

Systemes d'information pervasifs

Unité d'enseignement mutuelle

Frédérique Laforest

Frédéric Le Mouël

Qui suis-je?

- Frédérique Laforest
 - frederique.laforest@insa-lyon.fr
 - <http://liris.cnrs.fr/frederique.laforest>
- Maître de conférences INSA de Lyon
- Recherche au LIRIS
 - axe systèmes d'informations communicants
 - Mots-clés
 - Systèmes d'information pervasifs
 - Adaptabilité à l'utilisateur et au contexte
 - Interfaces utilisateurs
- Enseignement au département Télécommunications
 - Algorithmique, C, modélisation et programmation objet, java, web, bases de données

Qui suis-je?

- Frédéric Le Mouël
 - Frederic.le-mouel@insa-lyon.fr
 - <http://citi.insa-lyon.fr/~flemouel>
- Maître de conférences INSA de Lyon
- Recherche au CITI
 - axe middleware
 - Mots-clés
 - Systèmes pervasifs
 - Systèmes adaptatifs
 - Informatique répartie à large-échelle
- Enseignement au département Télécommunications
 - Middleware, modélisation et programmation objet, java, web

Plan

- Introduction
- Architectures des systèmes d'information distribués et pervasifs
- Trois niveaux de recherche en systèmes pervasifs
- Exemples de projets pervasifs
- Conclusion

Introduction

Systeme d'information

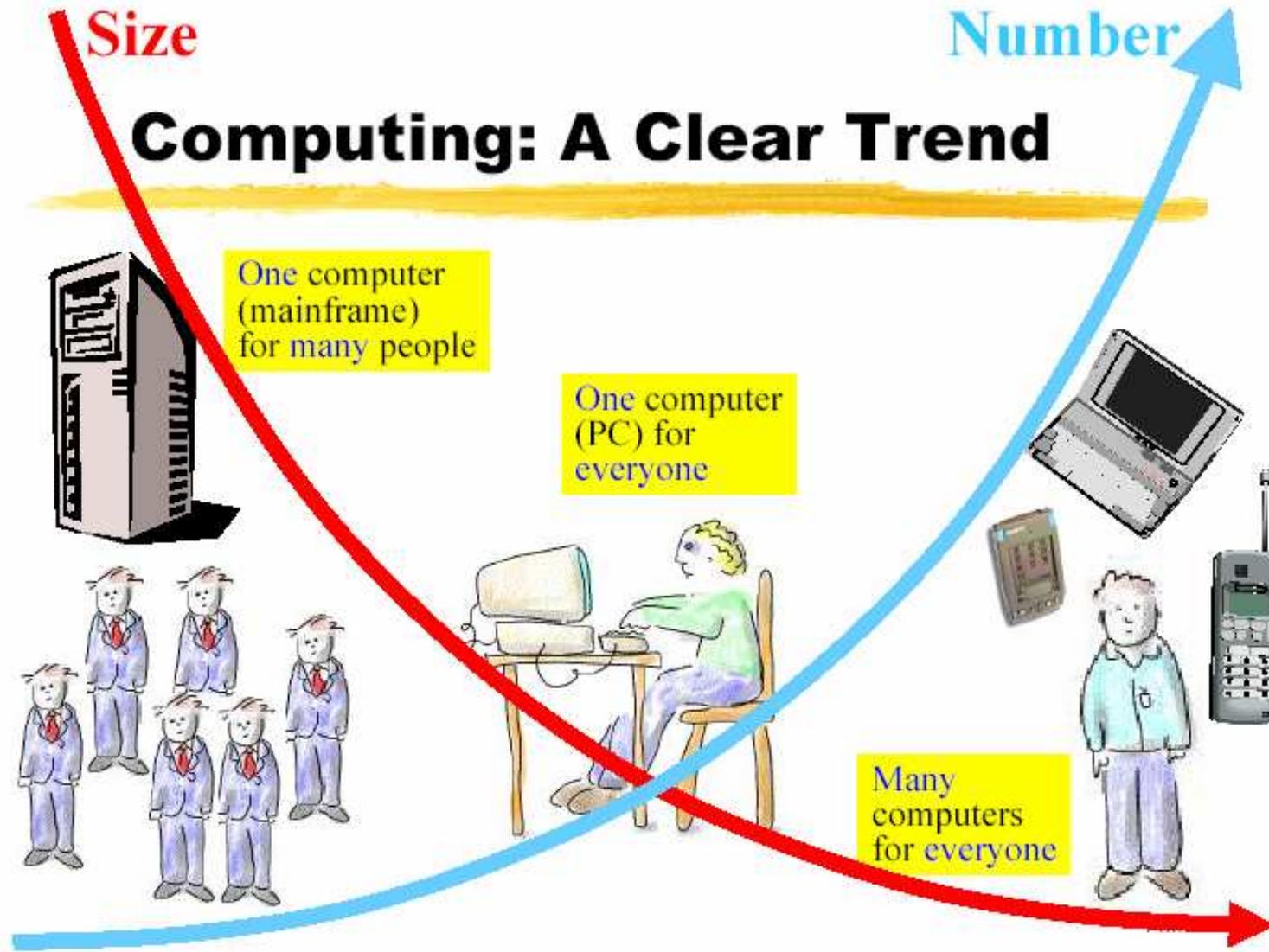
- Objectif
 - Production, collection, traitement, enregistrement et diffusion de l'information
- Composantes
 - Utilisateurs et administrateurs
 - Logiciels
 - Machines et matériels
 - Réseaux

Exemple de système d'information

- Suivi de patients hospitalisés à domicile
 - Utilisateurs et administrateurs
 - patients, médecins de ville, infirmières, praticiens hospitaliers, pharmacien, association de suivi
 - Logiciels
 - dossier patient électronique, workflow, rendez-vous, imagerie, mailing, saisie de données au domicile, télésurveillance...
 - Machines et matériels
 - matériel médical chez le patient (dialyseur, thermomètre électronique), matériel à l'hôpital (radiologie, chimiothérapie...), terminal patient, terminaux des acteurs de santé mobiles ou fixes
 - Réseaux
 - Liaison réseau privé de l'hôpital, association de suivi, domicile...

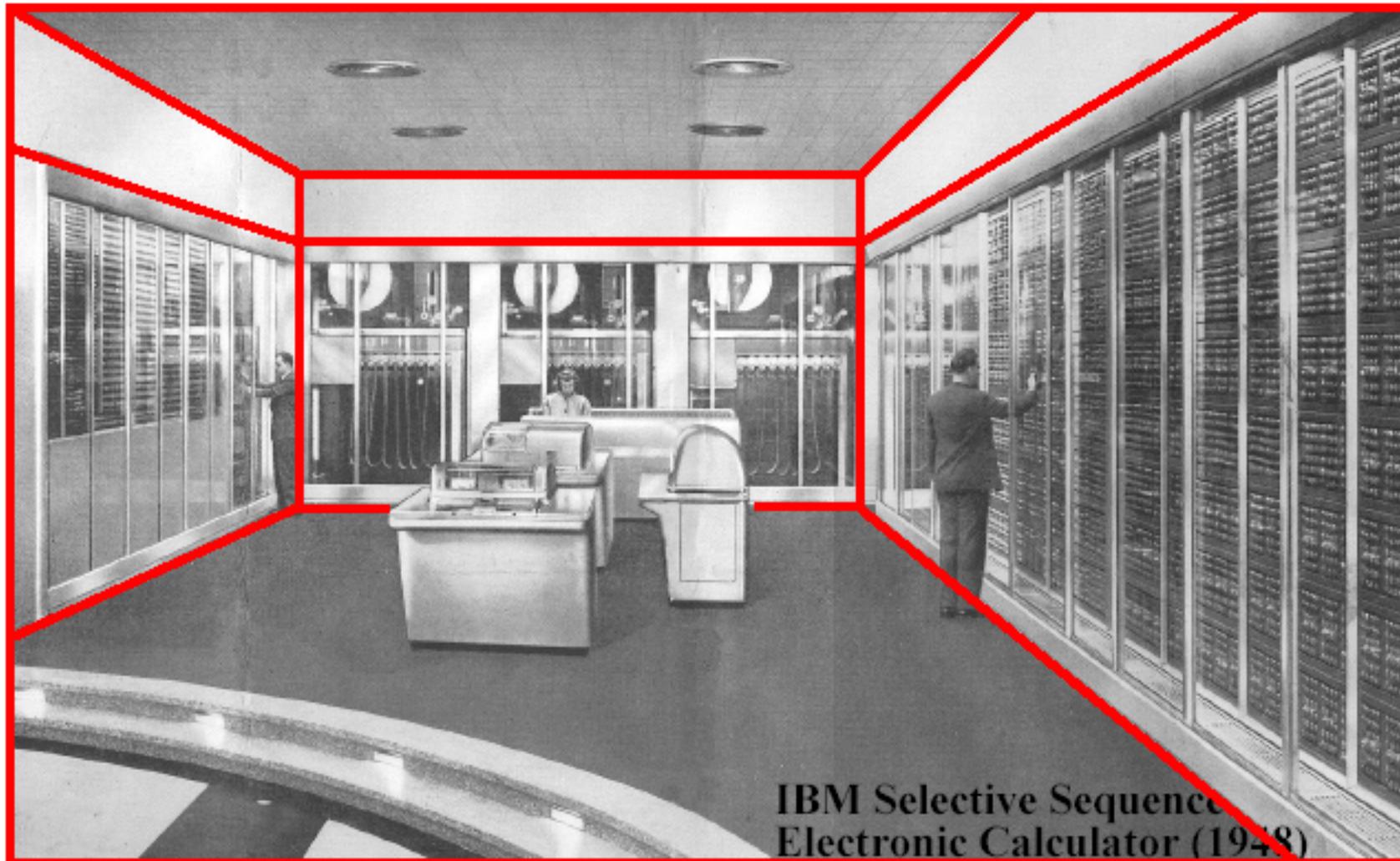
Une nouvelle vision de l'informatique

- Environnement informatique
 - Avant : environnement virtuel dans lequel nous entrons pour effectuer une tâche et dont nous sortons à la fin de la tâche
 - Aujourd'hui : espace physique amélioré de gestion de l'information
- Application
 - Avant : logiciel pour exploiter un matériel
 - Aujourd'hui : moyen pour l'utilisateur d'effectuer une tâche



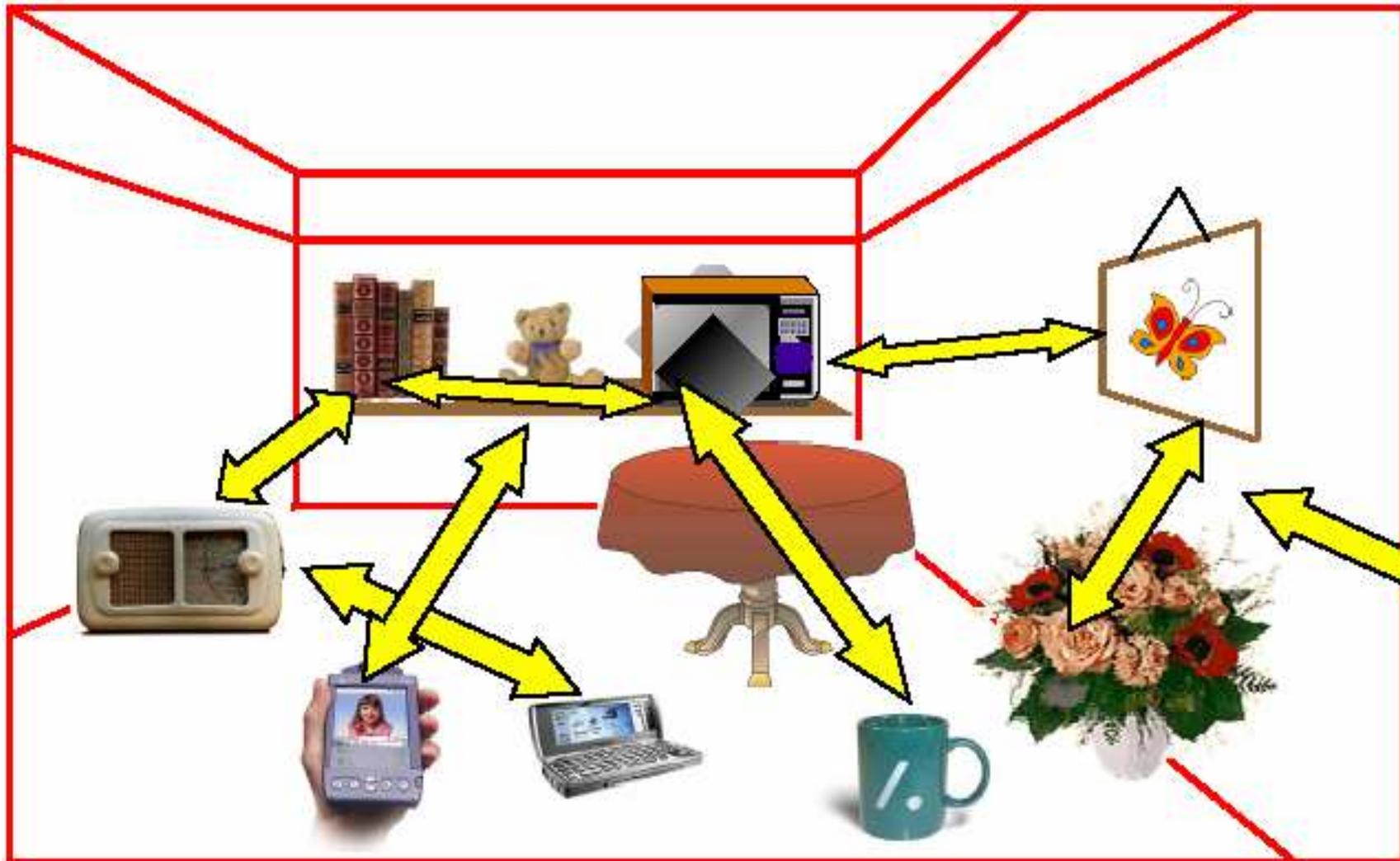
© F. Mattern

Les ordinateurs d'hier remplissaient les pièces



© F. Mattern

Ceux de demain aussi!!

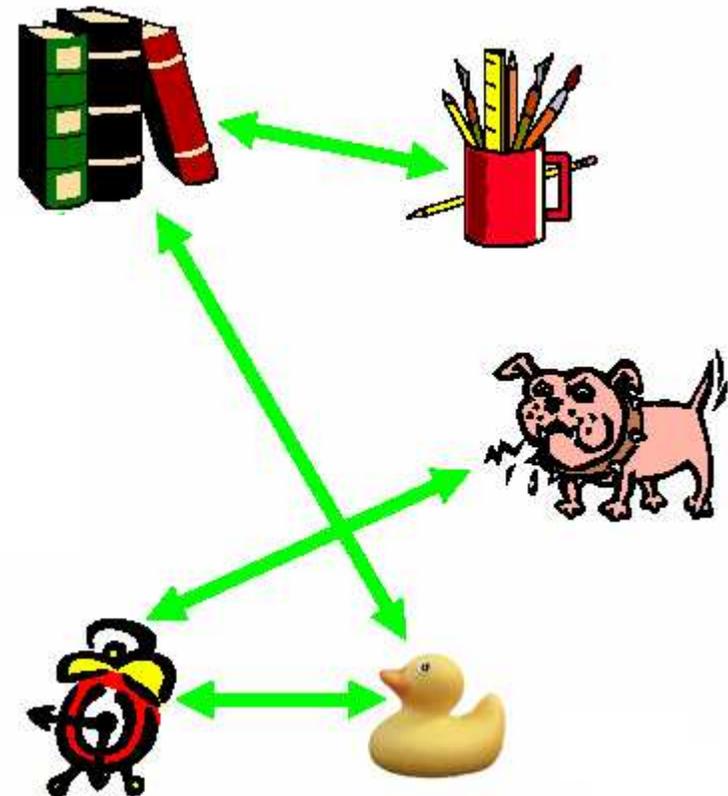


Informatique ubiquitaire

- Aujourd'hui, l'Internet connecte tous les ordinateurs
- Demain, tout objet sera « intelligent » (smart)
 - Processeurs embarqués
- Et ils seront tous interconnectés
 - Communication sans fil

L'informatique embarquée pour la coopération

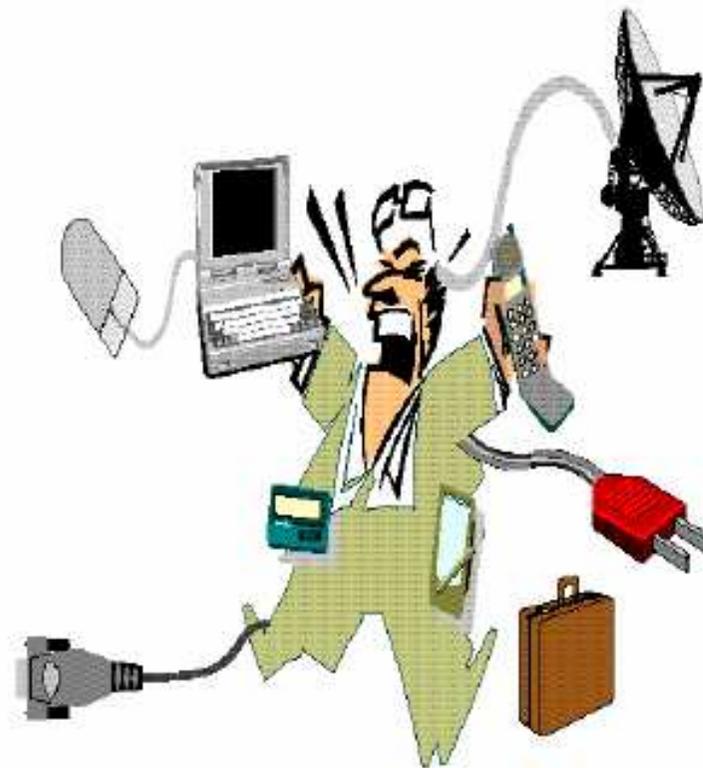
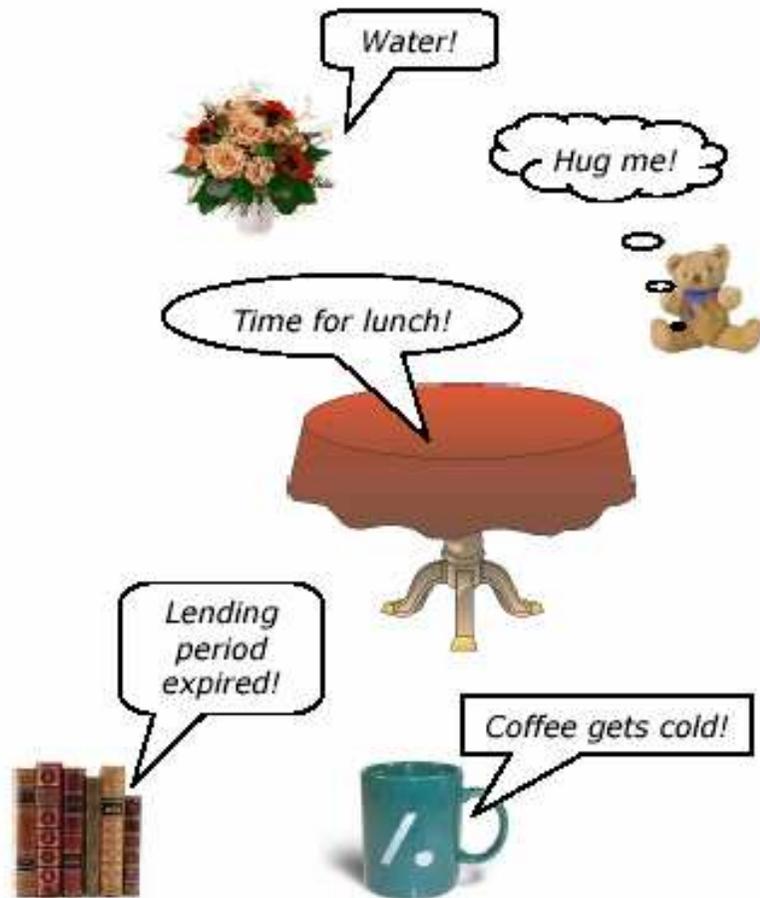
- Tout objet du monde réel sera enrichi de capacités de traitement de l'information
- Processeurs embarqués
 - Dans chaque objet quotidien
 - Petit, bon marché, léger
- Communications sans fil
 - Réseaux spontanés
- Capteurs



Les objets intelligents

- Peuvent se souvenir des événements importants
 - Mémoire
- Présentent un comportement dépendant du contexte
 - Capteurs
- Ils sont interactifs
 - Communiquent avec leur environnement
 - En réseau avec les autres objets intelligents

Ces objets sont-ils vraiment smart??



© F. Mattern

Systemes d'information pervasifs – la genèse

« Calm technology »

- Mark Weiser, 1991

- « A new way of thinking about computers in the world, one that takes into account the natural human environment and allows the computers themselves to vanish in the background »
- « The most profound technologies are those that disappear . They weave themselves into the fabric of everyday life until they are indistinguishable from it »

Systemes d'information pervasifs – la référence

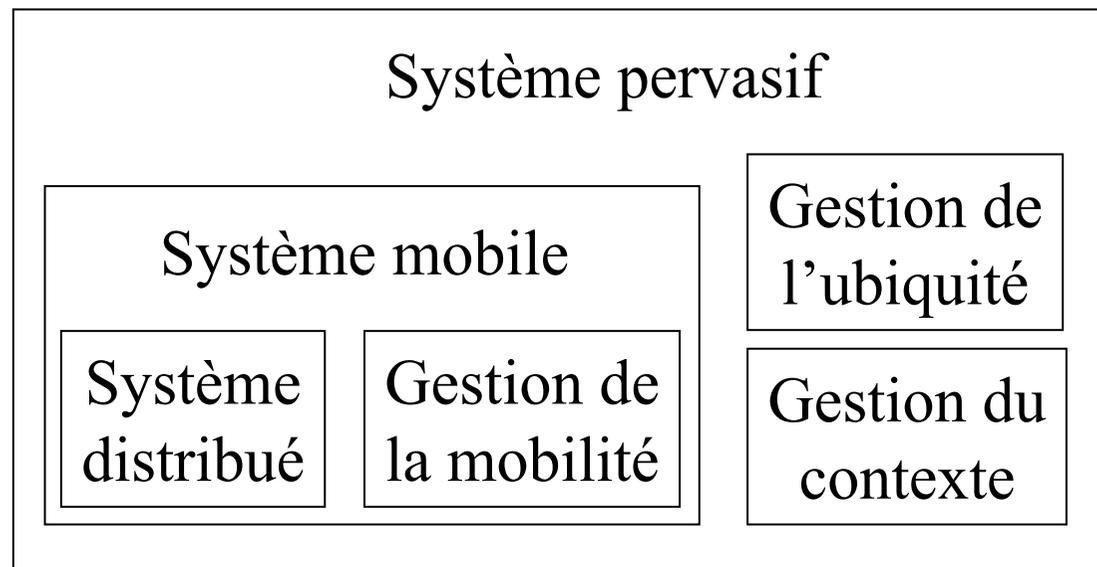
- M. Satyanarayanan, 2001
 - Étape suivante, après systèmes distribués et systèmes mobiles
 - 4 challenges
 - Utilisation effective des espaces intelligents (smart spaces)
 - Invisibilité
 - Scalabilité
 - Transparence des inégalités d'environnements
 - Pervasive computing environment = « one saturated with computing and communication capability, yet so gracefully integrated with users that it becomes 'a technology that disappears' »

Définitions

- Ubiquitaire
 - Accessible de n'importe où
- Mobile
 - Qui intègre les terminaux mobiles
- Context-aware
 - Qui prend en compte le contexte d'exécution
- Pervasif
 - Qui associe ubiquité, mobilité et context-awareness
- Ambient
 - Qui est intégré dans les objets quotidiens

Vue « système » d'un système pervasif

- Adapté de Saha & Mukherjee, 2003



Devoirs des SI distribués

- Persistance des données
- Échange de données entre applications hétérogènes
- Répartition des données sur des sites distants
- Gestion de la cohérence permanente des données
- Interopérabilité des plates-formes
- Portabilité des applications
- Gestion des accès concurrents
- Intégration des systèmes legacy
- Ouverture
- Sécurité

Devoir des SI pervasifs

- Ceux des SI distribués

ET

- Scalabilité
- Invisibilité
- Context-awareness
- Intelligence (« smartness »)
- Pro-action « all the time everywhere »

Scalabilité

- Passage à l'échelle
- Gérer des volumes de plus en plus grands de
 - utilisateurs
 - applications
 - appareils connectés
- Développer des applications dont le cœur est indépendant du volume, des utilisateurs et des appareils
- Utiliser des techniques d'adaptation pour pouvoir répondre à chaque cas

Invisibilité

- Nécessiter un minimum d'intervention humaine
- S'adapter seul aux changements
- Auto-apprentissage

- Par exemple,
 - reconfiguration dynamique des caractéristiques réseau d'un appareil
 - Accès aux ressources d'un « espace » en fonction de l'appartenance à la zone géographique de cet espace.
Sortie de l'espace = définition des limites de l'espace!

