L'informatique sensible au contexte

~ Sommaire ~

Introduction

Après l'arrivée et l'installation durable des ordinateurs dans nos foyers, l'informatique est appelée à être de plus en plus présente au quotidien. L'intégration de puces dans les voitures, l'explosion de la téléphonie mobile, l'arrivée sur le marché des ordinateurs portables et maintenant des ultra-portables, sorte de fusion entre un ordinateur portable et un organisateur, donnent une idée de ce qui peut nous attendre en informatique dans les années à venir : de plus en plus de mobilité et de communication et des utilisateurs s'attendant à ce que leurs appareils répondent à des attentes motivés aussi par l'environnement.

Ces projets sont dans les cartons des laboratoires depuis une dizaine d'années, sous le nom d'abord d'informatique ubiquitaire puis plus spécifiquement d'informatique contextuelle. La présente bibliographie a pour intention de présenter un état de l'art du domaine en analysant certains des articles clés publiés par les équipes de chercheurs qui ont pied dans ces nouvelles formes d'interactions homme machine.

Dans une première partie, je présenterai une synthèse des définitions données pour l'informatique contextuelle. Ensuite, je donnerai un tour d'horizon des utilisations possibles du contexte en présentant des projets de recherches. Enfin dans une troisième partie, je présenterai certaines méthodologies proposées pour l'organisation et la construction d'applications utilisant le contexte.

I) Définition du contexte

1.1) Vers la notion de contexte [4, 14, 8, 10, 2, 1, 16]

Les humains sont assez doués lorsqu'il s'agit de communiquer entre eux, de transmettre des idées, de réagir aux propos d'un interlocuteur. La richesse du discours permet de transmettre des notions très subtiles. Mais deux personnes ou un groupe de personnes communicantes ont aussi une sorte de notion implicite du monde qui les entoure et qui leur permet d'ajouter un cadre au dialogue. L'Homme se sert en permanence, et même sans s'en rendre compte, de tous ces artifices qui viennent enrichir et rendre le dialogue beaucoup plus simple et efficace.
C'est de cette prise en compte de l'environnement au sens large dans nos modes de communications que découle l'idée du contexte.

Son utilisation en informatique est nouvelle. Jusqu'ici, dans les interactions humains-machine, l'Homme réduit intentionnellement ses mécanismes de communications, en s'adaptant à la pauvreté des moyens de communication, simples périphériques d'entrée-sortie dont dispose la machine (ordinateur ou autre). Cet appauvrissement volontaire du contexte rend le dialogue entre homme et machine artificiel, complexe [4].

Figure 1 : application sans contexte

Les applications informatiques sont traditionnellement conçues comme des boîtes noires (*figure 1*), prévues pour réagir à des entrées et produire des sorties sur le mode stimulus-réponse. A une entrée donnée correspond un état de sortie. En fait, ce qui a fait la force de l'informatique a justement été la recherche d'une certaine indépendance au contexte d'utilisation. C'est la notion clef d'abstraction, le comportement d'un système doit être prédictif, quel que soit son cadre d'utilisation [14]. Cette idée a été revue depuis les années 80, avec l'apparition des premières interfaces graphiques, des logiciels re-configurables comme Emacs mais surtout avec MacOS des premiers Macintosh. Une première prise en compte du contexte d'utilisation a vu le jour dans les logiciels de bureautique par exemple où l'utilisateur pouvait enregistrer des préférences ou définir des modèles de documents. Le système s'en remettait toutefois complètement à l'utilisateur pour la définition de contextes d'utilisation [14]. Les premières utilisations réelles du contexte sont apparues au début des années 1990 avec les premières recherches sur l'informatique mobile ou embarquée dans les laboratoires Olivetti Research, avec le projet Active Badge et au PARC Xerox avec les premières expérimentations d'informatique ubiquitaire.[11]. L'utilisation du contexte a suivi dès lors deux voies parallèles :

- L'utilisation du contexte pour améliorer les interactions entre-homme et machine du point de vue de l'ergonomie des interfaces et de la connaissance d'informations sur l'utilisateur s'est considérablement développée avec l'arrivée du Web et l'utilisation des Cookies, la complétion automatique des champs de formulaires, la notion d'historique d'utilisation. Les interfaces graphiques suivent le même chemin et se re-configurent en fonction des fonctions les plus utilisées par l'utilisateur. Les menus contextuels savent proposer une série de fonctionnalités disponibles uniquement dans un contexte donné. Les traitements de textes actuels proposent certaines fonctions de corrections automatiques comme

l'accentuation ou la mise en majuscule des têtes de phrases. Les derniers systèmes d'exploitation proposent une gestion personnalisée par utilisateurs, avec des gestionnaires de coordonnées, de mots de passe qui vont pouvoir être réutilisés par le système pour renseigner des formulaires, envoyer du courrier... Cette approche du contexte se sert uniquement de l'expérience acquise avec l'utilisation ou de modèles contextuels conçus dès le départ (correcteurs automatiques) pour simplifier et guider l'utilisateur, celui-ci doit pouvoir se concentrer sur des tâches de plus haut niveau en laissant le soin au système de factoriser les tâches répétitives [14].

- L'informatique mobile avec d'abord l'arrivée des ordinateurs portables puis plus récemment les téléphones cellulaires, les organisateurs et enfin les ultra-portables ouvrent un champ d'expérimentations très vaste pour une utilisation du contexte basée non plus uniquement sur les caractéristiques de l'utilisateur mais aussi sur l'environnement géographique, le temps... Le contexte en informatique mobile fait un usage intensif de capteurs qui viennent compléter le système existant. C'est donc un par un enrichissement des appareils que l'on arrive à capter le contexte environnemental physique. L'exemple le plus répandu de l'utilisation du contexte en informatique mobile a été introduit par les ordinateurs portables et les systèmes de mise en veille automatique après une période d'inactivité [1]. L'utilisation de l'environnement physique en informatique mobile est très à la mode ; la localisation géographique de l'utilisateur est beaucoup utilisée, comme nous allons le voir, peut être parfois au détriment des autres éléments de contexte [1].

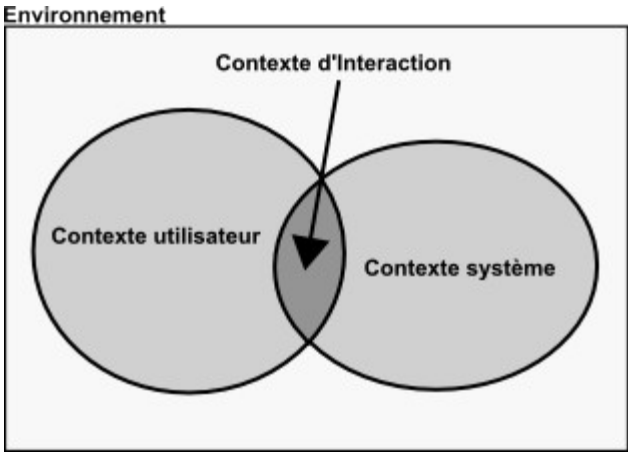
1.2) Définition formelle du contexte

Si chacun peut avoir une idée générale de ce qu'est le contexte, il est beaucoup plus dur de donner une définition précise à la notion de contexte. Il existe en fait à peu près autant de définitions de contexte qu'il y a d'équipes de recherche. Toutefois, certaines caractéristiques communes peuvent être retirées. Gaetan Rey et Joelle Coutaz proposent [2] quatre axes de définitions du contexte sur lesquels s'accordent une majorité des chercheurs :

- Il n'y a pas de contexte sans contexte : La notion de contexte doit se définir en fonction d'une finalité. Par exemple, on cherche à adapter dynamiquement les capacités interactives d'un système.
- Le contexte est un espace d'information qui sert l'interprétation : la capture du contexte n'est pas une fin en soi mais les données capturées doivent servir un objectif.
- Le contexte est un espace d'informations partagé par plusieurs acteurs : ici, il s'agit de l'utilisateur et du système.
- Le contexte est un espace d'information infini et évolutif : le contexte n'est pas figé une fois pour toutes mais se construit au cours du temps.

Ils définissent ainsi le contexte d'interaction comme étant l'intersection entre les connaissances contextuelles de l'ordinateur et celles de l'utilisateur ou l'idée que l'utilisateur se fait des connaissances contextuelles du système (*figure2*).

Figure 2 : Définition du contexte d'interaction



Le contexte est défini ici par un espace infini de variables non connues par avance par le système.

Selon Anind K. Dey et Gregory D. Abowd [4], cette définition est trop vague pour donner un cadre formel au contexte. Leur étude vise à donner une définition la plus précise possible en évitant les de se baser sur une énumération d'exemples de variables contextuelles et en évitant l'extrême inverse qui consisterait à donner une notion floue du contexte. Pour eux, le contexte est l'ensemble de toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité pouvant être un acteur, un lieu, un objet de l'environnement considéré comme utile à l'interaction entre la personne et le système.

Cette définition permet à un développeur de savoir instantanément ce qui est du contexte et ce qui ne l'est pas. Si une information peut être utilisée pour caractériser ou influer sur les interactions entre un utilisateur et le système, alors, cette information fait partie du contexte. Elle est hors contexte dans le cas inverse. Si l'on prend par exemple comme situation une personne dans son bureau en train de rentrer des données dans un tableur, les acteurs sont la personne et le système (logiciel tableur). S'il y avait à ce moment-là d'autres personnes en train de discuter dans le bureau, cela ne changerait pas les interactions entre les deux acteurs, ces autres personnes sont hors contexte. Si l'on imagine maintenant le même cas de figure avec cette fois-ci l'acteur humain en train de dicter les données du tableur plutôt que de les taper au clavier. Les autres personnes pourront avoir une influence dans les interactions entre les acteurs, pour peu qu'elles parlent trop fort. Elles sont intégrées au contexte, comme élément perturbateur.

Schilit et Theimer dans [16] ont les premiers introduit le terme "context-aware" (sensible au contexte) mais leur définition du contexte ne porte que sur des variables comme la localisation, l'identité des personnes proches et des objets ainsi que les changements dans ces objets. Avec cette définition, il est impossible pour un élément pris dans l'environnement, de savoir avec certitude s'il fait partie ou non du contexte. Cette définition est trop spécifique pour Dey et Abowd. Il est impossible d'énumérer de façon exhaustive toutes les variables d'environnement qui peuvent influer sur le contexte.

1.3) Catégorisation du contexte

Pour bien comprendre et appliquer cette définition du contexte, il est plus simple de passer par une catégorisation des variables du contexte. Selon Shilit *et al.* [8] , le contexte se décompose en trois sous-classes où chacune des variables répond à l'une des questions "Où suis-je ?", "Avec qui suis-je ?", "Quelles sont les ressources de mon environnement proche?". Ryan *et al.* [17] catégorisent le contexte en identité de l'utilisateur, ressources de l'environnement proche, localisation de l'utilisateur et période temporelle d'exécution de l'interaction.

