Respect de la vie privée et capture de l'activité

GUYAUX Maxime, MALIALIN Audréa

22/02/15

Abstract

Le projet ActivityHistory se propose d'enregistrer et de présenter les activités d'un utilisateur afin de permettre à celui-ci de facilement reprendre son occupation, que ce soit après une interruption ou pour retrouver des informations ou un contexte. Le sous-projet "Respect de la vie privée et capture de l'activité" consiste à ajouter une dimension "protection de la vie privée" au projet déjà existant.

Mots clés: Vie privée, capture d'activités, interruptions

1 Introduction

La lecture de ce rapport devrait vous prendre une vingtaine de minutes. Il est possible qu'au cours de cette activité vous soyez interrompu : un appel téléphonique, une question d'un collègue voir une baisse d'attention. À l'issue de cette interruption, vous risquerez d'éprouver des difficultés à vous replonger dans la lecture de ce document. Un outil simple, qui vous permettrait de revenir à votre activité est votre doigt (ou votre curseur) : accompagné de votre mémoire, il vous permettra de reprendre sans difficulté le fil de votre lecture. Qu'en est il d'une activité informatique plus générale ? D'une interruption longue, ou plus contraignante ? La réponse est simple : la reprise d'activité sera alors beaucoup moins facile.

Dans ce contexte, le Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS) ainsi que l'Université de Californie San Diego (UCSD) développent actuellement une application nommée ActivityHistory, permettant de faciliter le retour à une activité passée. Dans ce but, une grande quantité d'informations est enregistrée : c'est ici que s'immisce le problème de la vie privée : que peut-on enregistrer ? Au contraire, que ne faut-il absolument pas enregistrer ? Peut-on édicter des règles générales ou l'application doit elle être paramétrable afin de satisfaire chacun ?

2 État de l'art

2.1 Interruptions

Alors que les avancées technologiques nous permettent de réaliser de plus en plus de tâches en simultané, les capacités de mémorisation humaines, elles, sont limitées. Une fois une activité interrompue, il peut être difficile de s'y replonger.

Les interruptions se partagent en deux grandes familles : interruptions internes et externes [1]. Ces dernières sont le résultat d'éléments extérieurs tels qu'un téléphone qui sonne, une nouvelle notification, l'arrivée d'une personne. Les interruptions internes (self-interruptions) sont elles décidées par l'utilisateur par exemple quand d'une tâche émerge de nouvelles idées. Peu importe si une interruption fait partie d'une ou de l'autre de ces familles, elles sont par définition des évènements inattendues.

Les conséquences d'une interruption sont variables : si certaines engendrent une amélioration des performances d'autres ont l'effet inverse. L'effet Zeigarnik[1] nous montre que les activités au cours desquelles une personne a été interrompue marquent beaucoup plus l'esprit. En effet, sachant que telle tâche n'a pas été terminée engendre une motivation plus accrue, mais surtout la sensation d'un travail non terminé qui se traduit par une gêne que l'on va essayer de régler le plus rapidement que possible.

Si cet aspect positif a été constaté pour des tâches considérées comme simples [4], il a été montré que quand la complexité de celles-ci augmente il est fréquent qu'une interruption ait un effet néfaste. McFarlane [3] distingue quatre types d'interruptions : immédiates, négociées (choix de répondre/réagir ou pas), planifiées et médiées (un agent intermédiaire décide si l'interruption aura lieu ou pas). Si les trois dernières sont moins nuisibles car l'utilisateur a quand même un certain degré de contrôle, la première est la plus préjudiciable. Effectivement, le fait de devoir apporter de l'attention à autre chose occasionne la perte du processus de réflexion. Des études de Bailey, Konstan et Carlis[5,6] prouvent que plus une tâche est complexe plus la charge cognitive est grande et donc plus il est facile après une interruption de commettre des erreurs [2].

De plus, les interruptions étant par définition imprévisibles, incontrôlables, elles ne peuvent être évitées. La problématique n'est donc pas de les contourner mais d'aider l'utilisateur à se remettre dans un contexte précis.

2.2 Activity Based Computing

Une application seule ne peut répondre que rarement aux attentes d'un utilisateur. Ce dernier peut avoir besoin d'effectuer une recherche en utilisant un navigateur, ouvrir un document relatif à l'activité en cours, consulter ses emails et bien d'autres choses. Un utilisateur utilise donc une multitude d'applications

[1] Tendance à mieux mémoriser une tâche et ses aspects lorsque celle-ci n'a pu être terminée, se traduisant par un meilleur rappel des données concernant des problèmes ou des travaux inachevés. Source: http://www.definitions-de-psychologie.com/fr/definition/zeigarnik-effet.html

hétérogènes sur son ordinateur afin de réaliser une tâche [9]. Passer d'une tâche à une autre implique de changer complètement de configuration : ouvrir de nouvelles applications, de nouveaux fichiers etc. Il est donc nécessaire de considérer non pas le fichier ou l'application ouverte mais l'activité de l'utilisateur et ce dans un contexte précis.