Context-awareness

- Perception de l'environnement pour interagir plus « naturellement » avec l'utilisateur
- Capteurs de l'environnement physique
- Matériels auto-descriptifs
- Description des personnes
- Méta-données sur les applications

« Smartness »

- Smart = intelligent, à l'esprit vif, malin, débrouillard
- Percevoir le contexte d'exécution ne suffit pas
- Il faut utiliser efficacement les informations du contexte
 - Exemple : maison intelligente

Pro-action

- Suggérer, proposer des actions correctives à l'utilisateur en fonction du contexte présent ou prédit
 - Par exemple se déplacer de 100 mètres pour atteindre un réseau plus performant et ainsi accomplir une tâche dans un temps « correct »
- Attention à balancer avec l'invisibilité!
- Sous-entend de savoir
 - Prévoir un événement, une situation,
 - Évaluer une situation courante ou possible,
 - Comparer deux situations et juger de la meilleure
 - Que « ça vaut le coup » de transgresser l'invisibilité

Architecture des systèmes d'information distribués et pervasifs

- 1- ancêtres
- 2- middleware
- 3- systèmes mobiles
- 4- grilles de calcul
- 5- peer-to-peer

*Mes remerciements à Stéphane Frénot
pour ses précieux transparents!*

Les ancêtres : le PC

- A démocratisé l'informatique
- A permis une croissance gigantesque en matière de composants matériels
- A permis le développement des interfaces utilisateurs graphiques
- N'a pas fourni le potentiel attendu en matière de traitement de l'information

Les ancêtres : le Web

- Pionnier d'infrastructure de communication et d'information ubiquitaire
- A l'origine, n'a pas été pensé comme une infrastructure pour systèmes distribués
- A créé une culture qui a permis d'imaginer les systèmes pervasifs

Les ancêtres : le Client/Serveur

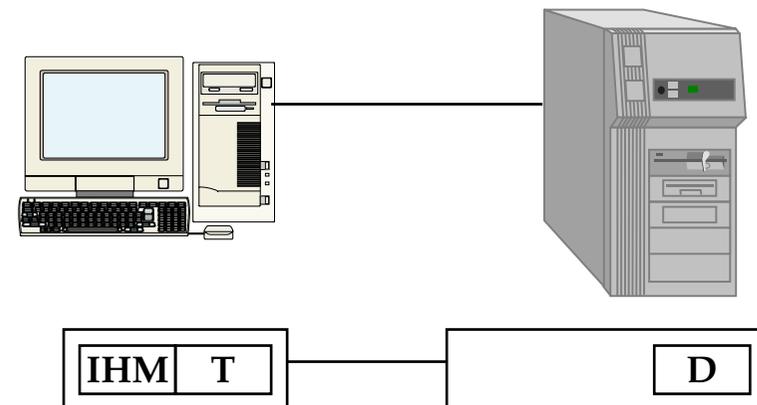
- C/S de présentation
 - Client : gestion de la présentation
 - Serveur : réalisation de l'ensemble des traitements
- C/S de traitement
 - C : Gestion de la présentation + traitements applicatifs
 - S : Gestion de l'accès aux BD
- C/S multi-tiers
 - C : Gestion de la présentation
 - Serveur applicatif : Connaissance des traitements métiers
 - Serveurs : gestion des accès aux BD

Client-serveur de présentation

- Déporter l'affichage sur un réseau
 - telnet
 - Xwindows
 - NTTerminal Serveur
- Le développement est « presque » centralisé

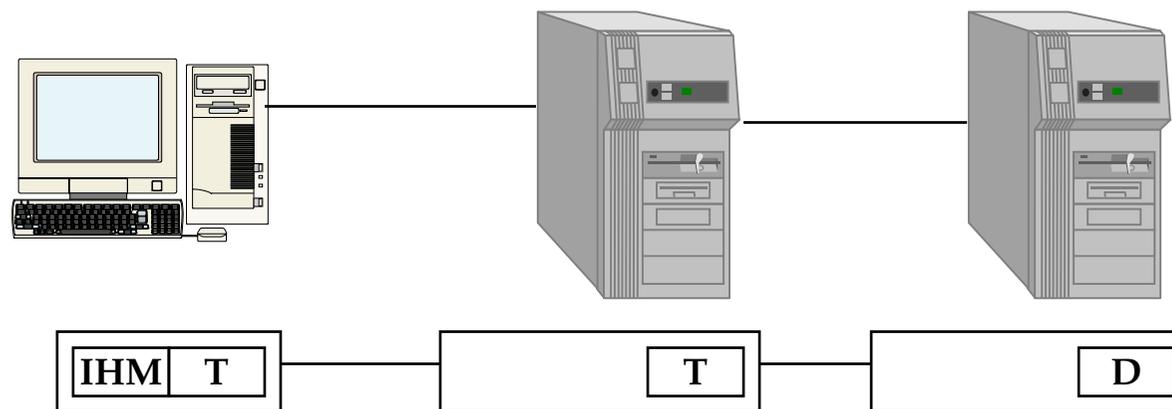
Client-serveur de traitement

- Le poste de travail héberge l'ensemble de la gestion d'interface homme-machine et le traitement,
- Le serveur est un serveur de base de données
- Architecture dénommée « client obèse »
- Avantage : Mise en œuvre efficace pour un nombre réduit de clients
- Inconvénients
 - Coûts de déploiement ?
 - Coûts de MAJ ?
 - Accès concurrents ?
 - Hétérogénéité sur BD ?



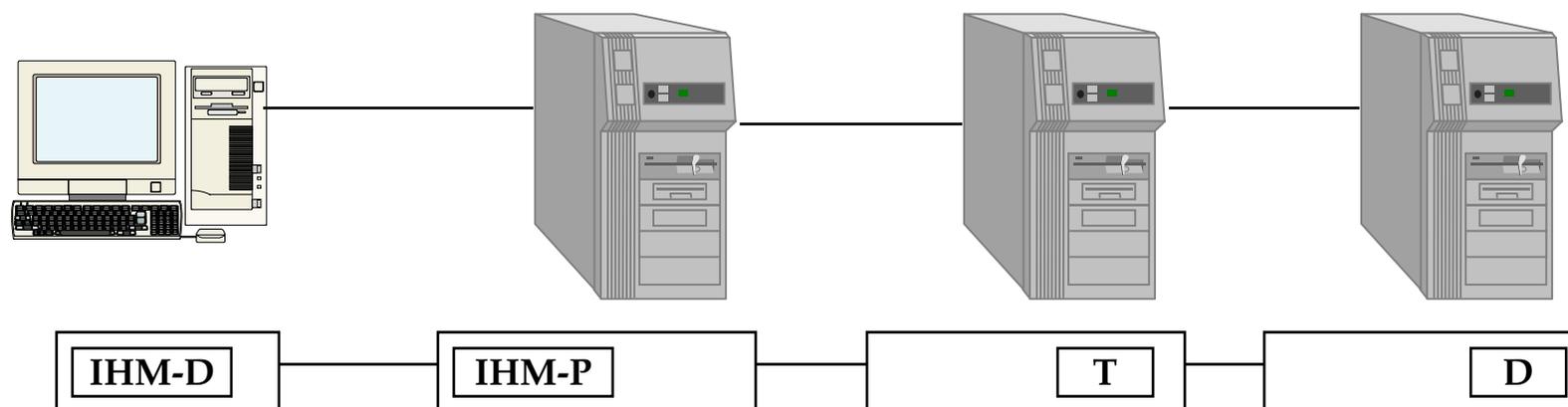
Client-serveur 3 niveaux (3-tier)

- Le poste de travail héberge la gestion de l'interface homme-machine et une partie des traitements,
- Le serveur d'applications gère l'autre partie des traitements
- Le serveur de données gère les accès aux données
- Architecture dénommée « traitements coopératifs »



Client-serveur 4 ou n niveaux (4-tier, n-tier)

- Le poste de travail héberge un navigateur standard,
- Le serveur HTTP gère la partie présentation de l'interface homme-machine
- Le serveur d'applications gère les traitements
- Le serveur de données gère les accès aux données
- Architecture de collaboration

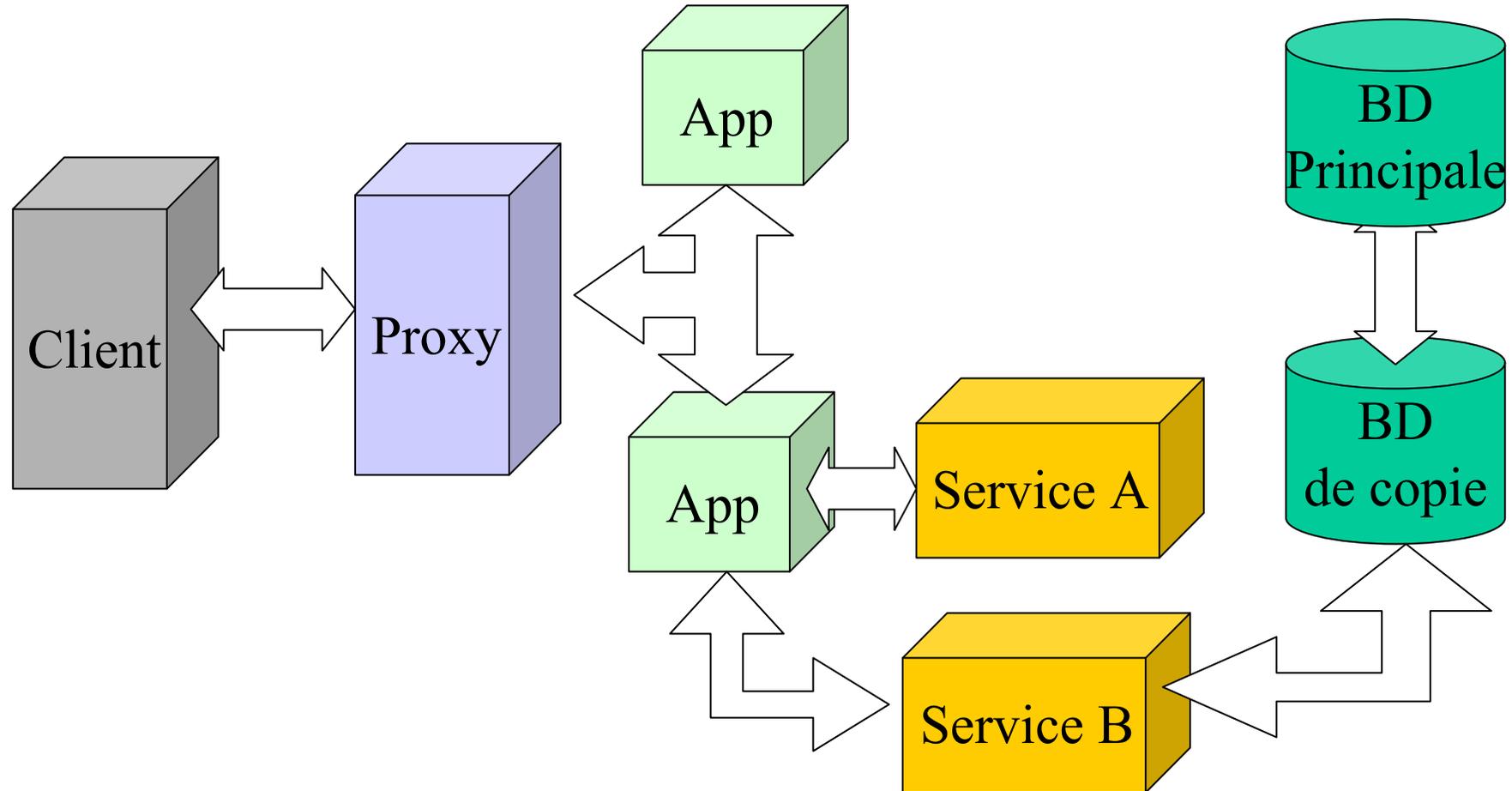


© S. Frénot

Le Client / Serveur

- Avantages
 - Première infrastructure informatique pour un travail coopératif
 - Centralisation des traitements au niveau du serveur
 - Pas de duplication des données (état global observable, gestion plus simple de la cohérence et de l'intégrité des données)
 - Maîtrise globale des processus
- Inconvénients
 - Relation directe C/S
 - Pas de transparence sur la localisation
 - Augmentation de l'hétérogénéité
 - Modèle rigide !
 - Ni portable ni inter-opérable

Client-serveur : la distribution à "l'ancienne"



Client-serveur : la distribution à "l'ancienne"

- Elle nécessite des compétences humaines précises
 - Systèmes propriétaires
 - Gestion de transactions
 - Définition de queues de messages
 - Réplication et Synchronisation de BD
 - Gestion des pannes
 - Sécurité des communications
 - Développement de clients
- Elle pose des problèmes techniques
 - Nécessite de nombreux serveurs pour l'équilibrage de charge
 - Nécessite une programmation complexe pour pouvoir évoluer

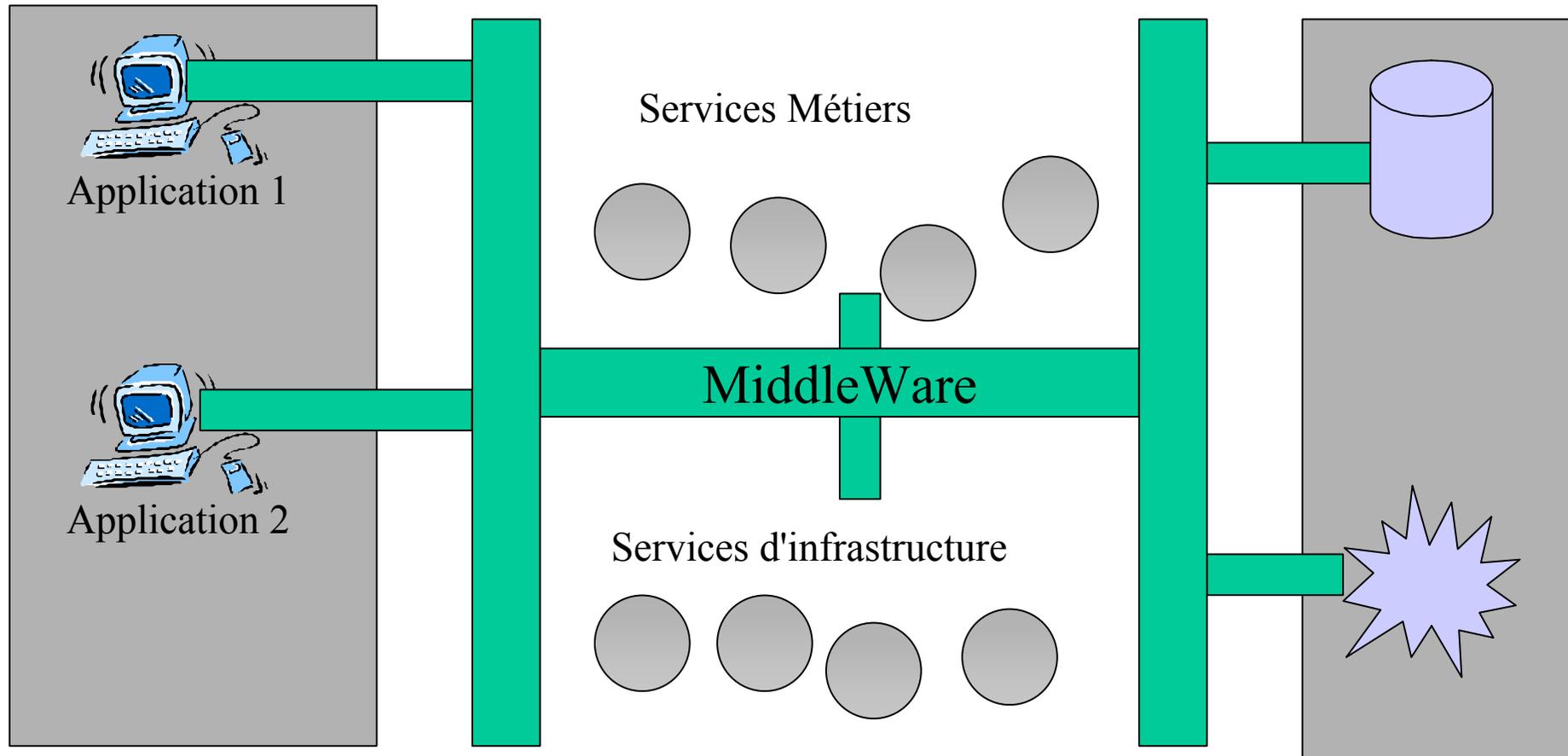
Du C/S au middleware explicite

- C/S
 - Le client et le serveur sont développés en collaboration

- Objet distant
 - Client et serveur sont liés par une interface
 - La couche réseau est masqué au client et au serveur

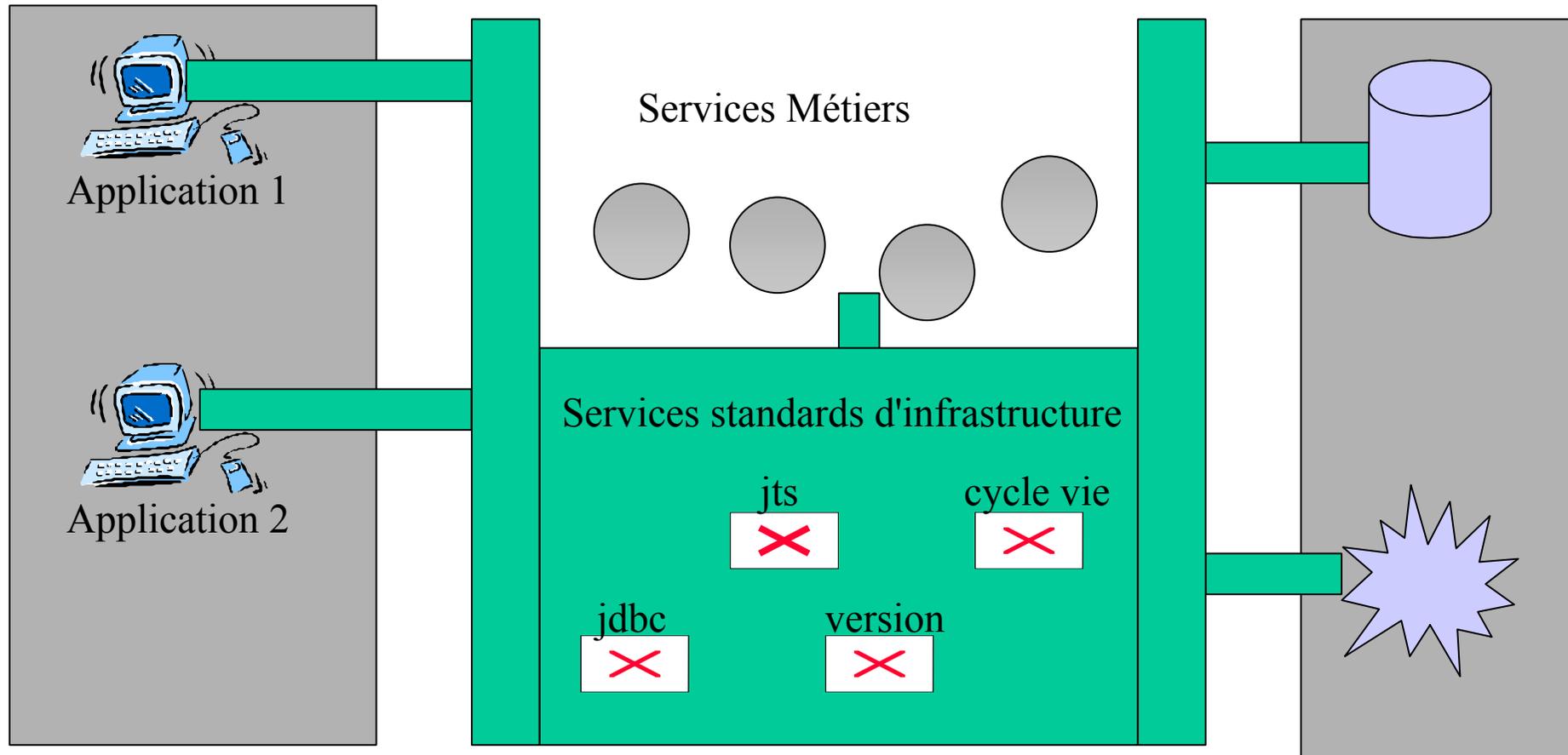
- \implies Notion de code applicatif/code non applicatif

Architectures middleware à objets distribués



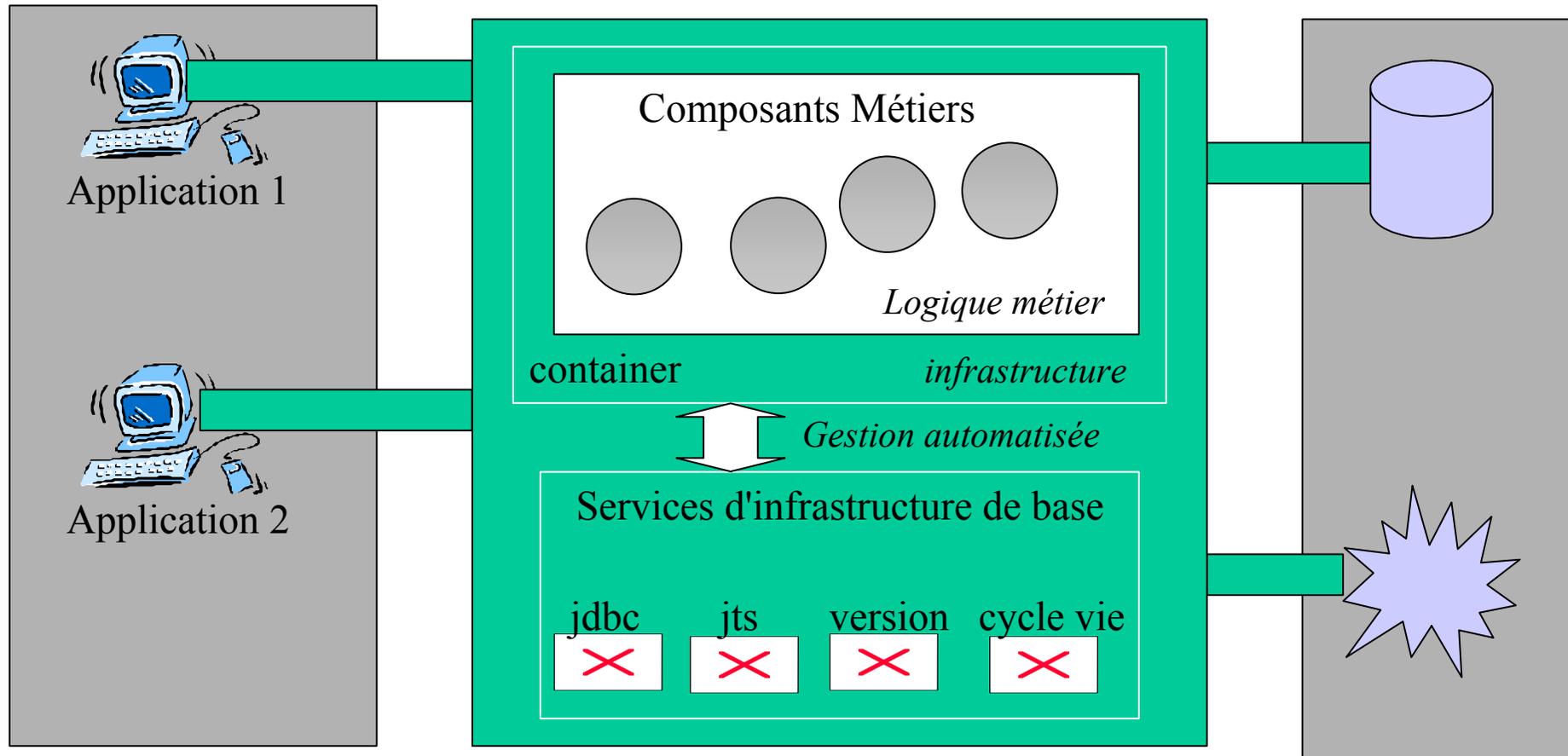
Exemples : DCOM, CORBA

Serveur middleware de composants de base



Exemple : OrbixWeb

Serveur middleware de composants intégré



Exemples : Tomcat (J2EE), Oscar (OSGi)