On voit apparaître en filigranes de toutes ces catégories, comme le résument Dey *et al.* [4], les notions de temps, d'identité, d'activité et de localisation. En fait toute variable contextuelle est utile pour éclaircir une ou plusieurs des questions "Quand ?, Où ?, Quoi ?, Qui ?". Le système ou les concepteurs du système vont utiliser ces informations catégorisées pour répondre au "Pourquoi ?" de l'occurrence d'une situation.

Un utilisateur équipé d'un système contextuel pour, par exemple une visite de ville ,et qui va s'approcher d'un monument, va voir apparaître sur ce guide des informations sur le monument. Le concepteur du système répond à la question "Pourquoi l'utilisateur s'approche-t-il ?" en lui envoyant des informations sur le monument. La variable contextuelle utilisée est la localisation de l'utilisateur mais aussi son environnement, c'est à dire que le monument fait aussi partie du contexte puisqu'il influe sur la réponse du système.

Temps, identité, activité et localisation sont des catégorisations primaires du contexte, ce sont les seules nécessaires pour expliciter une situation. Il est possible de les décomposer en sous-variables. De l'identité, on va pouvoir trouver l'adresse, le courriel et d'autres informations personnelles. Ces sous-catégories sont en fait considérées comme des attributs des catégories primaires : le numéro de téléphone d'une personne peut-être connue en passant par son identité dans un carnet de numéros.
L'utilisation des catégories et des attributs donne une première idée de l'organisation que peuvent avoir des variables du contexte.

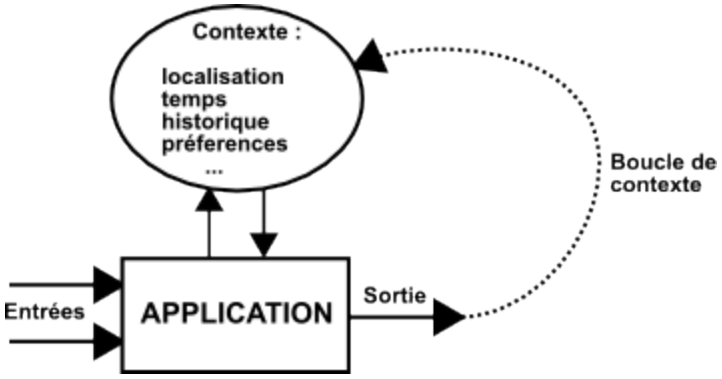
1.4)Définition de la sensibilité au contexte

On dit d'un système qu'il est sensible au contexte s'il peut tirer, interpréter et utiliser des informations issues du contexte et adapter sa réponse en fonction du contexte d'utilisation. Cette définition correspond à peu près à la première définition de la sensibilité au contexte proposé par Shilit et Theimer en 1994. Depuis beaucoup d'enrichissements ont étés apportés à cette définition. On fait maintenant le distinguo entre les applications qui utilisent le contexte, comme par exemple un service de météo qui aura besoin d'informations de localisation et de temps pour produire un bulletin et d'autres applications qui adaptent leur comportement en fonction du contexte. L'utilisation simple du contexte n'implique pas une modification du comportement du système.

Salbert *et al.* [18] définissent la sensibilité au contexte comme étant la meilleure capacité d'un système à agir en temps réel avec des données provenant du contexte. Dey limite sa définition à l'interface homme machine sans prendre en compte l'application en elle-même.
D'autres définitions sont plus orientées vers l'adaptation au contexte : Brown [19] dit d'une application sensible au contexte qu'elle doit automatiquement extraire de l'information ou effectuer des actions en fonctions du contexte utilisateur détecté par les capteurs.

Enfin Dey et Abowd [4] proposent qu'un système soit sensible au contexte s'il utilise des informations du contexte pour mettre à disposition des informations ou des services utiles à l'utilisateur, l'utilité dépendant de la tache de l'utilisateur. Avec cette définition, une application qui se contenterait d'afficher à l'écran le contenu des variables de contexte est une application sensible au contexte légitime, même si elle ne modifie pas son comportement.

Figure 3 : Application tenant compte du contexte



Hermann et Selker [14] complètent le schéma de la boîte noire du système informatique en ajoutant des éléments de contexte qui vont pouvoir agir sur l'application (*figure 3*). De même, il est possible pour l'application d'agir et de modifier le contexte. C'est ce qu'ils appellent la boucle de contexte où la sortie contextuelle de l'application va modifier les valeurs des variables contextuelles.
On rencontre dans la littérature beaucoup de synonymes de "sensible au contexte". on parle de systèmes "adaptatifs", "réactifs", "dirigés par l'environnement", "localisés". Toutes ces dénominations sont voisines de la notion de sensibilité au contexte.

Toute la difficulté de la sensibilité au contexte provient de la capture, la représentation mémoire et le traitement des données du contexte. La plupart du temps, il faut équiper les appareils de capteurs supplémentaires. Les données des capteurs doivent être filtrées et traitées avec souvent une utilisation de système d'intelligence artificielle. Enfin, l'utilisation du contexte se fait souvent en mélangeant les données en provenance de plusieurs capteurs. La déduction indirecte d'éléments du contexte est difficile pour un système automatique. Comment par exemple faire comprendre à une application que si trois personnes se rassemblent dans une pièce à un moment donné de la semaine (informations en provenance des capteurs de contexte), il peut s'agir d'une session de travail de groupe hebdomadaire.
Cette difficulté à l'abstraction explique que l'on ne trouve pas beaucoup de projets de recherche qui mélangent différentes sources d'information. En pratique, énormément de projets utilisent la localisation comme seule variable de contexte [1].

1.5) Possibilités caractéristiques des applications sensibles aux contextes [4]

- Les informations et les services peuvent être présentés à l'utilisateur dans le contexte courant. Ceci inclut la sélection d' informations de proximités ("où est la banque la plus proche ?") et de services ainsi que les commandes contextuelles (interface modifiable).
- Exécution automatique d'un service dans un certain contexte. Ceci inclut les actions déclenchées par des capteurs de contexte ("context-triggered actions"), si la lampe s'allume quand une personne rentre dans une pièce et l'adaptation automatique au contexte, avec par exemple un téléphone qui change son volume d'écoute en fonction du bruit ambiant
- étiquetages du contexte à l'information pour une restitution ultérieure.

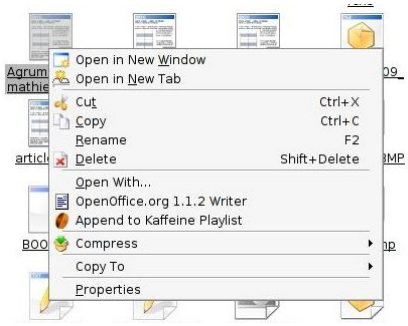
II) Exemples d'utilisation du contexte

Je présente ici des utilisations possibles d'éléments du contexte. Après une revue des simplifications de contextes existants dans les logiciels grand public, qui ont fait l'objet des premières recherches sur l'utilisation de contexte et sont à présent bien présents, je présenterai ensuite des systèmes mobiles utilisant le contexte qui sont plus documentés dans la littérature scientifique.

2.1) L'utilisation du contexte dans les logiciels grand public [14]

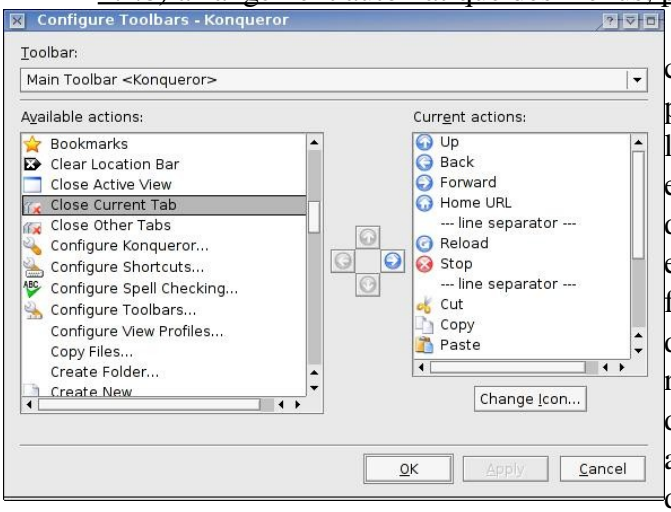
Historiquement, c'est vers la première catégorie d'utilisation du contexte (cf "vers la notion de contexte"), dans le but d'améliorer les interactions entre-homme et machine que se sont concentrées les premières utilisations de ce que l'on ne nommait pas encore prise en compte du contexte. Très liées à l'utilisation des interfaces graphiques, ces fonctionnalités mettant en oeuvre le contexte d'utilisation sont maintenant devenues transparentes et nous y sommes très habitués.

2.1.a)Les menus contextuels



C'est un widget particulier d'interface introduit par le Macintosh qui peut être appelé par la souris n'importe où au sein de l'écran ou d'une fenêtre et que son contenu dépend de cet endroit. Ici le contexte récupéré est celui de l'application, telle qu'il a été prévu par le concepteur de l'application, le contexte utilisateur n'influence pas sur le menu contextuel. L'utilisation de menus contextuel permet à l'utilisateur de disposer d'un panorama des possibilités offertes par l'application dans un contexte donné.

2.1.b) arrangement automatique des menus, personnalisation

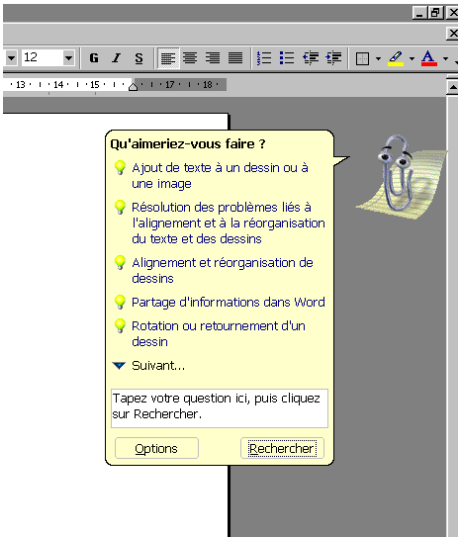


La plupart des logiciels professionnels du commerce donnent la possibilité à l'utilisateur de personnaliser les menus et les barres d'outils du logiciel pour l'adapter précisément à ses besoins. Il est possible de modifier la place des barres d'outils, d'en faire des fenêtres indépendantes, de rajouter ou enlever des boutons à celles-ci, de personnaliser la fonction associée à un bouton. C'est le premier type d'utilisation du contexte de l'utilisateur. On remarque que toutes les modifications du contexte d'utilisation sont à la charge de l'utilisateur et aucunement à l'initiative du système lui- même. Les dernières versions de Office de Microsoft proposent une simplification contextuelle des menus sans intervention de l'utilisateur en se basant sur l'historique d'utilisation (la variable contextuelle) pour masquer les fonctions peu utilisées.
Certaines applications enregistrent un profil par utilisateur et stockent les modifications d'interface opérées par chaque utilisateur pour que même si plusieurs personnes travaillent sur le même logiciel, chacune retrouve son interface personnalisée.

