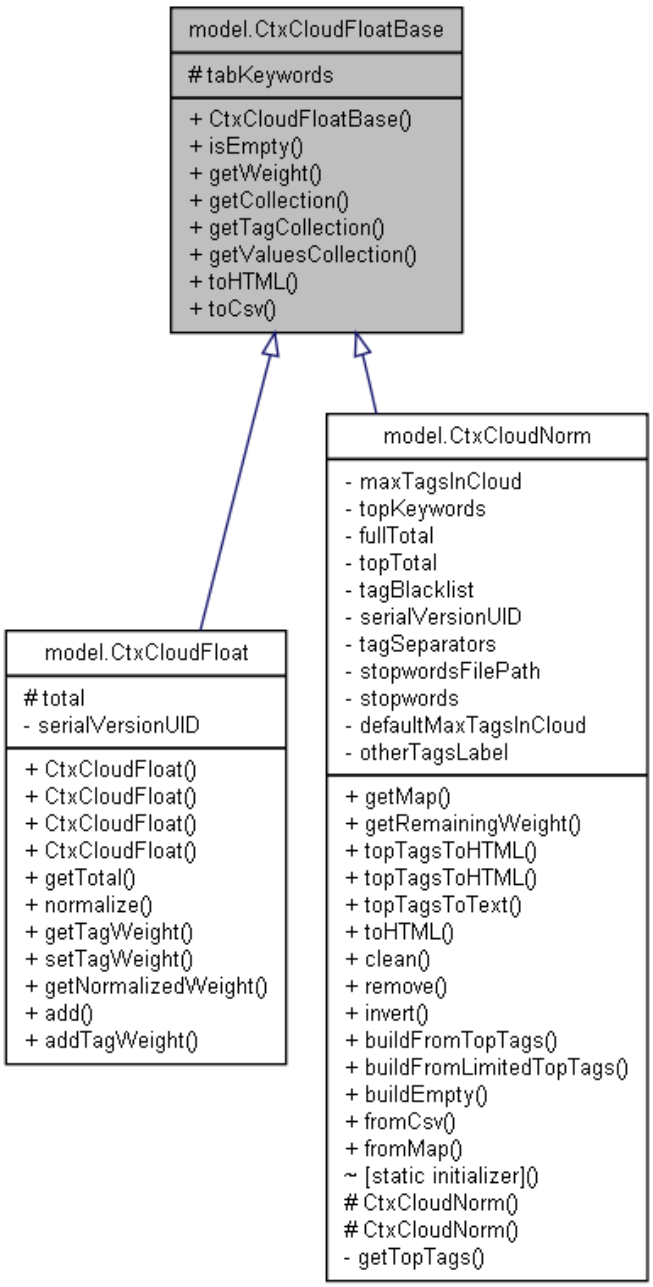

Figure 5.5: Hierarchy of classes encapsulating contextual clouds -

tag access functions returning either objects or output rendered as HTML or CSV. *CtxCloudFloat* extends *CtxCloudFloatBase* with tag-manipulation methods, including: weight retrieval (*get*), addition (*add*) and override (*set*). It also implements addition of contextual clouds. Calling the *normalize*() method normalizes and returns the contextual cloud as an instance of *CtxCloudNorm*, that also extends *CtxCloudFloatBase*. In that form, weighted tags are cleaned, filtered, sorted and projected as a vector of limited size, and additional import and export methods are exposed.

From events towards contextual indexing of social updates

In our framework, we have seen that a user's current context is represented as a contextual cloud. But the dynamicity and unpredictability of contexts imply that the framework must be able to receive updates at any time, in an asynchronous manner (e.g. from a context sniffer, or a context aggregator). In order to handle such updates/events, we defined a *CtxEventBase* class.

The *CtxEventBase* class encapsulates a contextual *cloud* (an instance of *CtxCloudFloat*), a *timestamp* and a description of the author (or *source*) of that cloud, and provides the corresponding getters and setters. Additional methods are also provided to import and export such events from/to hash map structures, HTML code and JSON objects.

Social updates are events thrown by social networking systems. In order to support contextual recommendation of social updates, the framework has to integrate those social updates, and also the context of their author at the time they were sent. For that reason, we extended *CtxEventBase* by two classes that combine a contextual event and a social update. Especially designed for external social updates, *CtxSocialUpdate* contains a *content* and a *link* attribute, respectively associated to the content of the social update to encapsulate, and its URL on the social networking system. We also added a *TweetEvent* class that only contains a *message* attribute, in order to support internal social updates, not relying on any third-party system. This internal social networking system will be further explained in the next chapter.

The aim of these combined structures is to ease the contextual indexing of social updates, so that previous social updates can easily be retrieved with their corresponding context.

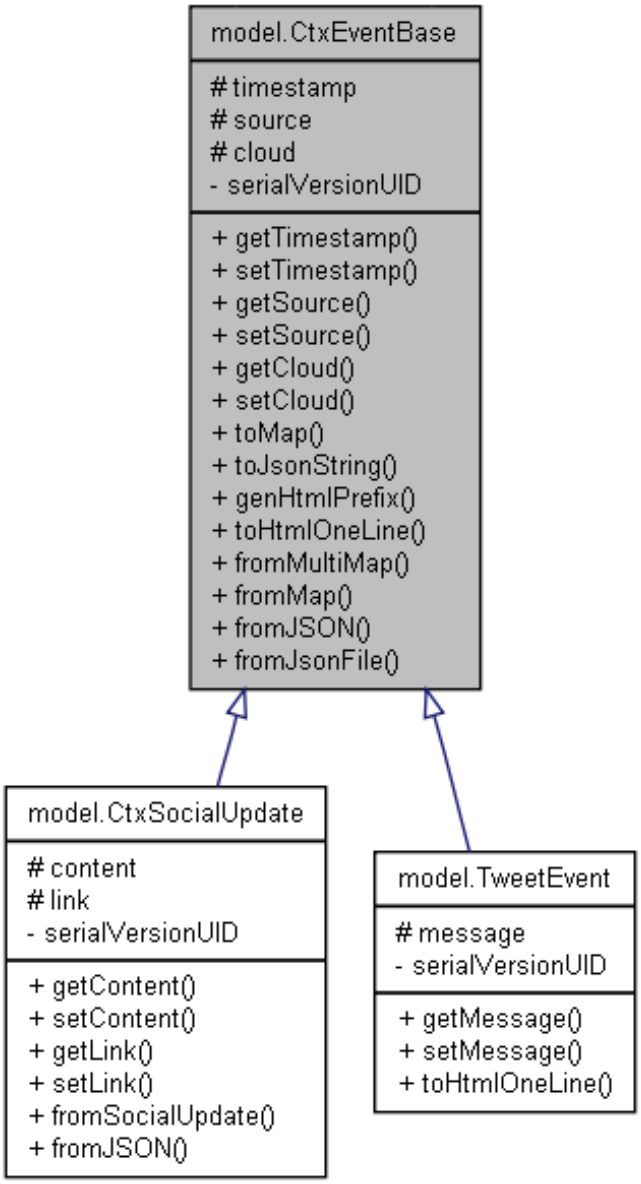

Figure 5.6: Hierarchy of classes encapsulating contextualized events - (including social updates)

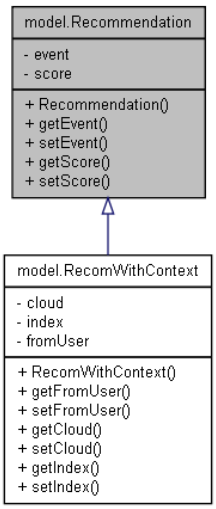

Figure 5.7: Hierarchy of classes encapsulating recommendations -

Recommending contextually relevant updates and people

When some social updates are indexed by context, and that the system receives a new context from one of the users, a set of relevant social updates are to be returned to this user, according to the similarity between the user's current context and the context of each social update.

In order to maintain of list of relevant social updates, a *Recommendation* class was defined, associating a contextualized *event* (e.g. a *CtxSocialUpdate* object) to the similarity *score* computed between the context of this event and the context of the user. Most relevant social updates can then be sorted by decreasing *score*, and notified to the user, in response to his/her new context.

An additional class, *RecomWithContext* extends *Recommendation* to also refer to the user whose this recommendation concerned, his/her contextual cloud, and its position in his/her contextual index. This class is useful for generating comprehensive recommendation reports from various users, and their previous contexts at once (e.g. for a batch recommendation process).

5.6.3 Using the framework for real-time matching

In this subsection, we present the classes and methods to use for creating a real-time awareness system, as depicted on Figure 5.4, relying on our framework.

Submitting contexts and social updates

As depicted on Figure 5.4, the filtering system developed upon our framework, is to be fed by user contexts and social updates. We recommend generating these contexts in two separate steps: context sniffing, and context aggregation. These processes are preferably implemented in dedicated modules, but they can possibly be implemented in a single module, as long as it can produce normalized contextual clouds and submit them to the system (as instances of the $CtxCloudNorm$ class). Nevertheless, we will explain this process in two steps.

A context sniffing module is responsible for observing contextual clues, and notifying a context aggregator of valuable events and information. An event is valuable when it can have an impact on a user's context. The implementation of sniffing modules, and their communication with context aggregators is independent of the framework.

A context aggregator is a module that synthesizes normalized contextual clouds from valuable events concerning a user, and submits substantial changes to the framework, after user confirmation. It can possibly receive such events asynchronously from various context sniffers. In that case, it is responsible for abstracting the meaning of those events into tags, and aggregate those tags progressively into a $CtxCloudFloat$ instance, using the $setTagWeight()$ and $addTagWeight()$ methods. When this contextual cloud contains comprehensive (or sufficient) description of the user's current context, it must be normalized into a $CtxCloudNorm$ instance, returned by the $normalize()$ method.

According to the requirements presented earlier in this chapter, contextual clouds must not be sent to a remote server (e.g. the filtering system) before having been shown and confirmed by the user. It is possible to implement a contextual cloud manipulation interface allowing the user to edit this cloud before sharing it, using the methods provided in the previous paragraph. Additionally, the user interface may allow the submission of a social update to be attached to this context. When the user confirms the submission of a contextual cloud and/or a social update, they must be sent as a $CtxEventBase$ instance (or derivative) to the filtering system. The interface of this filtering system is not specified in the framework, however the event must be passed in its normalized form to the $onEvent()$ method of a class which implements the $IEventListener$ interface.

In our implementation, we have developed two filtering servers exposing a RESTful interface. Events are to be submitted by making a HTTP POST request to the *event* entry point (url: '*http://filtering_server_host/event/*'). The *CtxEventBase* instance is expected to be serialized as a JSON string, in the body of the POST request, using the *toJsonString*() method. The *JsonSender* class implements this submission process, and is convenient to use with our servers.

For handling submitted events, our two servers rely on a common reference implementation of the *IEventListener* interface: the *ContextManager* class. This class holds the last events for each user (submitted by their respective context aggregator), evaluate the similarities between the corresponding contexts (using the '*Algo.getCosineDistance()*' function, and dispatch events, context updates and resulting matches to subscribers.

Subscribing to matches

In our reference implementation of the *ContextManager* class, three kinds of subscription are provided:

- The *subscribeToEvents*() method registers objects that implement the *IEventListener* interface. When a new event is received, it is transmitted to subscribed objects by calling the following method on them: *onEvent*(*CtxEventBase*).

- The *subscribeToMatch*() method registers objects that implement the '*IContextMatchHandler*' interface. When a similarity has been evaluated between two contexts, the following method is called on every subscribed object:
  '*handleContextMatch(String user1, String user2, CtxCloudFloat overlap, float similarity)*'.
  *user*1 and *user*2 identify the users whose contexts have been compared, *overlap* is a contextual cloud that contain the tags that appear in both contexts, and *similarity* is the relevance score in the range $[0; 1]$.

- The *subscribeToContext*() method registers objects that implement the *IContextListener* interface. When a new context is received from a user, the following method is called for each *tag* of this context and of the previous one from this user, on every subscribed object: '*onTagChange(String tag, Float weight)*'. *weight* is the weight of *tag*, in the new context. Tags that were appearing in the previous cloud only

are associated a *weight* value of 0. Notice that, in the current implementation, it is not possible to filter the subscription for a specific user only.

5.6.4 Using the framework to generate batch matching reports from logs

In order to evaluate the quality of contextual clouds and contextual relevance of social updates, an off-line process with intermediate output is required. In this process, users' actions and social updates are logged for being processed later, in a batch sequence.

Most tools we developed are independent from the type of contexts to process, except the users' activity logging tool and the processing of social sensors, which were required to evaluate the framework, as defined in the next chapter. In this evaluation, users' activities are web browsing events logged from a web browser. Those logs are timestamped lists of URLs which were opened, closed or focused. The social sensors that we developed extract tags describing the content located at those URLs.

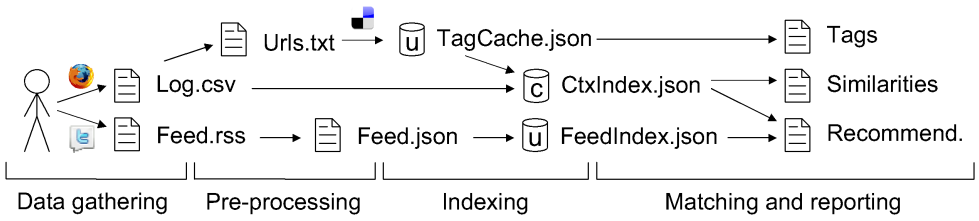

Figure 5.8: Batch sequence, from logs to reports -

The batch sequence, as depicted on Figure 5.8, consists of:

- extracting the list of URLs to submit to social sensors, from users' activity log (in CSV format);
- the social sensors process the list of URLs and store the resulting tags into a common tag cache (for all users, in JSON format);
- from the users' activity log and the populated tag cache, users' contextual clouds are generated and indexed on a common timeline to a JSON file for each user;
- the users' RSS feed containing their social updates are converted to a JSON file;

- social updates are indexing on the common timeline, and stored in another JSON file;
- finally, several reports can be generated about tags, contextual similarities, and contextual recommendations of social updates.

In this subsection, we present the tools, classes and methods of our framework, to log user activity, and to generate batch reports (and intermediate steps) from those logs.

Logging user activity

In order to gather contextual information about a user, some events related to this user are to be observed, including user's actions. In a real-time setup, those events are captured by context sniffers in order to synthesize a contextual cloud on the fly. In a sequential/batch process, those events must be logged for later processing.

For evaluating our application, as described in the next chapter, we adapted a context sniffer into a context logger. This context logger works as an extension of the Firefox web browser. Participating users have to install this extension, so that their browsing events (e.g. opening, closing, switching pages) are transparently tracked and logged into a local database. Another tool exports this log into a CSV file called '*user log*' or '*actions log*' that can be edited by users (for privacy reasons) and parsed from scripts provided with the framework, as explained in the following subsections.

Gathering and caching tags

Some context sensors rely on web services to turn user actions into meaningful tags. This particularly applies to what we call social sensors, because the corresponding information is extracted from web-based messages, but it also applies to some virtual and even physical sensors (e.g. geo-coding: transforming geographical coordinates from a GPS sensor into the name of the corresponding place). In a real-time application, these web services can possibly be queried on demand. On the other hand, generating batch reports from logs of past user actions in an experimental setup requires a stable mapping between actions and tags. One reason is that researchers need to be able to reproduce the exact same results from the same logs, whatever the technical problems

that can arise from querying web services (e.g. connection problems, interrupted services). This constraint motivates a pre-fetching and a caching of tags into a repository we call '*tag cache*'.

In order to populate this tag cache, URLs or web queries that would be invoked by context sniffers and aggregators must be gathered in a list, so that they can be invoked at once for all users logs. Notice that, gathering the URLs of all users also implies a removal of duplicates, and thus ensures that the same tags will be used for a same query. This effect will not necessarily be observed in a real-time setup, as each user's aggregator invoking a same web query might give different results, depending on the time of invocation, and of network conditions. This first step is achieved by a script called *extractURLs*. It generates a list of URLs that can be exported to a text file, extracted for a list of action log files (normally, one per user) provided as command line parameters.