Services du container d'objets métiers

- Services internes
 - Gestion de la charge du serveur
 - (cycle de vie, accès client, passivation...)
 - Service de nommage
 - Gestion des accès aux objets métiers
- Services externes
 - Gestion du mapping sur BD relationnelle
 - Gestion des transactions
 - Gestion des échanges de messages

Systemes mobiles

- Début des années 90
 - Essor des réseaux sans fil et des ordinateurs portables
- Quand des éléments du système sont mobiles et donc souvent non permanents
- Contraintes clés
 - Variations impromptues de la qualité du réseau
 - Déconnexions
 - Faibles confiance et robustesse des éléments mobiles
 - Limitations des ressources locales (poids, taille => écran, batteries...)
- Toujours un domaine de recherche actif

Systemes mobiles

- Résultats actuels
 - Techniques d'économie d'énergie (gestion de mémoire, rapidité processeur... en fonction de l'énergie disponible)
 - Réseaux mobiles (mobile IP, protocoles ad hoc, WIFI, Bluetooth, IRDa, GPRS...)
 - Accès aux informations (opérations déconnectées, contrôle de la cohérence des données...)
 - Support d'applications adaptatives (proxies, gestion adaptative des ressources système)
 - Calcul et gestion de la localisation
 - Systemes d'information
 - Wap et imode (C/S et téléchargement)

Grilles de calcul

- Système distribué qui permet le partage, la sélection et l'agrégation de ressources géographiquement distribuées pour la résolution de problèmes à grande échelle
- Ressources
 - Ordinateurs : PC, clusters, (éléments mobiles)...
 - Logiciels
 - Données et bases de données
 - Matériels spécifiques
- En fonction de leur disponibilité, capacités, coûts

Peer-to-peer

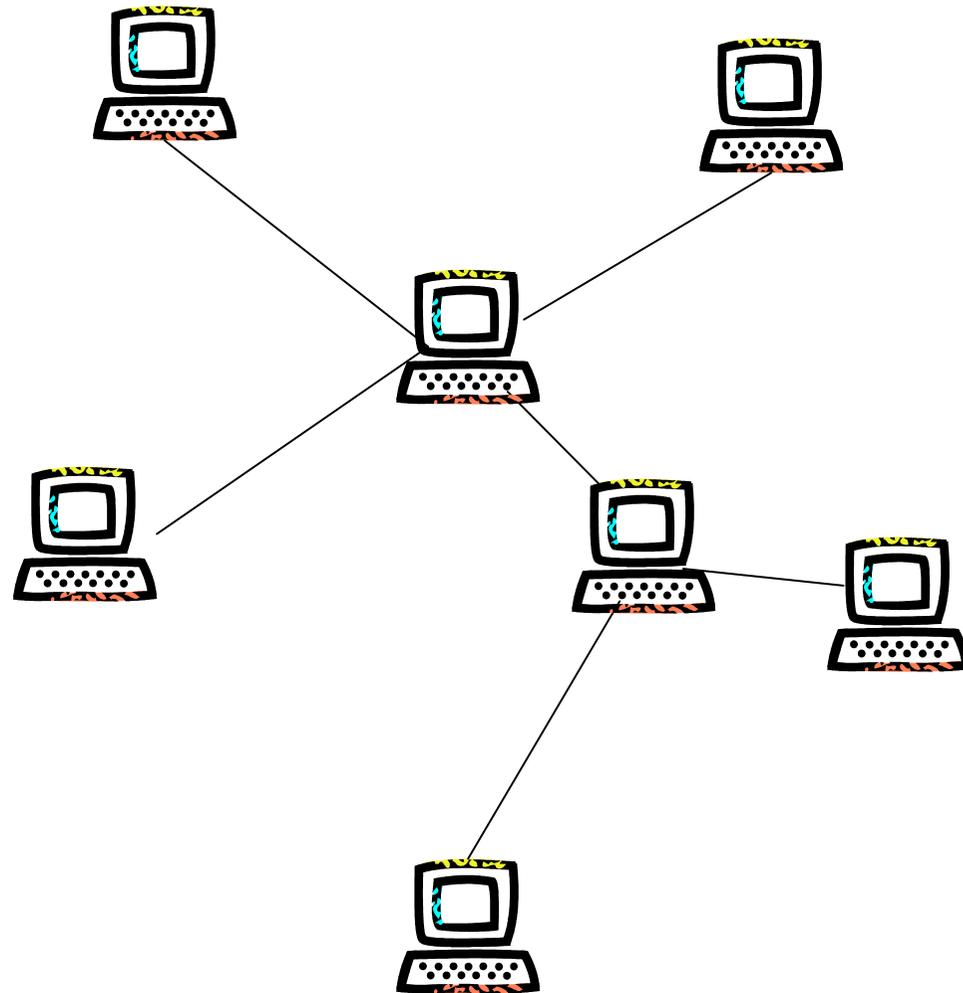
- Peer = « partenaire »
 - “...an entity with capabilities similar to other entities in the system.”
- Les applications sont réparties sur l’ensemble des peers
 - Pas de machine « maître », pas de serveur central
- peer-to-peer hybride
 - Lorsque certains éléments sont mobiles

Peer-to-peer vs. Client / Serveur

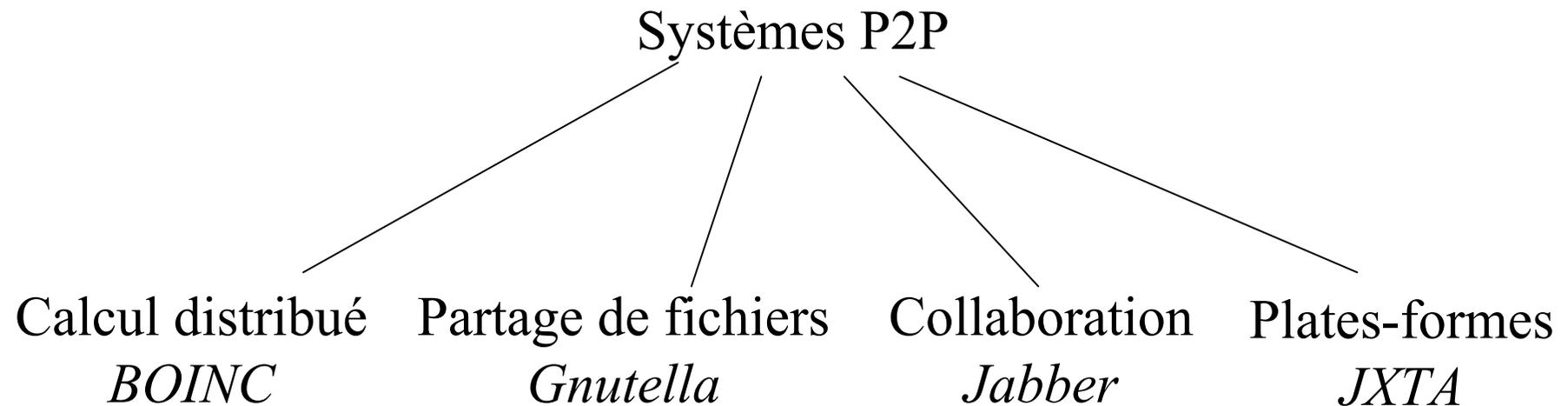
- Client / Serveur
 - Certains éléments sont fournisseurs de services (serveurs)
 - D'autres sont consommateurs (clients)
- Peer-to-peer
 - Chaque élément est à la fois client et serveur

Architecture peer-to-peer

- Les ressources d'un peer sont similaires à celles des autres participants
- Les peers communiquent directement les uns avec les autres et partagent des ressources



Taxonomie d'applications P2P



BOINC

- Berkeley Open Infrastructure for Network Computing
- plate-forme logicielle pour le calcul distribué utilisant les ressources matérielles de volontaires
 - pour des applications dont la tâche peut être décomposée en étapes indépendantes, et dont le temps de calcul local reste important par rapport au transfert des données locales sur Internet
 - exemples : SETI@home, climateprediction.net...
 - chaque projet a un ou n serveurs
 - chaque volontaire installe un client, et s'abonne à 1 ou n projets. Il définit les ressources associées à chaque projet
 - de nombreuses problématiques similaires aux réseaux mobiles : inclusion/exclusion de volontaires, déconnexions, faible puissance des clients...

<http://boinc.berkeley.edu/>

Gnutella

- Partage de fichiers en P2P
 - Installation d'un logiciel dit « client » qui sert à la fois de serveur et de client.
 - Choix de la partie du disque partagée sur le réseau
- Recherche d'une ressource par son nom
 - en local
 - demandée à un noeud connu
 - s'il ne la connaît pas, le noeud la propage à 4 « alliés »
 - etc... jusqu'à 7 hops max.
 - le noeud qui a la ressource la retourne à l'appelant et fait désormais partie des noeuds connus par l'appelant

<http://www.gnutella.com/>

Jabber

- Face émergée de l'iceberg
 - alternative Linux de messagerie instantanée comme ICQ
- Face immergée : protocoles de streaming fondés sur XML
 - permet à 2 entités sur Internet d'échanger des messages et toute sorte d'information structurée en temps presque réel.
 - Standard (Internet Engineering Task Force IETF)
 - Décentralisé : toute machine peut être serveur et/ou client
 - Sécurisé avec SASL et TLS
 - Ouvert, extensible grâce à XML
 - Flexible : outils de collaboration, syndication de contenu, partage de fichiers, gestion de systèmes à distance...

<http://www.jabber.com>

JXTA (juxtapose)

- Ensemble de protocoles permettant à tout type de terminal de participer à un réseau P2P
 - fournit aussi une implantation ouverte
- les protocoles standardisent la façon dont les peers :
 - se découvrent les uns les autres
 - s'organisent en groupes
 - communiquent entre eux de façon sécurisée
 - annoncent et découvrent des services
 - se supervisent

<http://www.jxta.org>

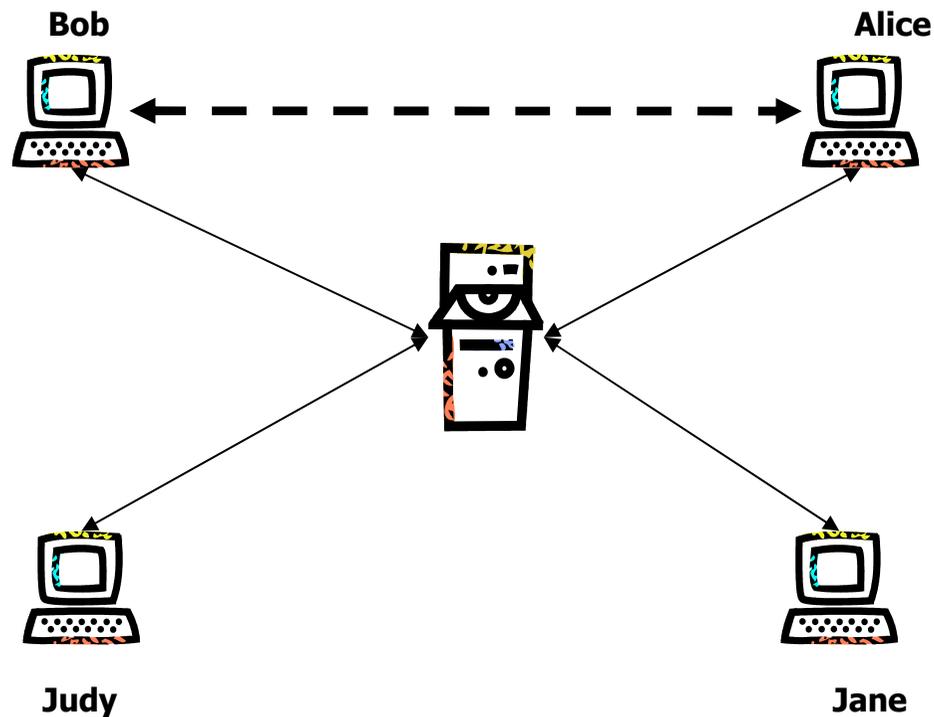
Challenges

- Découverte de peers et gestion des groupes
- Localisation et positionnement des ressources
- Temps de recherche d'une ressource
- Etat de chaque noeud
- Utilisation de la bande passante
- Sécurité, fiabilité
- Tolérance aux fautes

Approches

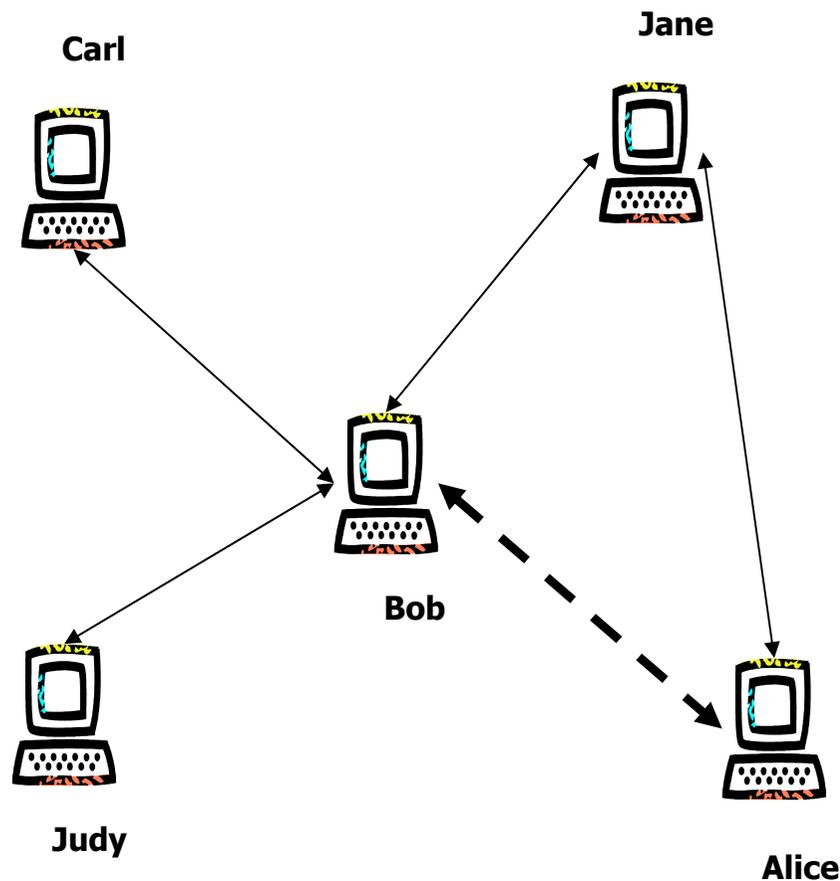
- Centralisée
- Flooding
- Document Routing

Centralisée, pseudo-P2P



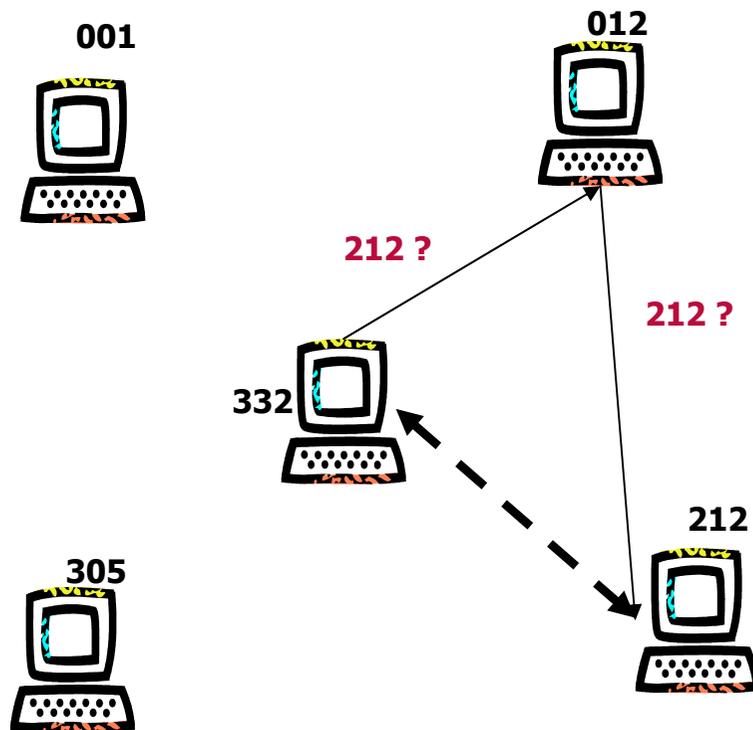
- Passage obligé par un point central
- Intérêts
 - Recherche efficace
 - Bande passante utile faible
- Inconvénients
 - point central problématique
 - Convient à faible échelle seulement

Flooding



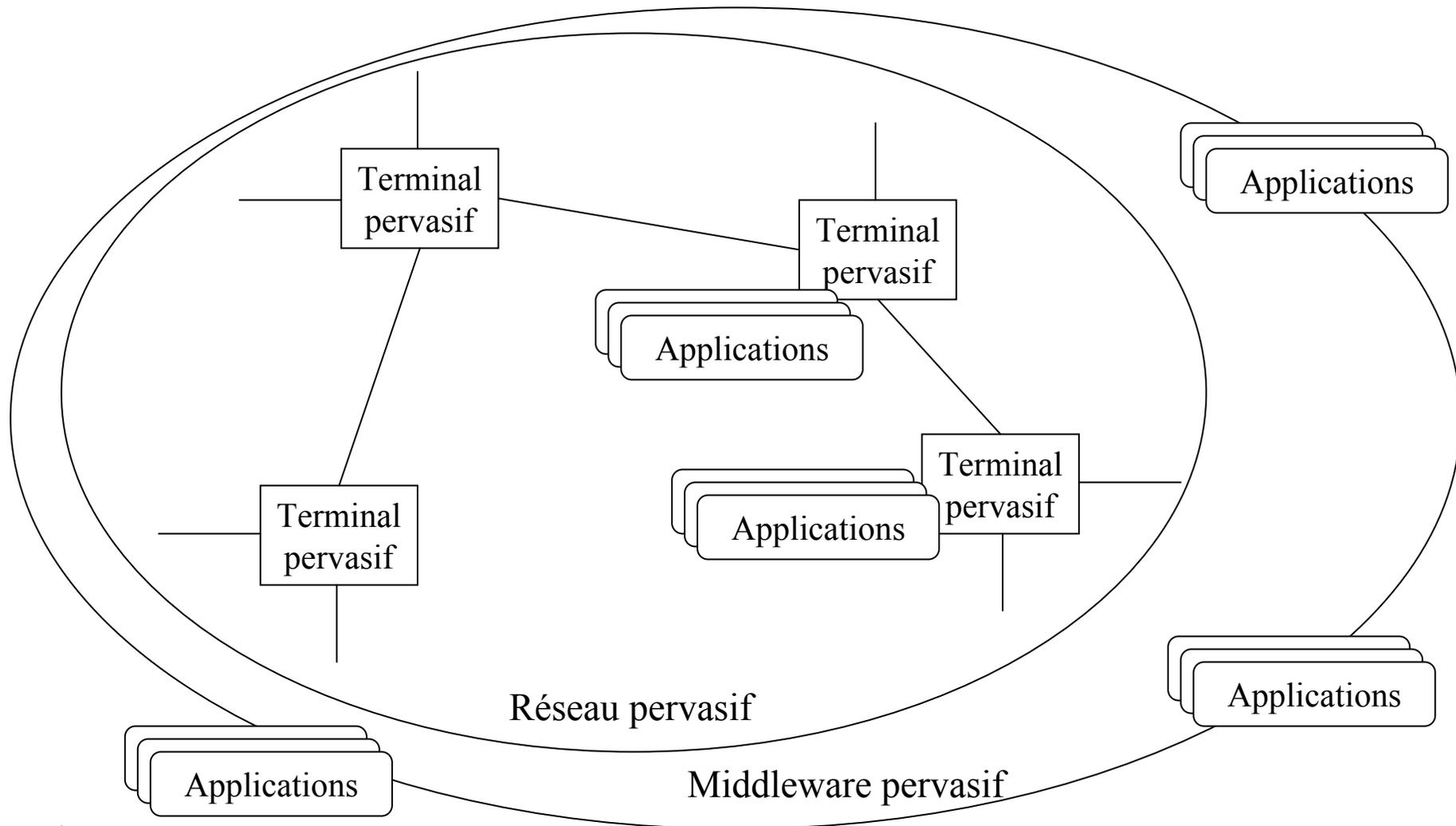
- Une demande est envoyée dans toutes les directions
- Intérêts
 - Pas de point central
- Inconvénients
 - Recherches lentes
 - Utilisation intensive de la bande passante

Document Routing



- Le peer demandeur prévoit le routage pour atteindre le peer cible
- Intérêts
 - Bande passante utile faible
- Inconvénients
 - Tolérance aux fautes limitée
=> redondance

Synthèse : framework de système pervasif



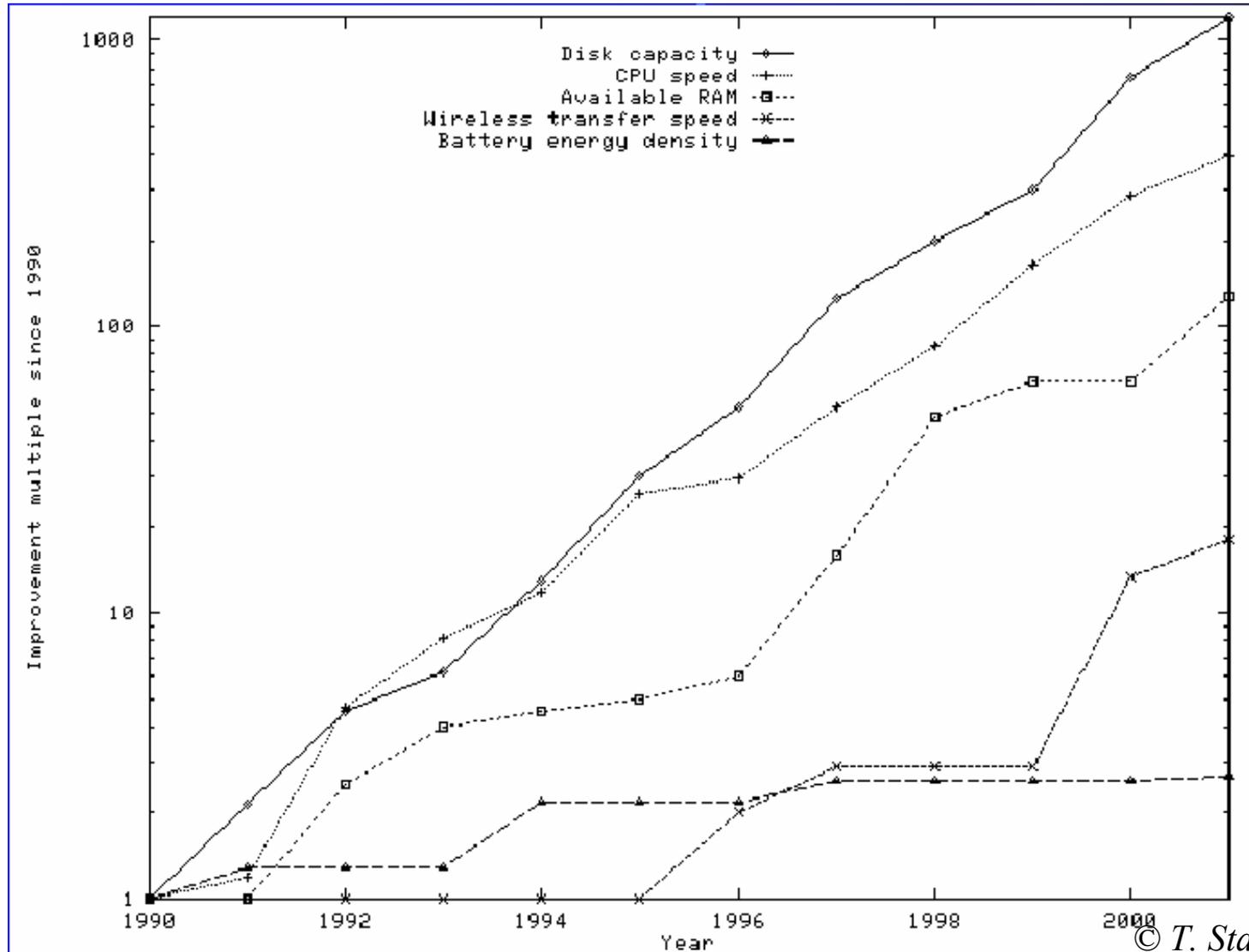
Trois niveaux de recherche en systèmes pervasifs