2.1.c) Les cookies des navigateurs Internet

Un cookie est une courte chaîne de caractères déposée dans un fichier de l'ordinateur par le navigateur web. Cette technologie a été introduite par Netscape dans son navigateur Navigator et récupérée par la plupart des autres navigateurs. Un cookie sert à stocker une information du contexte utilisateur pour pouvoir réutiliser cette information lors d'un prochain passage de l'utilisateur sur le site émetteur du cookie. Il peut s'agir d' informations personnelles recueillies par le serveur, de réponses à un formulaire, d'une date de passage ... Chacune des variables de contexte contenue dans un cookie est renseignée soit par le système (ici le serveur) soit par l'utilisateur qui peut par exemple renseigner les champs d'un formulaire qui sera stocké sous forme de cookie.

2.1.d) Assistants et compagnons d'utilisation



C'est ici une utilisation plus évoluée du contexte. Le compagnon a un rôle de guide et ne doit intervenir que dans un contexte précis. Le compagnon essaye d'avertir l'utilisateur s'il répète plusieurs fois des erreurs ou s' il effectue des actions répétées qui pourraient être effectuées plus simplement . L'exemple le plus connu est sans doute celui des compagnons de la suite office qui fonctionnent avec plus ou moins de bonheur. Pour déterminer le contexte d'utilisation, l'application compagnon se base sur l'historique d'utilisation de l'utilisateur mais il connaît aussi une base de donnée de contextes toute prête, construite par les développeurs, où il pioche simplement la réponse à afficher à l'utilisateur. C'est donc une interaction contextuelle un peu tronquée. De plus les compagnons ont souvent des comportements non souhaitables ou fournissent des réponses hors contexte. Sans utiliser plus de variable de contexte, il est difficile de simuler un comportement intelligent.

Ces quelques exemples ne représentent pas la totalité des utilisations de contexte réellement utilisées aujourd'hui ; on peut y inclure les systèmes de correction automatique, les systèmes de coloration syntaxique de code, les système d'exploitation avec gestion de comptes utilisateur (la plupart des OS modernes), les logiciels de détection de Spam et beaucoup d'autres. Dans aucun des exemples il n'a été nécessaire d'ajouter un mécanisme physique pour la capture de contexte. Tout est du domaine du logiciel.

2.2) Active Badge [11]

Expérimenté en 1990 au laboratoire de recherche Olivetti en angleterre, ce projet est généralement considéré comme le premier exemple d'application sensible au contexte. Initié par Want, Hopper, Falcao et Gibbons [11], le projet vise à une meilleure localisation et coordination d'un large groupe de personnes dans une entreprise. Dans ce type d'organisation, souvent un réceptionniste est chargé de transmettre les appels téléphoniques vers les postes internes, dans les bureaux des employés. Il dispose habituellement d'un plan avec les codes des téléphones associés aux utilisateurs.

Dans le centre de recherche Olivetti, ce système était très aléatoire, les membres des équipes de recherches étant rarement dans leurs bureaux, les appels transmis se perdaient.

La solution à ce problème à été la mise en place du système "active badge", un badge balise personnalisé porté par chaque employé qui va permettre de le localiser avec précision dans les locaux et ainsi de diriger les appels destinés à ce dernier vers le poste téléphonique le plus proche.

Design du badge

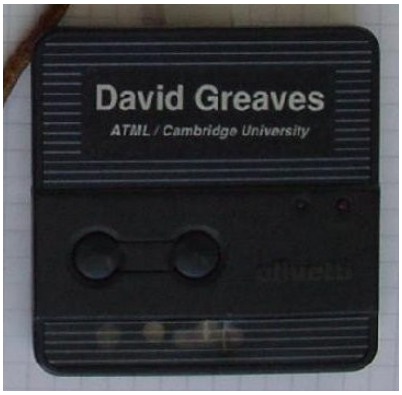


Figure 4 : prototype Active-Badge

Le active badge se présente comme un module électronique de 5 cm de coté et d'environs 40 grammes (*figure 5*). Il est capable d'envoyer un signal périodique d'un trentième de seconde toutes les 15 secondes environ. Ce signal est capté par un réseau de capteurs disséminés dans l'immeuble et relayé ensuite jusqu'à un serveur maître qui se charge de transcrire les données reçues en information de localisation utilisables par les clients. La transmission par infrarouge sert de support pour le signal. On peut utiliser un badge jusqu'à 6 mètres de distance d'un capteur. L'infrarouge ne traverse pas non plus les cloisons, ce qui permet d'éviter qu'un capteur situé dans une autre salle ne génère un signal alors que le porteur du badge n'est pas dans la pièce. Le matériel infrarouge est connu et largement employé dans l'industrie, son prix est aussi attractif.

Avec un signal toutes les 15 secondes, la consommation du badge est très faible et une autonomie d' un an est atteinte. Mais cela signifie aussi que la position de l'utilisateur est connue dans une marge de 15 secondes et que s'il se déplace rapidement, il peut être à un endroit différent de celui reporté par le système.

Le badge intègre aussi un capteur de lumière qui met hors tension le badge lorsqu'il fait noir, c'est à dire que l'utilisateur a fini sa journée de travail.

2.2.a) Implantation du réseau de capteurs

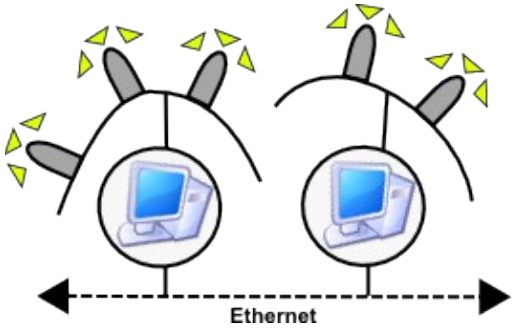


Figure 5 : réseau de capteurs reposant sur les postes clients

La topologie proposée exploite au maximum l'existant. Les capteurs sont connectés à des stations de travail du réseau, ce qui permet de faire transiter les informations sur l'Intranet de l'entreprise(*figure 5*). Le pré-requis à l'installation du réseau est de pouvoir placer des capteurs dans tout le bâtiment, il faut donc que le réseau Intranet de l'entreprise couvre une bonne partie des bâtiments. Idéalement, les capteurs sont placés au plafond, et aux entrées et sorties des couloirs. On doit pouvoir depuis un capteur voir le suivant pour être sûr qu'il n'y a pas de zones non couvertes. Les capteurs sont alimentés par le réseau.

2.2.b) Application test du réseau active badge

L'application de test est prévue pour aider le réceptionniste à localiser les employés et ainsi transmettre les appels téléphoniques au bon endroit.

La liste des employés associée à leur localisation remise à jour dynamiquement est affichée dans la fenêtre du logiciel. Une troisième colonne donne une idée de la probabilité de trouver l'utilisateur à l'endroit donné. En plus du simple affichage de la liste, une petite interface de commande permet de passer des ordres simples pour rechercher un employé, pour lister les personnes présentes dans une pièce ...

Si la plupart des utilisateurs ont trouvé l'application test et le transfert des appels téléphoniques utiles, il y a quand même des cas où le simple contexte de localisation n'est pas suffisant. Il n'est pas souhaitable par exemple qu'un employé reçoive un appel dans le bureau de son directeur ou lors d'une réunion. Le système devrait aussi pouvoir prendre en compte les horaires de repas. Les versions futures de active badge permettront de considérer ces cas.

Malgré les réticences compréhensibles des premiers utilisateurs, le système active badge a su gagner sa place dans les entreprises et les institutions (Xerox Parc, MIT Médialab, Olivetti). Toutefois les auteurs mettent en garde contre les dérives qui pourraient résulter d'une utilisation abusive de ce genre de systèmes de localisation. La question posée par ces travaux finalement n'est pas "peut on construire un réseau de localisation ?" mais bien "a-t-on envie de faire partie d'un tel réseau ?".

2.3) Cyberguide [12]

Les applications qui obtiennent des informations sur les alentours de l'utilisateur sont les plus connues des applications sensibles au contexte. Des premières expériences sur la localisation, comme ParcTab ou ActiveBadge, on va passer à des applications plus complètes qui vont pouvoir renseigner l'utilisateur sur l'endroit où il se trouve et bien plus.

Abowd, Atkeson, Hong, Long, Kooper et Pinkerton de l'institut de technologie de Géorgie proposent au milieu des années 1990 d'explorer les possibilités futures de la prise en compte du contexte dans les interactions entre un utilisateur et un terminal mobile [12]. Cyberguide est un assistant de visite virtuelle conçu à l'origine pour guider les visiteurs lors des portes ouvertes de l'institut de technologie.

2.3.a) Propositions d'utilisation du contexte

Avec l'arrivée des organisateurs, le MessagePad de Apple notamment, l'informatique mobile devient une réalité. Abowd *et al.* proposent leur vision de ce que devrait être un assistant personnel utilisant le contexte, future extension du prototype Cyberguide.

Un assistant personnel doit pouvoir être utilisable aussi simplement qu'un guide papier. Il doit rester de taille raisonnable, c'est à dire que l'on peut le tenir d'une main. On pourra utiliser un stylet ou un écran tactile comme système d'entrée, il pourra stocker et récupérer des informations (utilisation d'un Cédérom, d'une connexion Internet, ressources téléphoniques).

Pour prendre en compte du contexte, on utilisera :

- un système de reconnaissance et de synthèse vocale performant,
- une caméra vidéo pouvant au choix capter les mimiques de l'utilisateur ou des images de l'environnement,
- des capteurs d'orientation de l'appareil pour connaître l'orientation de l'utilisateur,
- Un récepteur GPS pour le positionnement précis de l'utilisateur.

Cet appareil pourra trouver sa place dans toutes les activités de l'utilisateur :

- Lors de visites de villes (utilisation en guide touristique),
- pour servir de guide de navigation,
- pour traduire des informations de l'environnement en langage naturel,
- pour supporter les activités de groupe en communiquant avec les autres assistants des membres du groupe.
- une utilisation plus futuriste pourrait être la reconnaissance faciale d'une personne et la restitution d'une fiche d'information.

