



# Introduction à l'Adaptation Dynamique Sensible au Contexte

---

## Équipe Rainbow

Stéphane Lavirotte ([Stephane.Lavirotte@unice.fr](mailto:Stephane.Lavirotte@unice.fr))

Jean-Yves Tigli ([tigli@essi.fr](mailto:tigli@essi.fr))

Diane Lingrand ([lingrand@essi.fr](mailto:lingrand@essi.fr))

I3S - UMR 6070

(CNRS - Université de Nice - Sophia Antipolis)



Laboratoire d'Informatique de Signaux et Systèmes de Sophia Antipolis - UNSA-CNRS  
930, Route des Colles - Bât Ecole Polytech'Nice Sophia, Dept. SI  
BP. 145 - 06903 Sophia Antipolis Cedex - France

---

# Le Contexte du Contexte...



# Historique: Informatique Diffuse

---

- Présentation de l'« *Ubiquitous Computing* » par Mark Weiser (Mark Weiser, « *The computer of the 21th century* », Scientific American, 1991)
  - Principe: Anywhere Anytime
- Autre terme très utilisé: « *Pervasive Computing* »
  - Informatique caché à l'utilisateur
- La Commission européenne utilise dans ces textes le terme « *Ambient Intelligence* »



# Historique: Sensibles au Contexte

---

- Premiers systèmes « Context-aware »
  - Fin des années 90
    - Plusieurs systèmes d'aide à la visite basés sur la localisation
  - Introduction en 1992
    - Want, Hopper et al., « *Active Badge Location System* », ACM Transactions on Information Systems, 1992
- « Context-aware »: sensible au contexte
  - Première apparition du texte en 1994: Schilit and Theimer, IEEE Network, 1994
  - Adaptation de l'application au contexte sans interaction explicite de l'utilisateur



# Pourquoi le Contexte ?

---

- Dans le communication humaine
  - Information situationnelle souvent implicite
  - Augmente la « bande passante » conversationnelle
- But
  - Augmenter l'efficacité (en terme d'interaction) et faciliter les interactions en prenant en compte le contexte environnementale
  - Technologie pour aider les autres applications
- Le contexte doit être largement pris en compte. La plupart des utilisateurs ne vont pas savoir quels informations sont pertinents.



---

# Qu'est ce que le Contexte ?



# Qu'est ce que le Contexte

---

## ○ Étymologie

- Contexte = « co-text » : ce qui est de part et d'autre d'une expression

## ○ Actuellement

- Acceptation plus large du terme
- Contexte: un ensemble de circonstances qui encadre un événement ou un objet
- Notion pluridisciplinaire
  - Corpus de plus de 150 définitions (Bazire and Brézillon, « *Understanding Context Before Using it* », LNAI 3554, Modeling and Using Context, 2005)
  - Psychologie, ..., Informatique
  - Difficile de trouver une définition satisfaisant toutes les disciplines



# Définitions du Contexte Appliquées à l'Informatique

---

- Définitions par énumération
  - Souvent utilisé dans les premières applications
  - Type de définition trop limitant
  - Localisation, environnement, identité, date et heure, ...
- Définitions par description
  - « Aspects de la situation actuelle »: trop généraliste
  - « Les éléments de l'environnement dont l'ordinateur à connaissance » [Brown 1996]



# Le Contexte selon Dey et Abowd

---

- « Le contexte est défini par toute information qui peut être utilisée pour caractériser une entité. Une entité peut être une personne, un lieu, un objet, pouvant être considérée comme appropriée dans l'interaction homme / application, incluant l'utilisateur et l'application eux-mêmes. »
- « Un système est dit sensible au contexte s'il utilise le contexte pour fournir les informations et/ou les services pertinents à l'utilisateur, la pertinence dépendant des tâches de l'utilisateur. »
- Les auteurs parlent de
  - Contexte primaire (localisation, identité, heure, ...)
  - Contexte secondaire (adresse email, ...)