From this list of URLs, the *populateTagCache* script generates a tag cache, consisting of a list of *CtxResourceReport* instances serialized as JSON strings. The *CtxResourceReport* class associates a resource identifier (in our case: the URL of a document that has been browsed by the user) to a *CtxEventBase* instance which contains the contextual tags associated with this resource, and the identifier of the weighting function (also called interpreters, or *enhancers*) that was used for extracting these tags. This output is thus human- and machine-readable, and will be leveraged by the next step of the process. The list of weighting functions that will be used to extract tags can be configured in the script itself.

As a tag cache might have been generated from a restricted set of weighting functions, we also provide a *mergeTagCache* script that aggregates two or more tag cache files into a common tag cache.

Indexing user contexts

In a real-time setup, context events are submitted asynchronously by users, and matches are based on each user's last submitted context. In a experimental log-based setup, users are not solicited, thus context events do not exist. In order to simulate these events, a common and synchronized pulse is to be defined. At each pulse, the actions that had been logged since the last pulse are considered as part of a new contextual

cloud, like if it was submitted by the user at the time of that pulse. That way, it is possible to index contextual clouds of several users on a common time-line.

For achieving this process, the framework provides a script called *indexContexts*. By passing the tag cache and a list of action logs as parameters, this script generates a indexed list of contextual clouds per action log. Each contextual cloud is represented by a JSON-serialized $CtxCloudNorm$ instance, prefixed with the index number corresponding to the timestamp of this contextual cloud (i.e. of the pulse, in our experimental setup). The index number is computed given the time-range of the experiment (in '*epoch*' format), using functions provided in the *Constants* class. Each index file begins with a heading line that contains the time-range of this index, and the frequency of the pulse, also serialized as JSON strings.

Importing and indexing social updates

Additionally to action logs, social updates posted by users must also be listed in a log, for processing. As most social networking sites expose RSS feeds of social updates, it is possible to gather a history of users' last social updates from every service they posted to. We developed $fetchFeed$, a script that can extract social updates from a RSS feed, and store them in a file as a list of $SocialUpdate$ instances, serialized as JSON strings.

As for contexts, social updates must then be indexed on the common time-line. For that, the *indexFeed* script transforms a list of JSON-serialized $SocialUpdate$ instances, provided as parameter together with their author's context index, into a list of $CtxSocialUpdate$ instances, also serialized as JSON strings. The $CtxSocialUpdate$ class extends $CtxEventBase$ with two additional attributes for attaching the *content* and *link* of the social update. This operation is very similar to the contextual indexing process described earlier. It results in a social update cache file, in which social updates are indexed by pulse number, and associated to the context of their author at the time of posting (or the last known context before posting).

Generating matching reports and charts

Once users' contexts and social updates are indexed on a common time-line, it is possible to generate reports, charts, and user surveys which include contextual recommendation of social updates, and context similarity analyses. We provide the following scripts:

- *genCtxReport* generates a CSV worksheet containing a list of most appearing tags, and most weighted tags, from one or several context index files.
- *genCtxLogChart* generates a CSV worksheet which can be easily rendered as a gantt-like chart to represent the repartition of users' not-null contexts on a time-line, given the corresponding context index files.
- *genCtxSim* generates two reports, given users' context index files. CtxMatchReport is a CSV worksheet that contains a matrix providing the cumulative number of contexts that are similar to other users' contexts, per user and per similarity threshold. CtxSimReport lists, for each user pair combination, the twenty most similar contexts, with their score and top common tags, between those two users.
- *genCtxSimChart* generates a CSV worksheet which can be easily rendered as a chart to represent the repartition of users' similar contexts on a time-line, given the corresponding context index files. The similarity threshold and the size of the temporal window can be set. The temporal window is the maximum temporal distance between two contexts which similarity will be evaluated, in number of pulses. From this worksheet, it is possible to render a dot-chart of similar contexts, in function of their similarity score and time.
- *recommender* generates a reports which lists the relevance scores between every context and social update, given context index and social update index files.
- *genSurvey* generates survey forms in HTML, given a set of user's context index and social update index files. For each user survey, five contexts are selected and displayed on the form. And for each context, three social updates from other users are selected and displayed on the form. Each user can rate contexts, and the relevance of proposed social updates to those contexts. At the bottom of the page, a button enables to send those ratings by email.
- *genCtxFeedReport* generates a survey form in HTML, given a user's context index and social update index files. For each social update posted by the user, the corresponding contextual cloud is also displayed, and the user can rate the perceived relevance of his/her social update with this context. At the bottom of the page, a button enables to send those ratings by email.

5.7 Conclusion

In this chapter, we have specified the requirements of our context management framework, proposed to model contextual information as '*contextual clouds*' (based on the vector space model), and a methodology for building and evaluating our filtering application based on this approach. Then, we identified several dimensions of context that can be leveraged for measuring the contextual relevance of users, and we designed an algebra for manipulating contextual clouds. We identified four filtering modalities, proposed a data flow and defined a general relevance matching function for our filter. Finally, we described and explained the implementation of this framework, the software modules involved in a real-time filtering application, and in the generation of experimental reports to evaluate the quality of the algorithms.

In the next chapter, we develop a social awareness application upon this framework, and propose an evaluation of the underlying contextual model and filtering model.

Chapter 6

A Social Awareness Application and its evaluation

In the previous chapter, we designed a tag-based context management framework that can gather contextual information from physical, virtual and social sensors, and a general social matching scheme that can be used for filtering social updates.

In this chapter, we propose to consider an enterprise environment as a case study, and we apply the general social matching scheme by implementing a social awareness application, called '*Enterprise Contextual Notifier*', for this specific environment. This application considers users' web-browsing context as relevance criteria for filtering social updates between employees.

6.1 Case study

As a motivating case, we propose to consider an enterprise environment where employees work on individual networked computers. Employees knows just a portion of other employees (mostly people working in the same department), and are spread across several offices in several cities. This traditional enterprise is organized following a hierarchical structure of managers to coordinate the efforts of workers and transfer information to the relevant parties.

We propose to set up an internal social networking system that allows every worker to share and be aware of the current status of relevant colleagues and their on-going professional activities. The time necessary for maintaining this network must be low,

and unnecessary task interruptions must be prevented. This social networking system must allow employees to reflect their current thoughts, activities, intentions and questions, support them for getting feedback and answers from relevant colleagues, to promote transversal communication and collaboration across hierarchical silos.

6.2 Application of the approach

6.2.1 Enterprise context model

As users are working on computers in the case study presented above, most contextual information about them (and their current activity) can be extracted from the software they use. In particular, we consider that their context is defined by the projects, subjects and topics covered by the web pages they are currently browsing. Indeed, the web is often used to find some reference on an ongoing task, and thus the description of these references can give a clue of the user's current activity. Therefore, users' contextual clouds will be populated with tags that describe the web pages currently being browsed, as depicted on Figure 6.1.

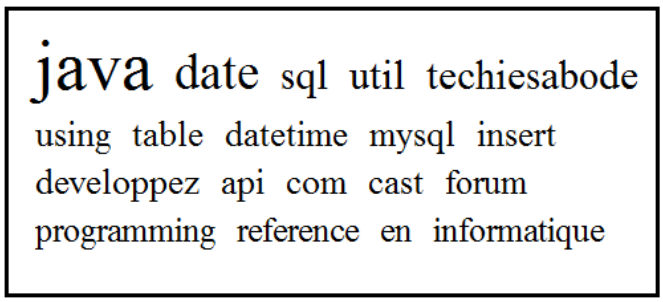

Figure 6.1: A sample contextual tag cloud, in an enterprise environment -

6.2.2 Web-browsing-based context sensors

In chapter 3, we have seen that documents are traditionally represented by a vector containing words that appear the most frequently in their content. In order to evaluate the quality of crowd-sourced/human annotations of web pages, we combine five additional techniques for extracting descriptive tags from web documents: social bookmarking tags (annotations given by human readers), search query terms (also entered

by humans), document metadata (provided by the authors of the web pages), completed with semantic content and resource location analysis (for web pages that have not been annotated by humans).

We define six functions to extract weighted tags from browsed documents:

1) the *Metadata* function

counts the number of occurrences of each term t with different coefficients (α, β, γ), depending of the position of this term in document d's metadata:

$$w_1(t,d) = \alpha * |t \in T_d| + \beta * |t \in K_d| + \gamma * |t \in D_d|$$

where $|t \in T_d|$ is the number of occurrences of the term t in the *title* of the document d, $|t \in K_d|$ in its *keywords* set, and $|t \in D_d|$ in its *description* text.

2) the *SearchQuery* function

counts the number of occurrences of term t in a search query Q_d, when the analyzed document d contains a list of search results:

$$w_2(t,d) = |t \in Q_d|$$

This function relies on the Google[1] search engine.

3) the *DomainNames* function

adds the domain names (including sub-domains) N_d from document d's URL as terms:

$$w_3(t,d) = |t \in N_d|$$

4) the *SocialBookmarks* function

counts the number of people who publicly bookmarked the document d using each term t as tag:

$$w_4(t,d) = \sum_{p \in P} tag(p,d,t)$$

where each person p is in the set of people P that are using this social bookmarking service, and where $tag(p,d,t)$ has a value of 1 or 0, whether or not this person p

[1]Google: http://www.google.com/

bookmarked the document d using term t as a tag. This function relies on the Delicious[1] social bookmarking service.

5) the *SemanticAnalyzer* function

counts the number of occurrences of semantically-defined entities (i.e. concepts and instances) that are represented by each term t, when they are identified in the document d:

$$w_5(t,d) = |t \in R_d|$$

$$R_d = [\forall e \in E_d, repr(e)]$$

where $repr(e)$ is the textual representation of a semantic entity e, R_d is the set of textually represented entities E_d found in the document d. This function relies on the SemanticProxy[2] web service.

6) the *KeyphraseExtractor* function

counts the number of occurrences of each term t of K_d, a set of keyphrases extracted from the document d:

$$w_6(t,d) = |t \in K_d|$$

This function relies on KEA, the Keyphrase Extraction Algorithm (Witten *et al.*, 1999).

6.2.3 A comparison of tags gathered from virtual and social sensors

In order to evaluate the quality of these tag extractors for synthesizing consistent contextual clouds that can represents users' context, based on the web pages that each user is browsing, we generated a contextual cloud with each sensor on a same web page and compared the results. We ran this experiment on a page entitled '*Getting started with extension development*', which is about Mozilla Firefox extension programming. From the resulting contextual tags depicted on Figure 6.2, we can draw the following conclusions:

[1]Delicious: http://delicious.com/
[2]SemanticProxy: http://semanticproxy.opencalais.com/

- The HTML-based Metadata sensor mostly emphasized keywords that were found in the title of the page. In our case, the page is quite well described by its title, that is why the results are satisfying. Nevertheless, this is not the case on all web pages.

- The KEA-based KeyphraseExtractor returned several relevant keywords (e.g. $firefox$, xul), including higher-level concepts (e.g. '*extension development*'), but also too many meaningless keywords (e.g. $ifest$, EM, $files$).

- The Semantic Proxy-based sensor picked up well-recognized terms such as names of standards. Indeed, RDF, DTD, XML and XUL technologies are very relevant keywords for the given page that are correctly emphasized by their high weight (based on the number of occurrences in the content). Lower weighed keywords are also relevant with the content of the page, but they do not really reflect its topic.

- Finally, the Delicious-based SocialBookmarks sensor returned fewer keywords but they better describe the topic of the page, and their weights are more heterogeneous, emphasizing the most consensual keywords (that are also the most descriptive, in this case), such as: $firefox$, *extension*, *development*. Even lower weighted keywords describe the topic and the utility of the page quite well, thanks to common sense emerging from the crowd of del.icio.us' users: *programming*, *tutorial*, xul.

Simple HTML meta	KEA	Semantic Proxy	Delicious
extension profile folder started development development resources knowledge install.rdf contents.rdf creating toolbar buttons base - adding items to menus extension development dev : tips : disable xul cache getting started with extension development mozillazine profile manager packaging extensions	helloworld/chrome.man ifest chrome EM Folder overlays Firefox files XUL extension development MozillaZine	Ted Mielczarek (1) Google (1) Mozilla Development Center (1) MDC (2) DTD (10) XUL (4) Javascript (3) US (2) XML (7) Linux (1) XUL Planet (1) rdf (12)	extension javascript tutorial mozilla firefox xul development programming howto extensions

Figure 6.2: Tag clouds extracted using four methods -

From this preliminary experiment, we conclude that the results from the Social-Bookmarks sensor are the most suitable for describing current user's context based on his/her web browsing session. Nevertheless, we must keep in mind that not every web page is tagged on this service. Therefore, it is necessary to include tags from other sensors, so that any web page can generate keywords that can be represented in the contextual cloud.

6.2.4 Software implementation

The application described above was implemented in a modular architecture in which software modules communicate through RESTful HTTP requests. In this section, we present these modules.

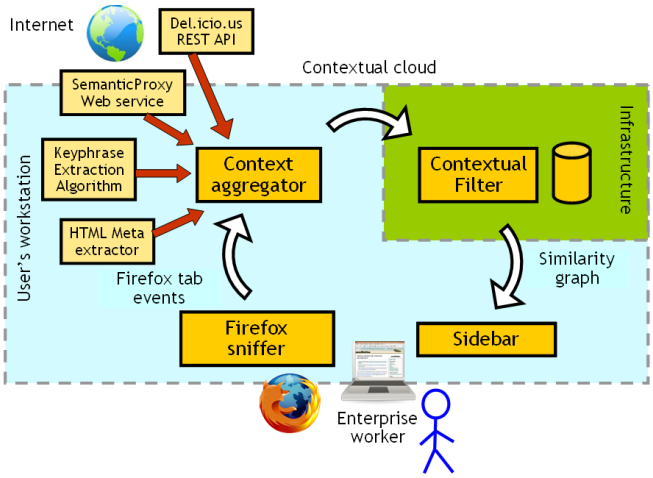

Figure 6.3: Flow of the contextual aggregation and recommendation process - in the enterprise application

Figure 6.3 depicts this architecture and flow as it is currently implemented in our application:

Firefox sniffer

The *Firefox*[1] *plug-in* is a JavaScript-based extension which hooks on the browser's events related to opening, closing and switching of web pages. Each occurrence of these

[1]http://www.firefox.com/

events is transmitted with the corresponding URLs to the local *Context Aggregator* for extracting tags from the sensors.

Table 6.1 lists the types of events being considered (extension of Mozilla's *visit_type* specification[1]).

actiontype	**Description**
1	The user opened a new page, by following a hyperlink
2	The user opened a new page, by typing its URL
3	The user opened a new page, by selecting a bookmark
5	The user was redirected to a new page (permanent redirect)
6	The user was redirected to a new page (temporary redirect)
7	The user downloaded a document
8	The user focused on a page (when several are opened at once, e.g. in tabs)
9	The user closed a page

Table 6.1: Types of browsing events handled from Firefox

Context Aggregator

The *Context Aggregator* handles events (triggered by users' actions) with their attached contextual information, and runs sensors' *weighting functions* on this information to produce an aggregated and normalized contextual cloud to be submitted to the *Contextual Filter*. It also behaves as a typical web server, in order to generate the user interface that is displayed locally in Firefox's side-bar, as explained later in this section.

Table 6.2 enumerates the parameters that are expected by the *logAction* REST service, handled by the Context Aggregator.