Ce sont quelques-unes des possibilités imaginées par Abowd *et al.* pour l'utilisation du contexte. Bien sûr, les possibilités sont infinies et les technologies ne sont pas encore assez au point pour pouvoir bien exploiter les données du contexte tels que les gestes ou la parole, même si se sont des thèmes de recherches très actifs.

Un autre problème d'un tel outil est qu'il n'existe pas encore de base de données assez grande pour pouvoir couvrir l'étendue des thèmes que peut proposer cet assistant idéal, même si l'utilisation des ressources Internet semble être une voie d'étude possible.

2.3.b) Les prototypes Cyberguide

C'est la famille d'outils développés par l'équipe dans l'idée de s'approcher de l'assistant idéal proposé ci- dessus. Ils proposent pour ces appareils quatre fonctions principales :

- Cartographie : utilisation de l'appareil comme un plan de visite. Sur le premier prototype, c'est en fait un plan du laboratoire dont une partie est affichée a l'écran en fonction du positionnement du visiteur (*figure 6*).
- librairie : contient la description des oeuvres, des monuments pour les afficher dans le contexte voulu, ce composant doit aussi permettre de répondre aux questions posées par l'utilisateur (questions contextuelles : "qui a peint cette toile ?"). Contient dans le prototype des informations sur les différents projets et démonstrations du laboratoire. Ces zones

d'information sont aussi marquées sur le composant de cartographie. Les fiches d'informations sont rédigées en utilisant la fonctionnalité "books" (visualiseur d'aide) du newton Apple (*figure 7*).

- Navigation : Composant permettant de localiser précisément l'utilisateur. Position et orientation. On utilise ce composant pour répondre à des questions de type "que suis-je en train de regarder ?". Le GPS a été choisi pour la localisation en extérieur. En intérieur, le signal GPS est trop faible et imprécis, il faut donc passer par un réseau infrarouge classique, comme dans le cas des Actives Badges (*figure 9*).
- Messagerie-communication : le touriste va pouvoir communiquer avec d'autres, l'exemple proposé est de faire passer des messages à un groupe de touristes équipés ("le bus part"). L'utilisateur pourrait aussi contacter l'artiste qui a peint la toile devant laquelle il se trouve. La première implémentation proposée consistait en l'utilisation du réseau appletalk avec un questionnaire contextuel rempli par l'utilisateur et qui est ensuite envoyé aux chercheurs par émail (*figure 8*).

Les versions futures des prototypes vont améliorer les fonctions de base en les enrichissant. CyBarGuide par exemple enrichit la librairie en offrant la possibilité à l'utilisateur de noter les sites visités, de les commenter. Outside CyberGuide va utiliser une cartographie enrichie où l'utilisateur va pouvoir savoir quels sont les sites où il est déjà passé...

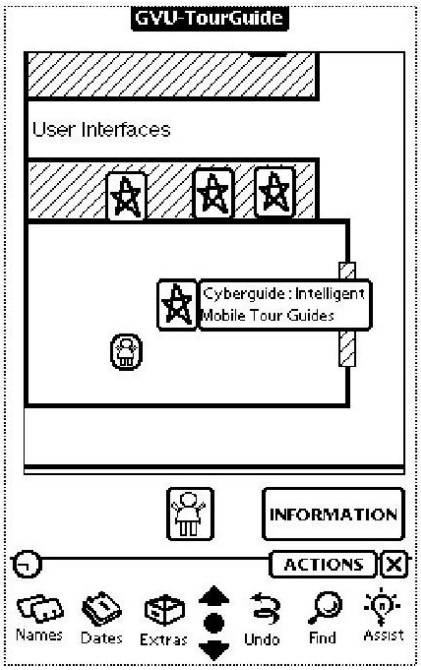


figure 6 : Cartographie

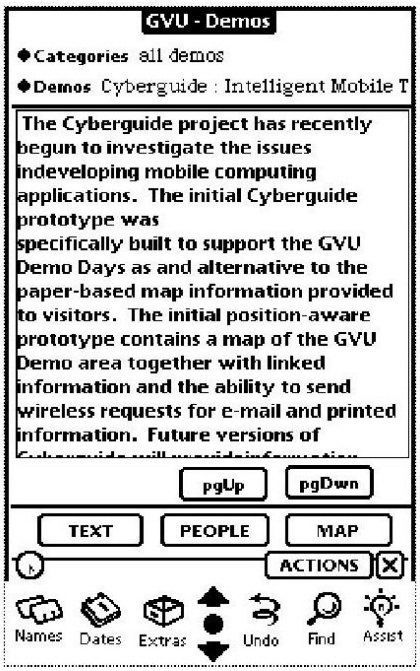


figure 7 : Librairie

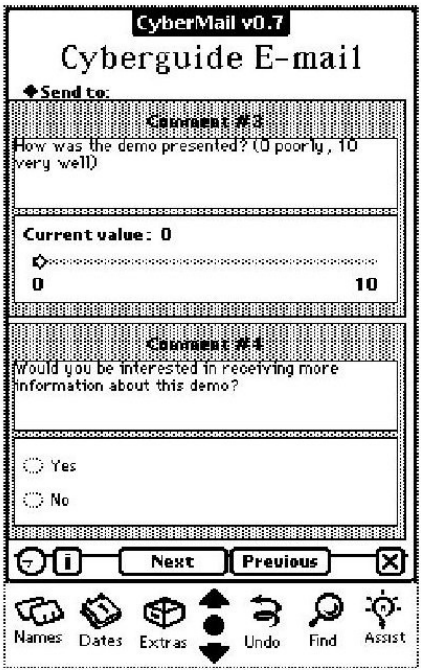


figure 8 : Communication

Figure 9 : Positionnement

L'équipe s'est concentrée ensuite sur l'utilisation de données en provenance du Web, sur l'utilisation de la vision et la communication multimodale avec l'assistant.

2.4) The Conférence assistant [6, 7, 3]

Les systèmes précédents se basaient principalement sur la localisation pour définir le contexte d'utilisation. Pourtant, avec peu de moyens supplémentaires, il est possible d'enrichir grandement le contexte. C'est ce que nous prouve le projet Conférence Assistant de Dey, Salber, Abow de l'institut de technologie de Georgie et Futakawa du laboratoire de recherche Hitachi [6].
Le but de ce projet est de combiner informatique vestimentaire et applications sensibles au contexte dans une sorte d'organisateur mobile portable pour aider les participant à organiser leur parcours dans les conférences de recherches.

L'informatique vestimentaire se prête bien à la capture du contexte en mobilité. Un module d'informatique vestimentaire doit être en fonctionnement quant on le porte (bien sûr), on doit pouvoir l'utiliser les mains libres, la plupart ont une notion du contexte via des capteurs et captent le contexte en permanence pour modifier leur comportement dynamiquement, enfin, ces appareils sont toujours allumés pour assurer la continuité dans l'évaluation du contexte.

2.4.a) Scénario d'utilisation proposé

Les grandes conférences de recherches sont très dynamiques, il y a de nombreuses sessions de conférences en parallèle, sur des sujets très variés impliquant forcément des contextes changeants. Le Conférence assistant est un programme chargé dans l'ordinateur vestimentaire du participant et qui va lui permettre de décider quelles conférences sont intéressantes pour lui, d'être au courant des activités de ses collègues, d'effectuer des interactions avec son environnement lors du déroulement de la conférence.

Les éléments de contexte utilisés sont très nombreux : temps, identité, localisation et activité. Le spectre des quatre classes de contexte de base est couvert et l'informatique vestimentaire se prête particulièrement bien à l'utilisation massive du contexte.

Lors de son arrivée à la conférence, le participant se voit remettre l'assistant sous la forme d'un logiciel qu'il va charger dans son ordinateur vestimentaire. Les informations qu'il a données lors de son inscription (centre d'intérêt, collègues présents, coordonnées) vont être intégrées par le logiciel assistant et serviront de contexte de base.

Lorsqu'il n'assiste pas à une conférence, l'application se présente à l'utilisateur sous la forme d'un planning de la conférence avec une présélection grisée des conférences qui peuvent l'intéresser (*figure 12*). Il peut aussi savoir à quelles conférences assistent ses collègues. Si le participant rentre dans une salle de conférence, l'assistant détecte un changement de contexte et réagit en affichant le nom du conférencier, le titre, le thème et d'autres informations sur la conférence. Lors du déroulement de la conférence, l'utilisateur reçoit via l'assistent les diapositives qui sont en train d'être passées et peut prendre des notes sur celles-ci (*figure 11*). Il lui est possible à la fin de la présentation, lors de questions, de désigner et faire apparaître la diapositive sur laquelle porte sa question sur l'écran du conférencier. Le participant a la possibilité de noter l'intérêt qu'il accorde à une conférence dans son planning, cette information est répercutée chez ses collègues qui peuvent savoir quelle sont les sessions intéressantes.

Il est enfin possible à l'utilisateur, une fois rentré chez lui de retrouver les présentations de la conférence en fonction des sessions auxquelles il a participé, annoté avec ses informations de contexte (Quand est-il arrivé dans la session ? Quelle question a-t-il posée ? Quand est il parti ?)(*figure 13*).

2.4.b) Conception du logiciel

Le logiciel est construit en prenant pour modèle d'architecture le "Context-Toolkit" [7] qui agrège des éléments de contextes dans des serveurs de contextes accédés au plus haut niveau par

l'application (*figure 10*).

A l'inscription du participant, un serveur de contexte est crée pour l'utilisateur, il va contenir toutes les informations contextuelles du participant pendant sa participation à la conférence et après. C'est à dire un certain nombre de widgets :

- Enregistrement : il acquiert les coordonnées, les intérêts et les collègues de la personne
- Memo : système d'acquisition des notes pendant les présentations
- Notation : détermine quelles sont les sessions intéressantes pour le participant, c'est plus qu'un simple widget de contexte puisqu'il interprète les données en provenance du widget Enregistrement.
- localisation : monitore les départs et arrivées du participant

On trouve aussi des serveurs de contextes propres à chacune des sessions, on y trouve toutes les informations sur la présentation de cette section (diapo, vidéo, contexte du conférencier) ici aussi les serveurs des sessions rassemblent un certain nombres de widgets :

- question : acquiert les questions du public
- localisation : monitore les départs et arrivées des utilisateurs aux conférences
- contenu : contient les présentation, détecte les changements dans les présentations
- Enregistrement : détecte si oui ou non une vidéo de la conférence est enregistrée

L'application Conférence Assistant ne communique pas avec les widgets de contexte directement mais passe par les serveurs et l'interpréteur pour récupérer les variables de contextes utilisées pour son exécution.