- 1- niveau matériel et système (survol)
- 2- niveau réseau (survol)
- 3- niveau système d'information
- 4- synthèse : relations entre les niveaux

Niveau matériel et système (survol)

- Limitations des ressources disponibles sur les terminaux légers
 - Critères primordiaux : poids et taille des terminaux
 - Contraintes sur tous les composants matériels
 - Le plus limitant : l'énergie (batteries)
 - Juste après : gestion de la chaleur
- Ces contraintes ont des répercussions sur
 - Le système d'exploitation
 - Les logiciels (voir niveau système d'information)

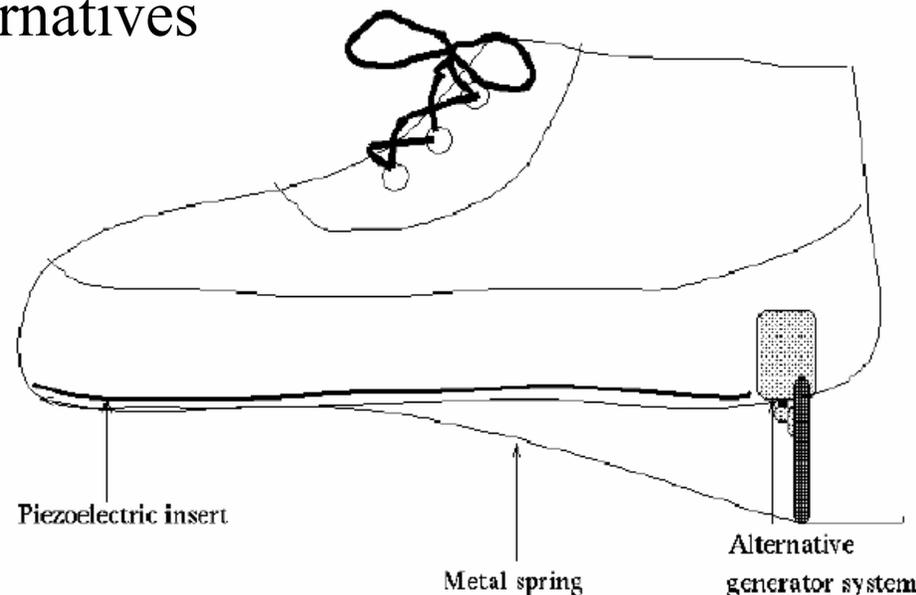
Évolution des capacités matérielles



© T. Starner, 2002

Recherches au niveau matériel et système

- Exemples pour la gestion de l'énergie
 - Adaptation dépendante de l'énergie
 - Ordonnancement du processeur à vitesse variable
 - Gestion mémoire dépendante de l'énergie
 - Energies alternatives



(Kymissis98 ISWC "Parasitic Power Harvesting in Shoes")

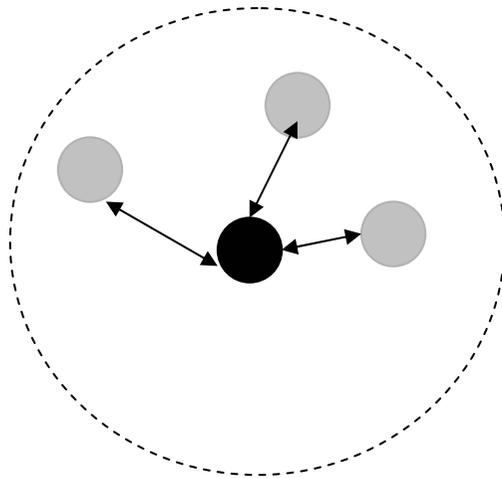
Niveau réseau (survol)

- Organisation des nœuds du réseau
 - Réseaux ad hoc mobiles
- Routage
 - Routage proactif
 - Routage réactif
- Multicast dans des réseaux mobiles

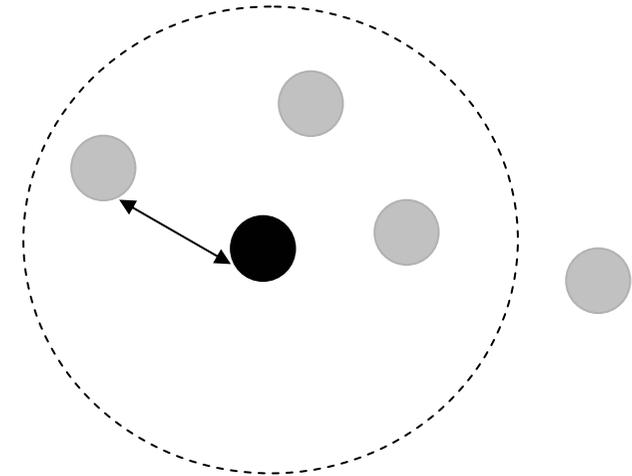
Réseaux ad hoc mobiles

- Pour former rapidement et simplement des liaisons entre appareils mobiles sans l'intermédiaire de structures fixes
 - Gestion du réseau répartie
 - Réseaux dynamiques (un nœud peut le rejoindre ou le quitter à tout moment) sans erreurs ou dysfonctionnements
- Nœud entrant = broadcast sur son rayon d'action
- Communication avec un autre nœud
 - Si dans le rayon d'action, comm directe (single-hop)
 - Sinon, routage (multi-hop)

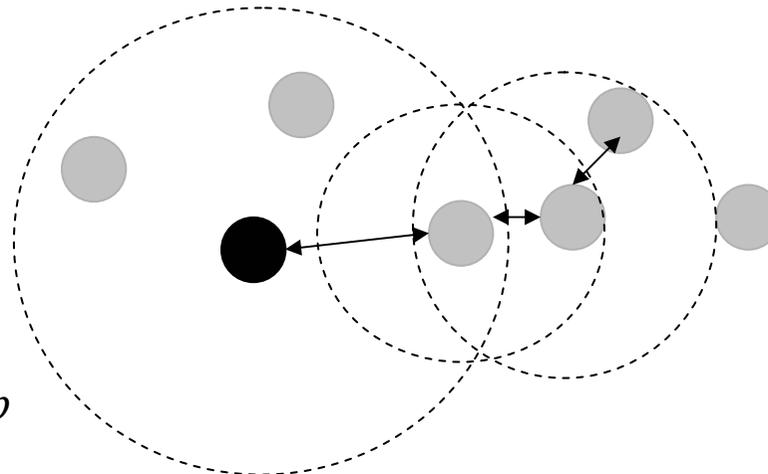
Exemples illustratifs



Broadcast à l'entrée dans le réseau



Communication single-hop



Communication multi-hop

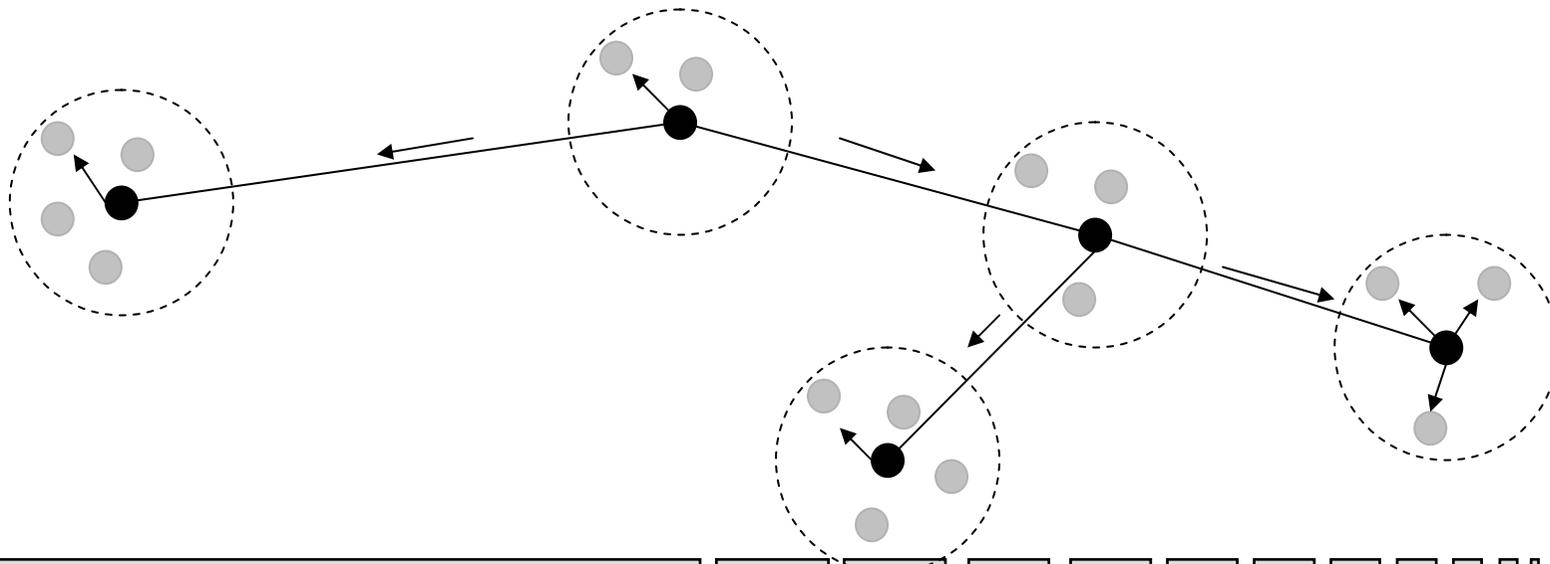
Routage

- Routage proactif
 - Apprend continuellement la cartographie du réseau en échangeant des informations sur la topologie
 - Gourmand en bande passante
- Routage réactif
 - Route calculée sur besoin par inondation de requêtes de chemins – choix de la route optimale parmi les réponses obtenues
 - Cache contenant les routes les plus récentes
 - Pb si réseau chargé
- Routage hybride
 - proactif en local et réactif à distance

Cas du multicast

- Pour atteindre plusieurs cibles en même temps
 - Exemple : téléconférence à plus de 2
- Pb : pas de routeurs => les nœuds sont routeurs
- Exemple TMTP (tree-based multicast transport protocol)
 - Découpage topologique du réseau en domaines
 - Un nœud par domaine se déclare gestionnaire de domaine (il crée le domaine – et le détruit lorsqu’il quitte le réseau)
 - Les nœuds les plus proches intègrent son domaine (par broadcast)

- (suite TMTP...)
 - L'échange de données inter-domaines se fait via les gestionnaire de domaine
 - Dissémination d'un message : dans un arbre dont la racine est l'émetteur, les nœuds internes sont des gestionnaires de domaine, les feuilles sont des membres des domaines.



Niveau système d'information

- De très nombreuses recherches en cours
- Sur des domaines très divers
- Synthèse difficile
 - => survol de certains points
 - => approfondissement d'autres

1- Découverte de services

- Un échange de données sous-entend de connaître la localisation des services fournissant les données
- Les questions
 - Comment faire connaître les services que l'on propose?
 - Comment trouver un service dont on a besoin?
 - Doit-on répartir ou centraliser la connaissances sur les services?
- Approches
 - Annuaire de services
 - Inondation
 - Routage sémantique

Découverte par annuaires de services

- Ensemble organisé de services disponibles
- Certaines machines du réseau fournissent un service d'annuaire
- Exemples
 - SLP
 - Jini
 - Salutation

Annuaire de services SLP

- Service Location Protocol
 - à base d'agents
- Service Agent
 - Représentant d'un service
- Directory Agent
 - Enregistre les SA dans un annuaire LDAP
 - Publicité multicast pour se faire connaître
 - Intermédiaire entre SA et UA pour la découverte
- User Agent
 - Représentant du client pour la découverte d'un service
 - Émet une requête multicast avec URI du service recherché

Annuaire de services Jini

- Jini
 - Promu par Sun, technologie Java
- Arrivée d'un nœud fournisseur de services
 - Le nœud broadcast son identité et son groupe
 - L'annuaire Jini lui répond avec une interface RMI
 - Le nœud envoie sur cette interface la déclaration de ses services
 - L'annuaire enregistre les services du nœud avec une durée de vie, ainsi qu'un proxy d'accès pour chaque service
 - Le nœud doit renouveler la déclaration de ses services avant la fin de durée de vie

Annuaire de services Jini

- Accéder à un service
 - Le nœud demandeur envoie une demande en multicast
 - Les annuaires recevant la demande répondent le cas échéant en fournissant la localisation et le proxy d'accès
 - Le nœud demandeur accède au service par l'intermédiaire du proxy
 - Les informations sur le service peuvent être mises en cache dans le nœud demandeur pour un accès ultérieur
- Technique de recherche par lookup
 - Les nœuds possédant l'information sont connus

Annuaire de services Salutation

- Annuaire plus réparti que Jini
 - Chaque nœud contient un agent annuaire qui enregistre un sous-ensemble des services disponibles
 - Ajout d'un service
 - inscription dans l'agent annuaire local
 - Et inscription dans les agents annuaires des nœuds voisins
 - Recherche d'un service
 - Recherche d'abord auprès de l'agent local
 - Puis recherche par broadcast le cas échéant
- Technique de recherche par découverte
 - Les nœuds possédant l'information ne sont pas connus

Bluetooth service discovery protocol

- Permet à un système Bluetooth de découvrir les services de son environnement
- fournisseur de service = sdp server
 - chaque service est décrit par un service record (liste de description des attributs du service)
 - les services sont regroupés en classes, une classe a un attribut identifiant universel (UUID)
- 2 modes :
- **recherche** d'un service par UUID
- **navigation** dans l'ensemble des services d'un serveur via la liste des UUID offerts

Découverte par inondation

- Chaque noeud enregistre sa liste de services disponibles
- Exemple : uPnP (consortium industriel)
 - ajout de noeuds avec zero configuration, communication, découverte automatique de services
 - Arrivée d'un nouveau nœud
 - Il demande une adresse IP (DHCP par exemple)
 - Il envoie par multicast un message XML fournissant la déclaration de son arrivée (ANNOUNCE) et la description des services qu'il fournit (OPTIONS).
 - Les réceptionnaires répondent au nouveau nœud pour qu'il ait connaissance de leur existence

Découverte par routage sémantique

- Choisir « intelligemment » les nœuds qui permettent d'atteindre le service requis.

- Allia
 - Olga Ratsimor, et al. Allia: Alliance-based Service Discovery for Ad-Hoc Environments, ACM Mobile Commerce Workshop, Sept 2002

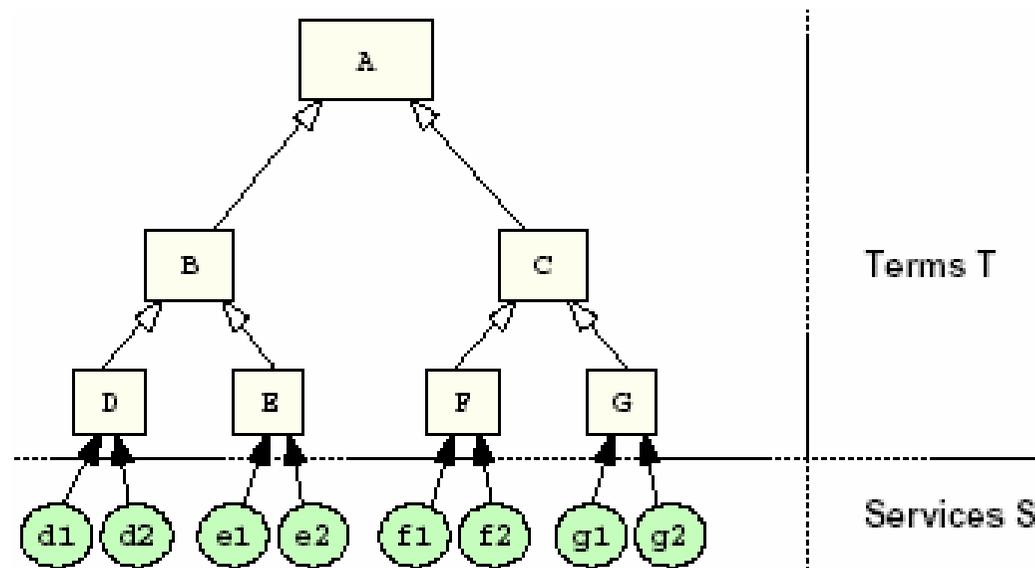
- Clusters multi-couche
 - Michael Klein, Birgitta König-Ries: Multi-Layer Clusters in Ad-hoc Networks - An Approach to Service Discovery. In: Proc. International Workshop on Peer-to-Peer Computing, Pisa, Italy, 2002.

Allia : Alliance-based service discovery

- Chaque peer
 - fournit un annuaire des services locaux
 - annonce les services qu’il fournit à ses voisins (multicast)
 - sélectionne parmi les annonces reçues celles qu’il gère dans son cache (avec une politique personnelle)
 - les peers dont des services sont mis en cache font partie de l’alliance du peer gérant le cache
 - un peer retire de son alliance tout peer qui « disparaît »
- Recherche d’un service
 - dans son annuaire local, puis dans son cache
 - puis demande à ses voisins (le voisin cherche en local ou transmet la demande si sa politique personnelle l’autorise)

Clusters multi-couche

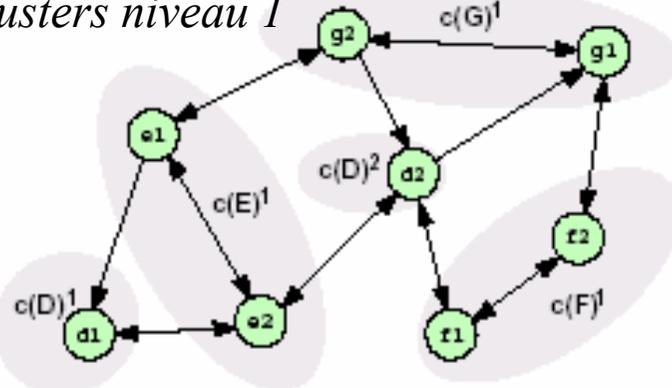
- Regroupement des peers en fonction de la proximité
 - géographique : en liaison directe (single hop)
 - sémantique : fournissent des services similaires
 - fondée sur une ontologie arborescente de termes



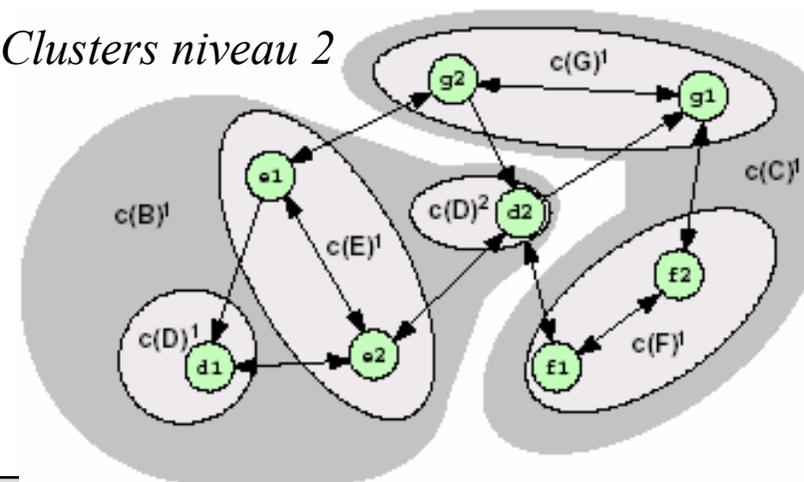
Clusters multi-couche

- deux peers sont dans le même cluster feuille (niveau 1) s'ils appartiennent au même terme et s'ils sont en liaison directe
- cluster niveau 2 : même terme de niveau 2 et géographiquement « atteignables »
 - atteignables : 2 nœuds des 2 clusters sont en relation directe
 - 3 hop max entre 2 nœuds du même cluster de niveau 2

Clusters niveau 1



Clusters niveau 2



Objets mobiles

- 3 façons pour transmettre un message à un objet mobile
 - serveur de localisation
 - un serveur stocke les couples objet/localisation
 - chaque objet informe lui-même le serveur de sa migration
 - poste restante
 - un message est envoyé à un intermédiaire fixe
 - l'objet mobile interroge l'intermédiaire pour prendre les messages
 - répéteurs
 - un objet quittant un site laisse derrière lui un répéteur
 - le répéteur fait suivre les messages jusqu'à la nouvelle position
 - Plusieurs migrations => une chaîne de répéteurs

Gestion de données en environnement pervasif

[Franklin 2001] Challenges pervasifs et besoins induits

- Adaptabilité et interaction utilisateur
 - Interaction temps-réel avec les données distantes
 - Flexibilité
- Mobilité
 - Délivrer et recevoir des données pendant le déplacement
 - Tolérer les déconnexions sans interrompre le service
 - Préparation des données qui seront utiles => prédiction des besoins
 - Location-awareness
 - Sélectionner les données en fonction de la localisation
 - Structures de données spécifiques à la localisation

Gestion de données en environnement pervasif

- Context-awareness
 - Analyser des flux de données provenant de capteurs temps réel
 - Passer d'une information brute d'environnement à une donnée de contexte qui a du sens pour l'application : interpréter les analyses
- Collaboration
 - Gestion de groupes de personnes dynamiques / ad hoc
 - Gestion de synchronisation et cohérence
 - Actions collaboratives de création, accès, modification de données partagées par le groupe
 - Gestion d'historique, log

Gestion de données en environnement pervasif

Exemples de projets

- **Projet Data Recharging**
 - Exploiter le profil de l'utilisateur pour délivrer automatiquement des mises à jour de données et nouvelles données pertinentes
 - Profil utilisateur = description des données pertinentes + priorités
- **Projet Telegraph**
 - Architecture adaptative de traitement de flux de données dans des environnements très dynamiques
 - Le plan de traitement du flux est dynamique (adaptation aux fluctuations telles que énergie et bande passante)

Gestion de données en environnement pervasif

- Exemple JavAne
 - JP Arcangeli & al. Development of flexible peer-to-peer information systems using adaptable mobile agents, DEXA 04 Workshop GLOBE (Grid and Peer-to-peer computing impacts on large scale heterogeneous distributed database systems)
- Gestion de réplicas
 - DEA de J. Gossa, encadré par J. M. Pierson et L. Brunie, LIRIS, 2004
- Exemple MoGATU
 - F. Perish et al. On data management in pervasive computing environments IEEE transactions on knowledge and data engineering, vol 16 n°5 may 2004

JavAne

- Agents mobiles pour le partage et la recherche de documents en environnement peer-to-peer
- Rôle serveur
 - Publier, gérer et partager des documents
- Rôle client
 - Rechercher et récupérer des documents
 - Stocker des informations sur les serveurs qu'il connaît
 - Naviguer de site en site
 - Des agents mobiles qui interrogent la base de chaque site

Basé sur JavAct : plate-forme de gestion d'agents mobiles adaptables

JavAne – code simplifié d'un agent chercheur

```
public class SearchBehavior extends JavAneBehavior implements Searcher,
    javact.util.StandAlone {
public SearchBehavior(client, clientInfo, energy, queryStamp, Params) {
...}
public void run() {
    if (energy <= 0) {suicide();return;}
    // Has the current place been already visited ?
    if (! markedPlace(myPlace(), queryStamp, client)) energy -= E_VISITED;
    else { // Database interrogation and sending of results to the client
        localResults = LocalDB.fileSelect(Params);
        if (localResults != null && localResults.length > 0) {
            send(new JAMprocessResults(place, localResults), client);
            energy -= E_FOUND;
        } else energy -= E_NOT_FOUND;
    }
    .....
```