# Le Contexte selon Chen et Kotz

---

- « Le contexte est un ensemble d'états et de paramètres qui soit détermine le comportement d'une application ou bien dans lequel un événement d'application se produit et est intéressant pour l'utilisateur. »
- Distinction de deux catégories d'informations
  - Critiques: le contexte actif (caractéristiques déterminant le comportement)
  - Appropriées: le contexte passif (approprié mais non critique)



# Le Contexte en Action

---

- Déduction du contexte => Utilisation du contexte (i.e. pour adaptation)
- Information
  - = service
  - = contexte
- Contexte
  - Explicite (fourni par l'utilisateur)
  - Implicite



# Types de Contexte

---

- 3 aspects importants dans le contexte d'après Schilit, Adams et Want
  - Where are you ?
  - Who are you with ?
  - What resources are nearby ?
- 4 dimensions du contexte d'après Gross et Specht
  - Localisation
  - Identité (intérêts utilisateurs, préférences, ...)
  - Temps (i.e. heures de travail)
  - Environnement ou activité



# Classification des Contextes

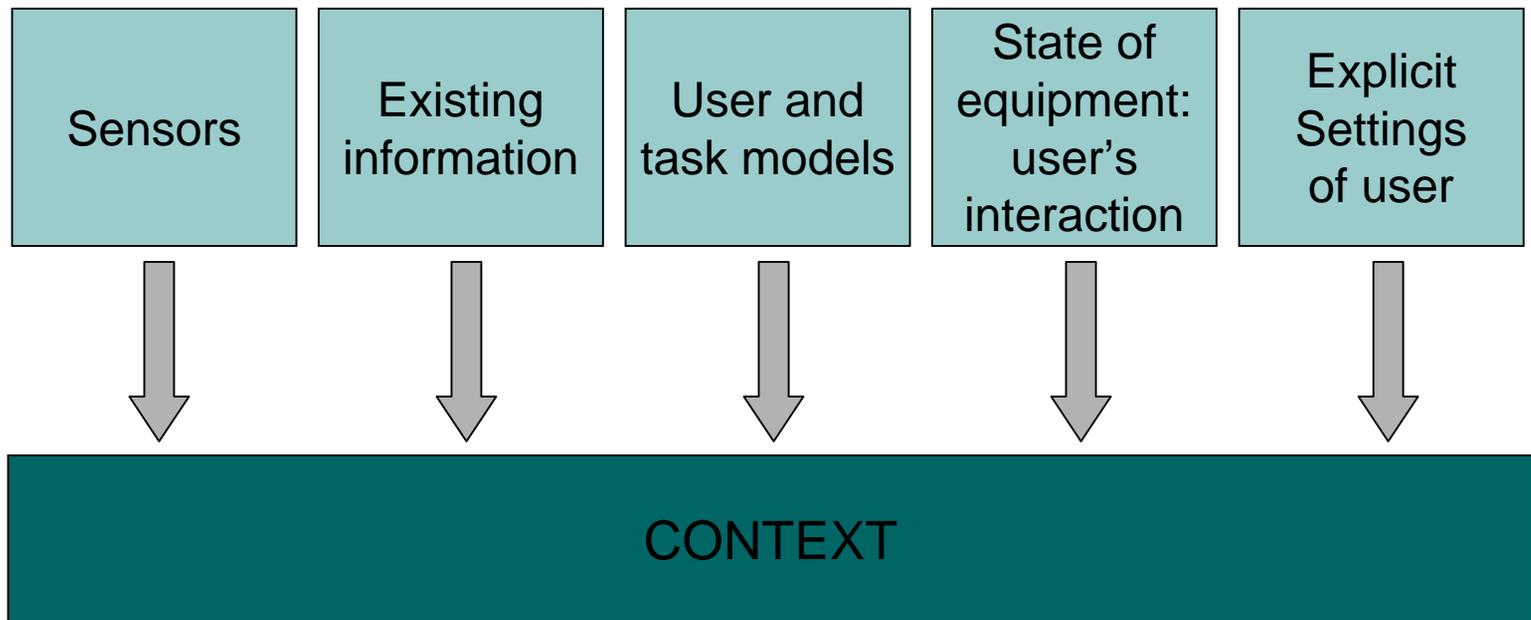
---

- Types de contexte
  - Physique: externe, mesuré par des capteurs matériels
  - Logique: interne, spécifié par l'utilisateur ou capturé par un système de monitoring
- Classification du contexte en familles:
  - **Contexte environnemental:** proximité, bruit, climat, ...
  - **Contexte utilisateur:** localisation, activité, émotion, ...
  - **Contexte machine:** énergie, ressources, occupation système, connexions, débits réseau, ...
  - **Contexte temporel:** date, heure, historique des actions, ...



# Déduction du Contexte

---



# Capture des Informations Contextuelles: Capteurs / Senseurs

---

- Capteurs physiques
  - GPS, accéléromètres, microphones, caméras, thermomètres, senseurs tactiles, Bio-senseurs
- Capteurs virtuels
  - Activité de l'utilisateur par analyse des mouvements de la souris et les interactions claviers
  - Localisation par utilisation de l'agenda électronique
- Capteurs logiques
  - Combinaison de plusieurs capteurs physiques et virtuels avec recours éventuel à une base d'informations