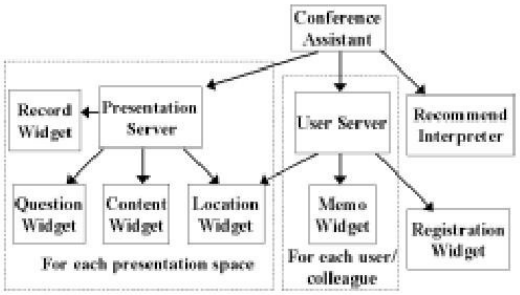


Figure 10 : architecture du logiciel

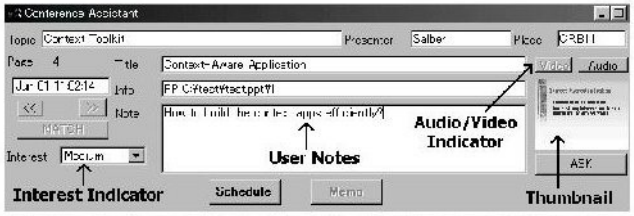


figure 11 : L'interface de prise de notes

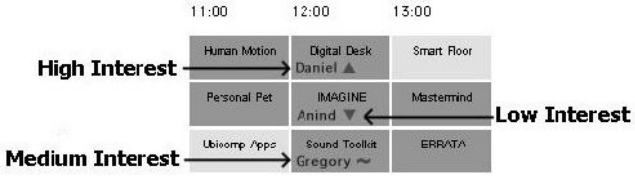


figure 12 : Planning avec les niveaux d'intérêts

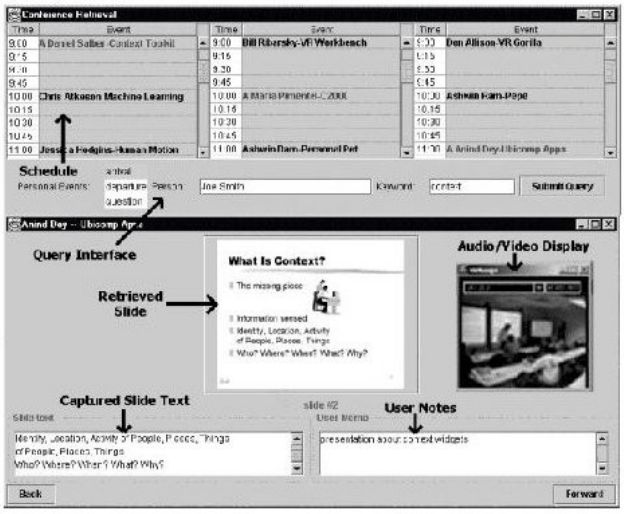


figure 13 : interface de restitution

Le Conférence assistant a été testé en laboratoire à l'état de prototype, sur des machines portables mais pas en informatique vestimentaire bien qu'il n'y ait aucune raison pour que cela ne marche pas. L'équipe prévoit un déploiement et un test en milieu réel du système après avoir enrichi celui-ci en fonctionnalités.

2.5) Téléphone cellulaire sensible au contexte : Sensay [9]

Le téléphone portable est un objet qui aurait beaucoup à gagner d'une meilleure utilisation du contexte. Tous les modèles proposés aujourd'hui laissent à l'utilisateur le soin de régler les volumes de sonneries et de haut- parleur, de savoir quand il peut être appelé ou non, de brancher ou non la sonnerie. Immanquablement, l'appareil se met à sonner quand il ne faut pas, au cinéma par exemple ou encore dans le métro ou le volume sonore empêche l'écoute.
Siewiorek, Smailagic, Furukawa, Moraveji, Reiger et Shaffer proposent avec Sensay [9] d'ajouter du contexte à un téléphone portable.

Sensay est un téléphone portable qui modifie son comportement en prenant en compte l'état de l'utilisateur et de son environnement. Il s'adapte dynamiquement à un contexte en évolution et sait aussi prévenir l'interlocuteur (la personne qui appelle) de l'état de l'utilisateur. Pour déterminer le contexte, Sensay va utiliser des capteurs de lumière, de son, de mouvement qui sont placés à des endroits-clé sur l'utilisateur. Tous ces capteurs sont reliés à la "sensor box", sorte d'unité centrale portable qui va interpréter les données des capteurs et calculer l'état dans lequel doit être mis le téléphone (*figure 14*).

2.5.a) architecture de Sensay

Sensay se décompose en 5 modules (*figure 15*):

- Capteurs : de bruits (voix et milieu), de mouvement, de température et de lumière visible
- Logique des Capteurs : C'est à ce module d'assurer l'interrogation périodique des capteurs. Le prototype interroge les capteurs toutes les secondes. Il envoie ses données au module de décision
- décision (*figure 16*): Ce module est au cœur du système, en fonction des données reçues des capteurs, il va appliquer un diagramme d'état pour savoir dans quel état faire passer le téléphone. Le module de décision stocke aussi les valeurs antérieures des capteurs et effectue les calculs pour les changements d'état sur les 10 dernières mesures des capteurs. On évite ainsi les interférences dues à un contexte qui changerait trop vite.
- Action : est chargé de l'interface entre le module de décision et le téléphone.
- Téléphone : c'est un téléphone cellulaire que l'on peut de préférence reprogrammer pour le mettre en lien avec le module d'action

Dans le prototype proposé les modules de décision, d'action et de capteurs tournent sur un ordinateur portable et seront intégrés à la Sensor Box, voire directement dans le téléphone portable.
Le téléphone peut être mis dans quatre états distinct sur ordre du module de décision ou sur demande expresse de l'utilisateur :

- In-interrompable : Quand l'utilisateur est engagé dans une conversation ou qu'il assiste à un cours, le téléphone entre dans ce mode. Il est calculé à partir des données du calendrier intégré au téléphone portable; on mesure aussi si l'utilisateur parle. Quand le téléphone est dans ce mode, un interlocuteur recevra un message SMS indiquant que l'utilisateur est occupé. Il a toutefois la possibilité, si l'appel à passer est très urgent de rappeler dans les 3 minutes, le mode est alors court-circuité.
- Dans ce mode, on fait aussi appel au capteur de lumière pour savoir s'il faut mettre le vibreur en route ou non : "peu de lumière" => "probablement dans la poche" => "vibreur ON". Pour le passage du mode In-interrompable vers un autre mode, on considère un historique des capteurs plus long pour éviter au maximum les effets de bord d'un faux changement de contexte.

- Actif : Sensay accorde le niveau de sonnerie et des hauts parleur en fonction du volume sonore ambiant, de même, si l'utilisateur est en déplacement, le vibreur va être augmenté au maximum. Ici aussi, avant de passer dans se mode, le module de décision inspecte un historique assez long pour que le simple passage dans une ambiance sonore importante ne provoque le passage dans ce mode.
- Disponible ("idle") : Périodes où l'utilisateur ne parle pas ou peu, où il n'est pas en situation in-interruptible, en général il se trouvera dans son bureau à ce moment-là. Sensay profite de ces moments pour informer l'utilisateur s'il a reçu des appels ou des messages. Bien sûr l'utilisateur est considéré comme interruptible dans ce mode. Ici le système considère un historique court pour qu'il soit facile de sortir de ce mode.
- Normal : Quand aucun des autres modes n'est actif. Ce mode correspond aux valeurs de sonneries et de vibreur par défaut du portable.

La décision de rentrer dans un état se fait selon un ordre de priorité donné aux variables. Dans leur étude Siewiorek *et al.* donnent les priorités suivantes (par ordre d'importance):

- Réunion prévue au calendrier
- L'utilisateur est en conversation
- Activité physique importante
- Volume sonore ambiant important
- faible volume, faible activité physique et peu de dialogue

Ainsi, si un utilisateur court alors qu'une réunion est prévue dans son agenda, le téléphone rentrera dans le mode in-interruptible plutôt que le mode activité.

La prochaine étape pour le Sensay est de proposer une version embarquée et réellement mobile. Pour la plupart des décisions, le système se fonde sur le fait que l'utilisateur parle ou non. Pour une version portable, il faudrait trouver un autre moyen de capturer cette variable de contexte, l'utilisateur ne portant plus les capteurs sur lui.

Figure 14 : Le système Sensay

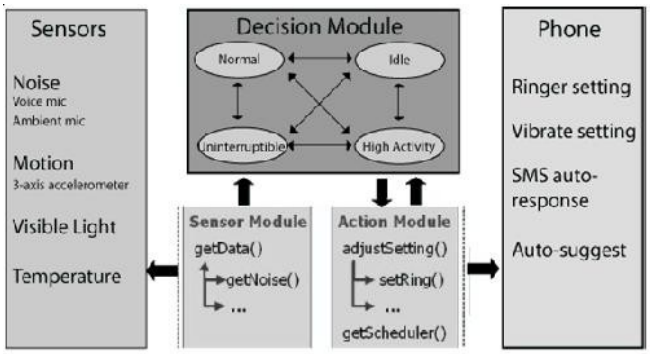


Figure 15 : architecture logicielle de Sensay

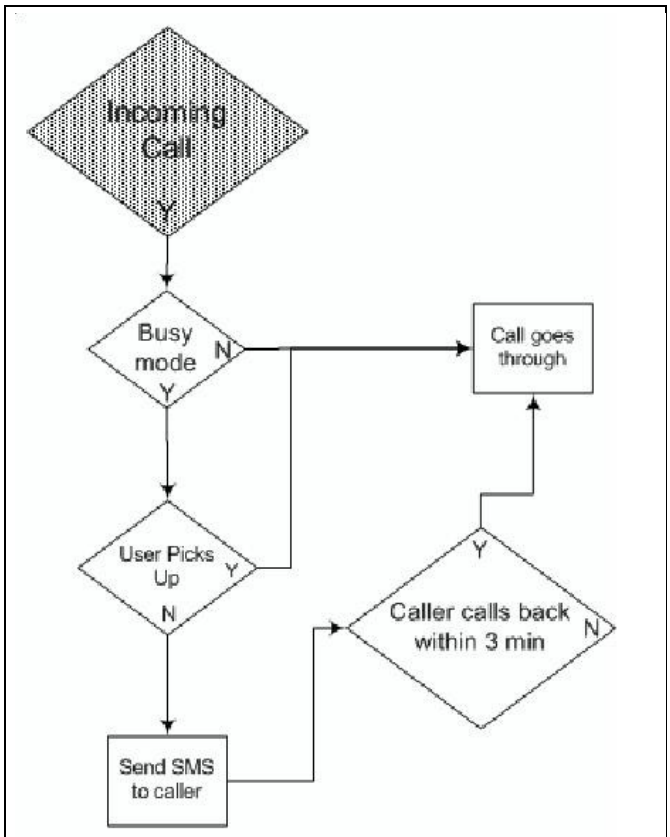


Figure 16 : Graphe de comportement

III) Organisation des données contextuelles

Les applications qui utilisent la notion de contexte sont de plus en plus nombreuses. Devant la multiplication et les avancées logicielles du domaine, des équipes de recherche se sont intéressées à la définition d'une architecture commune pour les applications contextuelles, celles-ci devant permettre l'interopérabilité et l'intégration de contextes de plus en plus riches sans pour autant avoir à reprendre toute l'architecture du logiciel de zéro.