C'est dans cette optique qu'a émergé le concept de bureau virtuel et notamment les Rooms system [7]. Une fenêtre est vite surchargée d'applications, l'utilisateur doit fournir un effort considérable afin de garder toutes ses fenêtres ouvertes et s'y retrouver au cours de sa navigation. Lui fournir un support lui permettant d'organiser de façon logique ses applications a donc été un premier pas vers le concept d'Activity Based Computing.

L'Activity Based Computing [9] doit soutenir l'utilisateur dans son activé en lui permettant notamment de définir une activité en la personnalisant, passer d'une activité à une autre, mais aussi d'enregistrer l'activité en cours afin de la reprendre après. Un second principe est le fait de considérer la collaboration et la communication comme faisant partie intégrante d'une activité. Il est donc nécessaire de permettre à utilisateur de pouvoir partager son activité.

Mais toutes ces notions sont à replacer dans un contexte, par exemple, l'emplacement physique d'un individu peut être utilisé pour déterminer la pertinence d'une activité. Il faut par conséquence noter que la description d'une activité dépend de l'espace et du temps [8], il est alors important pour un système basé sur l'activité de capturer à la fois l'histoire et le contexte de chaque activité car c'est cette combinaison qui révèle réellement l'activité humaine.

2.3 Vie privée

La vie privée est un concept difficile à définir mais peut être vu comme étant le droit de comprendre, choisir, contrôler les informations que l'on divulgue, mais aussi savoir avec qui elles sont partagées, pour combien de temps.

Cependant, ce qui est considéré comme privé peut être envisagé en différents niveaux selon la personne avec laquelle on accepte de partager les informations. C'est justement ce point qui explique le fait que l'on soit proche ou non d'une personne. La notion de vie privée doit être considéré dans un contexte précis [11].

L'avènement des réseaux sociaux ont quelque peu bouleversé ce concept : les liens d'amitiés proposés sont faibles. Combien de personne ont dans leur liste d'amis sur Facebook une personne qu'elles n'ont jamais rencontrée ? Alors pourquoi les utilisateurs partagent-ils leurs informations ? Selon une étude de Louise Barkhuus [11] les réseaux sociaux sont un moyen de se mettre en avant, d'avoir son heure de gloire. Des personnes se sentent importante quand elles reçoivent un nouveau commentaire sur un de leur post et ressentent le besoin de se mettre en avant par le biais de réseaux sociaux. Dans cette optique, elles n'hésitent à supprimer les identifications sur des photos qui ne seraient pas à leur avantage.

De plus, si la plupart des informations basiques (âge, nom, prénom) sont accessibles facilement, la création de groupe d'amis, la visibilité que l'on peut

mettre sur nos posts montrent que les développeurs des réseaux sociaux ont pensé au fait que les utilisateurs étaient prêts à partager leurs informations mais pas forcement tout, ni avec tout le monde. Cependant, même si cela semble utile pour des personnes sensibilisées à ces sujets, rares sont celles qui utilisent ces fonctionnalités[11].

Hormis les réseaux sociaux, nos informations sont collectées de nombreuses manières : cartes de fidélités, enquêtes, inscriptions, web tracking et bien d'autres encore [10]. Les entreprises récoltent de grandes quantités de données sur leur client afin de les exploiter et d'en créer si possible du profit. Si en Europe ces informations sont réglementées, aux Etats-Unis par exemple, la vente de données destinées au ciblage publicitaire n'est pas considéré comme illégale. Ce phénomène de « Data Brokers » apparu à la fin du XIXème siècle a connu son apogée avec les réseaux sociaux.

Après avoir récupéré ces données, il est possible de faire un travail d'agrégation et d'identification. L'agrégation consistant à regrouper des données afin de créer une représentation d'une personne. Si ces deux techniques ont des avantages certains par exemple la personnalisation des offres, de meilleurs dossiers sur les patients etc. C'est aussi une menace pour la vie privée car combiner les informations et les attribuer à une personne précise peuvent révéler des détails de la vie personnelle [10].

3 Perception de la vie privée

Dans le contexte générale de recherche autour du projet "Activity History", une série d'entretiens est réalisée, à la fois à Lyon et à San Diego afin de déceler des idées concernant trois thèmes principaux :

- L'utilité de l'application
- La manière de présenter les données[2]
- La question de la vie privée[3]

3.1 Protocole

Afin de comprendre comment était abordé le thème de la vie privée par diverses personnes, il fut nécessaire de les rencontrer dans le cadre d'un entretien.