JavAne – code simplifié d'un agent chercheur

```

if (energy > 0) { // Random moving (if it remains energy)
    String[] places = getPlaces(myPlace());
    String nextPlace = places[random.nextInt(places.length)];
    go(nextPlace);
} else suicide();
// Collection of meta-information
String[] placesInfos = getPlacesInfos(myPlace());
send(new JAMaddPlacesInfos(placesInfos), clientInfo);
}
}

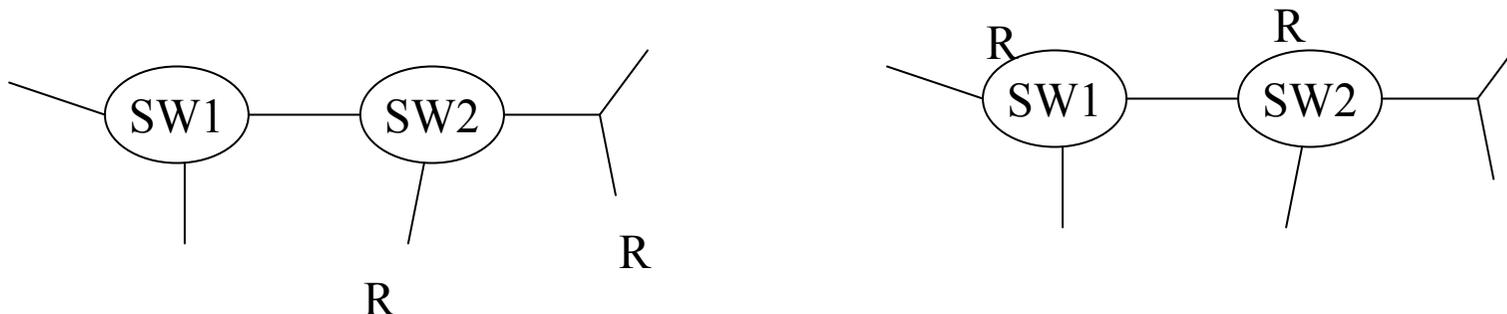
```

Gestion de caches sur le web

- Proxy-caches collaboratifs
 - une donnée est dupliquée sur le proxy qui la demande
 - gourmand et non « optimisé »
- Content Delivery Network
 - gestion de données avec distribution à très grande échelle
 - architecture fixe tentaculaire très coûteuse
- Push-caching
 - les serveurs « poussent » les réplicas vers les proxy de leur choix en fonction d'une politique qui leur est propre
 - le client s'adresse au fournisseur qui redirige vers le cache le plus adéquat

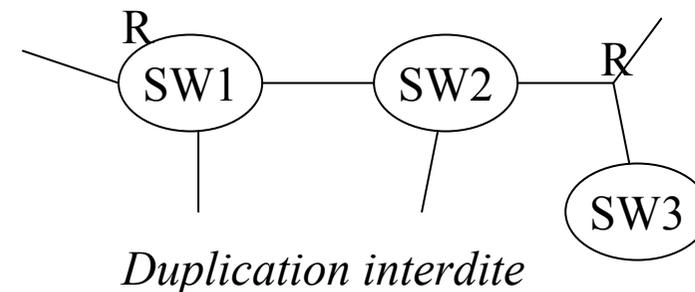
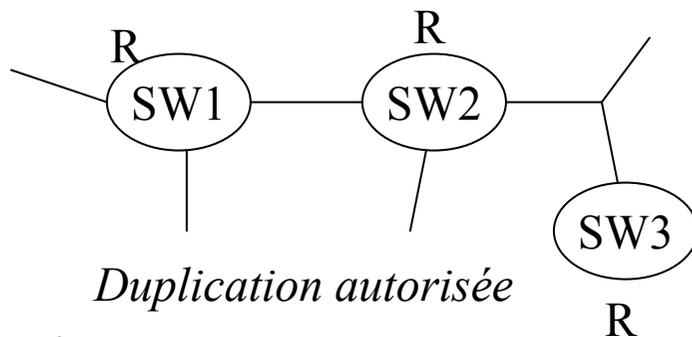
(Dé)placement de réplicas en environnement pervasif

- basé sur la théorie des small worlds (groupes d'intérêt)
 - un réplica doit se rapprocher des small worlds qui s'intéressent à lui
- A chaque demande d'une donnée, les réplicas se déplacent vers le foyer de demande



(Dé)placement de réplicas en environnement pervasif

- Apparition d'un nouveau groupe d'intérêt
 - si autorisé, duplication du réplica et déplacement du nouveau réplica vers le nouveau groupe
 - sinon, déplacement des réplicas existants jusqu'à équilibre entre les 2 groupes d'intérêt
- Disparition du groupe d'intérêt
 - déplacement ou suppression du réplica



(Dé)placement de réplicas en environnement pervasif

- Le (dé)placement ne se fait qu'avec une connaissance partielle du réseau
 - chaque peer ne connaît que les nœuds les plus proches!
 - Notion de vecteur d'attraction d'un réplica sur chaque lien sortant du cache sur lequel il est hébergé
 - augmenté à chaque demande provenant de ce lien
 - Diminué avec le temps
 - Déplacement effectif ou duplication
 - lorsque le vecteur d'attraction dépasse la distance entre le proxy hébergeant le réplica et le nœud contenant le proxy destination
 - Destruction du réplica quand le vecteur d'attraction reste nul pendant une durée seuil

MoGATU

- Plate-forme de gestion de données en P2P pur
 - une couche de gestion des données
 - une couche de communication
- Sur chaque peer, 3 types d'entités
 - Information provider : source de données
 - Information consumer : entités qui interrogent et mettent à jour les données
 - Information manager : informations sur les peers voisins, intermédiaire entre provider et consumer, gestion de cache en fonction d'un profil

MoGATU – information provider

- Gère et fournit une interface d'accès à une sous-partie locale des données
- Décrit son service à l'aide d'une ontologie
- Communique uniquement avec l'information manager local, qui « route » les messages entre ce provider et les autres peers
- Se déclare régulièrement auprès de l'information manager local, qui lui-même transmet la déclaration aux peers de l'entourage

MoGATU – information consumer

- Utilisateurs finaux ou autres agents
- S'enregistrent auprès de l'information manager local (pas d'annonce aux peers de l'entourage)
- Envoie ses requêtes à l'information manager local
- L'information manager transmet la demande aux information providers locaux ou distants correspondants, et récupère les réponses

MoGATU – information manager

- Communication réseau
 - Analyse des messages
 - Découverte et routage des messages en multi-hop
 - Requêtes proactives sur les autres peers
 - Broadcast
 - des annonces des providers (locaux et/ou voisins)
 - des requêtes des consumers locaux
- Maintenance d'informations
 - Informations sur providers et consumers locaux
 - Durée de vie, services, restrictions sur les requêtes
 - Informations sur peers de l'entourage
 - Id, services fournis, types d'informations fournies (leurs annonces)

MoGATU – information manager

- Profil utilisateur
 - Préférences et besoins
- Cache
 - réponses aux requêtes passées fournies par les peers de l’entourage
 - Choix de mettre en cache en fonction du profil utilisateur et des capacités du peer
- 4 niveaux d’information manager selon capacité
 - 1 : aucun cache
 - 2 : cache les annonces distantes
 - 3 : cache les annonces et les réponses aux requêtes passées
 - 4 : cache annonces et réponses et les rend accessibles aux autres peers

Généralisation aux caches en général

- Systèmes distribués de proxy-caches
 - ensemble de services interposées entre le « client » et le « fournisseur » d'un service effectuant un travail à la place du fournisseur ou à la place du client
 - adaptation de contenu,
 - délestage des serveurs,
 - optimisation de la sécurité,
 - traductions inter-protocoles,
 - stockage de copies de données (réplicas)...

Context-awareness

- M. Weiser
 - «The most profound technologies are those that disappear»
- Informatique context-aware
 - Prise en compte de l'environnement dans lequel se trouve l'utilisateur
 - Capture et accès automatique
 - Délivrer l'information pertinente
 - Quand?
 - Où?
 - Comment?

NB : Cette partie III-3-3 est largement tirée des travaux de Dey

Définition du contexte

[Salber,Dey,Abowd 99]

- « Environmental information or context covers information that is part of an application's operating environment and that can be sensed by the application. This typically includes the location, identity, activity and state of people, groups and objects. »

Définition du contexte

- Ce qui est intéressant
 - Le concepteur de l'application décide quelles informations contextuelles sont intéressantes
 - Dépend de l'application
- Le contexte
 - est souvent constitué d'un ensemble d'informations implicites auxquelles l'application n'a pas accès
 - rend les applications plus « smart »

Application context-aware

- « Context-aware applications sense context information and modify their behavior accordingly without explicit user intervention »
- 3 types de comportement
 - Afficher le contexte
 - Automatiquement exécuter ou adapter des services
 - Enregistrer et « indexer » l'information de contexte pour la retrouver au besoin

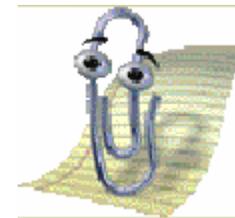
Afficher le contexte

- Exemples
 - Tableau des entrées et sorties du personnel
 - Cartes de localisation
 - Écrans d'état (météo, activité...)

FEL In/Out Board	
Gregory Abowd (Out 10:50 am)	Jen Mankoff (In 12:08 pm)
Jason Brotherton (In 9:28 am)	David Nguyen (In 11:09 am)
Anind Dey (In 12:08 pm)	Rob Orr (Out 1:25 pm)
M. Futekawa (In 12:00 pm)	Maria Pimentel (Out 5:54 pm)
Y. Ishiguro (Out 10:52 am)	Daniel Salber (In 10:14 am)
Rob Kooper (Out 5:26 pm)	Brad Singletary (Out 2:50 pm)
Kent Lyons (Out 12:27 pm)	Khai Truong (Out 1:25 pm)

Exécuter / adapter des services

- Sélectionner et exécuter le service pertinent parmi un ensemble de services
- Modifier le comportement d'un service ou son exécution
- Exemple : imprimer sur l'imprimante la plus proche, afficher de la documentation...



Enregistrer et « indexer » le contexte

- Utiliser le contexte dans le cas de recherches fondées sur le contexte
- Exemples
 - Forget-me-not
 - Assistant de conférence
 - Guides touristiques
 - Historique (revoir des informations « cochées » lors de la réunion, lors de la visite)

Conception d'une application context-aware

- Sans support
 - Spécifications
 - Acquisition et représentation
 - Diffusion
 - Réception et stockage
 - Action
- Avec support

Conception sans support

- Spécifications
 - Quel contexte utiliser
 - Quels comportements
- Acquisition et représentation
 - Installer les capteurs utiles
 - Créer ou apprendre les API des capteurs
 - Créer les modules d'acquisition des données de capteurs ou les modules de transmission des modifications de valeurs des capteurs
 - Stocker le contexte
 - Interpréter / abstraire le contexte

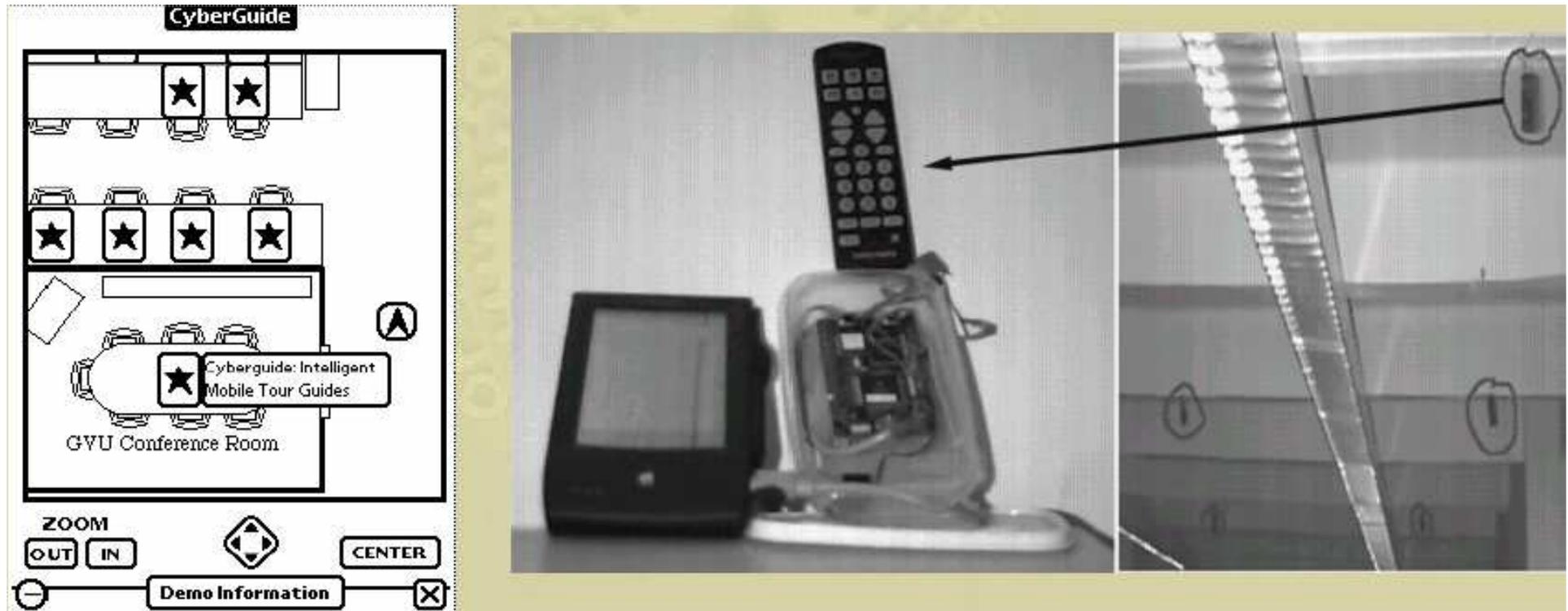
Conception sans support

- Diffusion
 - Mécanisme de transport de l'information des capteurs vers les services
 - Capteurs sources d'information nombreux et hétérogènes
 - Distribution à des services nombreux et hétérogènes
- Réception et stockage
 - Requêtes ou notifications?
 - Stocker tout le contexte? Durée de vie du contexte?
 - Interprétation des données brutes
- Action
 - Combiner le contexte reçu avec les anciennes valeurs
 - Effectuer une action en fonction des résultats de l'analyse

Exemple de conception sans support

- Visite guidée mobile
- Application:
 - Afficher la liste des démos non vues
 - Surligner les démos pertinentes
 - Afficher une carte centrée sur la localisation de l'utilisateur
 - Afficher de l'information sur la démo la plus proche
 - Application mono-utilisateur

Visite guidée mobile



Visite guidée mobile

- Spécifications
 - Contexte
 - Liste des démos faites
 - Localisation de l'utilisateur
 - Orientation de l'utilisateur
 - Comportement de l'application
 - Cf. transparent précédent

Visite guidée mobile

- Acquisition
 - Capteurs
 - Orientation : compas électronique
 - Localisation : badge actif
 - Liste des démos : badge actif + application
 - API
 - à construire pour orientation et liste de démos
 - À apprendre pour badge actif
 - Stockage des données dans un fichier local
 - Interprétation de la localisation pour savoir si près d'une démo

Visite guidée mobile

- Diffusion
 - Seul le badge actif est sans fil
 - Écrire la couche transport (RPC par exemple) pour acquérir les données
- Réception
 - Notification des changements
 - Interprétation pour recevoir la liste des démos
 - Ensemble de capteurs fini

Visite guidée mobile

- Action
 - Modification d'orientation
 - Modification de la carte
 - Modification de la localisation
 - Modification de la carte
 - Mise à jour de la liste des démos non vues
 - Surligner les démos pertinentes

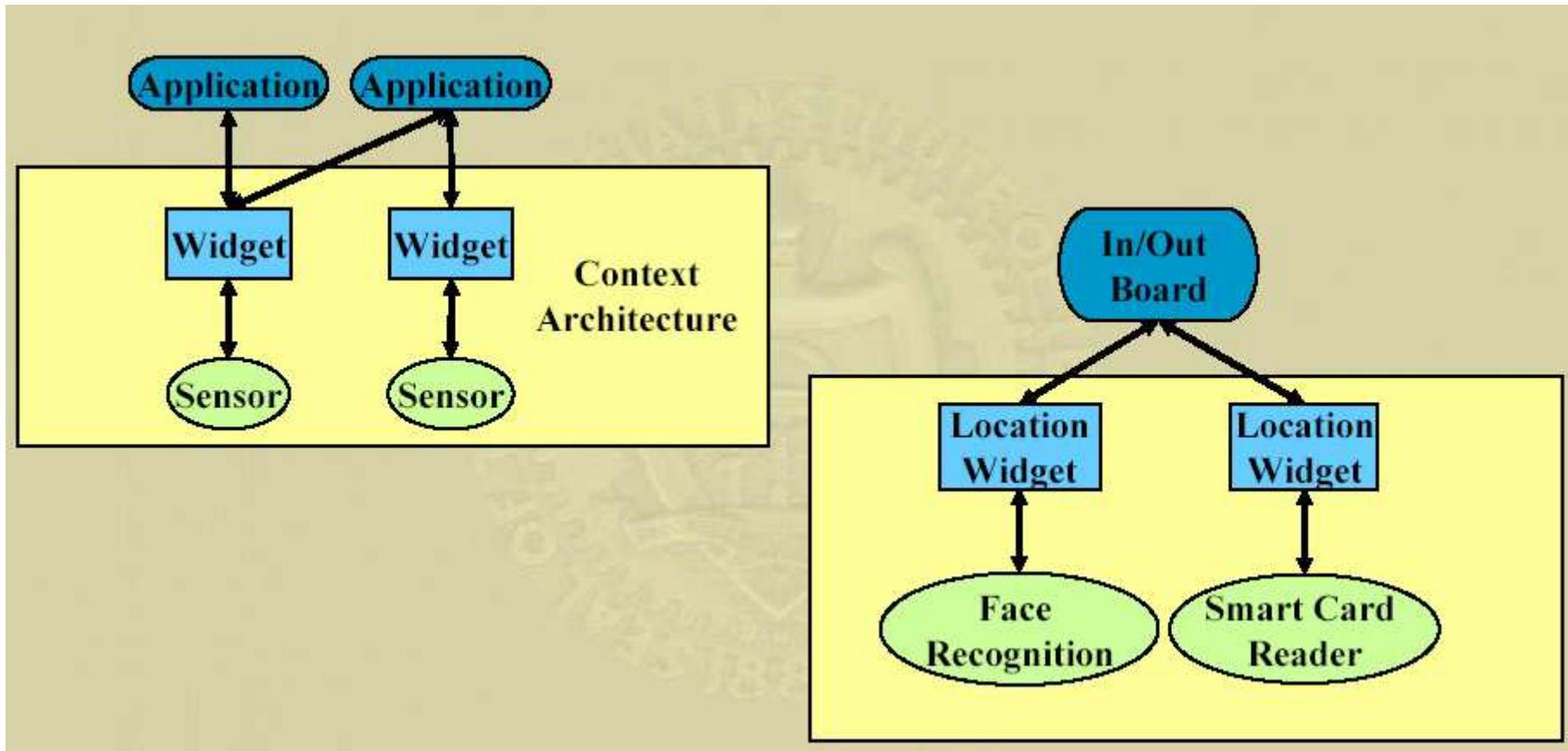
Conception d'une application context-aware

- Sans support vs. avec support
- Nécessité d'un support pour:
 - Spécifications : Spécifier le contexte nécessaire
 - Découverte : localiser les composants qui peuvent acquérir du contexte et agir en fonction du contexte
 - Séparation des problèmes : séparer l'acquisition du contexte de son utilisation
 - Stockage : importance des historiques de contexte
 - Interprétation : pour des informations de plus haut niveau
 - Communications transparentes
 - Disponibilité permanente du contexte

Briques de base de la plate-forme de Dey

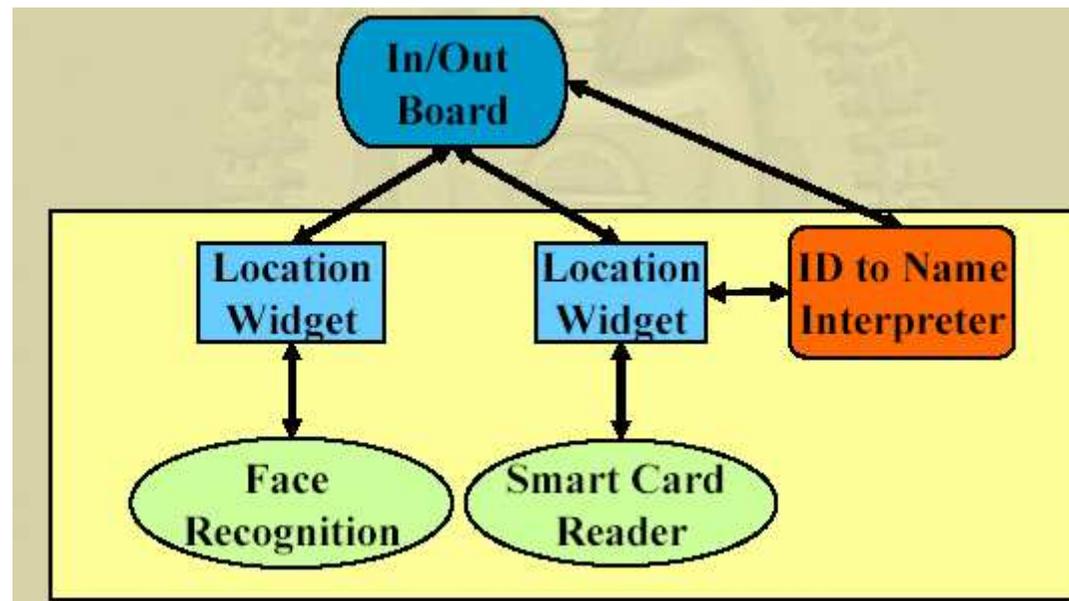
- Context Widgets
 - Par analogie avec les widgets des interfaces graphiques
 - Encapsulation, abstraction
 - But
 - Acquérir et abstraire les données des capteurs
 - Séparer les problématiques
 - Stockage

Context widgets



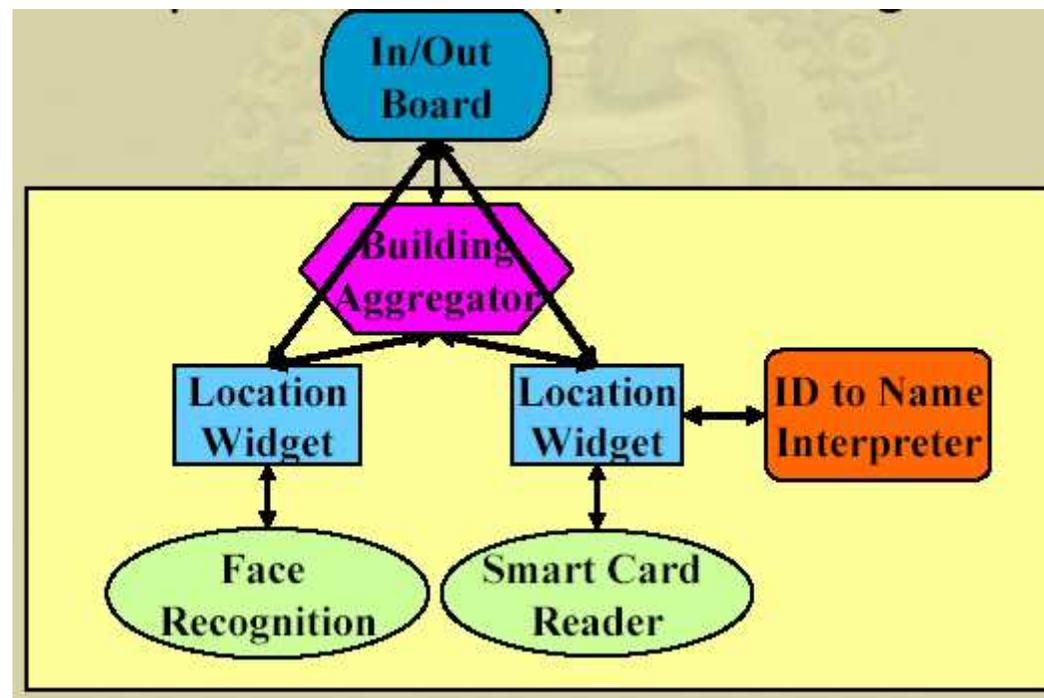
Interpretors

- Les données des capteurs sont rarement fournies au bon niveau d'abstraction
- Convertissent ou interprètent le contexte pour obtenir une information de plus haut niveau