La première tentative d'infrastructure a été présentée par Shilit dans sa thèse de 1995 [21web]. Il y décrit une architecture générale pour la prise en compte du contexte en se concentrant toutefois sur la localisation et en prenant modèle sur les Active Badges de Olliveti.

Depuis, beaucoup d'autres modèles ont vu le jour, nous en détaillerons quelques-uns parmi les plus connus dans cette section.

3.1) Le Context-Toolkit [3, 7, 15]

Conçu à l'institut de Technologie de Georgie et proposé en 1999, le Context-Toolkit est une tentative de boite à outils pour la création d'applications sensibles au contexte.

Partant de l'idée que pour généraliser et répandre l'utilisation du contexte dans les applications, il faut fournir aux développeurs un système d'abstraction par rapport au niveau physique (logique des capteurs) Dey, Salber, Abowd et Futakawa vont proposer l'utilisation d'une couche logicielle entre les capteurs et l'applications basée sur la notion de widgets de contexte.

3.1.a) Recherche du contexte[3]

L'un des problèmes majeurs de l'utilisation du contexte est que jusqu'alors, il n'existait pas d'uniformité dans la construction de ces applications, chacune d'entre elle était construite autour de sa propre architecture et on ne pouvait pas du tout envisager une quelconque interopérabilité entre deux applications contextuelles, voire l'utilisation des capteurs de contextes par une autre application que celle pour lesquels ils ont été mis en place.

Jusqu'ici, les premières expériences d'utilisation du contexte étaient basées sur une approche intégration du contexte ou sur une approche de type "serveur de contexte"

- Approche intégrative : Les drivers et la logique permettant d'acquérir les données du contexte sont directement codés au sein de l'application. Les développeurs doivent jongler avec le haut niveau de l'application, de l'interface et les protocoles bas niveau de communication et de données des capteurs. Cette approche rend la programmation très difficile et ne permet pas d'appliquer de bonnes méthodes de design d'application, puisqu'il faut mélanger acquisition et application. Enfin, il est impossible de réutiliser le contexte directement avec une autre application.
- Approche serveur : L'application va ici s'adresser à un serveur qui va effectuer l'abstraction nécessaire entre l'application et les capteurs. Le serveur gère toute la communication et la synthèse du contexte. Les applications n'ont plus qu'à demander les informations contextuelles au serveur pour les exploiter directement. Bien qu'apparemment, cette technique résolve les deux précédents problèmes, Dey *et al.* expliquent que l'utilisation d'un serveur par élément de contexte ne simplifie qu'en apparence l'acquisition des données

puisqu'en réalité, une application va vouloir communiquer avec autant de serveurs qu'elle nécessite d'éléments de contexte, sachant que chaque serveur a un mode de communication et transmet des données qui lui sont propres ; un peu comme si l'on exploitait directement les capteurs.

Dey *et al.* proposent une nouvelle approche en comparant l'acquisition du contexte à l'acquisition d'entrée de l'utilisateur. Il existe depuis longtemps des abstractions pour les entrées de l'utilisateur : depuis l'arrivée des interfaces graphiques et la programmation du même nom, on utilise des boîtes à outils d'éléments d'interface ("widgets") pour saisir les entrées de l'utilisateur, plus personne ne dessine boutons et menus à la main.

3.1.b) Un Context-Widget [15]

Un Context-Widget (non traduit pour éviter la confusion avec "élément de contexte") est un élément logiciel qui permet à une application d'accéder à un élément du contexte de la même manière qu'un élément d'interface lui permettrait d'effectuer une présentation à l'écran. Un context widget cache la complexité des capteurs à l'application : que l'on capte la position d'un utilisateur avec un GPS, un réseau infrarouge ou des senseurs de pression ne doit pas affecter l'application qui n'a à faire qu'à des informations de position.

Les widgets sont une première abstraction du contexte : si une personne se déplace dans un immeuble, on est par exemple intéressé par les sorties et les entrées de celle-ci dans une pièce. Le context widget associé aux déplacements va suivre la personne et donner l'alerte à l'application uniquement au moment opportun, c'est à dire lors des franchissements de porte.

Les widgets de contexte sont réutilisables et peuvent être combinés : les informations de localisation vont pouvoir être utilisées indifféremment par toute une gamme d'applications. Si l'on veut définir des éléments de contexte plus complexes, il est possible de créer des widget qui sont des combinaisons booléennes de widgets de plus bas niveau. Par exemple le widget contextuel "Présence par temps clair" va sûrement utiliser un widget "présence" (lui même défini par un "localisation" et un autre "identité") et un widget "quel temps fait-il ?".

L'encapsulation, le mécanisme d'héritage et la généricité sont trois points forts du modèles des boîtes à outils graphiques utilisant des éléments d'interface comme brique de construction. Le modèle de widget contextuel va lui aussi tirer parti des mêmes avantages. Deux différences majeures entre un élément d'interface et un Context-Widget apparaissent :

- Un Context-Widget est basée sur une architecture distribuée, puisqu'il doit pouvoir combiner ses sources d'informations en provenance d'origines diverses, en fait trois comme nous le verrons dans le détail de l'architecture.
- Un Context-Widget doit être actif tout le temps pour pouvoir surveiller et prévenir l'application à tout moment d'un changement dans le contexte.

3.1.c) Architecture pour l'utilisation des Context-Widgets [7]

Dey *et al.* commencent par définir les besoins d'une application contextuelle :

- Les applications doivent pouvoir accéder au contexte distribué et sur un réseau comme s'il s'agissait d'une machine locale. Le contexte peut venir d'un vaste réseau de machines, même si l'application ne tourne que sur l'appareil de l'utilisateur.
- L'architecture doit être portable sur plusieurs plates-formes et être programmée en différents langages.
- Supporter l'interprétation du contexte, c'est à dire qu'un simple widget de contexte ne peut

pas être utilisé tout seul, il faut lui adjoindre des éléments pouvant interpréter son information.

- Support de l'agrégation des informations contextuelles.
- Permettre l'indépendance et la persistance des Context-Widget
- Stocker un historique des états des Context-Widgets (CAD : un historique des contextes)

A partir de ces recommandations, ils proposent un modèle orienté objet consistant en trois types d'objets : les serveurs, les interpréteurs et les Context-Widgets (*figure 17*).

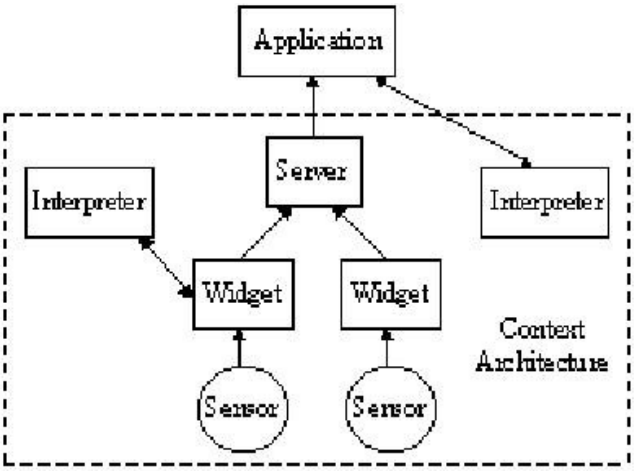


Figure 17 : Architexture du modèle Context-Toolkit

Chacun des objets est indépendant des autres, c'est à dire qu'il possède son propre thread d'exécution et peut être utilisé localement ou accédé via un réseau.

Un context widget se définit par un certains nombre d'attributs qui sont les variables liées au contexte en provenance des capteurs et par des déclencheurs d'événements liés à ces attributs et qui peuvent être nécessaires aux composants utilisant le widget. Tous les autres composants peuvent avoir accès aux déclencheurs et aux attributs mais aussi à l'historique des états du widget.

Les serveurs ont pour rôle de combiner les sources de contexte en provenance des widgets. On les utilise pour collecter l'ensemble des éléments de contexte d'un domaine donné (récupérer le contexte d'une personne : identité, localisation, activité, …). Un serveur simplifie la tâche du programmeur puisqu'il lui permet d'exploiter un seul serveur plutôt que d'adresser une multitude de widgets. Les serveurs héritent des déclencheurs et des attributs des widgets auxquels ils sont liés.

Les interpréteurs sont chargés de déterminer automatiquement un contexte à partir de widgets. Généralement le soin d'interpréter le contexte était laissé à l'application, mais on ne pouvait alors réutiliser une interprétation avec une autre application. Un interpréteur peut prendre des informations en provenance des widgets "présence de son", "présence d'image" et "ambiance lumineuse sombre" pour interpréter le contexte comme "au cinéma, en train de regarder un film" (Voir le projets Sensay).

3.1.d) Communication

La communication entre les objets s'effectue via TCP/IP, chaque objet est identifié par une URL et on utilise HTTP pour accéder aux attributs. XML est utilisé comme langage de définition des attributs. Ces technologies simples se prêtent bien à de l'embarqué, contrairement à des technologies d'objets répartis comme Corba ou Java RMI.

La Context-Toolkit est utilisée dans les projets d'informatique contextuelle de l'institut de technologie de Georgie, mais a aussi été adoptée par des entreprises comme British Telecom pour organiser les applications contextuelles. Les futures versions de la Context-Toolkit incluent des modules supplémentaires dans l'architecture, comme un service pour lister quels sont les widgets de contextes disponibles dans un environement donné (le "discoverer") ou encore des "services" permettant d'effectuer une action sur le contexte par ordre de l'application [15]

3.2) Le Contexteur [2]

Le modèle des contexteurs est proposé par Rey et Coutaz du laboratoire CLIPS-IMAG de Grenoble [2]. Leur but est de proposer une infrastructure basée sur des briques logicielles : les contexteurs pour répondre aux problèmes du modèle de la Context-Toolkit [15] et notamment le fait qu'il n'existe pas de notion de qualité dans la Context-Toolkit.