Our *Context Aggregator* also implements the *weighting functions* defined previously in this section: six different interpreters turn URLs into contextual clouds. The *Metadata* interpreter parses the *title*, *description* and *keywords* elements from the HTML source code of each web page to produce the corresponding weighted terms. The *SearchQuery* interpreter extracts query terms from Google Search result pages. The *SocialBookmarks* interpreter gathers tags given by users about a web page, when existing on the Delicious social bookmarking service. The *SemanticAnalyzer* gathers

[1] http://www.forensicswiki.org/wiki/Mozilla_Firefox_3_History_File_Format

Parameter	Description
source	Identification of the sniffer
	default: *webbrowser*
localtime	Local timestamp of the event
	format: UNIX/epoch timestamp (milliseconds)
actiontype	Type of event: *added*, $focus$ or *removed*
tabs	Array of URLs of currently opened web pages
$focus$	URL of the web page which currently has the focus

Table 6.2: Specification of parameters for events sent by the Firefox sniffer to the Context Aggregator

textual representations of semantic entities that were identified in the web page, thanks to the SemanticProxy web service. And the *Keyphrase Extractor* runs the Keyphrase Extraction Algorithm (Witten *et al.*, 1999) on the textual content of each web page.

Contextual Filter

The *Contextual Filter* handles contextual clouds gathered and interpreted by users' *Context Aggregator*, computes relevance scores between them, and recommend best-ranked social updates to each user. Social updates are gathered by subscription to the users' declared third-party social feeds/streams (e.g. their Twitter account). An internal social networking system has also been implemented in order to experiment the application without relying on third-party accounts. In our online implementation of the framework, the recommendations are displayed in near-real-time in a sidebar of Mozilla Firefox, thanks to an AJAX interface provided also by this server.

Side-bar (user interface)

The *side-bar* user interface is implemented as a XUL overlay, as a part of the Firefox plug-in introduced above. The content of this side-bar is generated as a HTML page by the local Context Aggregator. It consists of displaying the current contextual tag clouds, dynamically evolving as the user browses the web, a messaging field (called *tweet*) with a submission button, and a list of recommended social updates based on the last submission.

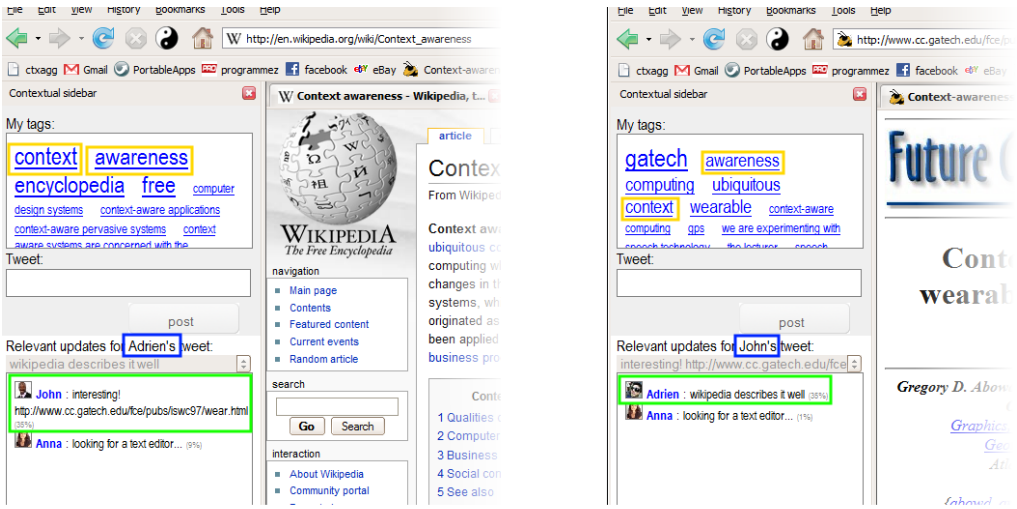

Figure 6.4: Screenshot of the *side-bar* user interface in Firefox - Simultaneous points of view for two users. The user's current contextual cloud is displayed on the top. It can be submitted to the Contextual Filter with an optional social update, by pressing the '*post*' button. Recommended social updates are listed below, according to the last submitted contextual cloud.

On Figure 6.4, we present the simultaneous points of view of two users: Adrien and John. Adrien is browsing web pages about *context-awareness*, including the definition of this concept on Wikipedia. John is browsing web pages about *ubiquitous computing*, including a page from *Georgia Tech* (known as *gatech*). Their contextual tag clouds respectively represent the terms corresponding to these pages, and relevant social updates are recommended according to their contextual tag cloud. On Adrien's screen, the first recommendation is a social update from John, who provides the URL of an *interesting* web page, apparently about *gatech*. In gray, the relevance score of this social update is shown: 35%. On John's screen, Adrien's last social update is recommended, also with a relevance score of 35%. This score represents the similarity between Adrien's and John's contextual tag clouds. Here, the terms *context* and *awareness* appear with a significant weight on both of them.

This online software implementation is functional and gives a good sense of the benefits of our approach.

6.3 Context-based relevance of social updates, an evaluation

In order to evaluate the validity of our hypothesis on relevance of contextually recommended social updates, we gathered browsing logs and social updates from volunteers during two weeks, ran our algorithms on these logs to generate 1846 contextual clouds, and asked the volunteers to rank the quality of selected results. In this section, we specify the requirements of the evaluation, describe the experimentation plan and setup we followed, explain the data processing phase, then discuss the results obtained.

6.3.1 Evaluation requirements

The evaluation of our hypothesis relies on the quality of two data processing steps: (i) the extraction of weighted contextual tags by sensors' contextual weighting and aggregation functions, and (ii) the recommendation function based on the similarity of those contextual clouds.

According to the eighth claim recommended by Terveen & McDonald (2005), '*Evaluations of social matching systems should focus on users and their goals*', we decided not to rely on existing evaluation data sets such as the ones from TREC, nor to follow a scenario-based experiment. Instead, we leverage usual browsing behavior of the users, and their own social updates, with their consent during the period of the experiment.

The scores expected from volunteers are threefold: (i) the representativity of contextual tag clouds, (ii) the accuracy of the relevance function for recommending social updates that are contextually relevant, and (iii) the relevance of social updates with the context of their author at the time of posting.

6.3.2 Experimentation plan

As explained above, we decided to let volunteers browse web sites and post social updates as they usually do, without constraints or guidelines. The evaluation period consists of a preparation phase, followed by three steps (as depicted on Figure 6.5):

1. The recruitment and briefing of volunteers: we presented and explained the following experimentation plan to potential volunteers and provided the necessary

software and support to install it for gathering log data. Additionally, we gathered volunteers' social feeds (e.g. their twitter accounts) to which they would post social updates, so that we can capture those for the experiment.

2. The data gathering period: during one week, volunteers browsed web pages using Firefox, while the provided sniffing extension was logging the required browsing events to a local database. At the end of this period, we provided a script to export these log entries to a human-readable text file, so that they could remove privacy-critical entries if needed (e.g. private activities, and other noisy data that is irrelevant to this study), and then send us this file.

3. The processing period: we ran our indexing and matching algorithms (as defined in the previous chapter) on the browsing logs and social updates provided by volunteers, to produce personalized survey forms containing ranked recommendations for each volunteer.

4. The volunteer feedback period: we asked each volunteer to fill two personalized surveys. In the first survey, for five contextual tag clouds generated from their own web browsing data, we ask each volunteer to rate the readability, representativity, need for modifications before sharing, and willingness to share these contextual tag clouds in an enterprise environment. In order to support them in remembering those contexts, we provide the list of web pages that were being browsed by the volunteer at that time. Still for each contextual tag cloud, we ask volunteers to rate the relevance of three random social updates in this context, one of those being a well-ranked match. In the second survey, volunteers rate the relevance of their own social updates with their contextual tag cloud at the time of posting.

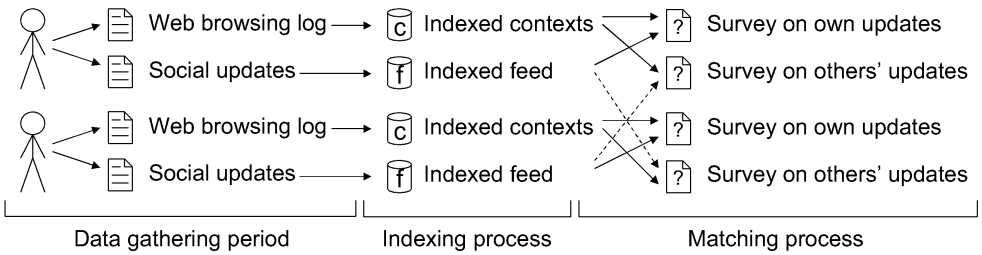

Figure 6.5: Experimentation plan -

6.3.3 Experimental setup and recommendation process

In order to gather experimental results, we generated personalized surveys for our eight volunteers, all being office workers (including researchers) of the same company. In this subsection, we present the process that we followed and the parameters that we set to generate these personalized surveys from the web browsing logs provided by volunteers.

Browsing logs contain a list of browsing events defined by: the timestamp of event, the type of event (i.e. page opened, focused, or closed), and the URL of the page concerned by this event. We parsed these log files to gather a global list of URLs, and executed the tag extraction and weighting functions. Some of those rely on querying web services (e.g. Delicious, SemanticProxy), and every page also had to be downloaded to extract tags from its metadata information (i.e. title, keywords and description). The *Metadata* weighting function was parameterized with factors $\alpha = 50$ per term appearing in the title, $\beta = 10$ in the *keywords* field, and $\gamma = 1$ in *description* field of a document. We stored the tag clouds resulting from the execution of tag extraction and weighting functions for each URL, in a file called *TagCache*, so that we could reproduce the execution of our indexing algorithm.

Because the experimental software was not interactive, we indexed contextual and social logs (provided as ATOM/RSS feed files by volunteers) on a common time line with a period of 10 minutes. When indexed, web browsing events are called *context snapshots*. A context snapshot is composed of a contextual tag cloud that aggregates all the weighted tags extracted from web pages that were opened, focused and closed since the previous snapshot. Indexing a social update consists of associating it with the contextual cloud of the last context snapshot at the time of posting this update. If there is no known context information in the previous snapshot, we use the one before the previous. Every indexed contextual cloud is processed to split multiple-word tags, cleaned from punctuation and other non-literal characters, filtered against a stop-words list, and then normalized so that the sum its tags' weights equals 1. Only the first 20 tags (with highest weights) are displayed. As shown on Figure 6.1, a contextual tag cloud can contain diverse kinds of terms, such as words in various languages, word combinations and acronyms.

Then, we ran the recommendation algorithm on the indexed contextual and social logs in order to produce a relevance matrix for each participant. In order to generate

a participant's personalized survey, we selected 5 indexed heterogeneous contexts (i.e. the most dissimilar to each other) that were matched (by the recommender) with at least one highly-ranked social update.

The second survey was simply generated from users' feed files.

6.3.4 Results

As specified in the evaluation requirements, the results are threefold: we gathered scores given by every participant on (i) the representativity and sharing of five contextual clouds selected from his/her own, (ii) the perceived relevance of three selected social updates (from other participants) for each of those contextual clouds, and (iii) the relevance of the participant's own social updates with the context of their posting.

Representativity and sharing

In order to rate the representativity and sharing of contextual tag clouds, we asked each participant to report on a 4-point Lickert scale their answers to the four following questions, for each of the five personalized contextual tag clouds they were given:

- Readability: *Does it contain English/French words?* Proposed answers are: 1 - none, 2 - few are words, 3 - list of words, or 4 - list of well-spelled words.
- Reflectivity: *How well do these words reflect your browsing activity?* Proposed answers are: 1 - not at all, 2 - might give a vague clue, 3 - quite well, or 4 - very representative.
- Privacy: *Who would you share this contextual tag cloud with?* Proposed answers are: 1 - no one ever, 2 - to someone specific, 3 - to a known set of people (friends, family, co-workers), or 4 - to anyone, I don't mind.
- Sharing in the enterprise: *Would you share such contextual tag clouds with other employees of your company?* Proposed answers are: 1 - never, 2 - would only keep it for personal usage, 3 - yes if it could support work, or 4 - yes, even in real time on a blog.

As depicted in Figure 6.6, the average readability is rated 3.05 (i.e. on our $[1, 4]$ grade rating range), corresponding to a score of 68.33% (i.e. translated to a [0,1] grade

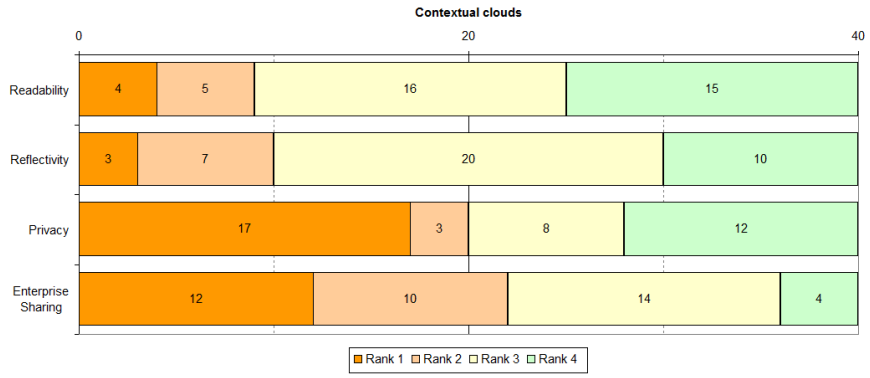

Figure 6.6: Distribution of ratings on contextual clouds -

range). We observe that these ratings are quite homogeneous for each participant, despite the heterogeneity of proposed tag clouds. According to the fact that some clouds were similar across participants, we can assume that this rating is rather a general one (i.e. applies to all tag clouds, as seen in websites). The average reflectivity is rated 2.93, corresponding to a score of 64.17%. Also quite homogeneous, we observe some exceptional values in these ratings, showing that the percieved reflectivity depends more on the contextual nature of our tag clouds, and thus the set of tags extracted from web pages. These results confort us in the sense that, despite the novelty of this representation for contextual information, the concept and visualization of contextual tag clouds is understandable by users.

With an average rating of 2.38 (45.83%), the privacy acceptance rating reveals heterogeneous intended audiences for all participants, depending on the contents of the contextual clouds. With an average rating of 2.25 (41.67%), the enterprise sharing ratings are lower, and not correlated with the privacy ratings. Except one participant who would not mind transparency of his/her contextual clouds in the enterprise, most users would rather keep those for personal usage. This can be explained by the category of volunteers: enterprise workers that are not all so familiar with social networking practices. It can also be explained by the fact that not all contexts (or some of their tags) were work-related, as it was not possible for them to manipulate their contexts in this experiment.

Relevance of recommendations

As explained previously, social updates proposed to users are voluntarily not all relevant. Our goal is to observe a correlation between the relevance scores given by participants and the rankings computed by the context-based recommender. Thus, we rely on a Mean Percentage Error (based on MAE, Mean Absolute Error) to define the following *accuracy* function:

$$accuracy = 1 - \sum_{q=1}^{Q} |sim(C_q, U_q) - rating(C_q, U_q)|$$

in which, for each proposed social update q, $sim(C_q, U_q)$ is the relevance score of the social update U_q with the contextual tag cloud C_q, as evaluated by the ranking algorithm. Whereas, $rating(C_q, U_q)$ is the actual relevance score, as given by the volunteer. Both scores are values in the range $[0, 1]$, represented as percents. As $rating()$ scores are given by volunteers in the $[1, 4]$ grade rating range, they are converted to percents with the following formula:

$$rating = \frac{(grade - 1)}{3}$$

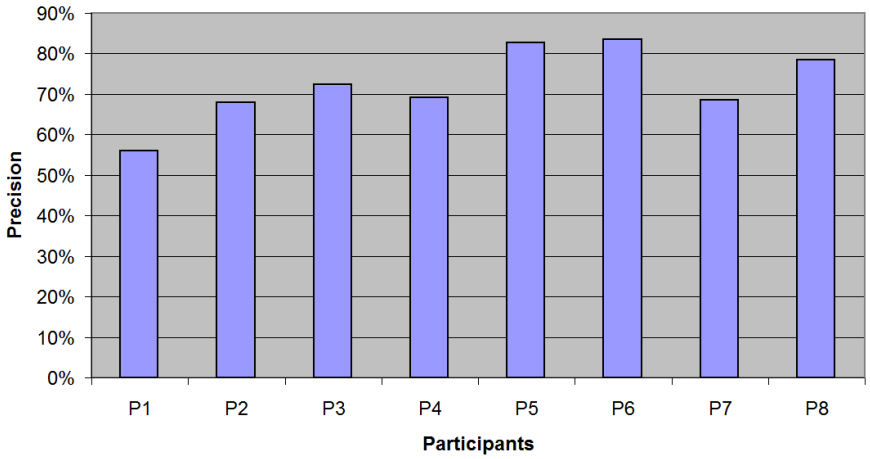

Figure 6.7: Average precision values for each participant -

As depicted on Figure 6.7, the precision values vary between 56% and 84%, with an average value of 72%.