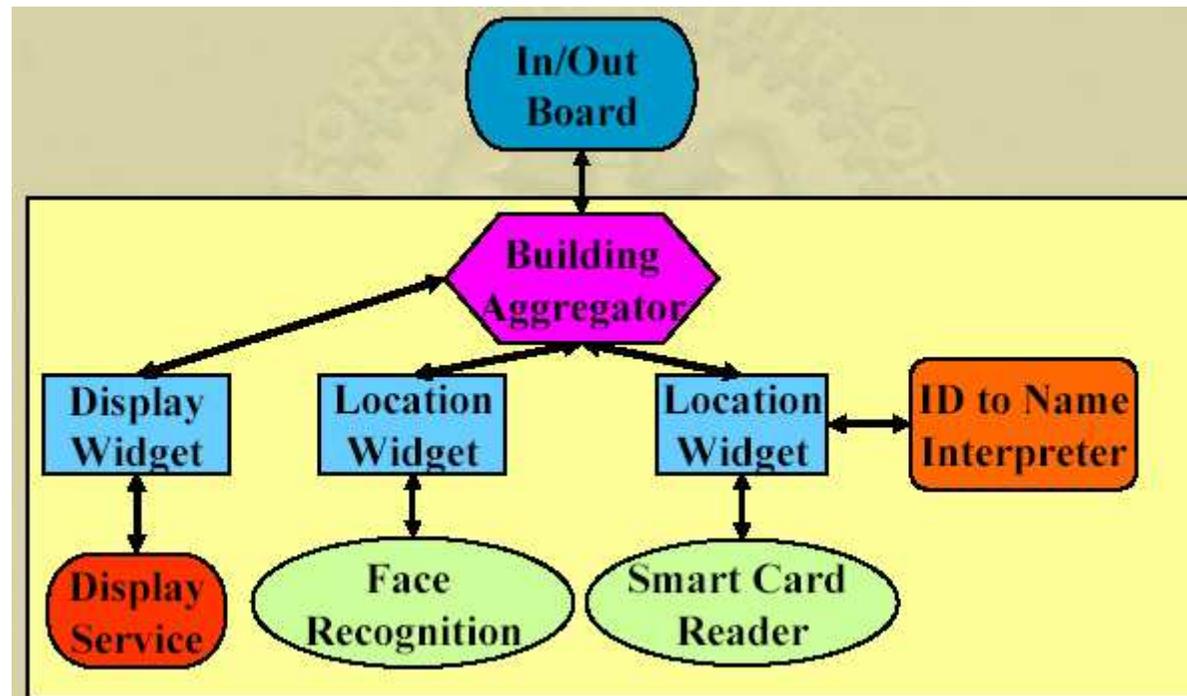
Agrégators

- Rassemble des données de contexte pour les rendre pertinentes pour certaines entités
- Simplifie la conception de l'application



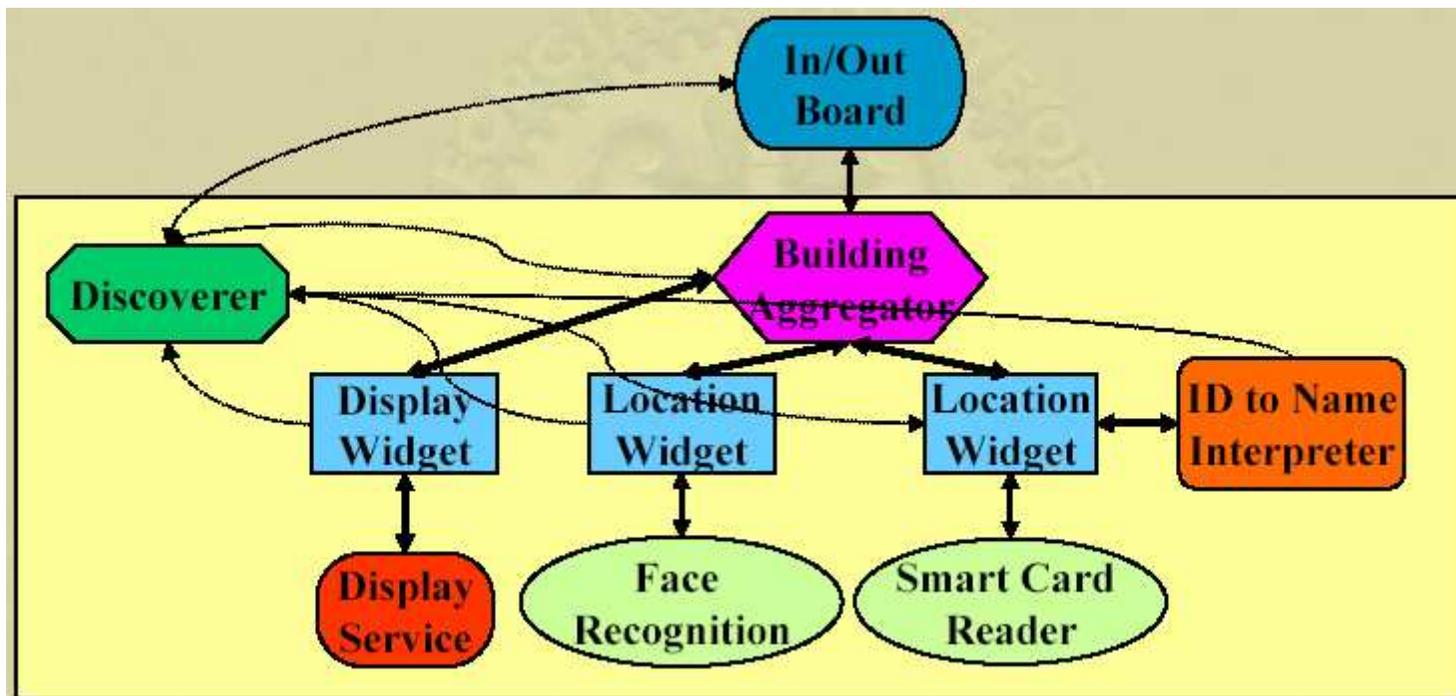
Services

- Effectue les traitements sur les données de contexte

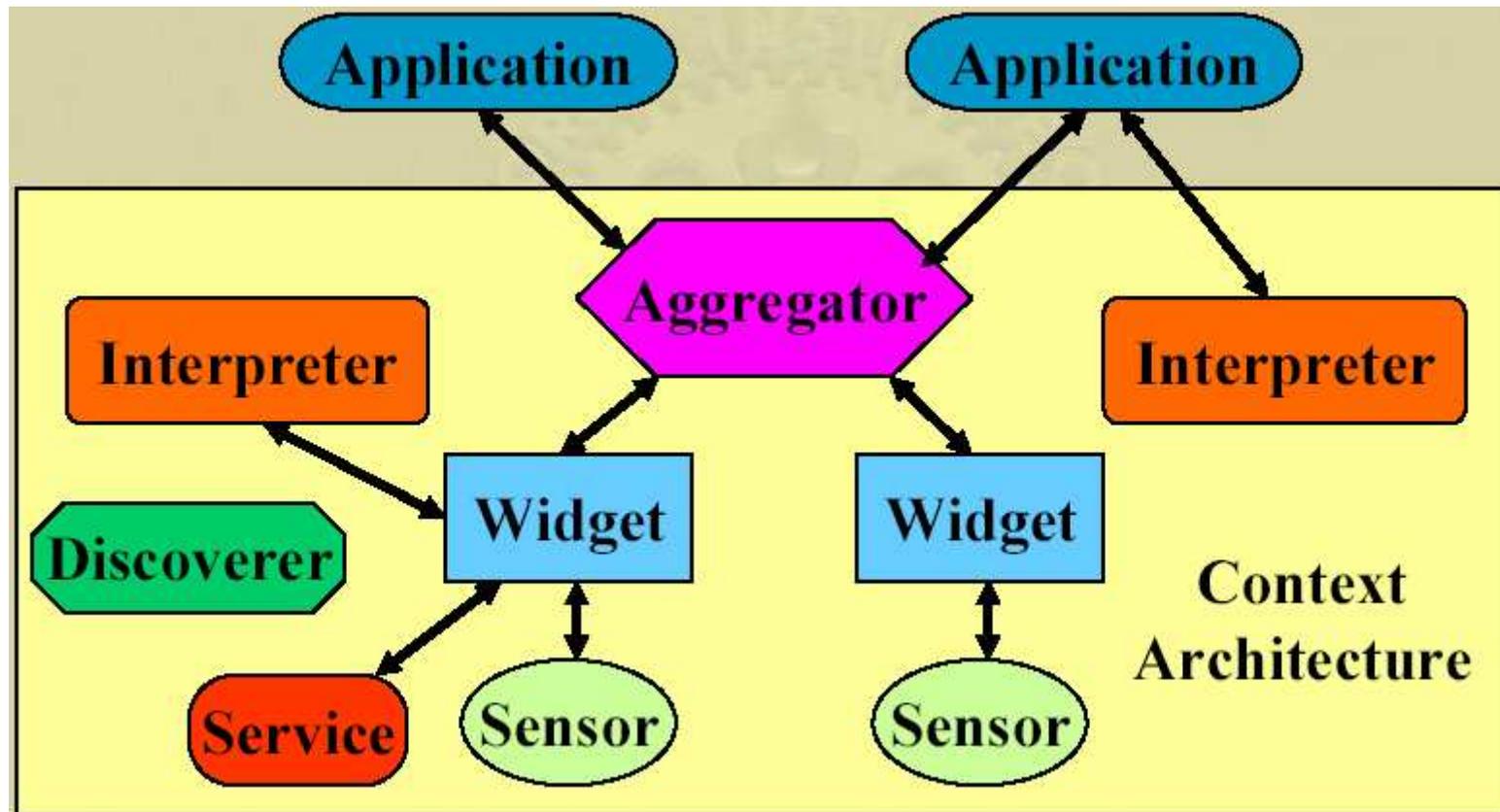


Discoverer

- Répertorie l'ensemble des composants de gestion du contexte



Synthèse de la plate-forme de Dey



Autres challenges en context-awareness

- Killer application
 - Besoin de quelque chose pour orienter les recherches
 - Besoin de quelque chose de concret à mettre entre les mains des futurs utilisateurs
- Taxonomie
 - Quel contexte est important?
 - Comment représenter le contexte?
 - Standards pour partager les composants entre groupes de recherche

Autres challenges en context-awareness

- Connaissances sur le monde réel
 - Comment le monde fonctionne
- Protection de la personne
 - Capture et collecte d'informations sur les gens, les lieux, les terminaux
 - Réponses sociales et technologiques à apporter
- Qualité de service : méta données
 - Cohérence, Confiance, Fréquence...
- « Smartness »
 - Quand le système doit-il décider, quand l'utilisateur doit-il décider?

Autres challenges en context-awareness

- Ambiguïté
 - Limitations des capteurs : vue réduite de la réalité
 - Comment réagir à une mauvaise interprétation automatique
- Modèle de l'environnement
 - Relations entre les localisations : granularité de la distance, hiérarchie de composition (batiment/piece...)
 - Relations entre les gens : famille, collègues
 - Relations entre les équipements?

Adaptation

- Contexte
 - Contenu
 - Modélisation
- Principes génériques d'adaptation
 - Statique
 - Dynamique
- 3 domaines principaux
 - Adaptation des interfaces utilisateurs
 - Adaptation du contenu
 - Adaptation des services

Contenu du contexte

- 4 axes
 - Utilisateur
 - Profil et préférences, localisation...
 - Terminaux et matériels
 - Taille de l'écran, résolution, couleurs, mémoire, API...
 - Réseau
 - Bande passante, connectivité, Qos
 - Méta-données de l'application
 - Taille, format, codage, langue, versions

Modélisation du contexte

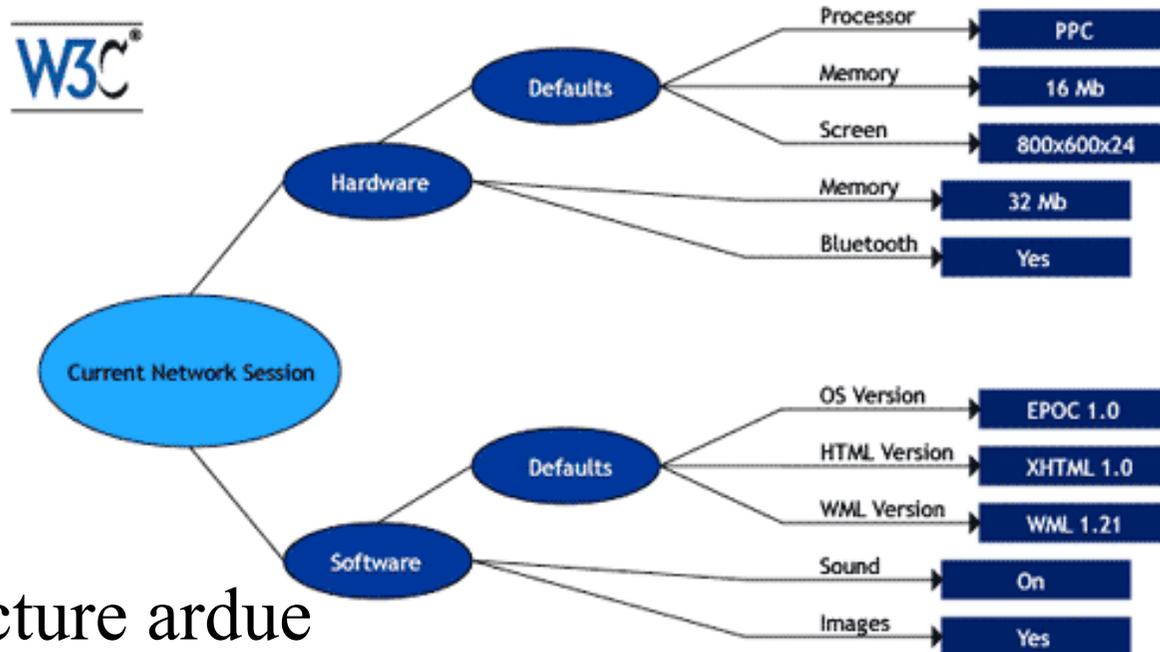
- Très souvent dépendante de l'application, encore peu de travaux « génériques » sur cette modélisation
- 3 approches
 - Couples attribut/valeur
 - Extension de CC/PP
 - Modélisation par ontologies

Modélisation du contexte par attribut/valeur

- Contexte = paires (attribut, valeur)
 - userName=flaforest
 - localisation = courseRoom
- Les paires sont indépendantes
- + Facilité d'implantation
- - Cohérence de l'ensemble
- - Sémantiquement pauvre

Modélisation du contexte avec CC/PP

- Composite Capabilities / Preferences Profile (W3C)
 - description des terminaux et des préférences utilisateur
 - Document RDF pour décrire les attributs
 - Modéliser le contexte = faire des extensions



- + standard
- - extensions => complexité, lecture ardue

Exemple de document XML CC/PP

```
<?xml version="1.0"?>
<!-- Checked by SiRPAC 1.16, 18-Jan-2001 -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#">
  <rdf:Description rdf:about="HWDefault">
    <rdf:type rdf:resource="HardwarePlatform" />
    <display>320x200</display>
    <memory>16Mb</memory>
  </rdf:Description>
</rdf:RDF>
```

Modélisation du contexte par ontologies

- Exemple CoOL [Strang & al. 03]

```
<instance xmlns=http://demo.heywow.com/schema/cool
xmlns:a=http://demo.heywow.com/schema/aspects
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<contextInformation>
<entity system="urn:phonenumber">+49-179-1234567</entity>
<characterizedBy>
<aspect name="GaussKruegerCoordinate">
<observedState
xsi:type="a:o2GaussKruegerType">367032533074</observedState>
<units>10m</units>
</aspect>
<certaintyOfObserver>90</certaintyOfObserver>
</characterizedBy>
</contextInformation>
</instance>
```

- + Sémantiquement riche
- + Utilisable dans des environnements d'envergure

Synthèse des modélisations de contextes

	Expressivité et richesse sémantique	Facilité d'implantation	Gestion des conflits
Paires Attribut / valeur	-	+	-
Extension de CC/PP	+	+	-
Ontologies	+	-	+

Principes génériques d'adaptation

- Adaptation statique
- Adaptation dynamique

Adaptation statique

- Préparer plusieurs versions d'une ressource avant son exploitation
- En fonctionnement, adaptation = choix de la version correspondant au contexte
- Beaucoup utilisée dans les débuts des applications multi-terminaux
 - Version existante pour terminaux standards
 - Nouvelles versions pour autres terminaux
- + simplicité et efficacité de fonctionnement
- - pb d'échelle, lourdeur de prise en compte d'une nouvelle version

Références de projets d'adaptation statique

- Eric D. Larson, Wireless Java Application Saves Women's Cancer Center \$500,000 per Year, J2ME case studies, september 2002.
<http://wireless.java.sun.com/midp/casestudies/wcc/>
- V. Massé, « La société MobilePlanet Europe fournit des terminaux mobiles à la Brigade des Sapeurs Pompiers de Paris (BSPP) pour l'équipement de ses véhicules d'intervention », Mobile Planet, juin 2002.
http://www.mobileplanet.fr/m_includes/press_release/2002_brigade.asp
- S. Benjaminsen, A. Djora « JetRek: How organisational identities slowed down speedy requisitions », BSA medical sociology conference, september 2002, York. <http://ww.sv.ntnu.no/iss/Aksel.Tjora/publications/Siri-york02-09.pdf>

Adaptation dynamique

- Transformations sur la ressource en cours de fonctionnement
- Exemples
 - CSS : transformations à la volée de la forme d'un document XML=>HTML
 - transformation de format de données
- + gain de temps pour prendre en compte une nouvelle version
- - complexité de l'outil (choix d'un « chemin d'adaptation » + adaptation)
- - temps de réponse

Références de projets d'adaptation dynamique

- TeraCom “Soluphone santé” 2002.
<http://www.soluphone.com/SoluSante.pdf>
- T. Lemlouma, N. Layaida, « A Framework for Media Resource Manipulation in an Adaptation and Negotiation Architecture », OPERA project, under submission, INRIA Rhône-Alpes, august 2001
- G. Menkhaus « Adaptive User Interface Generation in a Mobile Computing Environment », PhD Thesis, Salzburg University, 2002.

Trois domaines principaux d'adaptation

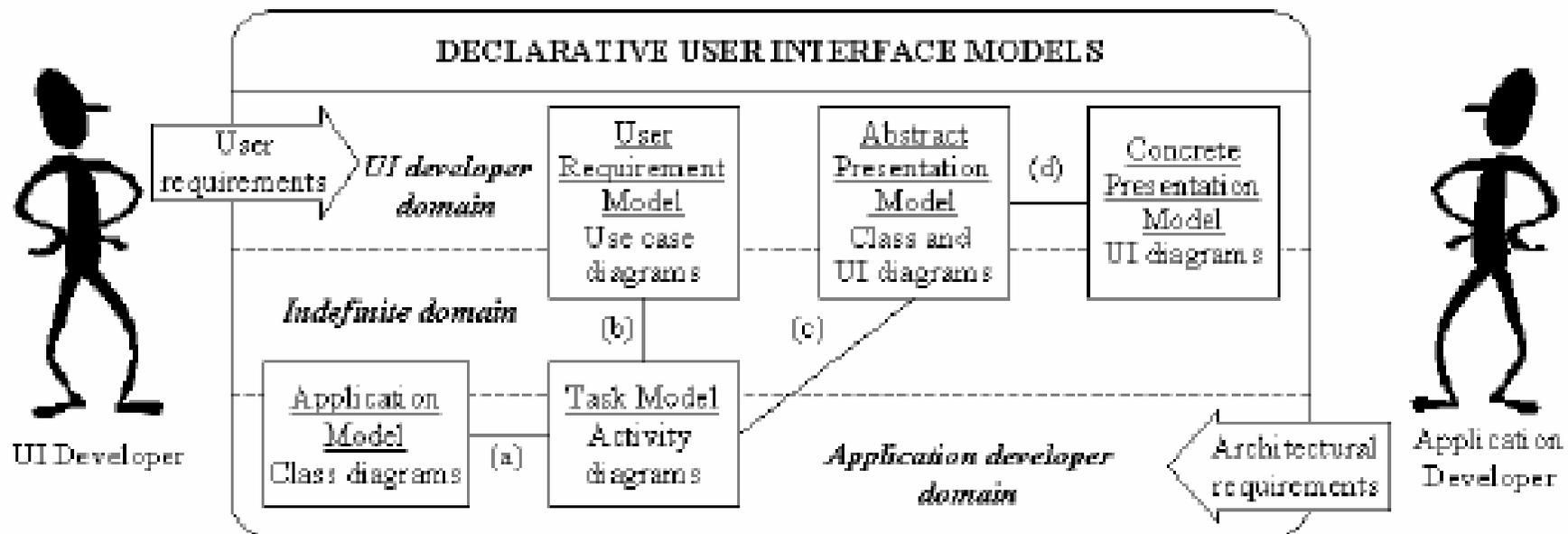
- Adaptation des interfaces utilisateurs
- Adaptation du contenu
- Adaptation des services

Adaptation des interfaces utilisateur

- Approches cognitives
 - Modèles utilisateur/tâche/dialogue
 - Généralement instrumentés par des plates-formes de génération automatique ou semi-automatique
- Approches d'ingénierie
 - Outils de conception et développement des interfaces homme-machine
 - Question : passer d'une problématique de visualisation/mise à jour à une solution concrète (code)
 - Généralement basé sur des modèles
 - UMLi
 - UiML

Modèle UMLi

- P. Pinheiro da Silva and N. W. Paton. UMLi: The Unified Modeling Language for Interactive Applications. In Proceedings of UML2000, volume 1939 of LNCS, pages 117-132, York, UK, October 2000. Springer
- Extension de UML: aspect interaction
- Diagrammes complexes et fastidieux



Exemple UMLi

Objets graphiques

- FreeContainers, 
- Containers, 
- Inputters, ▽
- Displayers, △
- Editors, ◇
- ActionInvokers, ▷

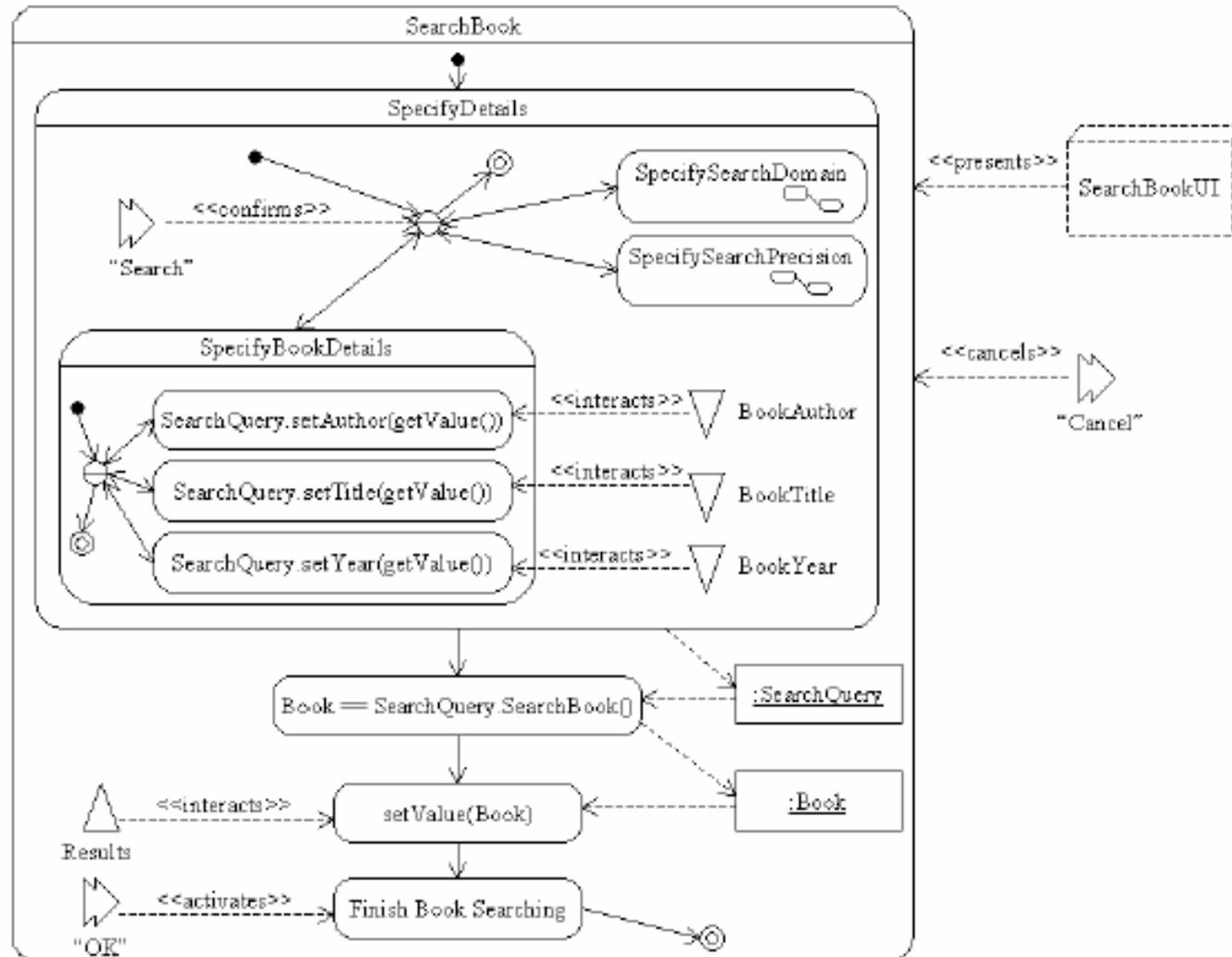


Figure 9: The SearchBook activity.