3.2.a)Description d'un contexteur

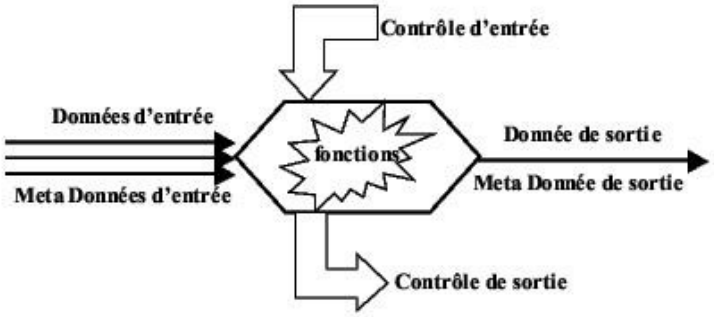


Figure 17b : Un bloc contexteur

Un contexteur comprend trois classes d'éléments (*figure 17b*):

- Les entrées : Les données en entrées sont les informations que le contexteur a la charge de traiter. Elle proviennent d'autres contexteurs. On leur associe systématiquement un flux de méta-données en entrée qui indique quel est le degré de confiance que l'on peut avoir sur cette entrée. En effet, dès que l'on se base sur une information en provenance de systèmes d'automatique, on doit pouvoir juger de la qualité de l'information reçue. Les pannes de capteurs ne sont pas rares (75% des pannes en automatisme) ; il peut aussi y avoir des problèmes de transmission, des réceptions incomplètes, des signaux redondants … Le contrôle d'entrée permet d'effectuer des opérations directes sur le corps fonctionnel du contexteur. Il est lié à d'autres contexteurs qui peuvent modifier le comportement et demander via le contrôle d'entrée à un contexteur reconnu comme déficient de s'arrêter.
- Les sorties : les données de sorties sont à destination soit d'autres contexteurs soit d'applications, ici aussi, le contexteur y associe une sortie de méta-données caractérisant la qualité de sa sortie. Le contrôle de sortie permet de modifier le comportement d'un contexteur cible. L'ordre reçu par le contexteur cible ne s'applique qu'entre l'émetteur et le récepteur. Si C1 émet en contrôle de sortie une demande d'arrêt vers C2, C2 va couper la communication avec C1 mais continuer à communiquer avec les autres contexteurs auxquels il est lié.
- Le corps fonctionnel : c'est la fonction associée à un contexteurs. Rey et Coutaz en proposent plusieurs :
  - Contexteur élémentaire qui intègre un capteur et ne possède pas de données d'entrée
  - Contexteur à seuil qui se déclenche au franchissement d'un niveau de contexte
  - Contexteur de traduction pour changer la représentation des données d'entrée
  - Contexteur à mémoire pour mémoriser un historique des méta-données et données en entrée
  - Contexteur de fusion : fusionne les entrées provenant de plusieurs contexteurs de même type pour en améliorer la qualité
  - Contexteur d'abstraction qui jouent le même rôle que la classe "interpréteur" de la context-toolkit, c'est à dire fournir une abstraction par rapport aux données des

capteurs en livrant une interprétation du contexte.

3.2.b) Organisation des contexteurs

Un contexteur utilisé en brique simple connaît et peut fournir des informations sur lui même : nom, classe, fonction. Il a aussi la capacité de modifier son comportement ou de se mettre en veille si un autre contexteur le lui demande.

Tous ces types de contexteurs peuvent être assemblés dynamiquement soit dans une organisation hiérarchique, soit dans une encapsulation

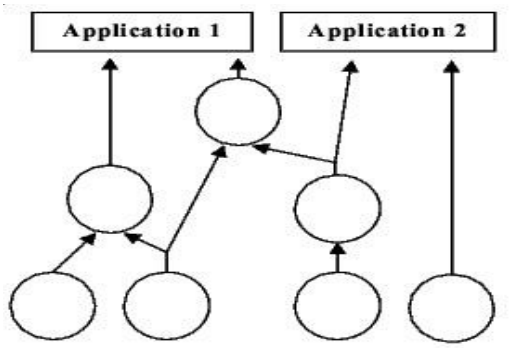


Figure 18 : Architecture hiérarchique

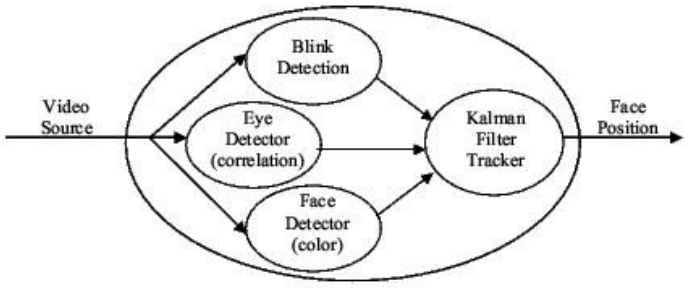


Figure 19 : modèle à encapsulation

- Assemblage hiérarchique (*figure 18*): les contexteurs sont organisés par niveau, chaque sortie d'un contexteur constituant l'entrée d'un ou de plusieurs contexteur de plus haut niveau. Au sommet du graphe, on trouve les applications. Cette organisation risque d'être sujette à des pannes ou dysfonctionnement si un contexteur de niveau supérieur n'offre plus la qualité de service nécessaire. De même, plus un contexteur est à un niveau hiérarchique élève (ie : proche de l'application) plus sa re-configuration est difficile, car elle entraîne en cascade la re-configuration des contexteurs de niveau inférieur.
- Encapsulation (*figure 19*): des groupements sont faits entre plusieurs contexteurs pour définir une variable de contexte. Ces groupements peuvent utiliser des contexteurs qui mesurent la même variable et en font une addition pour renforcer la qualité de cette variable ou alors des contexteurs qui mesurent des variables différentes pour en faire une composition d'une nouvelle variable de contexte. L'exemple de la figure [] propose d'utiliser trois informations de localisation du visage puis de filtrer ces informations pour en déduire sans doute la position du visage. Pour les autres contexteurs, ce regroupement agit comme un contexteur virtuel, c'est à dire que l'on peut récupérer sa sortie et modifier son comportement comme s'il s'agissait d'un contexteur simple.

Les contexteurs en tant que composants autonomes à la fois client et serveurs auprès d'autres contexteurs et applications s'articulent autour d'un protocole de communication réseau de type peer to peer développé spécialement par l'équipe.

La connexion des contexteurs se déroule en deux temps, une première phase de prospection permet à un contexteur de trouver sur le réseau les ressources (contexteurs simples ou assemblés) dont il a besoin pour assurer son fonctionnement et une phase de connexion où les contexteurs mettent en place les relations réseau entre eux. La communication dans un réseau de contexteurs est basée sur http et XML, tout comme la context-toolkit.

Avec la notion de met-adonnées, Rey et Goutaz espèrent proposer une architecture de prise en

charge du contexte robuste et capable de s'autogérer voire de s'auto-réparer. Apres les tests en laboratoire sur leur application "observatoire d'activité", avec une architecture de 5000 contexteurs ; les résultats convaincants permettent de prévoir un passage à plus grande échelle du modèle.

3.3) Stick-e documents [5]

L'idée des Stick-e Documents proposée par Brown de l'université de Canterbury est de donner un accès le plus simple possible à des variables de contexte. Les deux framework étudiés précédemment (Context-toolkit et contexteur) reposent sur des architectures orientées objets avec des notions avancées comme les composants répartis. Brown propose une architecture beaucoup plus simple devant permettre à n'importe quel développeur d'intégrer des éléments contextuels à son application.

Les stick-e documents peuvent être comparés à des notes de type "post-it" sauf que celles-ci vont se déclencher et "s'afficher" dans un contexte voulu.

3.3.a) Descriptions des stick-e notes

Chacune des notes contient un contenu, qui est ce qui va être affiché par la note et une partie de définition du contexte correspondant à l'environnement dans lequel doit être affichée la note.
La stick-e note va pouvoir réagir à des contraintes contextuelles comme la localisation, la présence d'objets, des états à seuil (température), des états internes (comportement de l'ordinateur), la présence de personnes, le temps et en règle générale à tout élément contextuel tel qu'explicité dans les propositions de définition du contexte. Bien sûr, chacun de ces éléments peut être combiné pour former des contextes enrichis ("Je me trouve à cet endroit, à cette heure ci avec telle personne").

Les documents sont soit personnels, c'est à dire disponibles uniquement sur l'appareil de l'utilisateur (PDA ou autre) ou alors téléchargés, partagés via le réseau Internet entre plusieurs appareils.

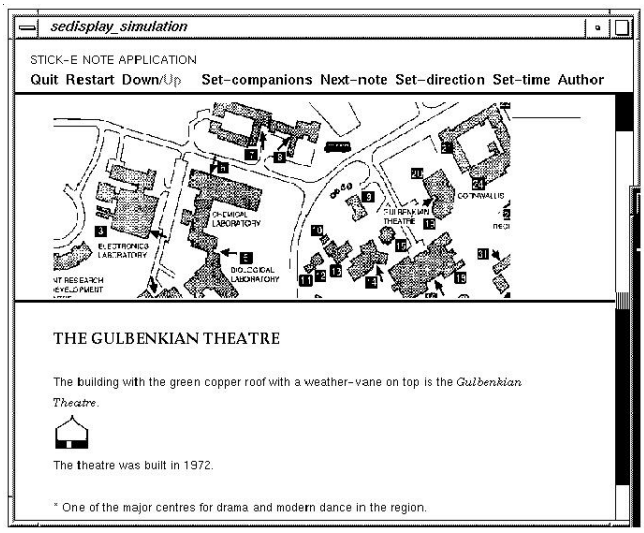
La note est écrite en SGML avec des balises qui expriment les contraintes contextuelles et le contenu de la note :

- <required> est la balise délimitant les informations de contexte, on y trouve des sous-éléments comme <with> ou <at> où sont exprimées les variables de contextes nécessaires au déclenchement de la note.
- <content> contient le contenu à de la note. Celui-ci peut être un simple texte à afficher, du contenu HTML qui sera affiché par un visualiser Web, ou encore lien vers un programme à exécuter en cas de rencontre du contexte,

La construction d'une note est donc aussi simple que de décrire une page HTML pour le Web. Typiquement, les utilisateurs vont avoir sur leur appareil une série de stick-e notes qui vont être activées ou désactivées suivant l'entrée ou la sortie dans des zones de contextes.

3.3.b) Logiciels

C'est aux application contextuelles d'utiliser directement les stick-e en exploitant les fichiers SGML pour créer les liens entre application et élément contexte. Les stick-e notes ne sont qu'une sorte de standardisation dans la façon d'expliquer et d'exploiter et le contexte. Brown définit quand même des recommandations pour l'utilisation complète des stick-e notes. Une application doit permettre aux auteurs de rédiger des notes, de gérer l'affichage du contenu des notes, elle doit contenir un mécanisme de déclenchement pour réagir aux changements de contexte, elle doit aussi contenir un mécanisme de



déclenchement d'évènements (transmissions de signaux, exécutions d'autres applications).
Des applications plus évoluées doivent permettent à l'utilisateur de modifier via une interface appropriée le contenu ou les éléments de contexte d'une note.