On Figure 6.8, we observe that the distribution of relevance ratings given by participants on the 120 recommended social updates is quite close to algorithm expectations.

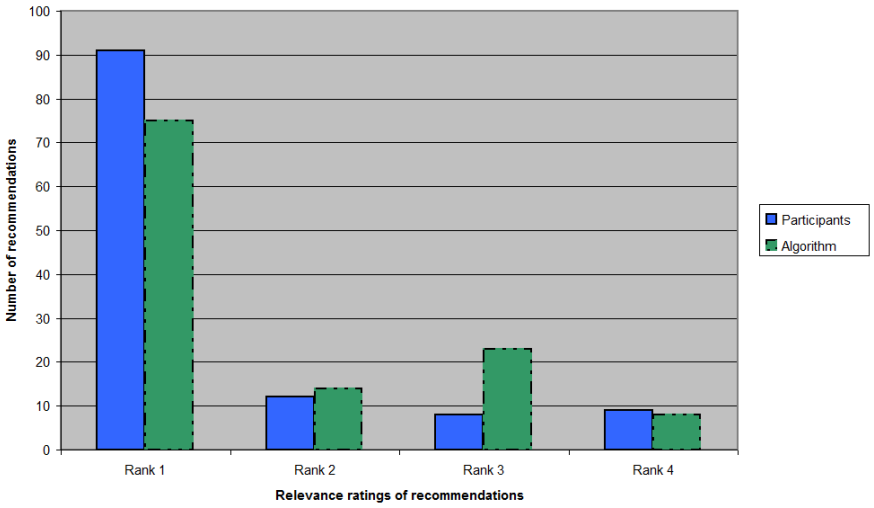

Figure 6.8: Comparative distribution of recommendation ratings - user-given and predicted

As a natural behavior of recommender systems, the best-ranked ratings (mostly in *Rank 3*) are slightly overestimated by the recommendation algorithm, whereas low relevance ratings (*Rank 1*) given by participants are higher than expected.

We observe that 63% of relevance ratings expected by the recommendation system are low ranked (*Rank 1*), whereas medium-high (*Rank 3*) ratings were expected for 19% of these ratings. The high number of low-ranked scores and the medium ranking of better scores expected by the algorithm reveals that highly similar contextual tag clouds were rare in our small scaled experiment. By increasing the number of participants, more similar contexts would be found, thus the average scores would naturally increase.

Relevance of contextualized social updates

For the third part of our study, in order to prove that contextual clouds are consistent reference documents for recommending social updates, we asked the participants to rate the relevance of each of their own social updates (e.g. their status updates and social bookmarks) to the contextual cloud representing their current situation at time of posting/sharing.

Over a total of 59 social updates, their authors rated an average relevance to context of 50.3%. Figure 6.9 shows the following distribution of ratings: 19 social updates were ranked 1 (low relevance), 10 were ranked 2, 14 were ranked 3, and 16 were ranked

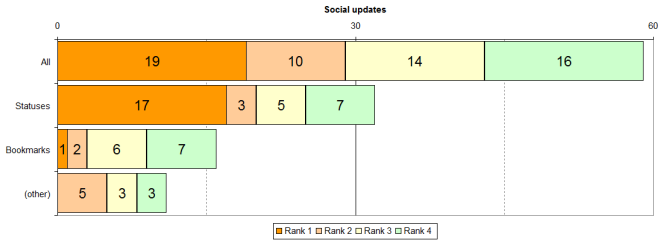

Figure 6.9: Comparative distribution of relevance ratings of social updates - to their context, per type

4 (high relevance). These social updates are gathered from several social feeds: 54% are status updates posted on Twitter, 29% are bookmarks instantly shared through Delicious (social bookmarking service).

By further analyzing these specific types of social streams, we discovered an average relevance score of 71% for shared bookmarks, and 38% for status updates from Twitter. It is natural that new bookmarks are more relevant to their context, as the web document that is bookmarked is usually being browsed by the user, and thus represented in the corresponding contextual tag clouds. Concerning status updates, Naaman et al. Naaman *et al.* (2010) proved that only 41% of social updates from twitter are actual statuses about the current activity of the person (categorized as "me now" by the authors). The similarity of this proportion with our average contextual relevance score for status updates gives some proof, although preliminary, about the consistency of our results.

6.3.5 Discussion

In this section, we studied the potential of our novel social update recommendation approach based on contextual clouds. Despite the small scale of our experiment, our results are promising. Although participants are still quite reluctant towards sharing contextual information, they showed a positive acceptance of contextual tag clouds for representing their browsing activity. The average accuracy of selected social updates: 72%, is significant for a web recommender system. With the average relevance of social updates to their context of emmission: 50.3%, we showed that our hypothesis is partly valid, depending on the type of social updates.

6.4 Matching contexts from virtual and social sensor tags, a comparative analysis

In the previous section, we evaluated the accuracy of social update recommendation based on contextual relevance. In order to estimate relevance, our system relies on similarities between contextual clouds, generated by aggregating descriptive tags associated to the web pages browsed by users. These associations are extracted from five mechanisms relying on:

- four virtual sensors: leveraging HTML metadata and semantic entities recognized in those web pages, names extracted from sub-domains of the resource locator (URL), and search query terms entered by the user;
- and one social sensor: leveraging tags given by users to describe web pages, on a social bookmarking service.

In this section, we analyze the differences between these sensors for enabling contextual matching, based on the experimental data introduced in the previous section.

6.4.1 Description of the data

Among a list of 14625 unique URLs accessed by participants during our experiment,

- semantic entities were recognized and turned to tags from 5292 URLs by SemanticProxy,
- sub-domain tags were extracted from 3979 URLs,
- meta-tags were extracted from the title, description and keywords annotated in the HTML source code of 2898 URLs,
- descriptive tags were given on 1395 URLs by users of the Delicious social bookmarking service,
- and search query terms were gathered from 1050 Google search URLs.

It is to be noted that a rather minor portion of those 14625 URLs have been successfully described by tags, using our virtual and social sensors. This ratio can be explained by several reasons which prevented the sensors from analyzing them:

- some of the URLs were local (stored on the user's computer), and thus invisible to some sensors;
- some pages were personal, or protected by credentials, thus also invisible to sensors;
- some pages do not contain any HTML content, e.g. pages embedding Flash or Java components;
- some pages are not referenced by any user in the Delicious social bookmarking service;
- and it is also possible that some of those URLs were unreachable by sensors at the time of our indexing process, e.g. temporary server breakdown, loss of connectivity, or even sensor failure.

These reasons further justify the use of additional sensors that do not necessarily rely on the Internet, but none were used in the frame of our experiment.

6.4.2 Analysis of results

In order to compare the efficiency of virtual and social sensors for enabling context matching, we generated several charts depicting the number of matched contexts for each user and the distribution of similarity scores. Here, we rely on Delicious to represent social sensor data, and on SemanticProxy to represent virtual sensor data. Indeed, we observed similar results between SemanticProxy and other virtual sensor data.

On Figures 6.10 and 6.11, we represent a cumulative histogram showing the number of context matches involving each user, for ten similarity threshold values. The total number of context matches depends of the number of 10-minute chunks that were represented by tags, extracted respectively by the SemanticProxy sensor (Fig. 6.10) and by the Delicious sensor (Fig. 6.11). The more chunks associated with tags, the higher the number of contexts that can be matched with other users' contexts. The average number of matched contexts is lower for the SemanticProxy sensor (127718) than for the Delicious sensor (156696); this observation also applies to other virtual sensors. Despite the lower number of URLs tagged on Delicious, this opposite difference is explained by the fact that the number of contexts of a user mostly depends on the

specific combinations of URLs he/she has browsed. For example, it is possible that all URLs were opened by only one user, in one 10-minute chunk, thus indexed into only one contextual cloud. In that case, no contexts would have been matched, despite the high number of URLs.

Concerning the distribution of the similarity scores of matched contexts over the threshold values, we observe that in average 31.46% of Delicious-based matches have a similarity score higher than 0.1 (a score of 1 being maximal, representing contextual equality), whereas this portion represent only 5.72% of SemanticProxy-based matches. This means that, despite the higher number of URLs described as tags by SemanticProxy, Delicious performed better in providing tags that can be used for matching contexts. This distribution is further emphasized on Figures 6.13 and 6.14. We observed similar distributions among all virtual sensors.

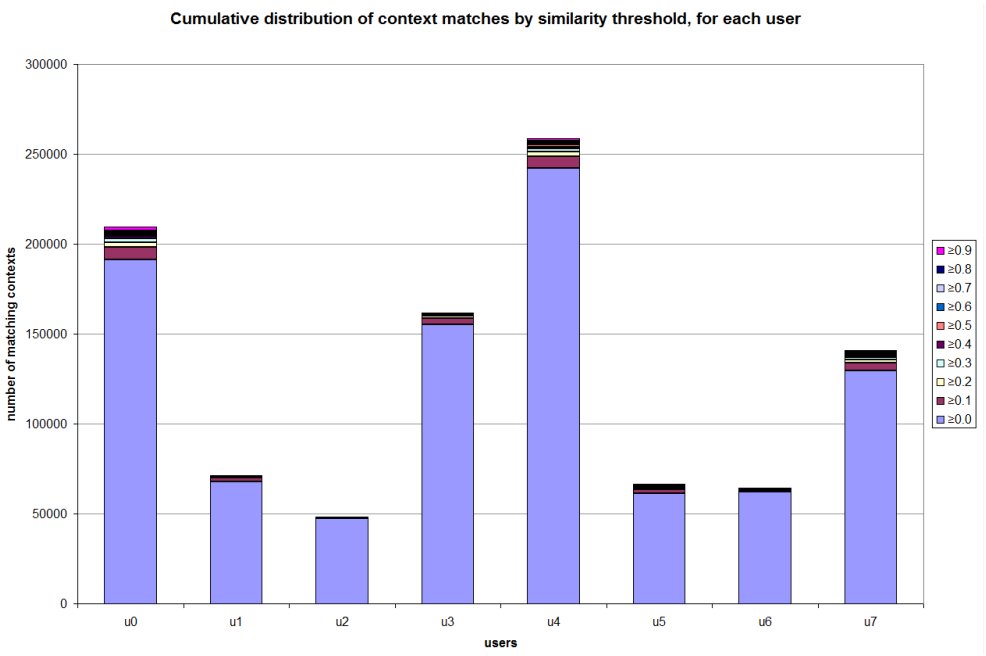

Figure 6.10: Tags from SemanticProxy: Cumulative distribution of context matches - by similarity threshold, for each user

Figures 6.12 and 6.15 show the same charts, after aggregation of tags from the five sensors (virtual and social). With an average value of 330228 matched contexts, from which 8.72% are above the 0.1 similarity score threshold, we observe that this aggregation actually reduces the number of matched contexts above this threshold:

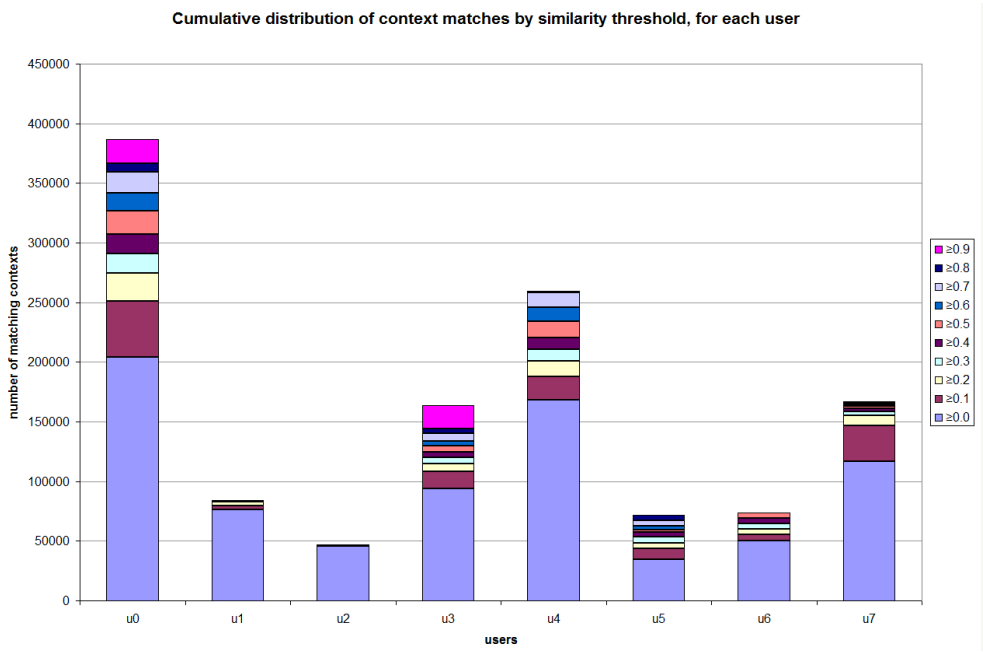

Figure 6.11: Tags from Delicious: Cumulative distribution of context matches - by similarity threshold, for each user

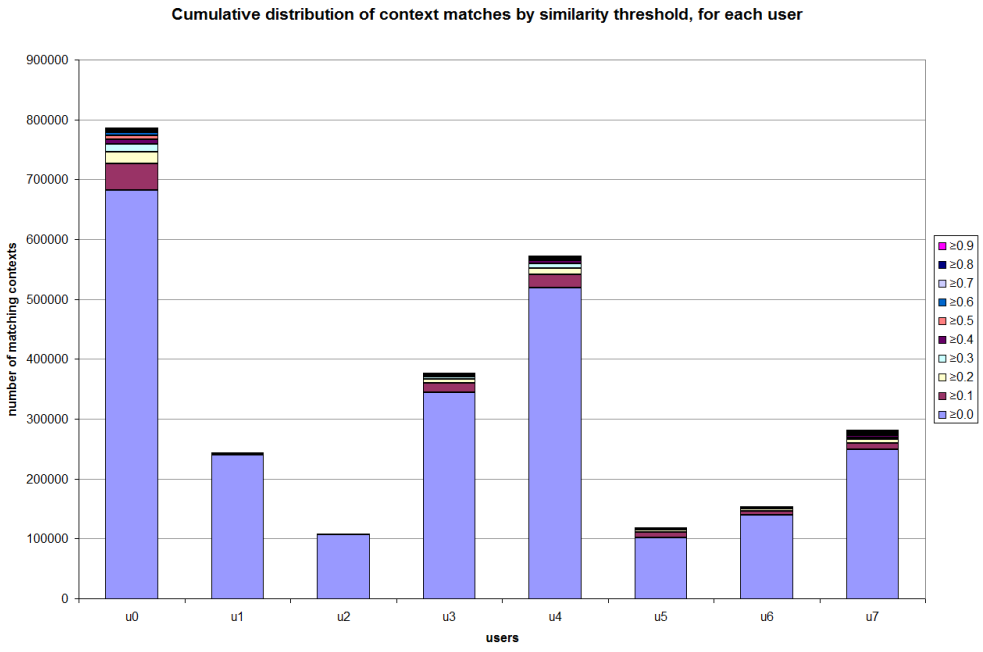

Figure 6.12: Tags from all sensors: Cumulative distribution of context matches - by similarity threshold, for each user