- Nécessité d'avoir un assemblage ou une composition



# Plusieurs Sources pour un même Type d'Information

---

- Localisation géographique:
  - GPS, GSM, badges IR, RFID, ultra-sons, caméras, ...
- Identification:
  - lecteur d'empruntes, reconnaissance vocale, caméras, code alpha numérique, ...
- Température:
  - capteur thermomètre numérique, internet (serveur météo), ...
- Heure:
  - Horloge interne ordinateur, radio, internet (ntp), ...



# Utilisation du Contexte

---

- Challenge: synthétiser des événements de haut niveau à partir:
  - d'événements de bas niveau
  - d'informations partielles (manques)
  - d'informations partiellement correctes (erreurs)
  - d'informations en contradiction
- Proposition de solution pour résoudre ces problèmes
  - Donner une priorité aux informations
  - Utiliser des modèles probabilistes



---

# Modélisation du Contexte



# Représentation du Contexte

---

- Différents types de structures de données ont été utilisés pour la modélisation ou représentation du contexte (Chen et Kotz)
- Généralement le contexte est décrit comme un ensemble d'attributs
  - Paires de nom/valeur
  - Brown pointe aussi que les valeurs des attributs comme « none », « any » et « not-working » sont nécessaires pour avoir tous les comportements souhaitables.



# Exemple: Individu Contextuel

---

- Dans le contexte position 2D, date, heure, existent plusieurs individus:
  - I1: (200, 300, 6 janvier 2006, 12h06)
  - I2: (5000, 320, 7 janvier 2006, 12h15)
  - I3: (2000, 400, 8 juillet 2005, 20h30)
- Problème: quel est le plus proche de I2 ?
  - Notion de distance
  - Notion de composition



# Formalisation de la Notion de Contexte: Fonctions de Distance

---

- Formalisation du contexte (Pauty, Couderc et Banâtre, 2004)

$$\mathcal{E} = \{ \{e_1, \dots, e_n\}, \{d_1, \dots, d_n\} \}$$

- Espace contextuel défini comme un espace d'états
- Un état contextuel est un vecteur constitué de composantes contextuelles  $e_i$
- Association d'une fonction de distance  $d_i$  à chaque composante contextuelle  $e_i$ , avec:

- Soit  $d_i: E \times E \rightarrow \mathbb{R}^+$  et  $\forall x, y, z \in E$ :

$$\left\{ \begin{array}{ll} d_i(x, y) \geq 0 & \\ d_i(x, y) = 0 \Leftrightarrow x = y & \text{(séparation)} \\ d_i(x, y) = d_i(y, x) & \text{(symétrie)} \\ d_i(x, y) \leq d_i(x, z) + d_i(z, y) & \text{(inégalité triangulaire)} \end{array} \right.$$



# Notion de Distance

---

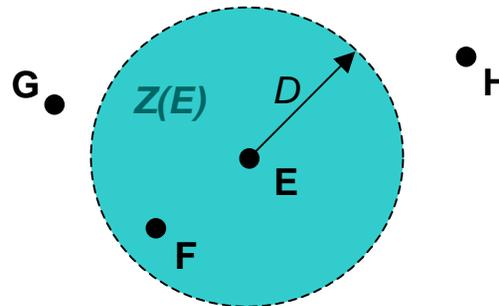
- Distance pour une composante contextuelle
  - $d_i$
- Comment combiner ces distances sachant que:
  - On veut pouvoir ajouter et retirer des composantes contextuelles sans perturber l'application
  - On veut pouvoir moduler l'importance de certaines composantes par rapport aux autres



# Notion de Zone Contextuelle

---

- Formalisation de la zone contextuelle
  - La fonction de distance sur l'espace contextuel permet de définir:
    - La proximité ou voisinage (qu'est ce qui dans mon contexte)
    - C'est à dire la zone contextuelle
      - $Z(E) = \{ F / d(F, E) \leq D \}$



[Pauty, Couderc, Banâtre:2004]