Le travail effectué par l'équipe de l'UCSD, qui réalise elle aussi cette même série d'entretiens, dû être adapté, évidement à la langue, mais aussi à la profession des personnes interrogées : pour exemple, certaines questions relatives à la hiérarchie professionnelle durent être remaniées afin d'en tirer des réponses équivalentes, à la fois dans le monde professionnel et étudiant.

Afin de contrecarrer la contrainte temporelle imposée, il fut nécessaire d'omettre la première prise de rendez-vous. En effet, selon le protocole strictement traduit

[2]Sous-projet "activity-visualizer".
[3]Sous-projet "ter 2015"

(cf Annexe 1.1) un premier rendez-vous de dix minutes doit être pris. Cette courte durée permit d'intégrer cet entretien dès l'accord des volontaires, les impliquant ainsi d'avantage.

Le protocole utilisé s'inscrit en trois phases :

- Un premier entretien vise à dessiner le profil de l'interviewé en matière de lifelogging[4] : une série de questions concernant leurs habitudes sont alors posées. À chaque type de données collectées le sujet est interpellé à propos de l'intérêt et du partage de ces informations que ce partage soit conscient ou au contraire tout à fait inconsidéré.

- La seconde partie est effectuée uniquement par le volontaire : il doit installer le logiciel SelfSpy[5] (modifié dans le projet ActivityHistory) et enregistrer au moins une heure de son activité habituelle, en essayant au maximum d'ignorer le fait qu'il soit enregistrer.

- La dernière partie repose sur un entretien dont la durée moyenne est de 40 minutes. Le sujet passe alors en revue l'ensemble des captures d'écrans de son enregistrement sans que ces images ne soient vus par d'autres que lui. Durant ce visionnage, le sujet est soumit à un ensemble de questions permettant de savoir si d'une part les enregistrements lui permettent de retrouver l'activité qui étaient en cours, mais aussi si il y a présence d'éléments qu'il ne souhaiterait pas voir enregistrer et enfin, la possibilité de partager ces données avec diverses personnes était envisageable. A la fin de ce questionnaire, le sujet est amené à compléter une frise chronologique (cf. Annexe 1.1 Figure 3) afin de relater la journée au cours de laquelle l'enregistrement a été fait : cette frise permet de remettre le l'enregistrement dans le contexte dans lequel il était et révéler les aspects intrusifs de la capture d'activité.

3.2 Résultats

Un total de dix personnes a été interrogé deux travaillant dans des entreprises, trois étudiants en informatique et cinq enseignants chercheurs. Certes l'échantillon n'est pas représentatif de la population mais considérant le domaine d'application du projet de recherche cela fut suffisant.

Les profils recherchés étaient les suivants: des personnes qui sont sensibles à la thématique de la vie privée qui partagent des informations, ou qui au ne souhaitent pas partager leurs données ou du moins limiter ceci. Si certains des sujets étaient des accrocs aux réseaux sociaux d'autres au contraire étaient très récalcitrantes à utiliser ces outils notamment certains enseignants chercheurs en Informatique.

[4]Le lifelogging est le processus de suivi des données personnelles générées par nos propres activités.

[5]Ce logiciel enregistre l'activité d'un utilisateur grâce à des captures d'écrans et une base de données où figurent le nom de applications utlisées ainsi que le nom des fenêtres ouvertes.

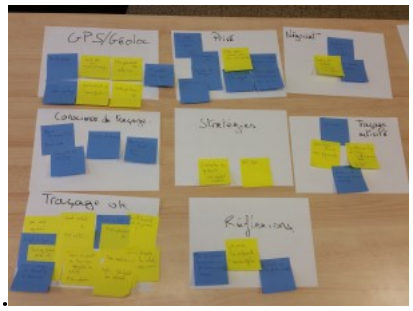
1.  2. 

Figure 1: 1. Premier diagramme d'affinité. Il a permit regrouper les informations en dix thématiques. 2. Une deuxième itération a permit de définir différemment les groupes d'idées et notamment de faire émerger les aspects liés à la géolocalisation.

Afin de tenter d'extraire des données fiables, s'appliquant à la majorité des interrogés, les données issues de chaque entretien individuel ont été extraites, ajoutées aux autres puis groupées au sein d'idées générales grâce à la méthode du diagramme d'affinité. Deux groupements sont présentés sur la figure 1.

Cette étude a permit de faire remonter de grandes idées telles que:

- La plupart des interrogés ont conscient de la vente d'informations personnelles en échange d'un service. Cependant, pour la majorité tant que le rapport praticité (service rendu) sur données personnelles reste correct il ne s'inquiètent pas plus du devenir de leurs informations.

- Une écrasante majorité des sondés n'ont aucun problème avec le traçage par géolocalisation. Cependant, les contradictions sur ce thème sont intéressantes: un sujet nous a expliqué que ce type de données n'était réellement personnel que si la personne concernée était de notoriété publique alors qu'un autre considère que connaître tous ses déplacements révélerait bien plus qu'il ne le souhaiterait.