Les stick-e notes ont été utilisées dans les projets du laboratoire pour l'intégration du contexte. Ce sont surtout des applications utilisant la localisation comme élément du contexte. Le projet "tour" est un guide virtuel sur PDA qui contient des e-notes se déclenchant lors du passage des monuments décrits dans les notes. L'interface comprend une partie d'affichage du plan de visite toujours présente et une partie qui va contenir les notes des stick-e. ou du contenu HTML. Dans cette application, les notes sont pré-enregistrées au moment de l'écriture du guide. Certains éléments contextuels sont enrichis et prennent en compte le temps de passage devant le monument.

D'autres projets comme "Mobile workers" ou "Personnal pagers" utilisent des contextes plus compliqués et permettent aussi à l'utilisateur de définir ou modifier des stick-e notes.

3.4) Utilisation étendue du contexte [1]

L'élément de contexte majoritairement utilisé est la localisation. Depuis les premières expériences avec Active-Badge jusqu'au récent Sensay, cet élément contextuel est celui qui de l'avis le plus répandu apporte l'information la plus directement utilisable en terme de prise en compte du contexte. On a aucun mal à imaginer des utilisations possibles de la localisation, notamment en informatique mobile.

Selon Schmidt, Beigl et Gellersen de l'université de Karlsruhe, il est préjudiciable de vouloir utiliser majoritairement la localisation [1]. Selon eux, le contexte est défini par un ensemble de variables assemblées et non pas par une simple localisation. Leur étude porte sur les utilisations possibles du contexte en informatique mobile et quels capteurs peuvent être utilisées en complément de la localisation.

Utilisée très fréquemment, la variable contextuelle de localisation est pourtant dure à mesurer. On utilise le réseau GPS, qui est la solution la plus simple mais ne permet pas des mesures précises de déplacements, Ceci exclut par exemple son utilisation à l'intérieur. Si l'on cherche à mesurer des déplacement dans un bâtiment ou des petites variations dans la localisation, il faut passer par la mise en place d'un réseau de capteurs, c'est à dire beaucoup de contraintes, physiques comme logicielles.

3.4.a)Modèle proposé du contexte

Schmidt *et al.* proposent un modèle d'organisation hiérarchique pour le contexte où les variables sont classées entre deux grandes catégories :

- Les facteurs humains, avec les informations sur l'utilisateur, son activité et son environnement social
- les conditions environnementales, avec la localisation, l'infrastructure (objets disponibles, ressources informatiques...) et les conditions physiques.

Cette classification des variables pouvant décrire le contexte sert au développeur à identifier rapidement quel est le jeu minimum de variables lui permettant d'intégrer le contexte. L'utilisation du contexte reste selon Schmidt *et al.* utilisé dans le domaine des périphériques ultra-portables qui sont les seuls à permettre des situations où le contexte est changeant. Les PCs de bureau sont la plupart du temps utilisés dans les mêmes conditions, par les mêmes personnes. Il en va de même pour les ordinateurs portables qui offrent certes la mobilité mais qui dans les faits sont utilisés toujours dans des contextes très semblables. Les téléphones cellulaires et assistants personnels sont donc les meilleurs candidats pour une utilisation étendue du contexte.

3.4.b) Technologie des capteurs

Les capteurs sont bien connus et utilisés dans le monde de la robotique ou dans la vision par ordinateur. Avec les derniers progrès en automatisme, il est possible d'intégrer des capteurs aux appareils ultra-mobiles sans pour autant en altérer les qualités : poids, taille, consommation, bruit, prix.

- Optique : diodes photosensibles, capteurs de couleurs, vision UV et infrarouge apportent des informations sur l'environnement visible autour de l'appareil. L'utilisation des capteurs infrarouges peut aussi servir à l'envoi et à la réception de données.
- Son : le microphone, très utilisé permet de capter une large gamme de l'environnement sonore. On l'associe toujours à des filtres qui permettent de faire le tri entre les données utiles et le bruit. On peut imaginer utiliser des capteurs ultra-sonique ou infra-sonique pour enrichir l'environnement de l'utilisateur par des sons non perçus mais faisant partie du contexte.
- mouvement : pour capter l'inclinaison, le déplacement, d'accélération. cela permet de déterminer quelles sont les conditions d'utilisation de l'appareil (vibrations, déplacement rapide ou plutôt utilisation statique)
- Position : Localisation de l'appareil dans l'espace, proximité d'autres utilisateurs .. les systèmes GPS et réseau de capteurs sont les plus utilisés. le périphérique mobile ne porte plus les capteurs mais va agir comme une balise en envoyant des signaux réguliers aux capteurs pour indiquer sa position.
- Biométrie : la mesure du pouls, de la tension, du souffle permet de construire le contexte physique de l'utilisateur. Beaucoup de logiciels d'entraînement sportif utilisent ces capteurs pour calculer et afficher les courbes d'effort d'un coureur par exemple.

Bien d'autres capteurs peuvent être utiles au contexte, on pense par exemple à l'humidité ambiante, aux gaz dans l'air, aux capteurs de pression, à l'altitude ... Suivant l'application conçue, il n'y a pas de limite à une utilisation très étendue du contexte.

Schmidt *et al.* ont conçu un prototype utilisant des capteurs angulaires au mercure et des capteurs de lumière ambiante pour un ultra-portable qui se met en veille quand la lumière s'éteint et qui affiche les fenêtres à l'écran toujours dans le bon sens, quelle que soit l'orientation de l'appareil (sens portrait ou sens paysage)(*figure 20*).

Figure 20 : exemple d'utilisation du sens de l'appareil pour le positionnement des fenêtres

L'utilisation des contextes enrichis proposée par Schmidt *et al.* est loin d'avoir été pleinement étudiée. La première étape serait de trouver une méthode efficace pour combiner les multiples sources d'informations. Mélanger des données de capteurs pour en faire des informations contextuelles ayant un sens autre que les valeurs brutes des capteurs est une opération compliquée dès lors que l'on utilise beaucoup de sources différentes.

Conclusion

Après avoir donné des éléments de définitions au contexte et à l'informatique sensible au contexte, j'ai présenté certaines des expérimentations et prototypes utilisant le contexte pour modifier leur comportement. enfin, quatre modèles logiciels pour permettre l'utilisation du contexte dans des projets à plus grande échelle ont été décrits.
On arrive aujourd'hui à un seuil d'acceptation des produits utilisant le contexte et déjà des entreprises comme Fujitsu se lancent dans la commercialisation d'ordinateurs portables enrichis de capteurs pour pouvoir tirer tout le bénéfice de l'utilisation du contexte. Les produits du marché restent toutefois encore assez simple et n'utilisent souvent que la localisation ou l'orientation pour modifier leur comportement.
Le défi pour la recherche est maintenant de trouver des méthode pour permettre à l'ordinateur de combiner les sources de contexte et savoir par lui même adapter son comportement face à des situations inattendues ou non prévues par le développeur.

Références Bibliographiques

1. Schmidt, A., Beigl M., Gellersen H. W. (1998). There is more to context than location. In Proceedings of the International Workshop on Mobile Computing (IMC'98). Rostock, Allemagne, novembre 1998.
2. Rey, G., Coutaz, J. (2003). Le Contexteur : une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte. In Proceedings of Interaction Homme Machine (IHM2002), p. 105-112.
3. Salber, D., Dey, K. A., Abowd, D. G. (1999). The Context Toolkit : aiding the development of context-enabled applications. In Proceedings of Computer Human Interactions (CHI'99), p. 434-441.
4. Dey, K. A., Abowd, D. G. (2000). Towards a better understanding of context and context-awarness. In Computer Human Interactions (CHI2000) Workshop on the What, Who, Where and How of Context-Awarness.
5. Brown P. J. (1995). The Stick-e document : a framework for creating context-aware application. In Proceedings of Electronic Publishing (EP'96), v. 8, p 259-272. septembre 1995.
6. Dey, K. A., Salber, D., Abowd, D. G., Futakawa M. (1999). The Conference Assistant : combining Context-awareness with wearable computing. In Proceedings of the 3rd International symposium on wearable computing. San Fransisco, Etats-Unis, Octobre 1999.
7. Dey, K. A., Salber, D., Abowd, D. G., Futakawa M. (1999). An architecture to support context-aware applications. In Symposium on User Interface Software and Technology (UIST'99) .
8. Schilit, N. B., Adams, N., Want, R. (1994). Context aware computing applications. In Workshop on Mobile Computing Systems and Applications. Sta Cruz, Etats-Unis, décembre 1994. p 85-90.
9. Siewiorek, D., Smailagid, A., Furukawa, J., Moraveji, N., Rieger, K., Shaffer, J. (2003). Sensay : A context-aware mobile phone. In proceedings of Symposium on wearable computers.
10. Brown, J. P., Bovey, D. J., Chen, X. (1997). Context-aware Applications : from the Laboratory to the marketplace. In Institute of Electrical and Electronics Engineers (IEEE) Personnal Communications. v. 4, p. 58-64.
11. Want, R., Hopper, A., Falcao, V., Gibbons, J. (1992). The active Badge Location System. In ACM Transaction on Information Systems. Octobre 1992. v. 10, p. 91-102.
12. Abowd, D. G., Atkeson, G. D., Hong, J., Long, S., Kooper, R., Pinkerton, M. (1997). Cyberguide : A mobile context-aware tour guide. In ACM Wireless Networks.
13. Dey K. A. (2001). Understanding and using Context. In Personnal and ubiquitous computing. Février 2001. v. 5, p. 4-7.
14. Lieberman, H., Selcker, T. (2000). Out of context : Computer systems that adapt to, and learn from, context. In IBM System Journal. Juillet 2000.
15. Dey, K. A., Salber, D., Abowd, D. G. (2001). A Conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. In Human Computer Interaction (HCI) Journal, special issue on context-aware computing. v. 16.
16. Schilit, B., Theimer, M. (1994). Disseminating active map informations to mobile hosts. In Institute of Electrical and Electronics Engineers (IEEE) Networks. v. 5, p. 22-32.
17 Ryan, N., Pascoe, J., Morse, D. (1997). Enhanced reality fieldwork : the context-aware archaeological assistant. In Computer applications in archaeology.
18 Salber, D., Dey, K. A., Abowd, D. G., (1998). Ubiquitous Computing : Defining an HCI

research agenda for an emerging interaction paradigm. In Georgia Tech GVU technical report. Janvier 1998.
19 Brown, P. J. (1998). Triggering information by context. In personnal Technologies. v. 2, p. 1-9.