# Problématique des Distances

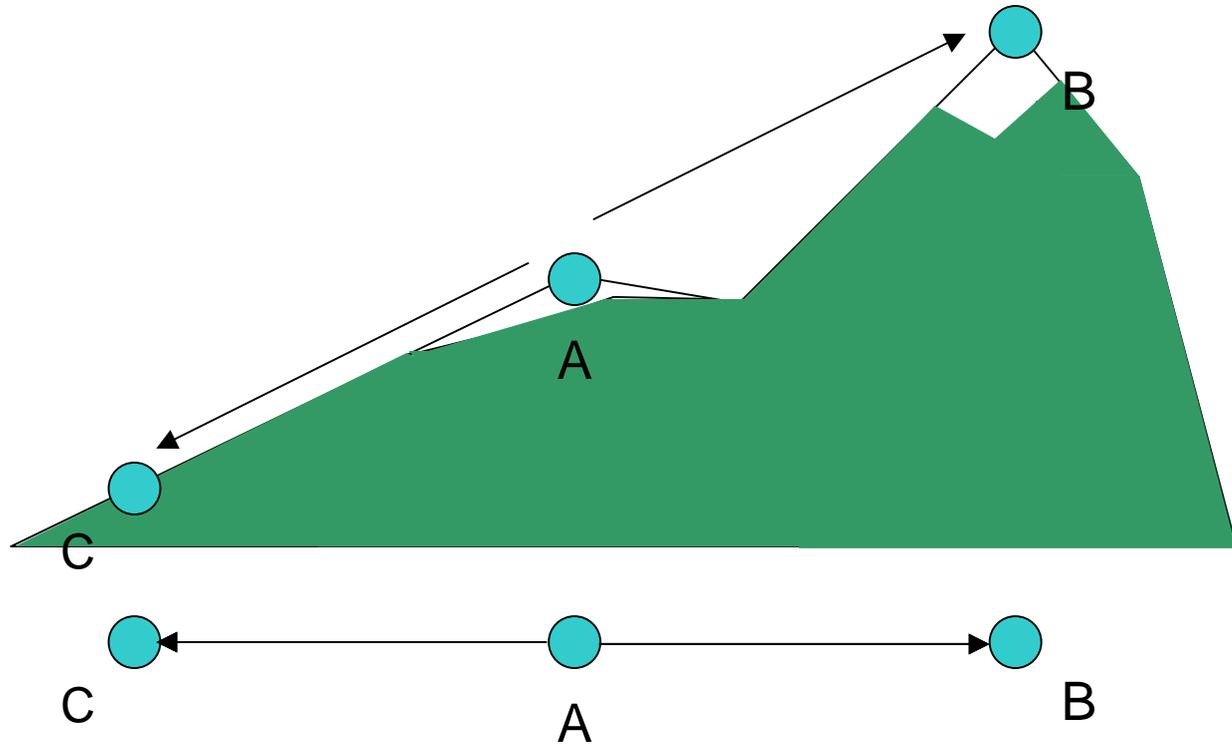
---

- Fonctions de distance:
  - Outil de mesure entre composantes contextuelles
  - Outil mathématique intéressant et bien maîtrisé
- Mais
  - Notion suffisamment large pour « tout » couvrir ?
  - Permet de quantifier correctement ce qui sépare deux composantes contextuelles ?



# Exemple de Limitations

---



# Modélisation à l'Aide de Fonctions de Coût

---

- Modélisation à l'aide de fonctions de coût (Lingrand, Lavirotte and Tigli, « *Selection Using non Symmetric Context Areas* », Workshop on Context-Aware Mobile Systems, LNCS 3762, 2005)
- Définition d'une *fonction de coût*  $c_i$ 
$$\left\{ \begin{array}{l} c_i(x, y) \geq 0 \\ c_i(x, y) \neq 0 \Rightarrow x \neq y \\ c_i(x, y) \leq c_i(x, y) + c_i(y, z) \end{array} \right.$$
$$\left\{ \begin{array}{l} d_i(x, y) \geq 0 \\ d_i(x, y) = 0 \Leftrightarrow x = y \\ d_i(x, y) = d_i(y, x) \\ d_i(x, y) \leq d_i(x, y) + d_i(y, z) \end{array} \right.$$
- La notion de coût permet
  - Une modélisation plus appropriée
  - Implique la perte de symétrie des zones contextuelles



# Zones Contextuelles et Méthodes de Sélection

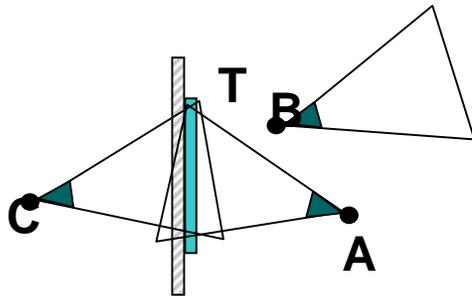
---

- Sélection des entités
  - Dans la zone contextuelle de E:
    - $S_{\text{endo}}(E) = Z(E)$
  - Dont la zone contextuelle contient E:
    - $S_{\text{exo}}(E) = \{ F / E \in Z(F) \}$
- Sélection des entités « en phase »
  - Dans la zone contextuelle de E et dont la zone contextuelle contient E
    - $S_{\text{bilatérale}}(E) = S_{\text{endo}}(E) \cap S_{\text{exo}}(E)$