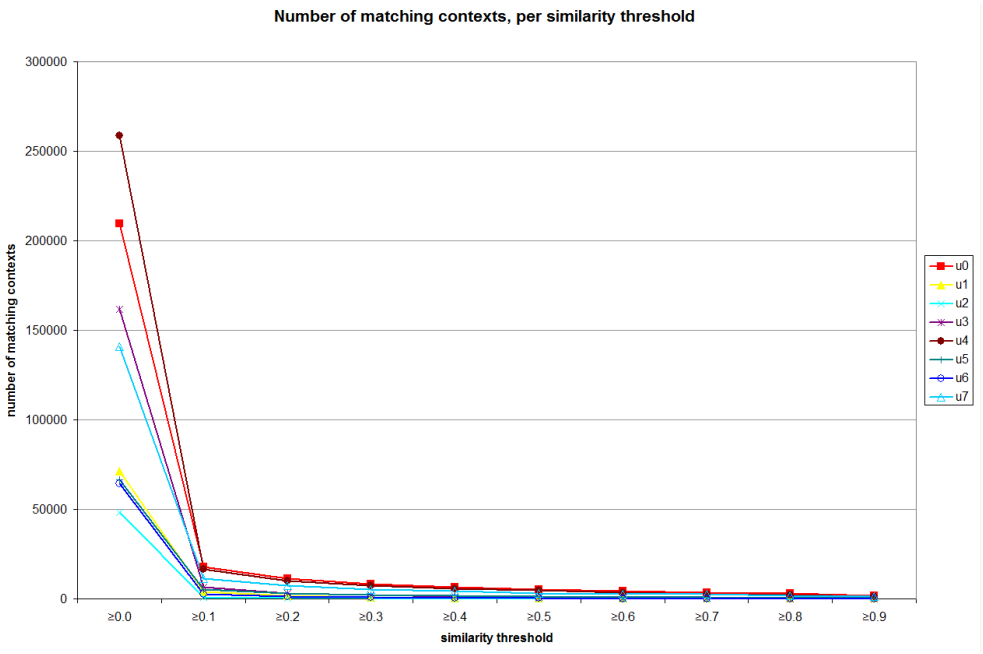

Figure 6.13: Number of matching contexts using only tags from SemanticProxy - per similarity threshold

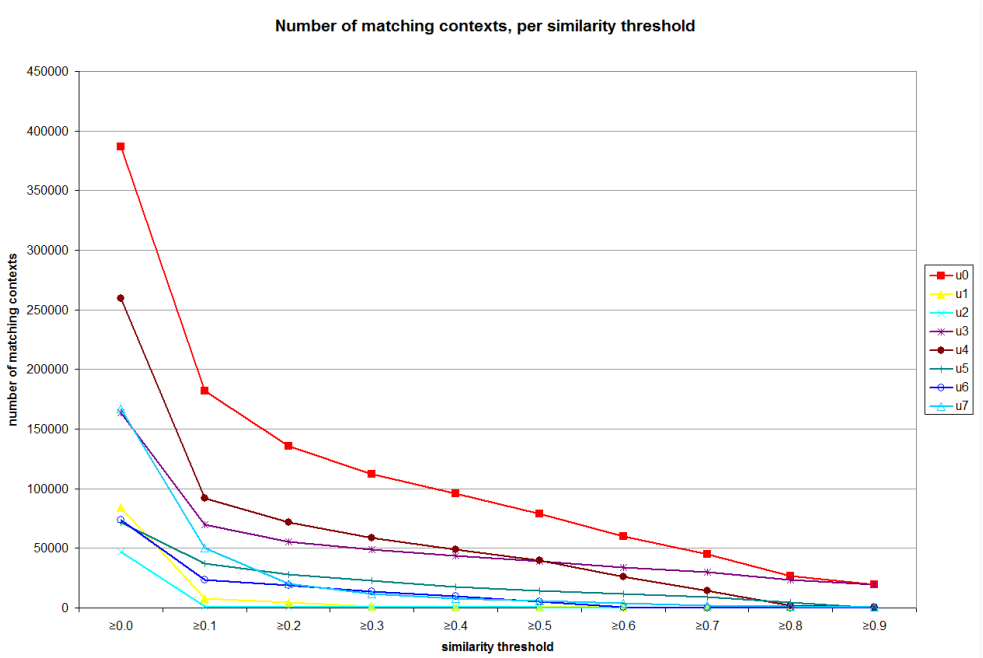

Figure 6.14: Number of matching contexts using only tags from Delicious - per similarity threshold

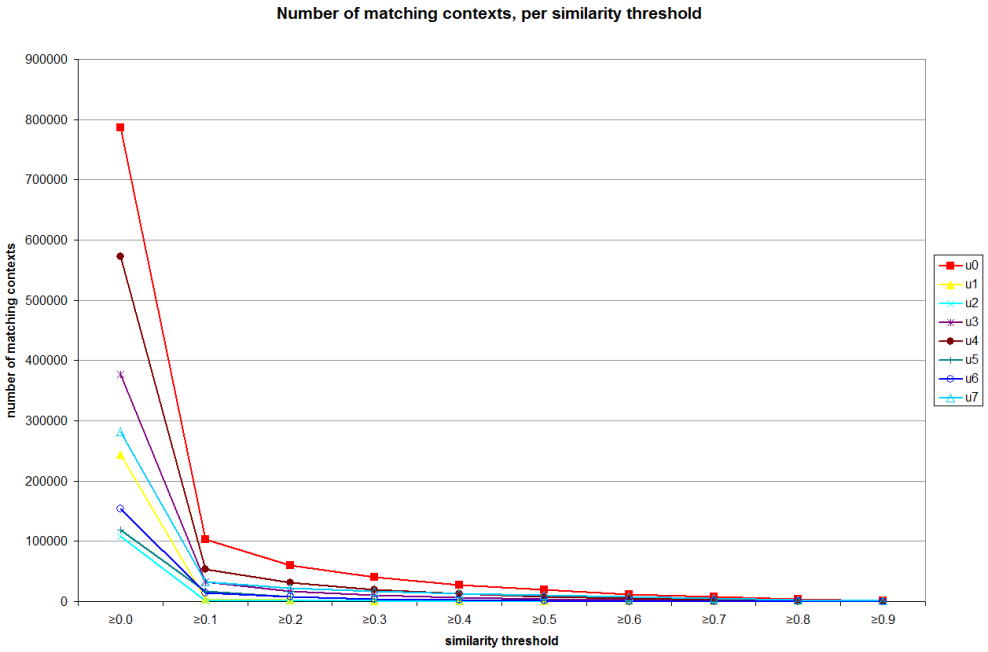

Figure 6.15: Number of matching contexts using tags from all sensors - per similarity threshold

32470, whereas 58008 were attained by using tags from Delicious only.

6.4.3 Discussion

With this analysis, we have proven the efficiency of tags gathered from social sensors for finding high similarity scores between users' contexts. In our case, we observed higher scores by relying on tags from the Delicious social bookmarking service than from virtual sensors: the SemanticProxy-based semantic entity extraction, tags extracted from HTML meta-data, URL sub-domain names, and Google search query terms.

This analysis also reflects that aggregating tags from various sensors implies lower similarity scores than relying solely on tags extracted by the Delicious social sensor. Indeed, adding tags extracted from various sensors imply a higher fragmentation of tags' weight. By applying the cosine similarity function on richer vectors (i.e. higher dimension space) with lower weights, the probability of resulting higher scores is inevitably reduced.

As explained earlier, the aggregation of complementary sources of tags (virtual and social sensors) is however necessary to ensure a maximal coverage of contextual entities.

Relying solely on social bookmarking services would not be enough, as a small portion of web pages are tagged by users. In order to increase the chances of finding higher scored similarities while leveraging several sensors, the dimension of resulting term vectors must be reduced. For example, tags could be clustered by semantic proximity (i.e. merge synonyms, hypernyms and hyponyms into one tag, or hierarchies), or noisy tags could be identified and removed. We have identified such techniques in Chapter 3, but we did not have enough time to apply them in this project. Nevertheless, these issues are interesting directions to explore in the future.

6.5 Conclusion

In this chapter, we applied our general context management framework and social update filtering scheme to develop a social awareness application for computer-based enterprise environments. In this application, contextual tags are gathered from the web pages currently browsed by users, using one social sensor based on the Delicious social bookmarking system, in addition to five virtual sensors. The architecture and the implementation of its modules were described, and an experiment was carried out to evaluate our framework, and support the underlying hypothesis.

Despite the small scale of our experiment, our results are promising:

- Firstly, the average accuracy of contextually-recommended social updates: 72%, is significant for a web recommender system. However, concerning the perceived relevance between their own social updates and their corresponding contextual cloud (at the time of posting), we observed an average score 50.3% from participants. We discovered that this score was lower than expected, because status updates (a subset representing 54% of the number of social updates posted during our experiment) were often not related to the context of their author. Therefore, our hypothesis could only be validated if most social updates were actually related to the context of their author, which usually is true for only 41% of status updates, according to Naaman *et al.* (2010).

- Concerning the efficiency of social sensors for extracting highly descriptive contextual information from crowd-sourced repositories, in contrast to more traditional content-based approaches, we appreciated the representativity of tags associated

by the Delicious social bookmarking site. We also observed that those tags implied more matches in looking for similar contexts, than tags extracted using virtual sensors. This result proves the efficiency of our novel context modeling approach based on crowd-sourced repositories.

Chapter 7

Conclusion

In this final chapter, we summarize our research, we recall our contributions and our original findings. Regarding our results, we propose recommendations and best practices concerning social awareness systems and tag-based information management. In addition to our answers to fundamental questionings, we also asked new other questions for which we propose some opportunities for future developments and improvements.

7.1 Research summary

7.1.1 Research context

In chapter two, and in (Joly *et al.*, 2009; Subercaze *et al.*, 2009), we provided a detailed presentation of various social networking systems (including microblogs), their common and specific characteristics and functions, and analyzed their usage through the enumeration of salient trends and major evolutions that we have observed since their appearance on the Internet. In particular, we presented the results of our survey about the cognitive disruptions (e.g. interruptions) caused real-time social communication platforms, such as the well-known Twitter micro-blogging service. In our analysis of those results, we made three major observations:

- More than half respondents (59%) follow the social feeds of more than 100 people, and 28% follow more than 250 people, which is higher than the Dunbar number;
- Many respondents (35%) are notified of new social updates in real time, and 62% read most of those updates;

- A majority of respondents (75%) did not set up any filtering mechanism, whereas more than 59% would like some support to filter social updates.

We supported that applying filtering techniques to those feeds could also reduce the frequency of cognitive interruptions (i.e. distraction) caused by real-time notifications of new updates. In particular, we identified that contextual information about users could be leveraged to filter social updates by relevance.

In chapter three, we reviewed several research efforts on improving social awareness, communication and collaboration using computer-based techniques. We identified information filtering and recommendation techniques that could be leveraged to regulate information streams between people. Inspired from content-based and collaborative filtering techniques, we proposed to develop a hybrid filtering system relying on tags inferred from documents currently opened by users, to represent their context, as published in (Joly, 2009a). Instead of a semantic representation, we chose a plain keyword representation for tags, as we identified that semantics could emerge from rich sets of tags (e.g. using co-occurrence and latent semantic analysis).

In chapter four, and in (Joly *et al.*, 2008), we proposed a state of the art of '*context-aware*' computing, as we have identified user context as a good criteria for filtering social updates by relevance. We identified scalability problems in most systems relying on semantic/ontology-based context models, supporting our choice for a keyword-based model for representing users' context. We also found out that many context management systems, contrarily to popular context-aware application and services that do not rely on a general context management system, imply privacy concerns, preventing users from adopting applications relying on those systems. We concluded that contextual information should remain in control of their users, allowing them to visualize and modify it before sending it to a server. We described three kinds of context sensors:

- physical sensors transmit information that is sampled from a physical environment (e.g. geographical positioning of users and other entities);
- virtual sensors synthesize contextual information from computer-based activities;
- and social sensors can provide emergent contextual information from crowd-sourced content by combining various web services.

7.1.2 Design, application and evaluation of a context management framework for social awareness

In chapter five, and in (Joly, 2009b), we specified our approach for designing a framework in order to validate our hypothesis:

> '*a social update shared by person U on a social networking system is relevant to another person X if the current context of X is similar to U's context at the time of sharing.*'

Following our requirements (e.g. crowd-sourcing, user control), we designed an interaction model, and a formal tag-based context model based on relevant sources of context for awareness applications. Then, we formalized four social matching schemes based on temporal combinations of users' contexts, and described the implementation of our context management framework for real-time and batch matching.

In chapter six, we applied our conceptual framework to develop a social awareness application for improving communication and collaboration between employees of large enterprises (e.g. a company, a research lab), motivated by a case study. In this application, users' context is inferred from their computer-based activities (e.g. browsed web pages). As an alternative to traditional content-based keyword extraction, we defined five additional techniques leveraging crowd-sourced/human annotations for extracting descriptive tags about web documents: social bookmarking tags (annotations given by human readers), search query terms (also entered by humans), document metadata (provided by the authors of the web pages), completed with semantic content and resource location analysis (for web pages that have not been annotated by humans). Then, we described the implementation of the application (relying on a context sniffing extension for Firefox, a local context aggregation server, and a common filtering server), and of those underlying techniques, integrated as virtual and social context sensors.

In the second half of this chapter, we evaluated the performance of our social awareness application (demonstrated in the CSCW'2010 conference (Joly *et al.*, 2010)), and of the underlying context management framework, by running context sensors and social matching algorithms on experimental data (social updates and web browsing logs) gathered from volunteers, and asking them to rate the results. This evaluation was twofold:

- Firstly, as explained in our upcoming publication (Joly *et al.*, to appear), we evaluated the perceived accuracy of contextually-recommended social updates, which average relevance rating was 72%. However, the relevance of social updates with the context in which they were posted, upon which their contextual recommendation relies, was rated 50.3% by their authors. It has been observed that, on Twitter, only 41% of status updates are related to the context of their author at the time of posting. Therefore, we stated that our hypothesis could only be validated if this ratio could be increased. In this regard, we observed that social bookmarking updates naturally imply a higher relevance-to-context ratio.

- Then, we evaluated the efficiency of social sensors (e.g. the Delicious-based SocialBookmarks weighting function), compared to more traditional content-based extraction techniques (implemented in virtual sensors), for extracting highly descriptive contextual information about users, from their web browsing logs. We observed that tags extracted by our social sensor implied more matches in looking for similar contexts, than tags extracted by virtual sensors. This result proves the efficiency of our novel context modeling approach based on crowd-sourced repositories.

7.2 Contributions

In this thesis, we presented the following contributions:

- a context-awareness framework based on '*tags*', instead of ontologies,

- an underlying multidimensional context model that can enable powerful end-user applications without sacrificing their privacy,

- a set of context sensors that can gather and interpret contextual information from human-generated content of several crowd-sourced repositories, instead of solely relying on physical, content-based and specific software sensors,

- and a social awareness application for large enterprises, in which contextual information is extracted from computer-based activities, in order to create new collaboration and communication opportunities, and maintain them seamlessly.

7.3 Findings

This thesis resulted in several findings:

- Contextual information can not only be extracted from physical sensors, humans can also be considered as a '*social sensor*' as long as they are many to contribute on structured crowd-sourced repositories (e.g. delicious).
- The descriptive tags emerging from social sensors (e.g. delicious) are promising features for social matching.
- Contextual similarity between users is a promising criteria for ranking social updates they produced.
- Users accept to share more private information about their context, as long as they can control and manipulate its contents.

7.4 Recommendations and best practices

Throughout our research, we identified some best practices that we recommend to users (and potential users) of social networking systems, creators of those systems, and to researchers.

Firstly, in order to facilitate filtering and recommendation of social updates (which are short, by nature), algorithms need to know whether or not a social update is related to the current context of its author. Several solutions and usages can be adopted in this regard:

- users can maintain separate feeds (e.g. Twitter accounts): one for context-related statuses (e.g. '*my train is late*'), and another one for non-related information (e.g. general news, information sharing and relaying, etc...);
- social networking tools, and particularly microblogging systems, can provide an option for users to precise whether or not their status is contextual, whenever they enter a new status;
- researchers can analyze social updates, to determine their relation with their author's context automatically.