Modèle UiML

- M. Abrams, C. Phanouriou, "UiML: an XML language for building device independent user interfaces", XML'99, Philadelphia, December 1999
- Spécification de la présentation
 - Avec des balises à la XML
 - Dépendante de la plate-forme cible
 - Long et fastidieux!

– Exemple

```
<?xml version="1.0"?> <!DOCTYPE uiml PUBLIC "-//UIT//DTD UIML 2.0 Draft//EN" "http://uiml.org/dtds/UIML20.dtd">
<uiml> <head>
  <meta name="Description" content="Use of templates for CreditCard entry form"/>
  <meta name="Author" content="Constantinos Phanouriou"/>
  <meta name="Email" content="uiml-editor@uiml.org"/>
</head>
<peers>
```

Exemple UiML suite

```

<presentation name="WML">
  <component name="Dialog" maps-to="wml:card"/>
  <component name="TextField" maps-to="wml:input"/>
  <component name="Button" maps-to="wml:DO type='ACCEPT'"/>
  <component name="InputEvent" maps-to="wml:event type='onenterforward'"/>
</presentation>
<presentation name="VoiceXML">
  <component name="Dialog" maps-to="vxml:form"/>
  <component name="TextField" maps-to="vxml:prompt"/>
  <!-- Button does not have a rendering in Voice -->
  <component name="Button" maps-to="QUOTE;/>
  <component name="InputEvent" maps-to="vxml:filled"/>
</presentation>
<presentation name="Java-AWT">
  <component name="Dialog" maps-to="java.awt.Dialog"/>
  <component name="TextField" maps-to="java.awt.TextField"/>
  <component name="Button" maps-to="java.awt.Button"/>
  <component name="InputEvent"
    maps-to="java.awt.event.ActionEvent"/>
</presentation>

```

Exemple UiML suite

```

<logic name="HTTP">
  <component name="CreditApp">
    <method name="SendCredit"
      maps-to="http://<server>:port/cgi-bin/acceptCredit.pl?">
      <param name="cnum"/>
    </method>
  </component>
</logic>
<logic name="Java">
  <component name="CreditApp" maps-to="myBiz.creditApp.class">
    <method name="SendCredit" maps-to="acceptCredit">
      <param name="cnum"/>
    </method>
  </component>
</logic>
</peers>

```

Exemple UiML suite

```

<template name="CreditCard">
  <part name="CreditContainer" class="CreditDialog">
    <style> <property name="title">Credit Card Entry Form</property> </style>
    <part name="CreditNum" class="number"/>
    <part name="AcceptNum" class="Accept">
      <style> <property name="content">Accept</property> </style>
    </part>
  </part>
</template>

<interface>
  <structure>
    <part name="ECommerceApp"> <!-- UI description -->
      <part name="MyCredit" source="#CreditCard"/>
    </part>
  </structure>

```

Exemple UiML suite et fin

```

<style> <property name="rendering" part-class="MyCredit.CreditDialog"> Dialog</property>
  <property name="rendering" part-class="MyCredit.number"> TextField</property>
  <property name="rendering" part-class="MyCredit.Accept"> Button</property>
  <property name="rendering" event-name="InputEvent"> InputEvent</property>
  <property name="rendering" method-name="SendCredit"> SendCredit</property>
</style>

<behavior>
  <rule>
    <condition> <event name="InputEvent" part-name="MyCredit.AcceptNum"/> </condition>
    <action>
      <method name="SendCredit">
        <param name="cnum"><property name="content" part-name="MyCredit.CreditNum"/> </param>
      </method>
    </action>
  </rule>
</behavior>

</interface>

```

```
</uiml>
```

Plates-formes de génération de code d'IU

- Fondées sur plus ou moins de modèles, parmi:
 - Domaine : description des services de l'application, modèle de BD...
 - Tâches : graphes d'activités
 - Utilisateur : préférences
 - Dialogue : partie interactive de l'interface
 - Présentation : partie visuelle de l'interface
- Une génération plus ou moins automatique du code
- Des modèles inter-reliés, long, complexes et fastidieux... il est souvent plus rapide de coder soi-même l'interface!

Références de plates-formes

- P. Sukaviriya, J. Foley, T. Griffith: A Second Generation User Interface Design Environment. In: S. Ashlund, et.al. (eds.): Bridges between Worlds. Proceedings InterCHI'93 (Amsterdam, April 1993). New York: ACM Press, 1993, 375-382.
- C. Märtin: Software Life Cycle Automation for Interactive Applications: The AME Design Environment. In: J. Vanderdonckt (ed.): Computer-Aided Design of User Interfaces. Namur: Namur University Press, 1996, 57-74.
- H. Balzert, F. Hofmann, V. Kruschinski, C. Niemann: The Janus Application Development Environment Generating More than the User Interface. In: J. Vanderdonckt (ed.): Computer-Aided Design of User Interfaces. Namur: Namur University Press, 1996, 183-205.
- F. Bodart, A.-M. Hennebert, J.-M. Leheureux, I. Provot, B. Sacre, J. Vanderdonckt: Towards a Systematic Building of Software Architectures: the TRIDENT Methodological Guide. In P. Palanque, R. Bastide (eds.): Design, Specification and Verification of Interactive Systems. Wien: Springer, 1995, 262-278.
- C. Janssen, A. Weisbecker, J. Ziegler: Generating User Interfaces from Data Models and Dialogue Net Specifications. In: S. Ashlund, et.al. (eds.): Bridges between Worlds. Proceedings InterCHI'93 (Amsterdam, April 1993). New York: ACM Press, 1993, 418-423.
- Guido Menkhaus, Wolfgang Pree: User interface tailoring for multi-platform service access. IUI 2002: 208-209

Références de plates-formes

- T. Elwert, E. Schlungbaum: Modelling and Generation of Graphical User Interfaces in the TADEUS Approach. In: P. Palanque, R. Bastide (eds.): Designing, Specification, and Verification of Interactive Systems. Wien: Springer, 1995, 193-208.
- P. Johnson: Human-Computer Interaction. London: McGraw-Hill, 1992.
- A. Puerta: The Mecano Project: Comprehensive and Integrated Support for Model-Based Interface Development. In: J. Vanderdonckt (ed.): Computer-Aided Design of User Interfaces. Namur: Namur University Press, 1996, 19-36.
- A. Puerta: Issues in Automatic Generation of User Interfaces in Model- Based Systems. In: J. Vanderdonckt (ed.): Computer-Aided Design of User Interfaces. Namur: Namur University Press, 1996, 323-325.
- F. Lonczewski, S. Schreiber: The FUSE-System: An Integrated User Interface Design Environment. In: J. Vanderdonckt (ed.): Computer-Aided Design of User Interfaces. Namur: Namur University Press, 1996, 37-56.
- P. Szekely, P. Sukaviriya, P. Castells, J. Muthukumarasamy, E. Salcher: Declarative interface models for user interface construction tools: the MASTERMIND approach. In: L. Bass, C. Unger (eds.): Engineering for Human-Computer Interaction. Proceedings of the IFIP TC2/WG2.7 working conference on engineering for humancomputer interaction (Yellowstone Park, August 1995). London: Chapman & Hall, 1996, 120-150.

Adaptation du contenu

- Adaptation des données : modifier une donnée pour
 - Qu'elle soit exploitable par le terminal cible
 - Qu'elle soit conforme aux règles de protection des données
 - ...
- Adaptation dynamique (à la demande) ou statique (plusieurs versions stockées)
- Localisation de l'adaptation
 - sur le client : ne convient pas aux terminaux légers
 - sur le serveur : si tient la charge
 - entre les deux : sur un « réseau actif » (encore difficile) ou via un ou n proxies

Adaptation selon le type de la donnée source

- Source texte
 - Conversion de format (html -> txt, doc -> pdf...)
 - Résumé
 - Traduction
 - Compression/décompression
 - Synthèse vocale
- Source image
 - Conversion de format (jpeg -> png)
 - Modification de résolution, nombre de couleurs...
 - Compression / décompression (e.g. sémantique jpeg ou brute zip)

Adaptation selon le type de la donnée source

- Source sonore
 - Conversion de format
 - Synthèse textuelle et reconnaissance vocale
 - Compression / décompression (e.g. sémantique MP3 ou brute zip)
- Source vidéo
 - Conversion de format (résolution, nb images/sec)
 - Décomposition / recomposition spatiale (zoom...)
 - Compression / décompression (e.g. sémantique MPEG4 ou brute zip)

Opérateurs pour l'adaptation de contenu

- Codage
 - transcodage de média (réduction nb couleurs)
 - transformation de modalité (texte -> audio)
 - compression (jpeg)
- Format (Wav->MP3)
- Structure (HTML ->WML)
- Remplacement (image par texte descriptif)
- Sélection (sélection d'images moins volumineuses)
- Intégration (données multi-serveurs)

Adaptation de contenu - RTP Mixer

- E. Mory et al. Adaptation de contenu multimédia aux terminaux mobiles. RTSI - ISI n° spécial systèmes d'information pervasifs n°9, 2004, Hermès: 39-60
- sens d'adaptation
 - la **distillation** réduit les données trop riches
 - le **raffinement** cherche à améliorer la qualité des données
- Adaptation dynamique par un proxy avec cache
 - transparente à l'application : effectuée par un intermédiaire durant le transport des données
 - utilisation d'un profil CC/PP
 - 2 services d'adaptation
 - redimensionnement automatique et recadrage d'images
 - traitement de vidéos : conversion de format, distillation

Adaptation de documents par services Web

- G. Berhe, L. Brunie, LIRIS Adaptation de contenus multimédia pour les systèmes d'information pervasifs RTSI - ISI n° spécial systèmes d'information pervasifs n°9, 2004, Hermès: 39-60
- Architecture d'adaptation à base de services Web
 - Local Proxies : calcul et gestion des adaptations
 - Content Proxies : frontaux des serveurs de données, communiquent avec les LP
 - Content Servers : sources de données
 - Adaptation Service Proxies : exécution des services
 - Adaptation Services Repository : répertorie les informations descriptives des services
 - Gestionnaire de profils

Adaptation de documents par services Web

- 4 types de profils
 - document
 - méta données physiques (type, taille, format...)
 - méta-données de stockage (versions disponibles, répartition...)
 - méta-données sémantiques (eg mots-clés)
 - client : utilisateur et terminal CC/PP
 - utilisateur : langues, domaines d'intérêt...
 - terminal : matériel (taille écran, mémoire, capacités d'affichage...)et logiciel (disponibles)
 - réseau
 - latence, bande passante entre proxy local et utilisateur
 - service
 - WSDL : type d'adaptation, type de média, performance, coût...

Adaptation de documents par services Web

- Proxy locaux
 - recueillir requête utilisateur
 - calculer profil client
 - transmettre requête aux proxies de contenu
 - comparer profil réponse avec profil client
 - en déduire un plan d'adaptation et lancer l'exécution du plan
 - intégrer les contenus reçus et adaptés et les transmettre au client
 - collaborer entre proxies locaux pour la gestion des caches

Adaptation de documents par services Web

- Plan d'adaptation
 - déterminer les contraintes d'adaptation : ensemble des conditions sur les attributs du profil
 - déterminer les opérateurs d'adaptation nécessaires
 - sélectionner une stratégie d'adaptation optimale
 - rechercher les services d'adaptation capables de réaliser les opérateurs d'adaptation nécessaires
 - négocier avec ces services (coûts et performance)
 - sélectionner les services
 - exécuter le plan d'adaptation

Adaptation de documents par services Web

- Notion de chemin d'adaptation
 - séquence d'opérateurs d'adaptation permettant de vérifier l'ensemble des contraintes définies
 - les chemins d'adaptation peuvent être pondérés par les coûts, performances...
- Notion de graphe d'adaptation
 - lorsque plusieurs adaptations peuvent être effectuées en parallèle sur des sous-ensembles de données (eg. image et méta-données sur les images - format DICOM)

Le projet SEFAGI

- Cadre applicatif : le projet SICOM
 - Système d'information communicant pour la santé
- Adaptation des interfaces utilisateurs au terminal
 - T. Chaari, F. Laforest, A. Flory Génération automatique d'interfaces graphiques pour la saisie et la consultation de données : Application au domaine médical. Int. Conf. on Sciences of Electronic, Technology of Information and Telecommunications SETIT 2004, Sousse, Tunisie, 15-20 mars 2004
 - T. Chaari, F. Laforest Génération et adaptation automatiques des interfaces utilisateurs pour des environnements multi-terminaux Le projet SEFAGI : Simple Environment For Adaptable Graphical Interfaces Revue Ingénierie des systèmes d'Information, n° spécial systèmes d'information pervasifs, volume 9 - n°2/2004:11-38

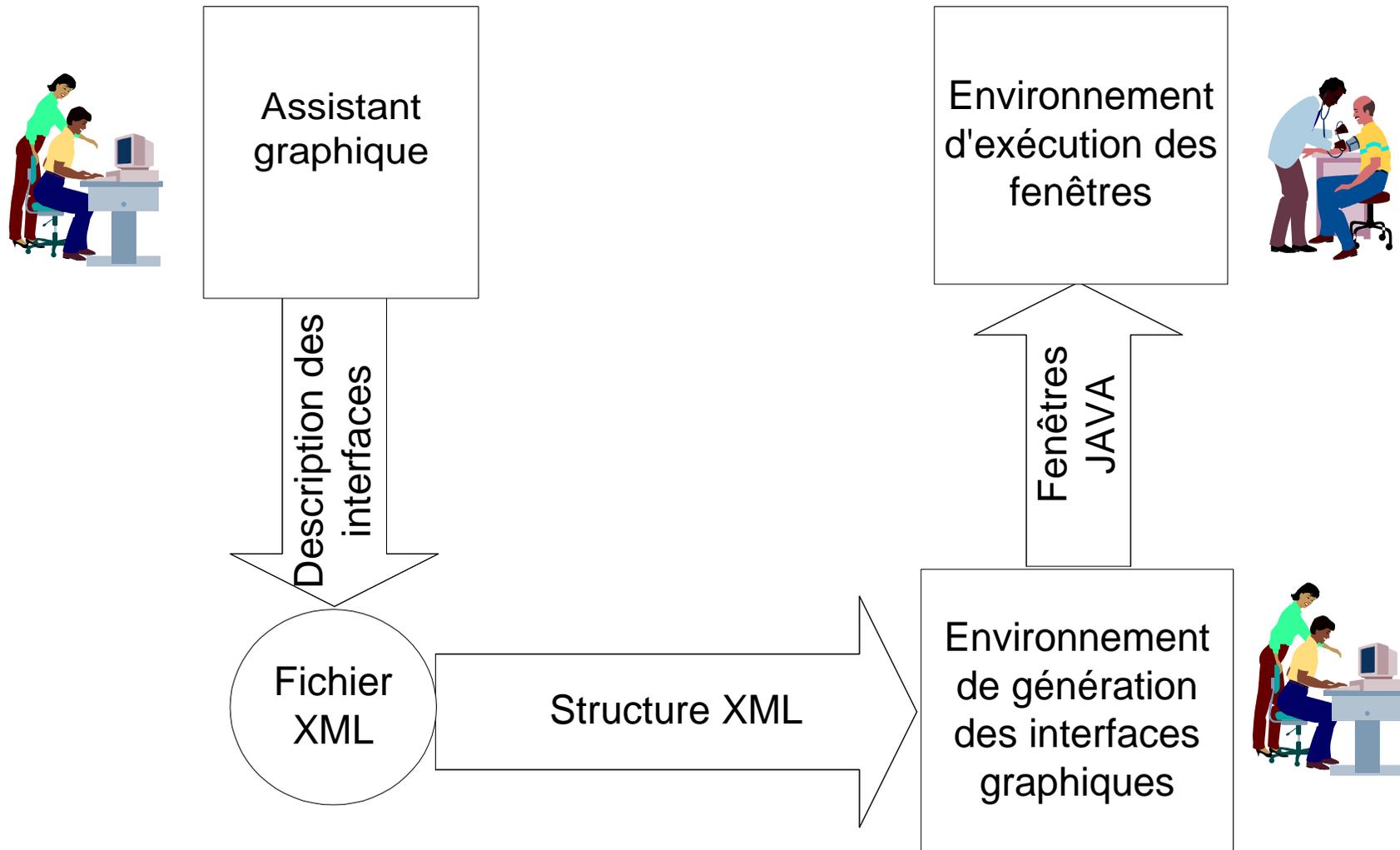
Motivation

- Interface graphique du Projet SICOM comporte
 - Une partie générique
 - Des parties spécifiques
 - Un ensemble de fenêtres pour chaque pathologie
 - Un ensemble de fenêtres pour chaque type d'utilisateurs de l'application
- ==> Nombre de fenêtres différentes élevé
- ==> Temps de développement assez important
- Génération automatique des fenêtres spécifiques pour des terminaux de types divers
 - PC
 - Terminaux légers faibles (MIDP)
 - Terminaux légers (CDC)

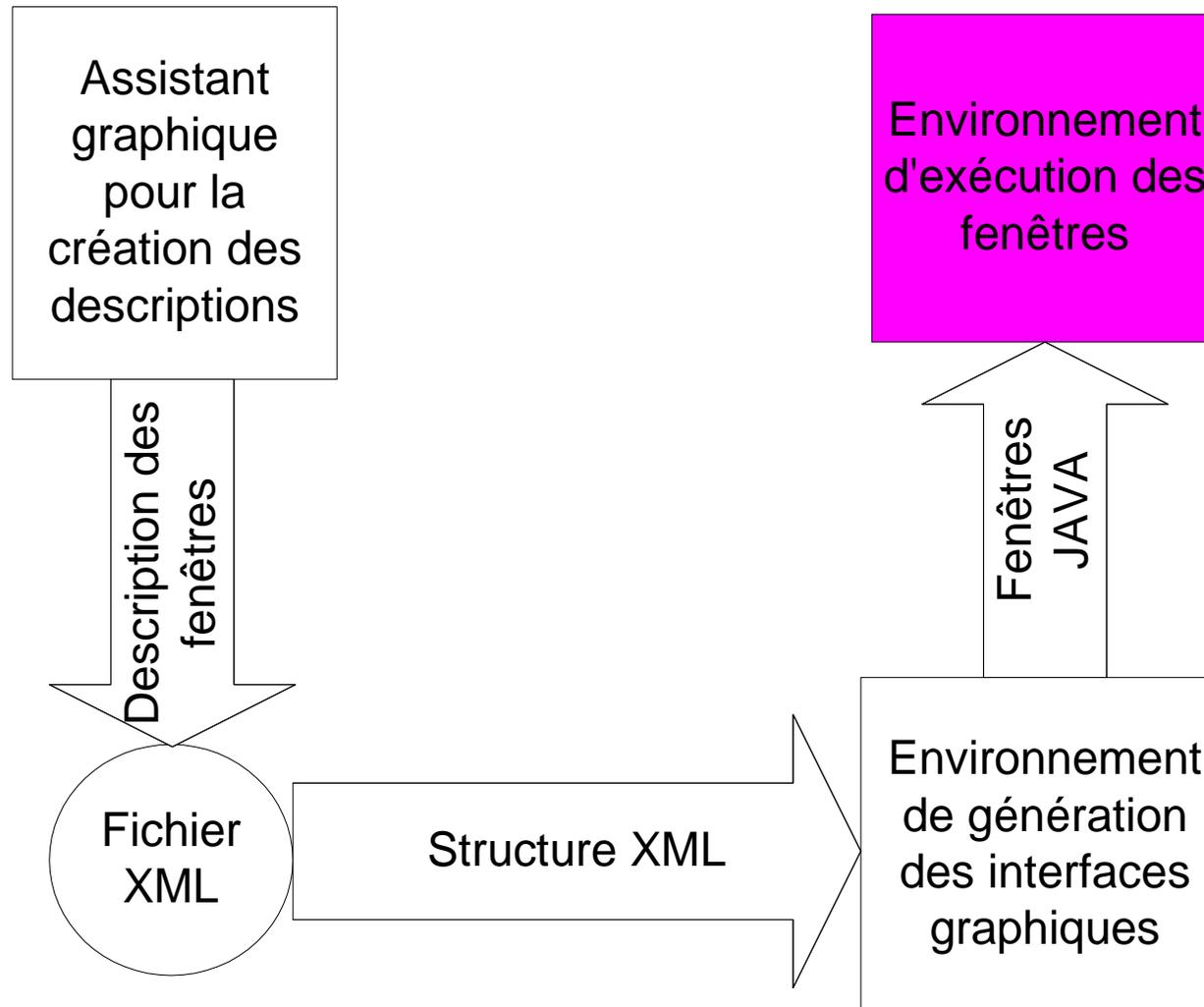
Objectifs

- Interface commune existante
 - Administration
 - Base de données existante
- Créer automatiquement des interfaces graphiques spécifiques
 - Affichage
 - Traitements
 - Intégrer dans le programme principal
 - Assurer l'échange d'informations inter-fenêtres