- La frontière données privées / non privées (professionnelles, publiques ...) n'a jamais pu être clairement défini :

  - D'un point de vue individuel, certains interrogés ne peuvent eux même la définir : pour eux il n'y a pas une seule dimension qui permettrait de diviser les données en parties distinctes mais une multitude de groupes qui se recouvrent mutuellement sur certaines de leurs parties. Plus précisément, un volontaire nous a fait remarquer qu'en plus de ses projets que l'on peut considérer comme professionnels et non privés, il a développé des projets personnels et donc pour lui, privés[6].

[6] « Tout ce qui est projet publique où il n'y a pas d'importance on pourrais les garder mais il faudrait éviter les projets perso » Prononcé par un sujet anonyme lors de la 3ème partie de l'expérimentation.

  - D'un point de vue plus global, en agrégeant les données récoltées, il n'est toujours pas possible de définir une règle : mêmes les parties qui ont pu être clairement défini par certains sujest se sont retrouvées dans la partie opposée chez une personne différente.

- Cependant, un petit groupe d'idées générales nous ont été suggérées par certains participants en vue d'améliorer le respect de la vie privée. Il est évident que si ces mêmes idées étaient soumises à d'autres personnes, elle se retrouveraient rapidement dans le cas précédent. Ces idées, comme la masquage des données de configuration ou de la taille des mots de passes (déjà masqués) ont été utiles.

- Enfin, une des questions les plus importante put être écartée : les applications pouvant être utilisées à la fois dans un domaine professionnel/publique et privé n'ont quasiment jamais retenu l'intérêt des participants concernant un éventuel traçage de celle-ci.

En plus d'une très grande diversité de réponses obtenues, le nombre d'interrogés reste insuffisant pour inférer sur une plus large population : néanmoins, cette diversité à conduit au choix, à la fois de la méthode de filtrage et de l'interface utilisateur.

4 Développement du module

4.1 Prototypage

Sous l'impulsion des résultats des entretiens, un module entièrement paramétrable a été imaginé, agissant à trois niveaux différents, chacun exécutant un certain nombre de filtres :

- Filtrage avant l'enregistrement
  - Heures autorisées
  - Géolocalisation : position où il est autorisé d'enregistrer
  - Applications non autorisées
- Filtrage durant l'enregistrement
  - Arrêt / lancement manuel de l'enregistrement
  - Suppression des dernières minutes d'enregistrement
- Filtrage après enregistrement
  - Mots-clés qui ne doivent pas être présents durant l'enregistrement
  - Ainsi qu'un effet rétroactif sur tous les autres filtres, utilisable manuellement.

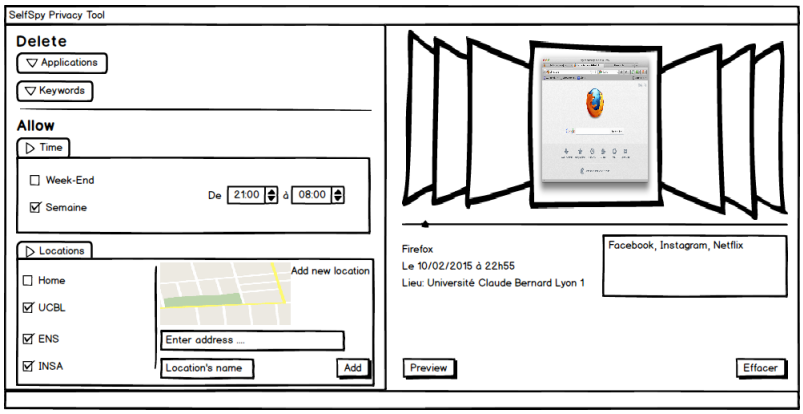


Figure 2: Prototype retenu.

Afin de mieux visualiser le travail à réaliser, un travail de prototypage a dû être réalisé. Pour ce fait deux séries de maquettes ont été réalisés : un premier jet a permis de mettre en forme les idées en permettant à l'utilisateur de configurer au maximum ses filtres. Cependant, après réflexions, l'outil proposé semblait trop compliqué, non seulement au niveau de la mise en oeuvre mais aussi au niveau IHM. A partir de cela des simplifications ont été apportées (cf figure 2) et le développement a pu commencer.

L'idée est de fournir un outil permettant de configurer le filtrage avant et pendant tout en étant . De plus, ce module devra être capable d'aider l'utilisateur à bien configurer les filtres mais aussi lui permettre de visualiser son activité afin d'en supprimer une partie manuellement, à l'aide ou non de filtres.

N'ayant pas les moyens temporels et logistiques (l'application ne fonctionnant que sous MAC OS), seule la partie de filtrage "après enregistrement" fut développée. Néanmoins, l'intégralité des filtres proposés (heure, localisation, application au premier plan et mots-clés) a été développé en Python (même langage que le projet ActivityHistory) et un appel à une ces fonctions au moment voulu permettra la mise en place de la solution de protection de vie privée en supprimant les données (éléments de la base de données et screenshots) considérées comme personnelles par l'utilisateur.