Secondly, we observed that some tagging interfaces (e.g. delicious bookmarking dialog) recommend most popular/frequent tags to a user when he/she is invited to provide tags to a resource that has already been tagged by others. Because it's easier for users to select predefined tags than to write down tags that they could naturally find to describe a resource, such recommendations lead to overweighting tags that are already high-weighted (i.e. because they are general and/or consensual), which implies a reduction of the variety, subjectivity and granularity of folksonomies. In this regard, we propose that:

- users try to enter tags more spontaneously, and avoid as much as possible to select recommended tags;
- tagging systems remove or hide recommended tags, or that they recommend less weighted tags in a more random manner.

Thirdly, we found that social bookmarking systems are a great way for tagging resources. However, we deplore that too few other crowd-sourced sites support tagging. In order for us to be able to leverage tags from more crowd-sourced sites, we propose that:

- users intentionally provide lists of tags (preferably preceded by a '*tags:*' prefix), instead of plain text annotations on crowd-sourced sites, whenever possible;
- crowd-sourced sites add new fields for users to enter tags instead of full text annotations.

Finally, we would like to address some feature requests to Twitter, other microblogging services, and third-party service developers that rely on social feeds:

- Microblogging services (including Twitter) provide each user a feed which contains status updates posted by people that he/she decided follow, and the search feature allows one to subscribe to new status updates that conform a query with given criteria (e.g. presence of a keyword/hashtag). We believe that aggregating followed feeds and subscribed searched into one feed would be a nice feature to add.

- The respondents of our survey revealed a need for subject-based filtering of social updates from their followed feeds. Today, it is already possible to subscribe to a keyword/hashtag-based query. However, some social updates do not use common keywords when they post about a same subject. Some semantic query enrichment would be welcome to help those users.

- Additionally, it is common that the same news are shared and relayed by many followed feeds, which increases the number of received updates. Clustering social updates that relate to the same information (e.g. a same article), or that link to the same content, would be an appreciated feature to reduce information overload.

- Furthermore, tag-cloud based visualizations of most popular keywords (e.g. twitscoop) are a great way to watch trending topics, and thus to summarize the activity of social feeds. However, we have not found any service that applies the same visualization for one's followed feeds only.

7.5 Opportunities for improvement

In this section, we recall several limitations we have encountered, and opportunities for improvement that we have identified:

- In order to increase the chances of finding higher scored similarities while leveraging several sensors, the dimension of resulting term vectors must be reduced. For example, tags could be clustered by semantic proximity (i.e. merge synonyms, hypernyms and hyponyms into one tag, or hierarchies of tags), or noisy tags could be automatically identified and removed. In chapter 3, we have identified several techniques that could be leveraged in this regard.

- Currently based on computing cosine similarity between every users' context, the complexity of our matching algorithm is to be reduced in order to suit high-scale deployments of our application. We believe that online clustering techniques should be integrated to our framework to classify new contexts, so that the matching process could scale linearly.

- Giving back to users the control of their context imply a need for effective contextual clouds visualization, browsing and manipulation interfaces, especially on mobile devices.

References

Adomavicius, G. & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, **17**, 734–749. 43

Agosto, L. (2005). *Optimisation d'un Réseau Social d'Échange d'Information par Recommandation de Mise en Relation*. Ph.D. thesis, Université de Savoie. 45

Amazon (2008). What's a fact. http://amapedia.amazon.com/view/Meta%3AFact/id=120433. 48

Amelung, C.J. (2005). *A context-aware notification framework for developers of computer supported collaborative environments*. Ph.D. thesis, University of Missouri. 41

Arrington, M. (2006). Facebook users revolt, facebook replies. http://www.techcrunch.com/2006/09/06/facebook-users-revolt-facebook-replies/. 20

Auer, S. & Lehmann, J. (2007). What have innsbruck and leipzig in common? extracting semantics from wiki content. In *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, 503–517, Springer-Verlag, Innsbruck, Austria. 47

Bailey, B.P., Konstan, J.A. & Carlis, J.V. (2001). The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *Proceedings of INTERACT*, vol. 1, 593–601, Citeseer. 37

Balabanovic, M. (1997). An adaptive web page recommendation service. *Proceedings of the first International Conference on Autonomous Agents*. 43

Balabanovic, M. & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Commun. ACM*, **40**, 66–72. 43

Baldauf, M. & Simon, R. (2010). Getting context on the go: mobile urban exploration with ambient tag clouds. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*, 1–2, ACM, Zurich, Switzerland. 53

Baldauf, M., Dustdar, S. & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, **2**, 263–277. 54

Baldauf, M., Fröhlich, P. & Reichl, P. (2009). The ambient tag cloud: A new concept for Topic-Driven mobile urban exploration. In *Proceedings of the European Conference on Ambient Intelligence*, 44–48, Springer-Verlag, Salzburg, Austria. 87

Barkhuus, L. & Dey, A.K. (2003). Is context-aware computing taking control away from the user? three levels of interactivity examined. *UbiComp 2003: Ubiquitous Computing*, **2864/2003**, 149–156. 27

Bauer, T. & Leake, D.B. (2001). Real time user context modeling for information retrieval agents. In *Proceedings of the tenth international conference on Information and knowledge management*, 568–570, ACM, Atlanta, Georgia, USA. 44

Beky, A. (2008). Facebook prend l'ascendant sur MySpace ! http://www.neteco.com/140768-facebook-prend-ascendant-myspace.html. 16

Bellegarda, J.R. (2007). Latent semantic mapping: Principles & applications. http://www.morganclaypool.com/. 49

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The semantic web. *Scientific American*, **284**, 28–37. 154

Bernstein, M., Kairam, S., Suh, B., Hong, L. & Chi, E.H. (2010). A torrent of tweets: managing information overload in online social streams. In *CHI 2010 Workshop on Microblogging*, Atlanta, GA, USA. 33

Berry, P. (2010). Chirp: Ce qu'il faut retenir de la conférence twitter. http://www.20minutes.fr/article/397962/Web-Chirp-Ce-qu-il-faut-retenir-de-la-conference-Twitter.php. 28

REFERENCES

Biehl, J.T., Czerwinski, M., Smith, G. & Robertson, G.G. (2007). FASTDash: a visual dashboard for fostering awareness in software teams. In *CHI 2007 Proceedings*, ACM, San Jose, CA, USA. 42

Bielenberg, K. & Zacher, M. (2005). Groups in social software: Utilizing tagging to integrate individual contexts for social navigation. Tech. rep., Digital Media. Bremen, Germeny, University Bremen. Master of Science in Digital Media. 45

Bolchini, C., Curino, C.A., Quintarelli, E., Schreiber, F.A. & Tanca, L. (2007). A data-oriented survey of context models. *SIGMOD Rec.*, **36**, 19–26. 63

Boyd, D. (2009). Twitter: "pointless babble" or peripheral awareness + social grooming? http://www.zephoria.org/thoughts/archives/2009/08/16/twitter_pointle.html. 22

Broens, T., van Sinderen, M., van Halteren, A. & Quartel, D. (2007). Dynamic context bindings in pervasive middleware. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, 443–448. 53

Brooks, C.H. & Montanez, N. (2006). Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th international conference on World Wide Web*, 625–632, ACM, Edinburgh, Scotland. 49

Budzik, J. & Hammond, K.J. (2000). User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th international conference on Intelligent user interfaces*, 44–51, ACM, New Orleans, Louisiana, United States. 74

Budzik, J., Fu, X. & Hammond, K.J. (2000). Facilitating opportunistic communication by tracking the documents people use. In *CSCW 2000 Workshop on Awareness and the WWW. Retrieved January*, vol. 30, 2005. 44, 45, 74

Cantador, I. (2008). *Exploiting the Conceptual Space in Hybrid Recommender Systems: a Semantic-based Approach*. Ph.D. thesis, Universidad Autnoma de Madrid (UAM), Madrid, Spain. 46

Carter, J. & Dewan, P. (2009). Automatically identifying that distributed programmers are stuck. In *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, 12, IEEE Computer Society. 74

Cattuto, C., Benz, D., Hotho, A. & Stumme, G. (2008). Semantic analysis of tag similarity measures in collaborative tagging systems. http://arxiv.org/abs/0805.2045. 48

Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F. & Chakraborty, D. (2004a). Intelligent agents meet the semantic web in smart spaces. *IEEE Internet Computing*, **8**, 69–79. 53, 55, 56, 64

Chen, H., Perich, F., Finin, T. & Joshi, A. (2004b). SOUPA: standard ontology for ubiquitous and pervasive applications. *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, 258–267. 56, 57

Chen, M.Y., Sohn, T., Chmelev, D., Haehnel, D., Hightower, J., Hughes, J., LaMarca, A., Potter, F., Smith, I. & Varshavsky, A. (2006). Practical Metropolitan-Scale positioning for GSM phones. *Proc. 7th Int. Conf. Ubiquitous Computing (UbiComp)*. 70

Christopoulou, E. (2008). Context as a necessity in mobile applications. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*. 53

Ciesielski, K., Klopotek, M.A. & Wierzchon, S.T. (2007). Histogram-Based dimensionality reduction of term vector space. In *Proceedings of the 6th International Conference on Computer Information Systems and Industrial Management Applications*, 103–108, IEEE Computer Society. 49

Cope, A.S. (2007). Discussing machine tags in flickr API. http://www.flickr.com/groups/api/discuss/72157594497877875/. 48

Damian, D., Izquierdo, L., Singer, J. & Kwan, I. (2007). Awareness in the wild: Why communication breakdowns occur. In *Second IEEE International Conference on Global Software Engineering, 2007. ICGSE 2007*, 81–90. 41

Damme, C.V., Hepp, M. & Siorpaes, K. (2007). Folksontology: An integrated approach for turning folksonomies into ontologies. *Bridging the Gap between Semantic Web and Web*, **2**, 57–70. 46, 48

Dean, M. & Schreiber, G. (2004). OWL web ontology language reference. 55

Decker, S., Erdmann, M., Fensel, D. & Studer, R. (1999). Ontobroker: Ontology based access to distributed and semi-structured information. *Database Semantics: Semantic Issues in Multimedia Systems*, 351–369. 55

Delalonde, C. (2007). *Mise en Relation et Coopération dans les Equipes Distribuées de R&D. L'application de l'Actor-Network Theory dans la Recherche de Connaissances*. Thèse de doctorat, Université de Technologie de Troyes. 42

Dey, A.K. (2000). *Providing Architectural Support for Building Context-Aware Applications*. Ph.D. thesis, Georgia Institute of Technology. xiii, 3, 53, 64, 91

REFERENCES

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**, 97–166. 54, 95

Dourish, P. (2001). Where the action is: the foundations of embodied interaction. *Cambridge: Massachusetts Institute of Technology*. 41

Dourish, P. (2004). What we talk about when we talk about context. *Personal and Ubiquitous Computing*, **8**. 64, 84

Dourish, P. & Bellotti, V. (1992). Awareness and coordination in shared workspaces. *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, 107 – 114. 40

Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L. & Herlocker, J.L. (2005). TaskTracer: a desktop environment to support multi-tasking knowledge workers. In *Proceedings of the 10th international conference on Intelligent user interfaces*, 75–82, ACM, San Diego, California, USA. 44, 74

Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J. & Burgelma, J.C. (2001). Scenarios for ambient intelligence in 2010 (ISTAG 2001 final report). Tech. rep., ISTAG. 8

Dunbar, R.I.M. (1993). Co-evolution of neocortex size, group size and language in humans. *Behavioral and brain sciences*, **16**, 681–735. 22

Economist, T. (2007). Social graph-iti. *The Economist, print edition*. 16

Ejigu, D., Scuturici, V. & Brunie, L. (2007). Semantic approach to context management and reasoning in ubiquitous Context-Aware systems. In *The Second IEEE International Conference on Digital Information Management(ICDIM 2007)*, Proceedings of ICDIM'07, 500–5005. 65

Ellis, J.B., Wahid, S., Danis, C. & Kellog, W.A. (2007). Task and social visualization in software development: Evaluation of a prototype. In *CHI 2007 Proceedings*, ACM, San Jose, CA, USA. 42

Ellison, N.B., Steinfield, C. & Lampe, C. (2007). The benefits of facebook friends: Social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication*, **12**. 14

Erickson, I. (2008). The translucence of twitter. *EPIC 2008, Ethnographic Praxis in Industry Conference*, 58. 22

Erickson, T. & Kellogg, W.A. (2000). Social translucence: an approach to designing systems that support social processes. *ACM Transactions on Computer-Human Interaction*, **7**, 59–83. 41

Ertzscheid, O. (2008). Indexation sociale et folksonomies : le monde comme catalogue. 8, 76

Facebook (2010). Facebook statistics. http://www.facebook.com/press/info.php?statistics. 25, 27

Fitzpatrick, G. (1998). *The locales framework: Understanding and designing for cooperative work*. Ph.D. thesis, The University of Queensland, Brisbane. 41

Flanagan, J.A. (2006). An unsupervised learning paradigm for Peer-to-Peer labeling and naming of locations and contexts. *International Workshop on Location and Context-Awareness, Dublin, Ireland*. 74

Franke, J.L., Daniels, J.J. & McFarlane, D.C. (2002). Recovering context after interruption. In *Proceedings 24th Annual Meeting of the Cognitive Science Society (CogSci 2002)*, 310–315, Citeseer. 37

Fritzke, B. (1995). A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems*, **7**, 625—632. 50

Gaonkar, S., Li, J., Choudhury, R.R. & Cox, L. (2008). Micro-Blog: sharing and querying content through mobile phones and social participation. *Proceeding of the 6th international conference on Mobile systems, applications, and services*. 54

Gay-Bellile, V., Dupont, R. & Naudet-Collette, S. (2010). A vision based hybrid system for real-time accurate localization in an indoor environment. In *International Conference On Computer Vision Theory and Applications*, Angers, France. 73

Gillie, T. & Broadbent, D. (1989). What makes interruptions disruptive? a study of length, similarity, and complexity. *Psychological Research*, **50**, 243–250. 37

Golder, S. & Huberman, B.A. (2005). The structure of collaborative tagging systems. *Arxiv preprint cs.DL/0508082*. 8, 45, 76

Gruber, T.R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, **5**, 199–220. 153

Gu, T., Pung, H.K. & Zhang, D.Q. (2004a). A bayesian approach for dealing with uncertain contexts. *Proceedings of the Second International Conference on Pervasive Computing*. 60

REFERENCES

Gu, T., Wang, X.H., Pung, H.K. & Zhang, D.Q. (2004b). An ontology-based context model in intelligent environments. *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, **2004**. 53, 54, 59

Hjrland, B. & Christensen, F.S. (2002). Work tasks and socio-cognitive relevance: A specific example. *Journal of the American Society for Information Science and Technology*, **53**, 960–965. 2

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 50–57, ACM, Berkeley, California, United States. 49

Horrocks, I. (2002). DAML+OIL: a description logic for the semantic web. *IEEE Data Engineering Bulletin*, **25**, 4–9. 59

Hotho, A., Jschke, R., Schmitz, C. & Stumme, G. (2006). Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, 411–426, Springer. 47, 48

Hung, C.C., Huang, Y.C., Hsu, J.Y. & Wu, D.K.C. (2008). Tag-Based user profiling for social media recommendation. In *Workshop on Intelligent Techniques for Web Personalization & Recommender Systems at AAAI2008*. 45

Iqbal, S. & Horvitz, E. (2010). Notifications and awareness: a field study of alert usage and preferences. In *CSCW '10: Proceedings of the 2010 ACM conference on Computer supported cooperative work*, 30, 27, ACM, Savannah, Georgia, USA. 37, 38

James, W., Burkhardt, F., Bowers, F. & Skrupskelis, I.K. (1981). *The principles of psychology*. Harvard University Press. 2

Java, A., Song, X., Finin, T. & Tseng, B. (2007). Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, 56–65, ACM. 21

Joly, A. (2009a). Plateforme de filtrage contextuel des interactions médiées. Forum Jeunes Chercheurs INFORSID. 136