# Méthodes de Sélection

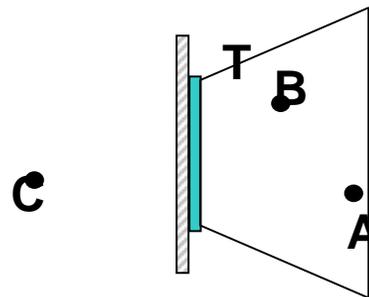
## Sélection Endo



Les entités A et C voient T

$$S_{\text{endo}}(E) = Z(E)$$

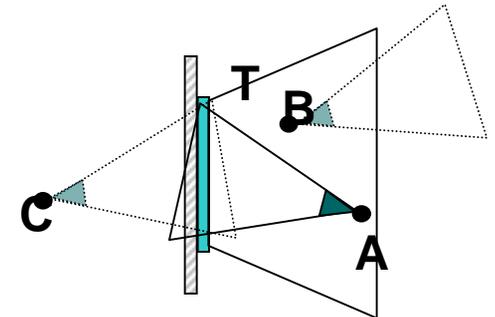
## Sélection Exo



T peut être vu par A et B

$$S_{\text{exo}}(E) = \{ F / E \in Z(F) \}$$

## Sélection Bilatérale



Seul A voit T

$$S_{\text{bilatérale}}(E) = S_{\text{endo}}(E) \cap S_{\text{exo}}(E)$$

[Lingrand, Lavirotte, Tigli:2005]



---

# Applications Sensibles au Contexte



# Caractéristiques Souhaitées

---

1/2

- Rhodes et Maes soulignent que la pertinence peut se corrélérer avec l'utilité mais pas obligatoirement:
  - Le récupération d'un document pertinent mais déjà bien connu n'est pas utile
- L'utilisateur est intéressé par un contexte immédiat
  - (Jones and Brown, SIGIR Workshop on Mobile Personal Information Retrieval, 2002)
- Pour que les utilisateurs adoptent les applications sensibles au contexte, il est nécessaire:
  - qu'il y ai un bénéfice
  - que l'utilisateur puisse garder le contrôle et la capacité de faire un autre choix (Lamming and Newman, Personal Computers and Intelligent Systems Information Processing, 1992).



# Caractéristiques Souhaitées

---

2/2

- Brown et al. notent, concernant la sécurité et le respect de la vie privée, que:
  - Les utilisateurs doivent avoir le contrôle sur les informations qui les concernent
  - La manière dont les informations sont utilisées doit être clairement identifiée
  - Les utilisateurs peuvent seulement récupérer les informations



# Contexte et IHM

---

- Traditionnellement
  - Interaction par actions explicites
    - Pointer la souris sur l'entrée d'un menu
    - Cliquer sur un bouton
  - Contexte implicite
    - Paramétrage par défaut
- Context-aware computing
  - Utilisation du contexte de manière explicite
  - Nécessité de redéfinir les interfaces et les interactions



# Applications Mobiles

---

- Les utilisateurs d'applications mobiles sont déjà occupés par leurs activités
  - Ne pas submerger l'utilisateur d'informations
  - Privilégier la précision plutôt que la redondance
- Dans un situation de mobilité:
  - L'environnement change vite (contexte dynamique)
  - L'information de contexte est généralement appropriée si elle est proche spatialement et temporellement (Huang, « *Supporting Context-aware Computing in Ad Hoc Mobile Environments* », Technical Report WUCS-02-36, 2002)



# Problèmes des Applications Sensibles au Contexte

---

- Pascoe, Ryan et Morse:
  - Gourmand en ressources
  - Coût de développement élevé
  - Environnements informatique divers
- Les applications sensibles au contexte existantes sont majoritairement monolithiques. Manque un modèle conceptuel partagé (Huang)
- Des outils et des standards doivent être développés pour faciliter la tâche



# Classification des Applications Sensibles au Contexte

---

1/2

- Applications réactives et proactives
  - Exemple d'application proactive: Just-In-Time Information Retrieval agent (JITIR) (Rhodes et Maes)
- Applications « continues »
  - Changement en continu suivant l'évolution du contexte
  - Applications discrètes plus simples: moteur de suivi des changements du contexte (Brown, Personal Technologies 2, 1998)



# Classification des Applications Sensibles au Contexte

---

2/2

- Classification des applications selon Dey et Abowd
- 4 types de contextes primaires:
  - Activité
  - Identification
  - Localisation
  - Temps
- Catégories d'applications sensibles au contexte:
  - Présentation d'information et de services
  - Exécution automatique de services
  - Étiquetage du contexte pour la récupération ultérieure