4.2 Filtrage

4.2.1 Temporel

Il s'agit du plus simple filtre mis en place : l'utilisateur indique dans l'interface du module de vie privée les horaires auxquels il accepte d'être enregistré. Ce filtre est particulièrement utile "avant enregistrement". En effet, il évite d'enregistrer, de stocker des informations que l'utilisateur ne souhaitera pas garder. Il permet donc un gain à la fois de temps mais aussi d'espace et de calcul.

4.2.2 Applications

Le projet ActivityHistory est basé sur le projet open source SelfSpy qui enregistre notamment l'évolution des fenêtres actives dans une base de données. Le travail consiste donc à mettre en relation les screenshots (contenant les dates) et les entrées de la base de données afin de récupérer les applications actives à chaque instant. Ce filtre peut être utilisé durant l'enregistrement afin de le mettre en pause automatiquement lorsque certaines fenêtres sont actives. Dans le cadre de ce projet, il permet l'utilisation rétroactif du filtrage par applications.

4.2.3 Localisation

Le filtre par localisation permet par exemple d'enregistrer uniquement l'activité effectuée au travail. A cette fin, l'utilisateur doit définir les lieux autorisés et non pas l'inverse.

4.2.4 Mots-clés

Il s'agit du filtrage le plus stricte, mais aussi le plus compliqué à mettre en place. La seule solution utilisable actuellement pour l'extraction de mots-clés est la reconnaissance optique de caractères (OCR) basé sur les captures d'écran (Le choix de l'outil utilisé est expliqué en Annexe 2.1).

Cette méthode étant très coûteuse en temps (Tentatives d'améliorations : cf. Annexe 2.2), une solution a du être adoptée afin de limiter le nombre d'OCR à effectuer : cette solution consiste en la classification d'images par lot. Ces lots d'images correspondent à un changement minimal (définit empiriquement) de l'histogramme de couleurs entre deux séries de screenshots. Ainsi, une seule OCR est effectuée par série de screenshots.

De plus, afin d'éviter à l'utilisateur d'avoir à patienter un temps trop long, chaque OCR effectué enregistre son résultat qui pourra être réutiliser plus tard, ce qui s'avère particulièrement utile lorsque l'utilisateur paramètre son filtre et effectue plusieurs prévisualisations.

4.3 Architecture

Le projet ActivityHistory étant développé en Python, la partie "active" du module, contenant à la fois les filtres et les fonctions de nettoyage (supprimant toutes les données - images, textes et bases de données concernant une période) a été développé dans ce même langage. Cela permettra, à terme la fusion des deux outils.

- Chaque filtre est représenté par une classe Python (permettant l'activation ou non de ce dernier).
- Une classe Grouper permet de grouper les images en fonction de leurs distances

- Deux scripts permettent respectivement le filtrage et le nettoyage de données

Le tout s'articule autour d'une classe imageSet, représentant une suite de screenshots, associée à toutes les autres données (temps, localisation, application au premier plan et texte extrait)

L'interface utilisateur permettant la configuration, la prévisualisation et le filtrage "après enregistrement" a été codé en JavaScript, au sein d'une application NodeJS utilisant le framework Sails. L'utilisateur entre les données permettant la configuration de l'outil de protection de la vie privée, qui sont ensuite enregistrée dans la base de données sqlite via la partie serveur de l'application en utilisant des requêtes AJAX. Le but étant que l'utilisateur voit le module comme une page de configuration et non pas comme une page web qui se rechargerait après chaque envoi de formulaire.

A l'inverse, pour ce qui concerne la partie prévisualisation de l'application, le serveur NodeJS appelle le script Python puis retourne les informations à la partie client qui met forme les données reçues.

5 Conclusion

Malgré la très grande diversité d'opinions face au sujet, nous avons tenté, en développant un outil totalement paramétrable, de répondre à la problématique de la vie privée au sein d'une application enregistrant l'activité.

Bien loin d'avoir pu intégrer la totalité des fonctionnalités que nous souhaitions et encore moins ce qu'il eût fallu nécessaire afin de satisfaire l'intégralité des éventuels futurs utilisateurs, nous nous sommes concentré sur la réalisation d'une application fonctionnelle dans le temps imparti, mais aussi réutilisable et convenant au plus grand nombre.

Un grand nombre de technologies a été utilisé dans la mise en oeuvre de ce projet et en particulier certaines que n'avions jamais rencontré, en particulier NodeJS, ses frameworks Sails et Express, Node-webkit et Python. De plus, nous avons approché une démarche de recherche dans son ensemble, de l'état de l'art à la production, en passant par l'expérimentation. Celle-ci nous a permis de faire des rencontres qui furent, pour la plupart, très enrichissante autant pour l'évolution du projet dans sa globalité mais aussi dans notre sous-projet et pour nous même.