Joly, A. (2009b). Workspace Awareness without Overload: Contextual Filtering of Social Interactions. In *Smart Offices and Other Workspaces, workshop of the Intelligent Environments 2009 conference*, Ambient Intelligence and Smart Environments, 297–304, IOSPress. 137

Joly, A., Maret, P. & Bataille, F. (2008). Leveraging semantic technologies towards social ambient intelligence. In D. Stojanovic, ed., *Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications*, IGI Global, Disseminator of Knowledge. 136

Joly, A., Maret, P. & Daigremont, J. (2009). Context-Awareness, the missing block of social networking. *International Journal of Computer Science and Applications*, **4**. 135

Joly, A., Maret, P. & Daigremont, J. (2010). Enterprise Contextual Notifier, Contextual Tag Clouds towards more Relevant Awareness. CSCW 2010, the ACM Conference on Computer Supported Cooperative Work. 137

Joly, A., Maret, P. & Daigremont, J. (to appear). Contextual Recommendation of Social Updates, a Tag-based Framework. In *International Conference on Active Media Technology (AMT'10)*. 138

Kagal, L., Finin, T. & Joshi, A. (2003). A policy based approach to security for the semantic web. *Proceedings of the 2nd International Semantic Web Conference*, **2870**, 402–418. 57

Kernchen, R., Boussard, M., Hesselman, C., Villalonga, C., Clavier, E., Zhdanova, A.V. & Cesar, P. (2007). Managing personal communication environments in next generation service platforms. *Mobile and Wireless Communications Summit, 2007. 16th IST*, 1–5. 62

Kleek, M.V., Karger, D.R. & mc schraefel (2009). Watching through the web: Building personal activity and Context-Aware interfaces using web activity streams. In *Proceedings of the Workshop on Understanding the User - Logging and Interpreting User Interactions in Information Search and Retrieval (UIIR-2009)*, Boston, USA. 44

Koch, M. (2005). Supporting community awareness with public shared displays. In *Proc. Bled Intl. Conf. on Electronic Commerce*, Bled, Slowenia. 84

Koolwaaij, J., Tarlano, A., Luther, M., Nurmi, P., Mrohs, B., Battestini, A. & Vaidya, R. (2006). Context Watcher-Sharing context information in everyday life. In *Proceedings of the IASTED conference on Web Technologies, Applications and Services (WTAS). IASTED*, Calgary, Canada. 54, 72

Lamantia, J. (2008). Tag clouds evolve: Understanding tag clouds. http://www.joelamantia.com/blog/. 47

Lampe, C., Ellison, N. & Steinfield, C. (2006). A face (book) in the crowd: social searching vs. social browsing. *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 167–170. 14

Lassila, O. & Khushraj, D. (2005). Contextualizing applications via semantic middleware. In *Mobile and Ubiquitous Systems: Networking and Services, 2005*, 183–189. 54

REFERENCES

Leake, D.B., Bauer, T., Maguitman, A. & Wilson, D.C. (2000). Capture, storage and reuse of lessons about information resources: Supporting Task-Based information search. *In proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems*, 33—37. 46

Leake, D.B., Maguitman, A. & Reichherzer, T. (2003). Topic extraction and extension to support concept mapping. *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, 325—329. 49

Lederer, S., Mankoff, J. & Dey, A.K. (2003). Who wants to know what when? privacy preference determinants in ubiquitous computing. In *CHI '03 extended abstracts on Human factors in computing systems*, 724–725, ACM, Ft. Lauderdale, Florida, USA. 64

Leong, L.H., Kobayashi, S., Koshizuka, N. & Sakamura, K. (2005). CASIS: a context-aware speech interface system. *Proceedings of the 10th international conference on Intelligent user interfaces*, 231–238. 53

Liechti, O. (2000). Awareness and the WWW: an overview. *ACM SIGGROUP Bulletin*, **21**, 3–12. 40, 42

Lin, X., Li, S., Yang, Z. & Shi, W. (2005). Application-oriented context modeling and reasoning in pervasive computing. *Proceedings of the The Fifth International Conference on Computer and Information Technology*, 495–501. 65

Lin, Y., Sundaram, H. & Kelliher, A. (2008). Summarization of social activity over time: people, actions and concepts in dynamic networks. In *Proceeding of the 17th ACM conference on Information and knowledge management*, 1379–1380, ACM, Napa Valley, California, USA. 42

Lugano, G. (2007). Mobile social software: Definition, scope and applications. In *EU/IST eChallenges Conference*, The Hague (The Netherlands). 26

Marlow, C., Naaman, M., Boyd, D. & Davis, M. (2006). Position paper, tagging, taxonomy, flickr, article, toread. *in proceedings of Collaborative Web Tagging Workshop at WWW'2006*, 31—40. 47

Mathes, A. (2004). Folksonomies-cooperative classification and communication through shared metadata. *Computer Mediated Communication*. 47

McFarlane, D.C. & Latorella, K.A. (2002). The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction*, **17**, 1–61. 37

Millen, D.R., Feinberg, J. & Kerr, B. (2006). Dogear: Social bookmarking in the enterprise. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 111–120, ACM, Montréal, Québec, Canada. 44

Mischaud, E. (2007). *Twitter: Expressions of the Whole Self*. MSc dissertation, London School of Economics and Political Science. 21

Miyata, Y. & Norman, D.A. (1986). Psychological issues in support of multiple activities. *User centered system design*, 265–284. 37

Morris, M.R., Teevan, J. & Panovich, K. (2010). What do people ask their social networks, and why?: a survey study of status message q&a behavior. In *Proceedings of the 28th international conference on Human factors in computing systems*, 1739–1748, ACM, Atlanta, Georgia, USA. 23

Mynatt, E., Adler, A., Ito, M. & O'Day, V. (1997). Design for network communities. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, 217, 210, ACM, Atlanta, Georgia, United States. 6

Naaman, M. & Nair, R. (2008). ZoneTag's collaborative tag suggestions: What is this person doing in my phone? *Multimedia, IEEE*, **15**, 34–40. 47

Naaman, M., Boase, J. & Lai, C. (2010). Is it really about me?: message content in social awareness streams. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, 189–192, ACM, Savannah, Georgia, USA. 23, 125, 132

Nagata, S.F. (2003). Multitasking and interruptions during mobile web tasks. In *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 47, 1341–1345, Human Factors and Ergonomics Society. 37

News, C. (2010). Twitter, facebook use up 82 percent. http://news.cnet.com/8301-1023_3-10457480-93.html. 28

Niwa, S., Doi, T. & Honiden, S. (2006). Web page recommender system based on folksonomy mining for itng'06 submissions. In *Information Technology: New Generations, 2006. ITNG 2006. Third International Conference on*, 388–393. 45, 47

Nurmi, P., Martin, M. & Flanagan, J.A. (2005). Enabling proactiveness through context prediction. *Proceedings of the Workshop on Context Awareness for Proactive Systems, Helsinki*. 53

Oliver, N., Horvitz, E. & Garg, A. (2002). Layered representations for recognizing office activity. *Proceedings of the International Conference on Multimodal Interaction (ICMI 2002)*, 3–8. 53

REFERENCES

OpenSocial & Group, G.S. (2009). OpenSocial specification v0.9. http://www.opensocial.org/Technical-Resources/opensocial-spec-v09/OpenSocial-Specification.html. 18

O'Reilly, T. (2005). What is web 2.0: Design patterns and business models for the next generation of software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. 154

Orwant, L. (1995). Heterogeneous learning in the doppelganger user modelling system. *User Modelling and User Adapted Interaction*, **4**, 107–130. 61

Pazzani, M.J. (1999). A framework for collaborative, Content-Based and demographic filtering. *Artif. Intell. Rev.*, **13**, 393–408. 46

Pera, M.S., Lund, W. & Ng, Y. (2009). A sophisticated library search strategy using folksonomies and similarity matching. *J. Am. Soc. Inf. Sci. Technol.*, **60**, 1392–1406. 48

Perich, F., Avancha, S., Chakraborty, D., Joshi, A. & Yesha, Y. (2005). *Profile Driven Data Management for Pervasive Environments*. Springer. 58

Posner, M.I. (1980). Orienting of attention. *The Quarterly Journal of Experimental Psychology*, **32**, 3–25. 2

Quesada, J. (2008). Human similarity theories for the semantic web. In *The 7th International Semantic Web Conference*, 1, Karlsruhe, Germany. 46

Ranganathan, A., Al-Muhtadi, J. & Campbell, R.H. (2004). Reasoning about uncertain contexts in pervasive computing environments. *Pervasive Computing, IEEE*, **3**, 62–70. 59, 64

Rattenbury, T., Good, N. & Naaman, M. (2007). Towards automatic extraction of event and place semantics from flickr tags. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 103–110, ACM, Amsterdam, The Netherlands. 77

Roda, C., Ach, L., Morel, B., Nabeth, T., Angehrn, A.A., Rudman, P., Zajicek, M., Kingma, D., Molenaar, I. & Vanhala, T. (2006). AtGentive deliverable d1.2, state of the art report. Tech. Rep. D1.2, IST AtGentive. 37, 38

Rosenbush, S. (2005). News corp.'s place in MySpace. http://www.businessweek.com/. 14

Salton, G. & McGill, M.J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc. 42, 43, 90

Sarma, A., Redmiles, D. & van der Hoek, A. (2008). Empirical evidence of the benefits of workspace awareness in software configuration management. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, 113–123, ACM New York, NY, USA. 42

Schilit, B.N., Adams, N., Gold, R., Tso, M.M. & Want, R. (1993). The PARCTAB mobile computing system. In *Proceedings Fourth Workshop on Workstation Operating systems (IEEE WWOS-IV)*, 2, Citeseer. 54

Schilit, B.N., Adams, N. & Want, R. (1994). Context-Aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, USA. 7, 53

Sieb, R.A. (1990). A brain mechanism for attention. *Medical Hypotheses*, **33**, 145–153, PMID: 2292975. 40

Sinha, R. (2005). A cognitive analysis of tagging. http://rashmisinha.com/2005/09/27/a-cognitive-analysis-of-tagging/. 47

Snoeck, N. (2007). Plan recognition in smart environments. In *ICDIM'07: The 2nd International Conference on Digital Information Management*, Villeurbanne. 53

Specia, L. & Motta, E. (2007). Integrating folksonomies with the semantic web. *Lecture Notes in Computer Science*, **4519**, 624. 45, 46

Speier, C., Vessey, I. & Valacich, J.S. (2003). The effects of interruptions, task complexity, and information presentation on computer-supported decision-making performance. *Decision Sciences*, **34**, 771–797. 37

SPICE (2006). SPICE d4.1: Ontology definition of user profiles, knowledge information and services. 61, 64

SPICE (2007). SPICE unified architecture. Tech. rep., FP6 IST SPICE. 61, 62, 64

Stan, J., Zsigmond, E.E., Joly, A. & Maret, P. (2008). A user profile ontology for Situation-Aware social networking. In *3rd Workshop on "Artificial Intelligence Techniques for Ambient Intelligence" (AITAmI'08)*, 51–55, Patras, Greece. 50

Stankovic, M., Laublet, P. & Passant, A. (2009). Directing status messages to their audience in online communities. In *Pre-proceedings of Coordination, Organization, Institutions and Norms Workshop in agent systems in on-line communities (COIN)*, Springer LCNS, Torino, Italy. 76

REFERENCES

Strang, T. & Linnhoff-Popien, C. (2004). A context modeling survey. *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp*, 34–41. 54

Strang, T., Linnhoff-Popien, C. & Frank, K. (2003). CoOL: a context ontology language to enable contextual interoperability. *Distributed Applications and Interoperable Systems: 4th Ifip Wg6. 1 International Conference, Dais 2003, Paris, France, November 17-21, 2003, Proceedings*. 55

Strassner, J., O'Sullivan, D. & Lewis, D. (2007). Ontologies in the engineering of management and autonomic systems: A reality check. *Journal of Network and Systems Management*, **15**, 5–11. 153

Strohbach, M., Bauer, M., Kovacs, E., Villalonga, C. & Richter, N. (2007). Context sessions: A novel approach for scalable context management in NGN networks. In *Middleware for Next-generation Converged Networks and Applications*, Newport Beach, California, USA. 61

Subercaze, J., El Morr, C., Maret, P., Joly, A., Koivisto, M., Antoniadis, P. & Ihara, M. (2009). Towards Successful Virtual Communities. In *International Conference on Enterprise Information Systems*, Lecture Notes in Business Information Processing, 677–688, Springer Berlin Heidelberg. 135

Sysomos (2009). In-Depth look inside the twitter world. Tech. rep., Sysomos. 25

Szomszor, M.N., Cantador, I. & Alani, H. (2008). Correlating user profiles from multiple folksonomies. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, 33–42, ACM New York, NY, USA. 48

Tan, J.G., Zhang, D., Wang, X. & Cheng, H.S. (2005). Enhancing semantic spaces with Event-Driven context interpretation. *Pervasive Computing: Third International Conference, Pervasive 2005, Munich, Germany, May 8-13, 2005, Proceedings*. 65

Tapia, E.M., Intille, S.S. & Larson, K. (2004). Activity recognition in the home using simple and ubiquitous sensors. *Proc. Pervasive*. 53

Terveen, L. & McDonald, D.W. (2005). Social matching: A framework and research agenda. *ACM Trans. Comput.-Hum. Interact.*, **12**, 401–434. 40, 43, 118

Tesconi, M., Ronzano, F., Marchetti, A. & Minutoli, S. (2008). Semantify del. icio. us: automatically turn your tags into senses. In *Social Data on the Web, workshop at the 7th International Semantic Web Conference*, Karlsruhe, Germany. 45, 46, 48

Thompson, C. (2008). I'm so totally, digitally close to you. http://www.nytimes.com/2008/09/07/magazine/07awareness-t.html?_r=1&pagewanted=all. 21, 22

Thornycroft, P. (2009). Location based services for cellular phones using wi-fi: The university of cincinnati's system for emergency call location. White paper, Aruba Networks. 71

Truong, B.A., Lee, Y. & Lee, S.Y. (2005). A unified context model: Bringing probabilistic models to context ontology. *Lecture notes in computer science*, 566–575. 61

Tsatsaronis, G. & Panagiotopoulou, V. (2009). A generalized vector space model for text retrieval based on semantic relatedness. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, 70–78, Association for Computational Linguistics, Athens, Greece. 49

Tversky, A. (1977). Features of similarity. *Psychological Review*, **84**, 327–352. 46

Vogiazou, Y., Dzbor, M., Komzak, J. & Eisenstadt, M. (2003). Buddy space: Large scale presence for communities at work and play. In *Proc. ECSCW03 Workshop*, 1–7. 41, 82

Vyas, D., van de Watering, M., Eliëns, A. & van der Veer, G. (2007). Engineering social awareness in work environments. In *Universal Access in Human-Computer Interaction. Ambient Interaction*, 254–263, Springer. 42

Wang, X., Dong, J.S., Chin, C.Y., Hettiarachchi, S.R. & Zhang, D. (2004a). Semantic space: An infrastructure for smart spaces. *Pervasive Computing, IEEE*, **3**, 32–39. 53

Wang, X., Zhang, D., Gu, T. & Pung, H. (2004b). Ontology based context modeling and reasoning using OWL. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, 18–22. 60

Want, R. (2008). RFID explained: A primer on radio frequency identification technologies. http://www.morganclaypool.com/doi/abs/10.2200/S00040ED1V01Y200602MPC001. 8

Want, R., Hopper, A., Falco, V. & Gibbons, J. (1992). The active badge location system. *ACM Trans. Inf. Syst.*, **10**, 91–102. 54

Weil, K. (2010). Measuring tweets. http://blog.twitter.com/2010/02/measuring-tweets.html. 28

REFERENCES

Weiser, M. (1991). The computer for the 21st century. *Scientific American*, **265**. 7

Wetzker, R., Umbrath, W. & Said, A. (2009). A hybrid approach to item recommendation in folksonomies. In *Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, 25–29, ACM, Barcelona, Spain. 45, 49

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C. & Nevill-Manning, C.G. (1999). KEA: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, 255, ACM. 112, 116

Wroblewski, L. (2009). The impact of social models. 25

Yi, K. (2008). Mining a web2.0 service for the discovery of semantically similar terms: A case study with del.icio.us. In *Digital Libraries: Universal and Ubiquitous Access to Information*, 321–326, Springer. 46

Zhdanova, A.V., Zoric, J., Marengo, M., van Kranenburg, H., Snoeck, N., Sutterer, M., Rck, C., Droegehorn, O. & Arbanowski, S. (2006). Context acquisition, representation and employment in mobile service platforms. *Proc. 15th IST Mobile & Wireless Communications Summit*. 53

Zijlstra, F.R.H., Roe, R.A., Leonora, A.B. & Krediet, I. (1999). Temporal factors in mental work: Effects of interrupted activities. *Journal of Occupational and Organizational Psychology*, **72**, 163–185. 37

Appendices

Appendix A

Introduction to ontologies and the Semantic Web

In their study, Strassner *et al.* (2007) define ontologies as '*a formal, explicit specification of a shared, machine-readable vocabulary and meanings, in the form of various entities and relationships between them, to describe knowledge about the contents of one or more related subject domains throughout the life cycle of its existence.*' Semantic technologies, including ontologies and semantic description languages, are quite similar to human thinking and memorization: they allow the definition of concepts and instances (of these concepts) that are related with each other using semantically qualified links. They also allow to develop an inferred knowledge from the reasoning on this knowledge (Gruber, 1993). Applying such approach to information technologies enable machines to understand the actual meaning of data which is formulated using a distributed and evolving vocabulary. That way, ontologies fill the gap between ambiguous/fuzzy human thinking (e.g. in natural languages, a word can have different meanings) and formalized digital data (i.e. stored using specific formats and interpreted by specific applications for a specific purpose).