---

# Architecture pour Applications Sensibles au Contexte



# Terminologie

---

- **Architecture** = structure globale, composants logiques, et relations logique du système
- **Librairie** = ensemble d'algorithmes relatifs à un sujet (orienté réutilisation de code)
- **Framework** = structure commune pour une certaine classe d'applications (orienté réutilisation de designs)
- **Toolkit** (Boîte à outils) = collection de composants pour des fonctionnalités communes (framework)
- **Infrastructure** = ensemble bien défini, fiable et publiquement accessible de technologies qui agissent pour d'autres systèmes
- **Service infrastructure** = technologie middleware accessible par le réseau



# Architectures Sensibles au Contexte

---

- Accès direct aux capteurs
  - Dispositifs accessibles localement
  - Pas adapté aux applications réparties
  - Systèmes difficilement extensibles
- Middleware pour la gestion du contexte
  - Introduction d'une couche dans l'architecture logicielle
  - Facilite l'extension des systèmes et la réutilisabilité
- Serveur de contexte
  - Accès concurrents par de multiples clients



# Approche Horizontale / Verticale

Applications					Capteurs / Senseurs	Capteurs / Senseurs
Framework / Environnement						
Librairies / Interfaces de programmation						
Machine étendue	OS Threads Mem virtuelle	Env. Graphique	Multiplexage	Systeme Fichiers		
Gestion		Interruptions	Pilotes			
Matériel	CPU + Mem	Dispositifs IHM <i>Écran, Clavier, Souris</i>	Com Dcom	Données		



# Architecture: « C/S »

---

- Pascoe, Ryan et Morse écrivent:
  - le cœur d'une infrastructure supportant les applications sensibles au contexte est un « *Context Information Service* »
- Les objectifs d'un CIS:
  - Rassembler, Modéliser, Fournir les informations contextuelles
- « Guidelines » pour un CIS:
  - Modèle contextuel orienté objet
  - Accès partagé aux ressources
  - Composants réutilisables et extensibles
  - Infrastructure de service multicouches
  - Support de l'extensibilité
  - Indépendant de la plate-forme



# Architecture: « *Context Toolkit* »

---

- Salber, Dey et Abowd définissent une boîte à outils pour le contexte (Proc. Conf. Human Factors in Computing Systems, CHI'99, 1999)
  - Context Widget: obtient un type d'info contextuelle
  - Context Server: rassemble les infos d'une entité (proxy)
- Description complète du système dans:
  - Special Issue on Context-Aware Computing of Human-Computer Interaction (2001)
  - 70 pages



# Avantages d'une « *Service Infrastructure* »

---

- Grande variété de dispositifs et d'applications peuvent être utilisés
  - Indépendance vis à vis de la plate-forme matériel
  - Indépendance vis à vis du système d'exploitation
  - Indépendance vis à vis des langages de programmation
- Les capteurs ou senseurs peuvent être changés
  - Indépendance vis à vis des autres capteurs
  - Changement dynamique (couches middleware indépendantes)
- Les dispositifs peuvent être plus simples
  - Partage de capteurs et de puissance de calcul
  - Partage de données et de services



# Challenges pour une « Service infrastructure »

---

- Formats de données et protocoles de communication
  - autoriser l'interopérabilité
  - Supporter différents types de capteurs et de contextes
- Construire les services de base dans l'infrastructure
- Trouver le bon équilibre entre dispositifs et infrastructure
- Assurer la sécurité et l'intimité des données contextuelles
- Passage à l'échelle pour un grand nombre de capteurs, de services, de dispositifs et d'utilisateurs



# Expérimentation: UPnP

---

- Technologie UPnP utiles pour les applications sensibles au contexte:
  - Adressage (dispositif réseau)
  - Description
  - Découverte
  - Commandes
  - Présentation
- Modélisation
  - Dispositifs
  - Services



# Expérimentation : Plateforme Wcomp