Après fusion au projet de base et inclusion des fonctionnalités de filtrage avant et après enregistrement, il pourrait être intéressant d'effectuer des tests utilisateurs. De plus, comme indiqué précédemment, l'OCR est le talon d'Achile du module : l'utilisation d'un outil d'accessibilité permettrait de déplacer la détection de mots-clés au moment de l'enregistrement, et donc éviter la lenteur et la moyenne qualité de l'OCR.

6 Références

1. González, V. M., & Mark, G. (2004). Constant, constant, multi-tasking craziness: managing multiple working spheres, In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 113-120). ACM.

2. Adler R. F., Benbunan-Fich R. (2014) The effects of task difficulty and multitasking on performance, Interacting with Computers, First published online: March 9, 2014, DOI: 10.1093/iwc/iwu005

3. McFarlane D. C., Latorella K. A. (2002) The scope and importance of human interruption in human-computer interaction design,Human-Computer Interaction, 17 (1),

4. Speier, C., Valacich, J. S., & Vessey, I. (1997, December). The effects of task interruption and information presentation on individual decision making. In Proceedings of the eighteenth international conference on Information systems (pp. 21-36). Association for Information Systems.

5. Bailey B. P., Konstan J. A., Carlis J. V. (2000) Measuring the effects of interruptions on task performance in the user interface, in:IEEE International Conference on Systems, Man, and Cybernetics 2000: Cybernetics Evolving to Systems, Humans, Organizations, and Their Complex Interactions, Vol. 2, Piscataway: Institute of Electrical and Electronics Engineers, 757-762

6. 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: M. Hirose (Ed.) Human-Computer Interaction - INTERACT 2001 Conference Proceedings. Amsterdam: IOS Press, 593-601.

7. Henderson Jr, D. A., & Card, S. (1986). Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. ACM Transactions on Graphics (TOG), 5(3), 211-243.

8. Houben, S., Bardram, J. E., Vermeulen, J., Luyten, K., & Coninx, K. (2013, April). Activity-centric support for ad hoc knowledge work: a case study of coactivity manager. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2263-2272). ACM.

9. Bardram, J. E. (2005, September). Activity-based computing-lessons learned and open issues. In ECSCW 2005 workshop, Activity-From a theoretical to a computational construct.

10. Christopher Johnson, Rakesh Agrawal: "Intersections of Law and Technology in Balancing Privacy Rights with Free Information Flow", 4th IASTED International Conference on Law and Technology, Cambridge, MA, Oct. 2006.

11. Barkhuus, L. (2012, May). The mismeasurement of privacy: using contextual integrity to reconsider privacy in HCI. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 367-376). ACM.

7 Remerciements

Nous tenons à remercier :

- Pour leur temps, leur aide et leur accueil, toutes les personnes ayant accepté de nous recevoir pour effectuer les entretiens.
- Pour leur accueil et leur café, l'équipe SICAL.
- Pour leur accueil et leur aide, l'équipe de San Diego travaillant sur ce projet.
- Pour son implication, sa disponibilité et son aide, notre encadrant M. Tabard.

8 Annexes

8.1 Entretiens

8.1.1 Script d'entretien

Est présenté ici un bref résumé des questions posées au cours des trois parties des entretiens :

- Expérience en lifelogging (10 min)
  - Question principale de cette partie : Quel type de lifelogging avez-vous expérimenté ? Pourquoi ? Savez-vous ce qu'il est advenu de vos informations ?
  - Série de questions :
    - Suivi automatique ou manuel ?
    - Pourquoi avez-vous suivi ces informations ?
    - Avez-vous arrêté ? Pourquoi ?
    - Savez-vous quelles données étaient stockées ? Où ? Par qui ? Comment étaient-elles utilisées ?
  - Sujets abordés :
    - Santé, nutrition, fitness
    - Financier
    - Géographique
    - Temps
    - Autres (réseaux sociaux, cartes de fidélités, achats en ligne)
- A la fin du premier entretien, le logiciel Selfspy modifié (dans le projet ActivityHistory) est installé sur la machine du sujet, des instructions lui sont donnés quant à l'enregistrement et le respect de sa vie privée.

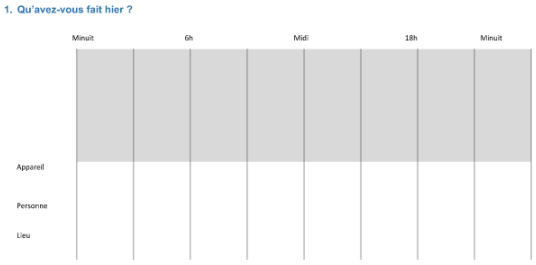


Figure 3: Frise chronologique proposée aux sujets.