One of the benefits of using semantic languages is to allow progressive/incremental modeling of a system, reflecting the natural progression of conceptual understanding of domains. Ontologies can ease the communication between heterogeneous entities (i.e. using different languages/protocols) by matching similar portions of the semantic graph of the sender's knowledge with the recipient's knowledge.

On the other hand, we would like to prevent the reader to make the naive assumption that semantic technologies are a magic solution to empower machines with autonomic intelligence. It may seem possible to model our universe as an ontology, allowing computers to understand the human world, but it is actually impossible. Indeed, modeling is always relative to a point of view, and integrating ontologies from experts of several domains would necessarily lead to inconsistencies. There is also a usual confusion about the so-called "Semantic Web" (Berners-Lee *et al.*, 2001). This expression does not mean that Internet users will have to deal with semantic languages to communicate on the web, but it refers to a set of languages and tools that would allow web resources (i.e. web pages and services) to be described semantically in order to allow seamless processing of knowledge distributed among heterogeneous sites. Today, with the rise of the '*Web 2.0*' (O'Reilly, 2005), users are already able to create '*mash-ups*' relying on several components and data streams hosted on different sites. However, the next step is possibly to automatize (or, at least, to ease) the development of such mash-ups, assuming that web data and components are semantically described.

Appendix B

Samples of personalized surveys

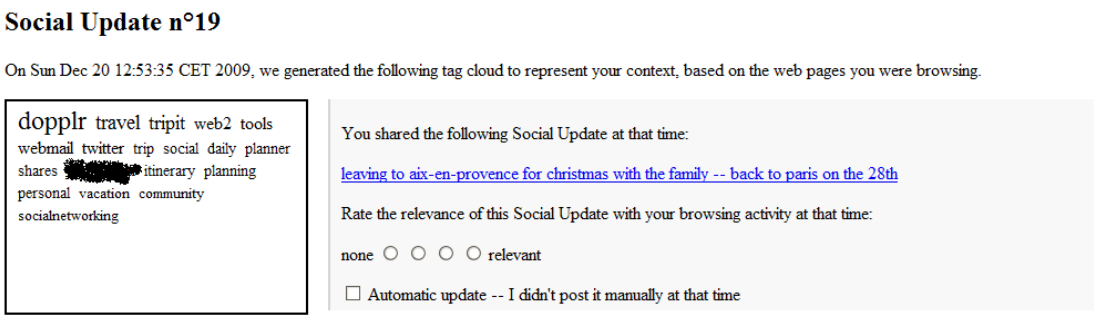

Figure B.1: A social update from a participant's personalized survey -

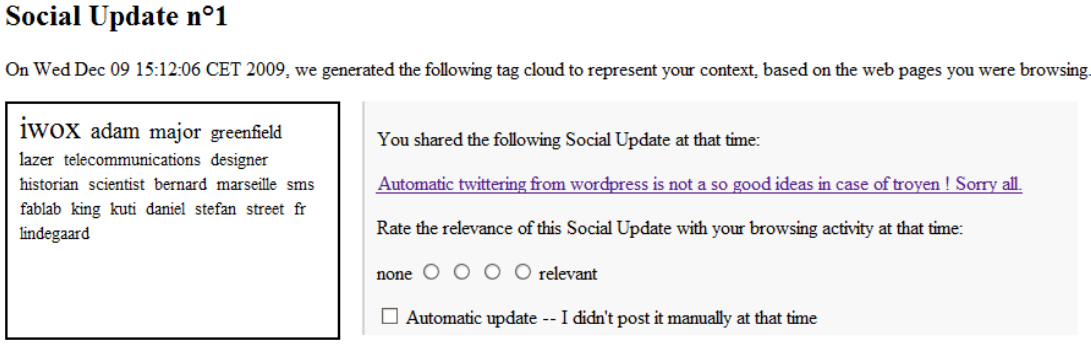

Figure B.2: Another social update from a participant's personalized survey -

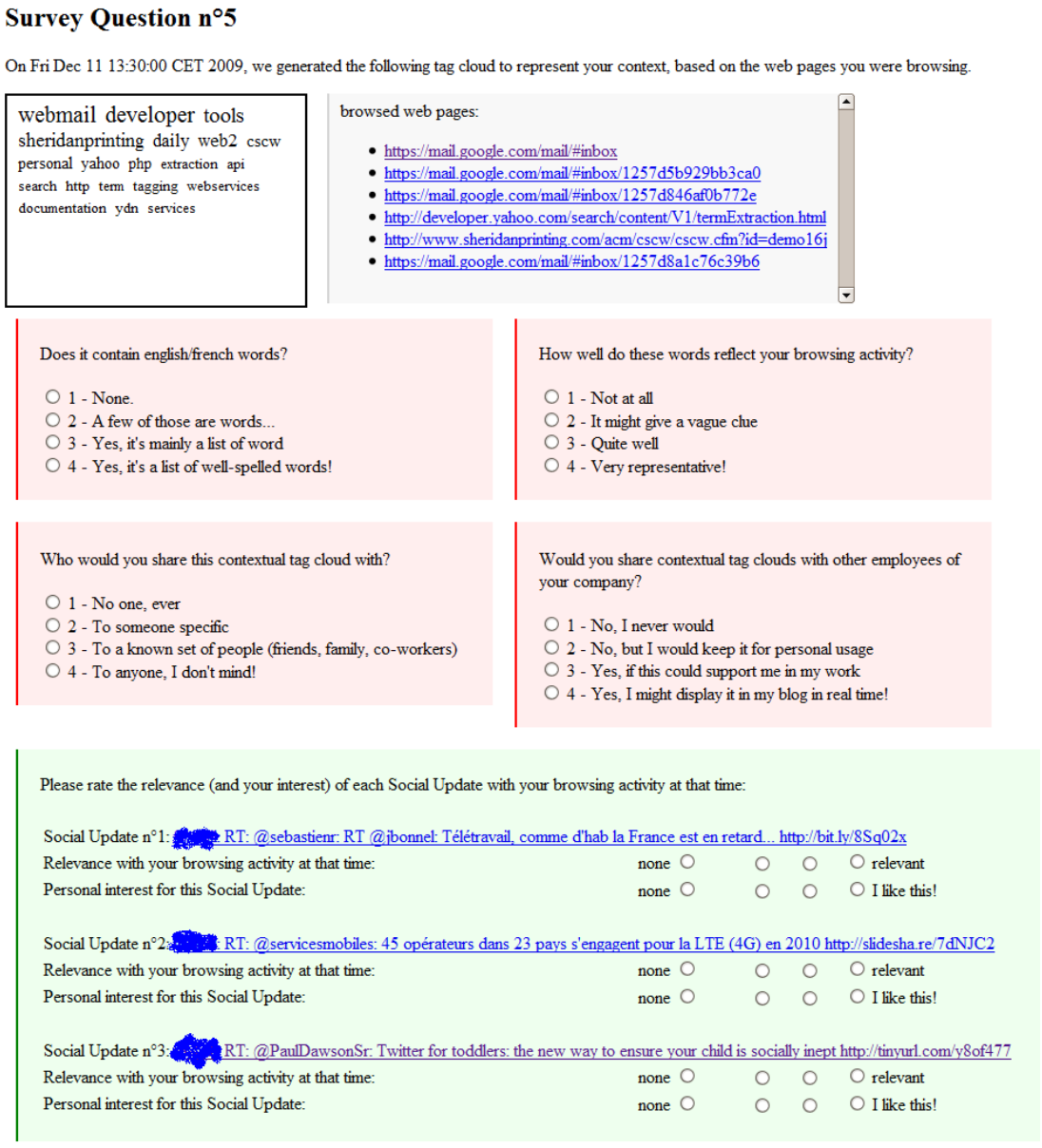

Figure B.3: A sample context from a participant's personalized survey -

Appendix C

Survey on usage of real-time microblogging

Twitter (or other microblog) usage poll

With this survey, we intend to gather usage preferences in order to evaluate the impact of microblogging usage on people's attention and productivity. Please select (or specify) the microblogging platform that receives most of your attention all day long, and answer to the following questions, according on your own usage of this platform.

This survey should take you less than 10 minutes to complete.

Thanks a lot for your help! :-)

Adrien Joly
PhD candidate
University of Lyon, France

E-mail: (first name) . (last name) (at) liris.cnrs.fr

* Required

A) Among microblogging platforms that you use, which one takes most of your attention? *
(please select the most appropriate answer)

- ○ A1 Twitter
- ○ A2 Yammer
- ○ A3 Friendfeed
- ○ A4 Identi.ca (or other laconi.ca-based)
- ○ Other:

B) What do you use microblogging for? *
(please check all appropriate answers)

- ☐ B1 For work / professional activity and awareness
- ☐ B2 For real-time news
- ☐ B3 For networking (e.g. annoucements, asking for advices...)
- ☐ B4 For a personal cause, a hobby
- ☐ B5 For keeping in touch with friends/contacts
- ☐ B6 Personal improvement
- ☐ B7 For fun
- ☐ Other:

How many people do you follow on this platform? *
(please select the most appropriate answer)

- ○ less than 10
- ○ 10-50
- ○ 50-100
- ○ 100-250

- ○ more than 250

Do you usually receive real-time notifications of microblogging updates, and how?
even if you receive Atom/RSS updates, please indicate how you are actually notified of new updates

- ☐ Mobile notifications (e.g. SMS)
- ☐ Computer pop-ups / "toaster" notifications (e.g. installed AIR software, or plug-in)
- ☐ I keep my feed visible on the screen
- ☐ I keep the number of new unread updates visible on the screen

How systematically do you read new updates? (using real-time notifications) *
by "reading an update", we mean only "read the tweet", not even opening linked content or retwitting

- ○ I read all updates as they appear (i.e. when I'm notified)
- ○ I read most updates as they appear, depending on my ongoing activity and availability
- ○ I am aware of new updates in real-time, but I ignore them most of the time
- ○ I am not aware of new updates in real-time / I don't receive notifications

When reading an update, what is the probability that you consult attached content? *
Attached content can be: a video, a blog post, or a link to any other resource that takes time to consult...

	0	1	2	3	4	5	6	7	8	9	10	
Never (0%)	○	○	○	○	○	○	○	○	○	○	○	Systematic (100%)

When reading an update, what is the probability that you respond on the microblogging platform? *
A "response" includes: replying (RE), re-tweeting (RT), or tweeting about something related

	0	1	2	3	4	5	6	7	8	9	10	
Never (0%)	○	○	○	○	○	○	○	○	○	○	○	Systematic (100%)

Please estimate the average usefulness of updates for your current activity, at the time you read them *

	0	1	2	3	4	5	
Useless (0%)	○	○	○	○	○	○	Needed (100%)

Please estimate the usefulness of updates that you read, for the longer term *

	0	1	2	3	4	5	
Useless (0%)	○	○	○	○	○	○	Needed (100%)

How are you filtering updates to read? *
(please select the most appropriate answer)

- ○ No filtering at all! I follow every contribution (I'm using another microblogging platform than Twitter)
- ○ I simply read updates from the people I follow
- ○ I mostly use a service to watch the trends (e.g. twitscoop)
- ○ Yes, I mostly use a service to emphasize the updates that are most relevant to me (e.g. my6sense)
- ○ Yes, I specified my own filtering rules (e.g. yahoo pipes filtering mash-up)

If there was a simple way to filter notifications from microblogging platforms, what criteria would you chose? *
E.g. Twitter natively sorts updates chronologically, filters by people you follow, and allows you to create lists of people. What criteria would you add?

- ☐ Filter by type of update (news, professional, personal, fun...)
- ☐ Filter by type of attached content (videos, blog posts, geographical check-ins...)
- ☐ Filter by subject (hashtags...)
- ☐ Filter by geographical location (local updates...)
- ☐ Filter by relevance with current activity
- ☐ Filter by relevance with future activities (trip plans, goals...)
- ☐ Filter by relevance with current interests
- ☐ Other:

Link to your microblogging profile
E.g. Twitter username, or the URL of your profile on the microblogging platform which usage you described in this survey

Let's keep in touch!
If you are interested in this study, or in a novel microblogging filtering service relevant to the needs you described in the survey, please provide your email address

Additional details to share?
If there is something specific about your usage, please let us know

Submit

Powered by Google Docs

FOLIO ADMINISTRATIF

THESE SOUTENUE DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

NOM : JOLY
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 14/10/2010

Prénoms : Adrien

TITRE : M.

NATURE : Doctorat

Numéro d'ordre : 2010 ISAL 0081

Ecole doctorale : InfoMaths

Spécialité : Informatique

Cote B.I.U. - Lyon : T 50/210/19 / et bis

CLASSE :

RESUME :

« A Context Management Framework based on Wisdom of Crowds, for Social Awareness applications »

At a time when social networking sites revolutionize the usages on the Web 2.0, it has become rich, easy, and fun to share private or professional content. Sharing personal information in real-time (such as news, moods, etc…), supports the maintenance of social ties at a high scale. However, the information overload which emerged from the growing quantity of signals exchanged on these services, often in real-time, motivates a regulation of these signals (called "mediated interactions"), in order to reduce the temporal cost for maintaining social networks, and implied interruptions, which have a negative impact on productivity on tasks that require long-lasting attention.

In the frame of this thesis, we have developed a filtering and recommendation system that relies on contextual similarity between users that produce and consume social signals, as relevance criteria. In our approach, contextual information is aggregated and interpreted on users' terminal(s), before being submitted on-demand to a server in the form of a set of weighted tags. In this thesis, we present a broad state of the art on context-awareness, social networks and information retrieval, we propose a formalization of our filtering problem, and we implement and evaluate its application for enterprise social networking.

MOTS-CLES : information retrieval, context-awareness, recommender systems, social networks, information overload

Laboratoire (s) de recherche : LIRIS + Alcatel-Lucent Bell Labs France

Directeur de thèse: Pr. Pierre MARET

Président de jury : Hervé MARTIN - LIG, Laboratoire d'Informatique de Grenoble

Rapporteurs :
- Anind K. DEY - Carnegie Mellon University, PA, USA
- Eddie SOULIER - UTT, Université de Technologie de Troyes

Examinateurs :
- Fréderique LAFOREST - INSA de Lyon
- Bernd AMANN - LIP6, Laboratoire d'Informatique de Paris 6