- Revue de l'enregistrement (20 min)
  - Question principale : Que trouvez-vous important d'enregistrer ? Qu'êtes-vous prêt à partager ?
  - Le fait d'être enregistré a-t-il modifié votre façon de travailler ? Si oui, comment ?
  - Avez-vous éteint l'enregistrement à un quelconque moment avant la fin ? Si oui, pourquoi ?
  - Pour chaque activité :
    * Selon vous est-ce utile d'enregister cette activité ? En quoi ?
    * Est-elle privée ? En quoi ?
    * Serait-ce utile ou accepteriez-vous de partager cet enregistrement avec [supérieur/collègue/subalterne/inconnu...] ?

8.1.2 Frise chronologique

Les interrogés sont invités à remplir la frise chronologique présentée en figure 3, à l'aide d'un calendrier ou de leurs notes, afin de remettre en contexte leur enregistrement, le plus souvent du jour passé.

8.1.3 Entretiens

L'intégralité des données receuillies étant anonymes, nous ne pouvons entrer plus en détails sur le contenu de celle-ci. Néamoins, voici quelques citations intéréssantes receuillies auprés de personnes différentes :

- "De nos jours, on écrit pas trop de mails sensibles."
- "A partir du moment où je peux paramétrer, supprimer, ce que je ne veux pas donner, j'accepterais de partager."
- "Pour protéger mes données personnelles, je n'installe rien sur ma machine, et utilise le moins de service possible."

8.2 Reconaissance optique de caractères

8.2.1 Choix de l'outil

Le Tableau 1 montre une comparaison entre différents OCR : on voit rapidement se détacher deux logiciels : abbyocr et tesseract. Abbyocr étant un logiciel propriétaire, et tesseract sous licence Apache, nous avons choisi de travailler principalement avec tesseract.

Cependant, afin d'effectuer rapidement des tests comparatifs, nous avons aussi utilisé abbyocr comme référence : nous avons en effet remarqué la grande qualité des résultats de ce logiciel.

	abbyyocr	cuneiform	gocr	ocrad	tesseract
License	Proprietary	BSD	GPL2	GPL3	Apache 2.0
Version	8.0	0.9.0	0.48	0.19	SVN r402
Input-Format	PNG[1)]	PNM	PNM	PNM	TIF[2)]
Recognition rates and time spent:					
courier/black	100% (2.92s)	61% (1.11s)	67% (0.09s)	21% (0.02s)	81% (0.63s)
courier/gray	100% (2.85s)	✖	67% (0.09s)	21% (0.03s)	81% (0.63s)
justy/black	11% (3.62s)	3% (1.14s)	31% (0.11s)	1% (0.02s)	15% (0.61s)
justy/gray	14% (3.45s)	✖	31% (0.10s)	1% (0.02s)	15% (0.60s)
times/black	100% (2.80s)	96% (1.07s)	76% (0.16s)	82% (0.03s)	92% (0.74s)
times/gray	100% (2.87s)	✖	76% (0.16s)	82% (0.03s)	92% (0.74s)
verdana/black	100% (2.90s)	95% (1.07s)	98% (0.10s)	98% (0.03s)	98% (0.45s)
verdana/gray	100% (2.85s)	✖	98% (0.10s)	98% (0.02s)	98% (0.46s)

Table 1: Comparatif de quelque logiciels d'OCR. Source : http://www.splitbrain.org/blog/2010-06/15-linux_ocr_software_comparison

8.2.2 Tentatives d'optimisation

Les résulats obtenus par OCR étant assez loin de nos espérances, que ce soit en terme de temps ou de qualité, nous avons dans un premier temps tenté d'améliorer celui ci : nou avons effectué plusieurs séries de benchmark. Quelques exemples sont visibles sur les figures 4 et 5.

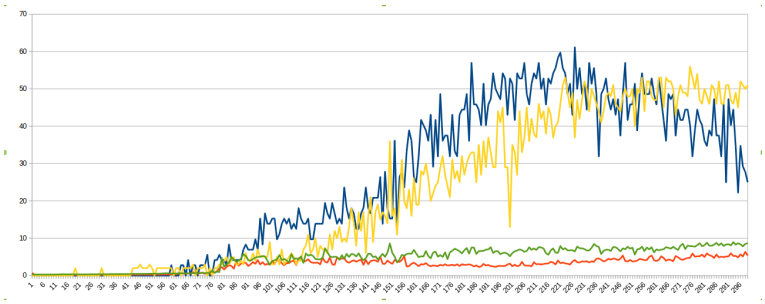


Figure 4: Evolution de la reconaissance et du temps en fonction de la taille : Bleu : taux de réussite en noir et blanc; Jaune : tax de réussite en couleur; Rouge : temps d'execution en noir et blanc; Vert : temps d'execution en couleur.

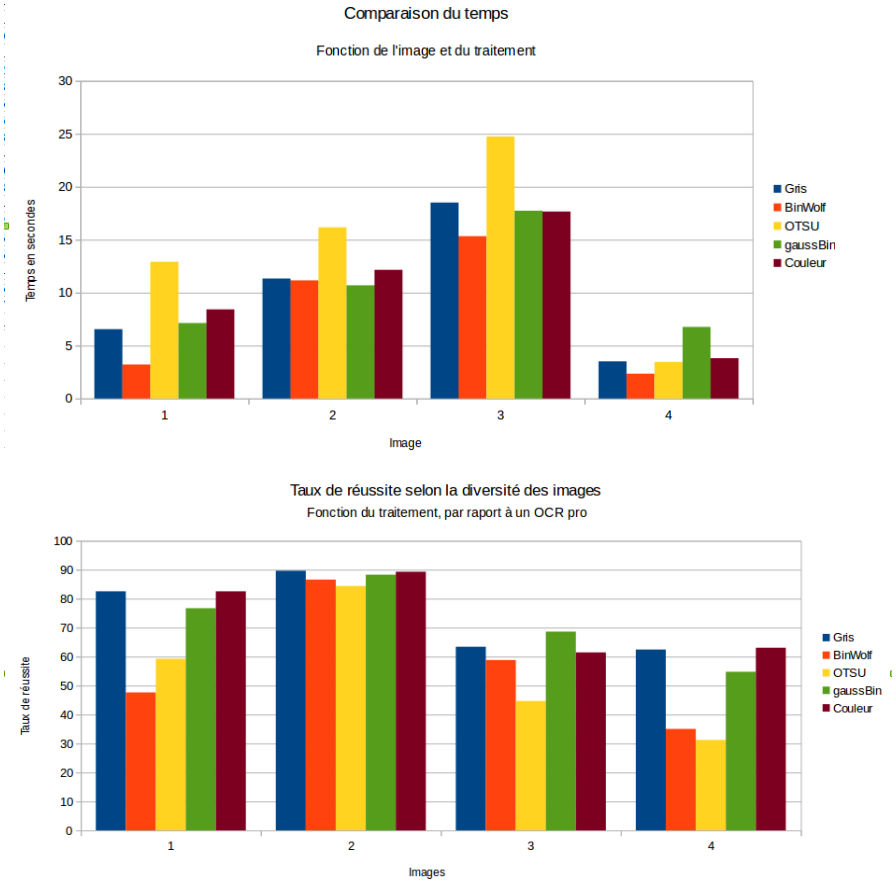
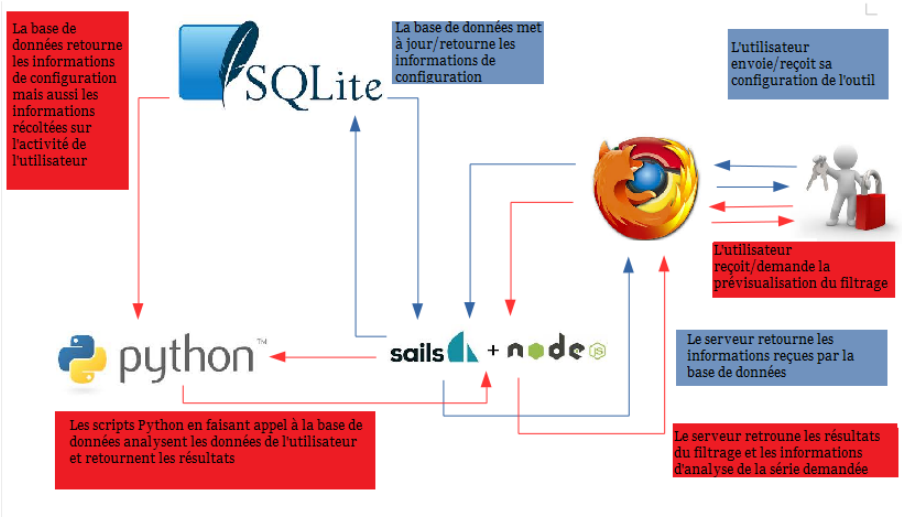


Figure 5: Taux de réussite (en comparaison avec les résultats de abbyocr) et temps pris pour l'analyse de 4 images différentes, avec 5 traitements différents : Niveaux de Gris, Binérisation (Tests effectués avec l'algorithme fourni par Christian Wolf : http://liris.cnrs.fr/christian.wolf/software/binarize/), méthode d'OTSU, flou gaussien et binérisation, ou aucune modification : couleur.

Malheureusement, aucune solution miracle n'a été trouvé : devant la diversité de screenshots : le choix le plus simple fut retenu : utiliser tesseract directement, il se charge seul d'un minimum d'optimisations.

8.3 Module développé

8.3.1 Plan général du module



8.3.2 Résultat