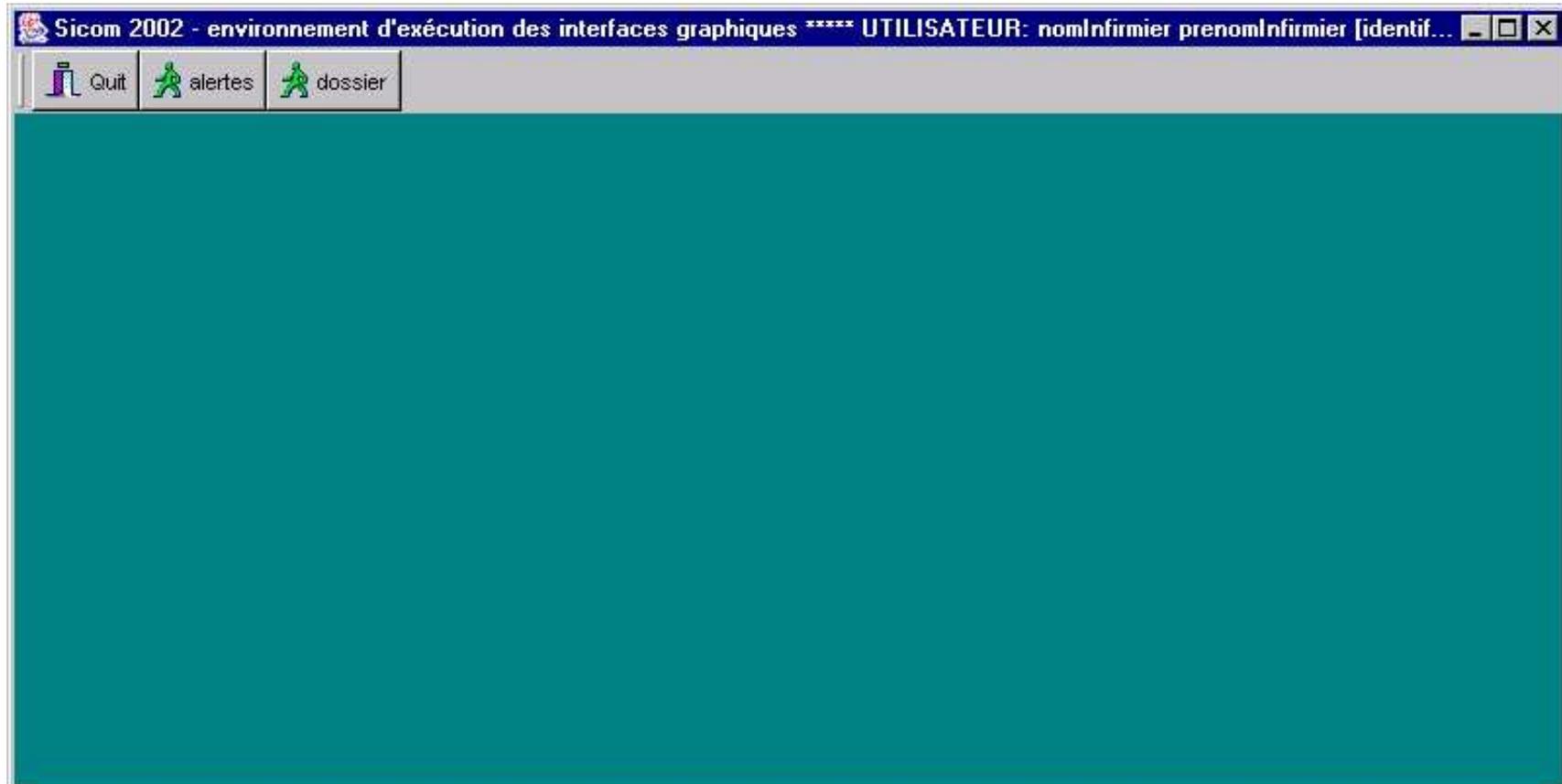
Architecture logique



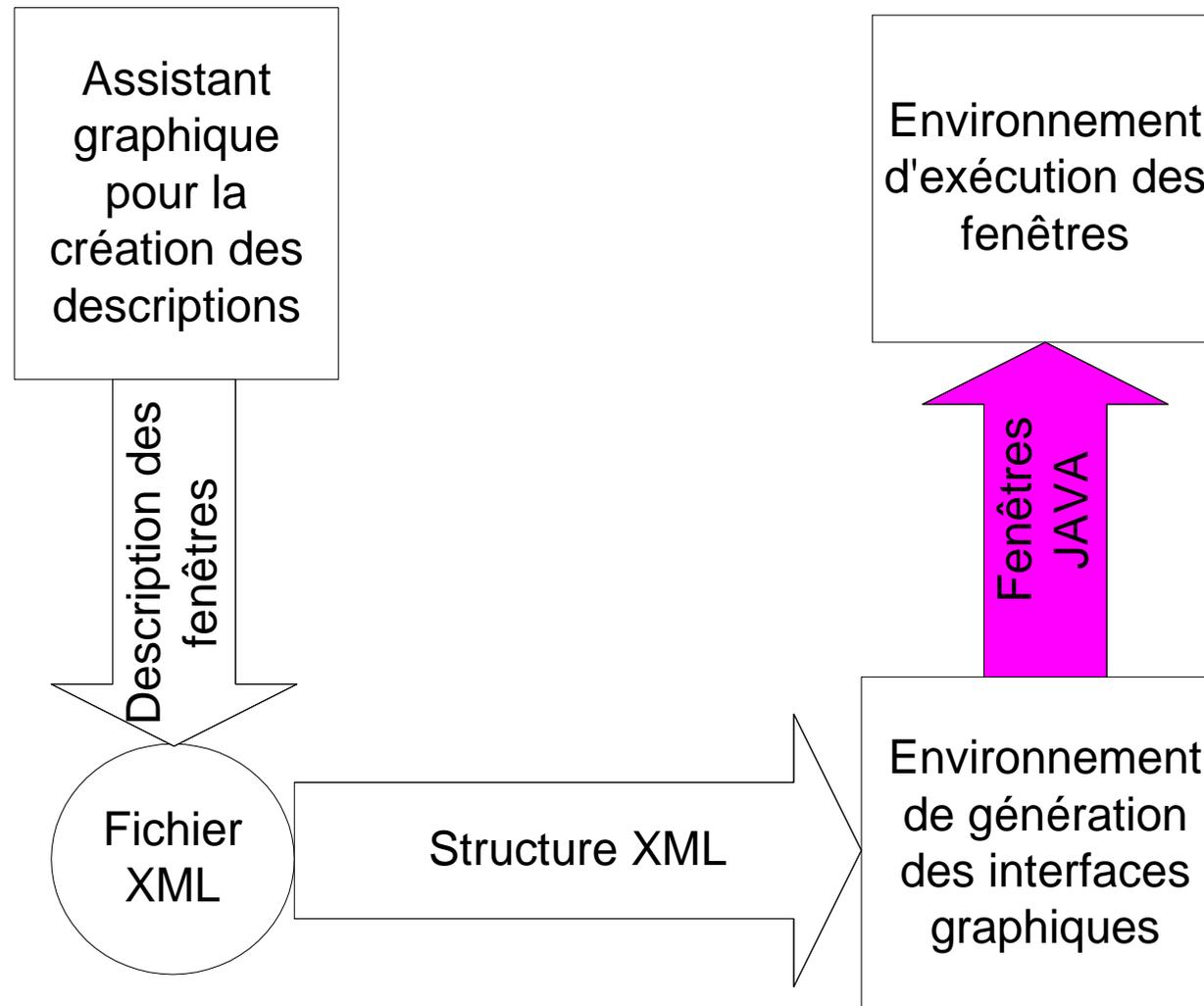
Environnement d'exécution



Environnement d'exécution sur PC



Description des fenêtres à générer



Description des fenêtres à générer

- Une fenêtre = une liste de panneaux
- Un panneau
 - Une représentation graphique dépendant du type de terminal
 - Des services associés (getData / setData)
- Types de panneaux pré-définis
 - 6 pour l'instant, extensible
 - Chaque type de panneau correspond à une classe dans la bibliothèque des composants graphiques

Sicom 2022 - environnement d'exécution des interfaces graphiques ***** UTILISATEUR: nomInfirmier prenomInfirmier [identif...]

Quit | alertes | dossier

alertes

Numéro Alerte	Numéro Dossier	Date	Libellé	Vu par
15	3	2002-02-24 00:00:00	Anomalie Cathéter	nomInfirmier
16	3	2002-02-24 00:00:00	Poids élevé	nomInfirmier
17	3	2002-02-24 00:00:00	Tension élevée	nomInfirmier
18	3	2002-02-24 00:00:00	Anomalie Cathéter	medecin1
25	3	2002-03-11 00:00:00	Température élevée	medecin1
26	3	2002-03-11 00:00:00	Poids élevé	medecin1
27	3	2002-03-11 00:00:00	Tension élevée	nomInfirmier
28	3	2002-03-11 00:00:00	Tension élevée	nomInfirmier
3	1	2002-02-24 00:00:00	Anomalie Cathéter	nomInfirmier
4	1	2002-02-24 00:00:00	Anomalie Cathéter	nomInfirmier

voir description | marquer message

Panneau de type tableau

- Saisie et consultation des données à partir du clavier
- Une case = un widget de base (zone de saisie, liste énumérée, cases à cocher...)
- Contraintes associées à chaque case

Températures	
general.dates	general.temperatures
2002-03-11 00:00:00	41
2002-03-12 00:00:00	39
2002-03-13 00:00:00	40
2002-03-14 00:00:00	41
2002-03-15 00:00:00	39
2002-03-16 00:00:00	38
2002-03-17 00:00:00	38
2002-03-18 00:00:00	37
2002-03-19 00:00:00	37

PC standard

Veuillez choisir une valeur

2002-03-11 00:00:00

2002-03-12 00:00:00

2002-03-13 00:00:00

2002-03-14 00:00:00

2002-03-15 00:00:00

2002-03-16 00:00:00

2002-03-17 00:00:00

2002-03-18 00:00:00

2002-03-19 00:00:00

2002-03-20 00:00:00

2002-03-21 00:00:00

Sortie

Suivant ^A

** Valeurs **

general.dates

general.temperatures

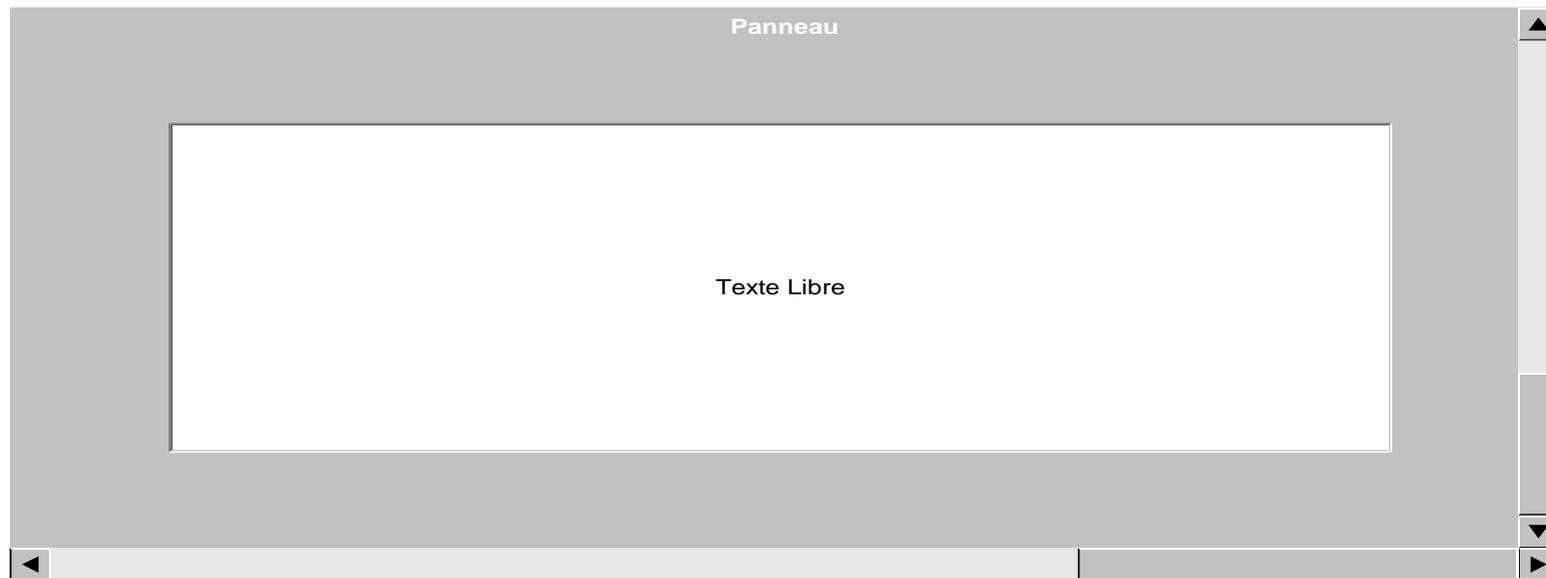
Sortie

Précédent ^A

PC de poche

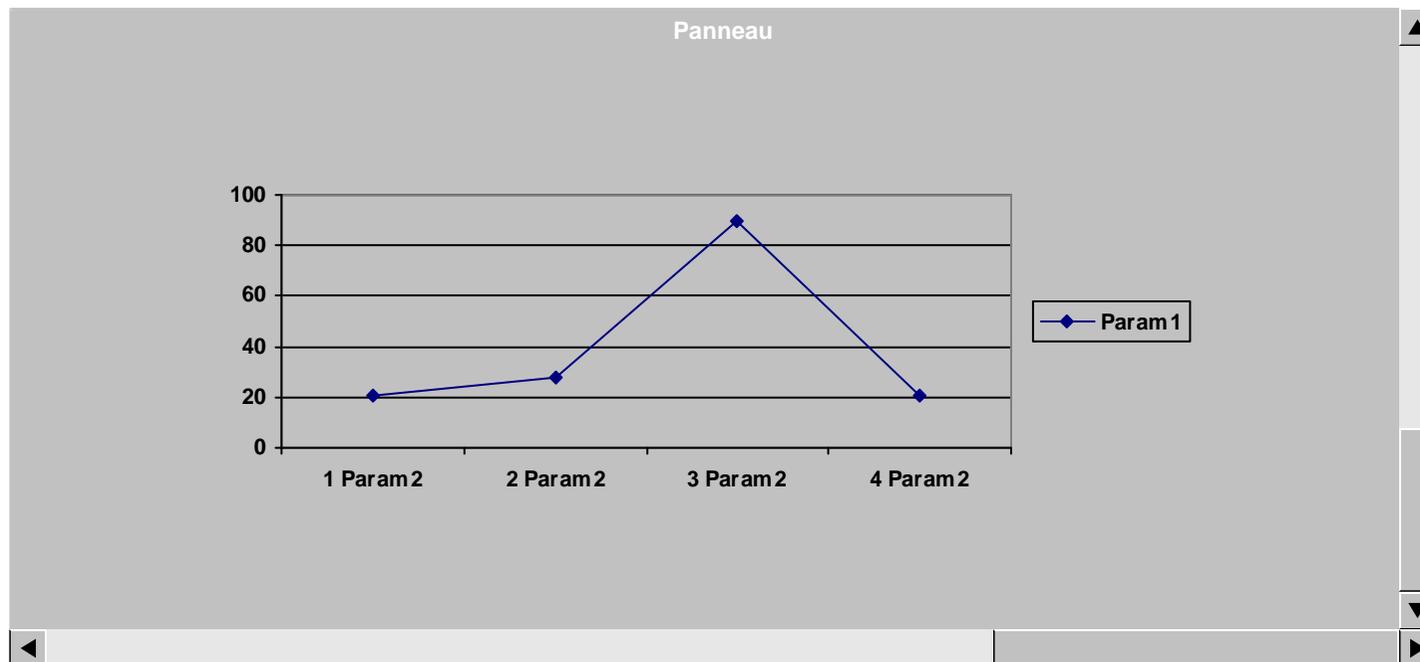
Panneau de type texte

- Grande zone texte (commentaires ,texte multiligne).



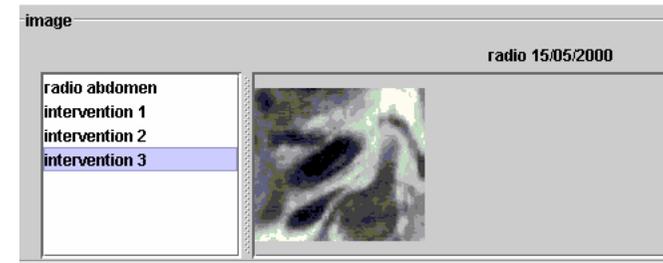
Panneau de type graphique

- Présentation de courbes 2D ou de graphiques bâtonnet



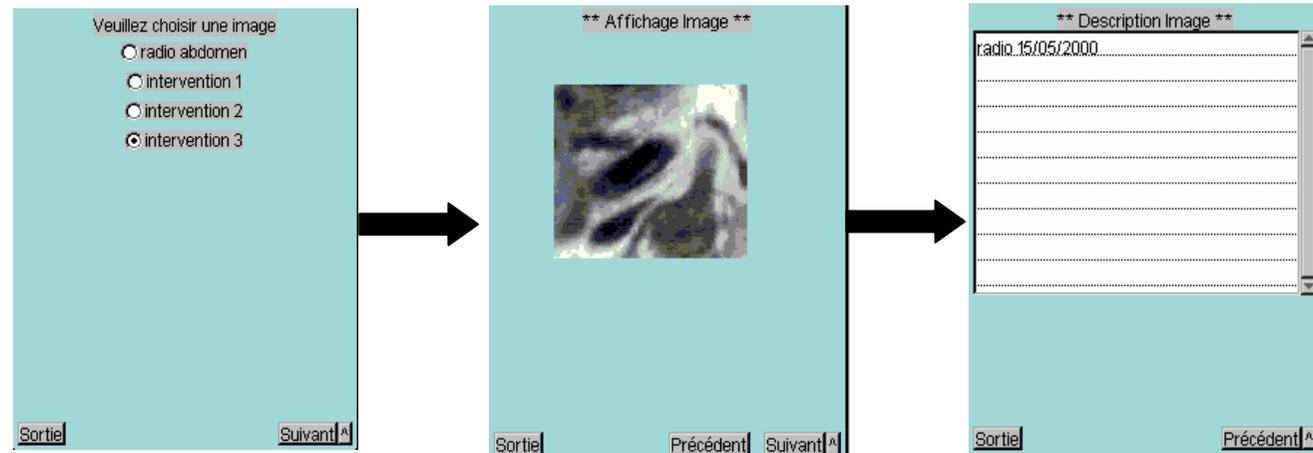
Panneau de type image

- Présentation des images
 - Liste des images
 - Image sélectionnée
 - Texte descriptif associé



PC standard

PC de poche



Panneau de type vidéo

- Présentation des extraits vidéo
- Liste des vidéos
- vidéo sélectionnée
- Texte descriptif associé



Panneau de type commande

- Boutons pour lancer l'appel des services associés aux panneaux de la fenêtre

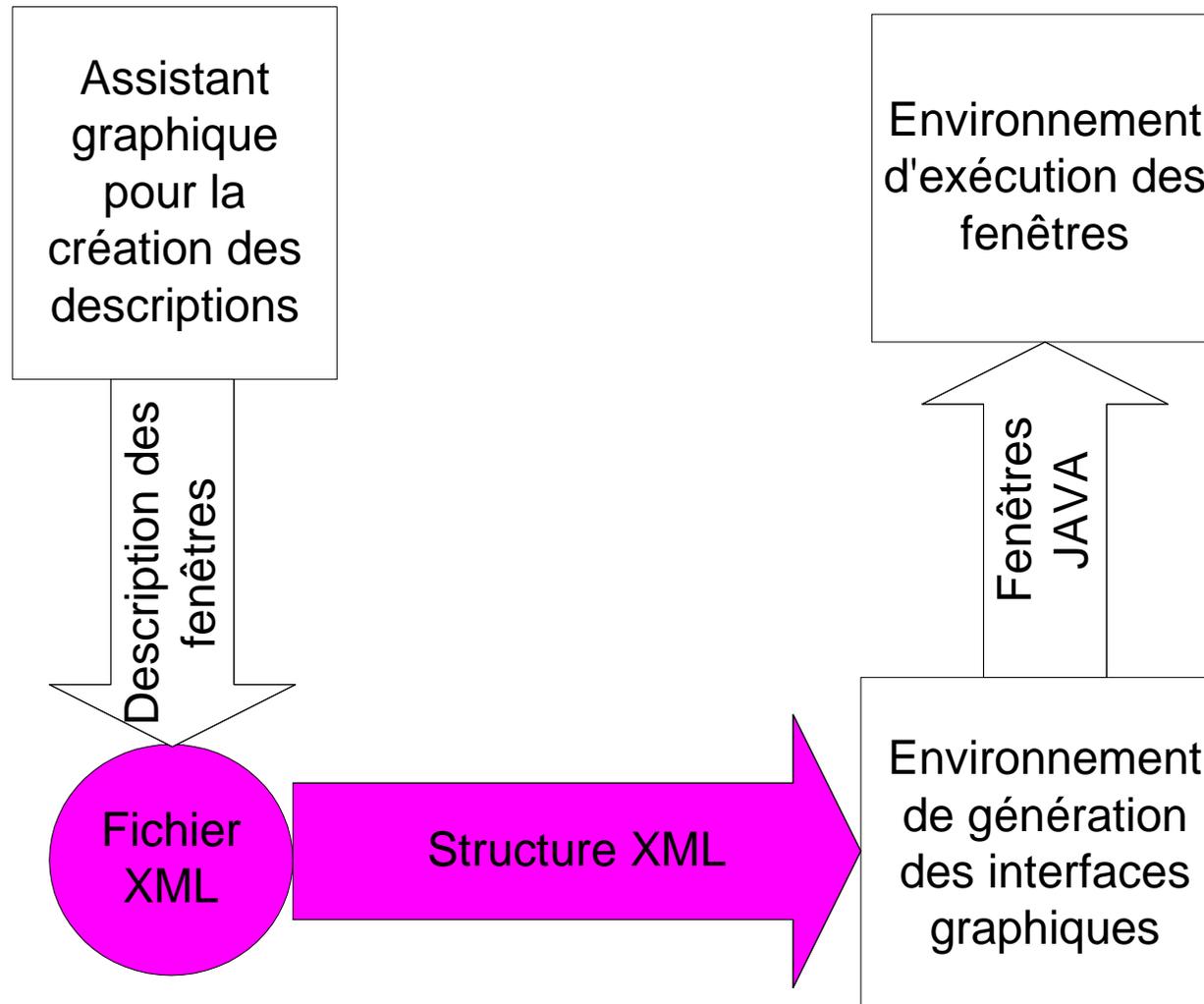


PC standard



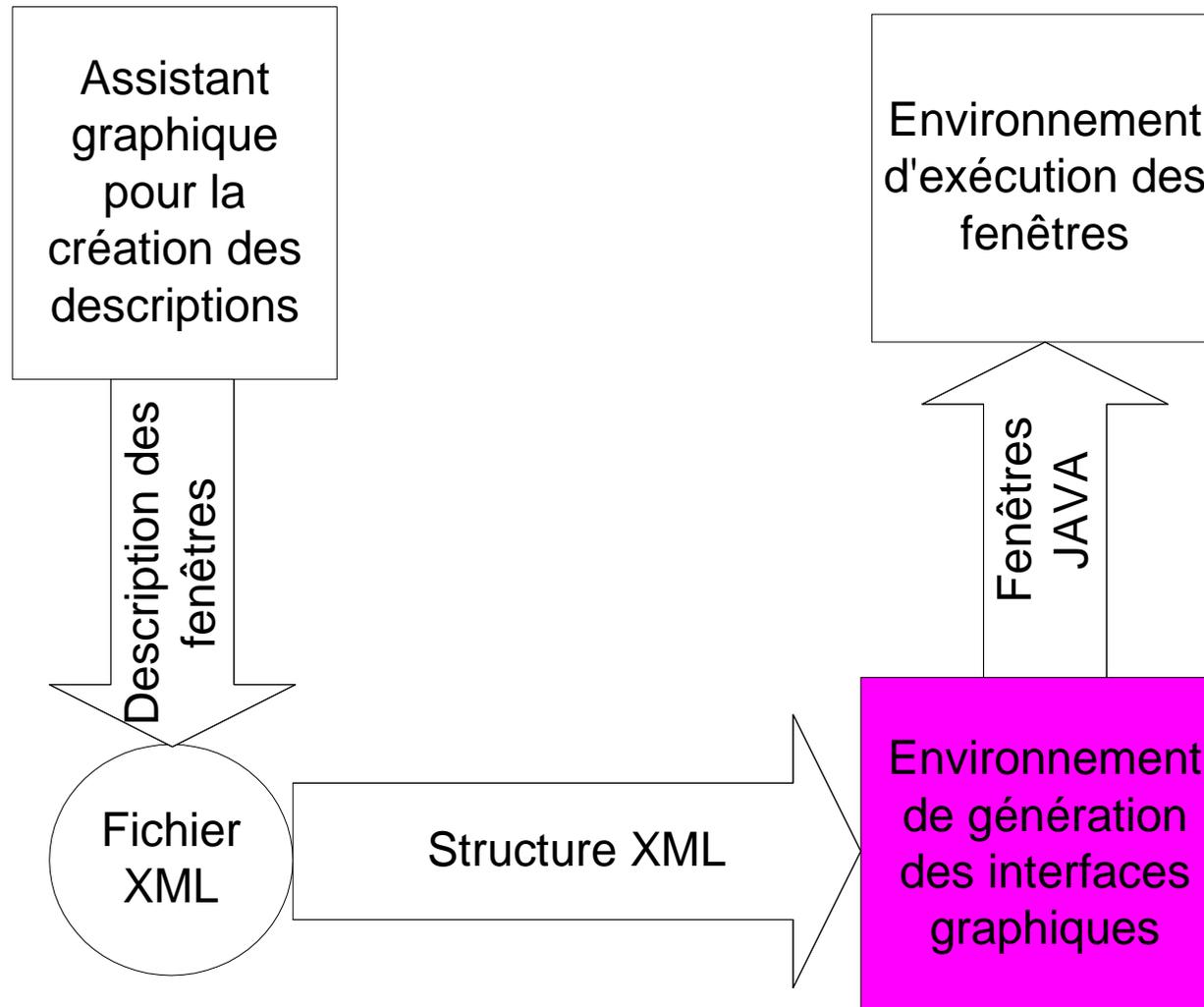
PC de poche

Structure XML



Structure	Values
interface	
nom	nom de la fenêtre
titre	titre de la fenêtre
panneau	
id	identifiant unique du panneau
type	type du panneau (tableau, images,...)
servicepardefaut	le service par défaut du panneau
titre	titre du panneau
service	
id	identifiant unique du service (nom.méthode)
server	serveur hébergeant le service (serveur:port)
parametre	
type	type du parametre (entrée, sortie, entrée/sortie)
nom	nom du parametre
valeur	
type	type de la valeur
localisation	composant ou variable (lien XPointer)
reference_groupe	1 iere référence sur la localisation
reference_specifique	2 ieme référence sur la localisation
val_initiale	valeur par défaut du parametre
contraintes	
type	énumération, intervalle...
contrainte	1 ère contrainte
contrainte	2 ème contrainte...
affichage	
composant	
type	type du composant (texte, numérique, liste,...)
editable	oui / non
visible	oui / non
id	identifiant unique du composant
titre	titre du composant
action	
panneaucible	id d'un panneau cible
servicecible	id du service cible

Environnement de génération

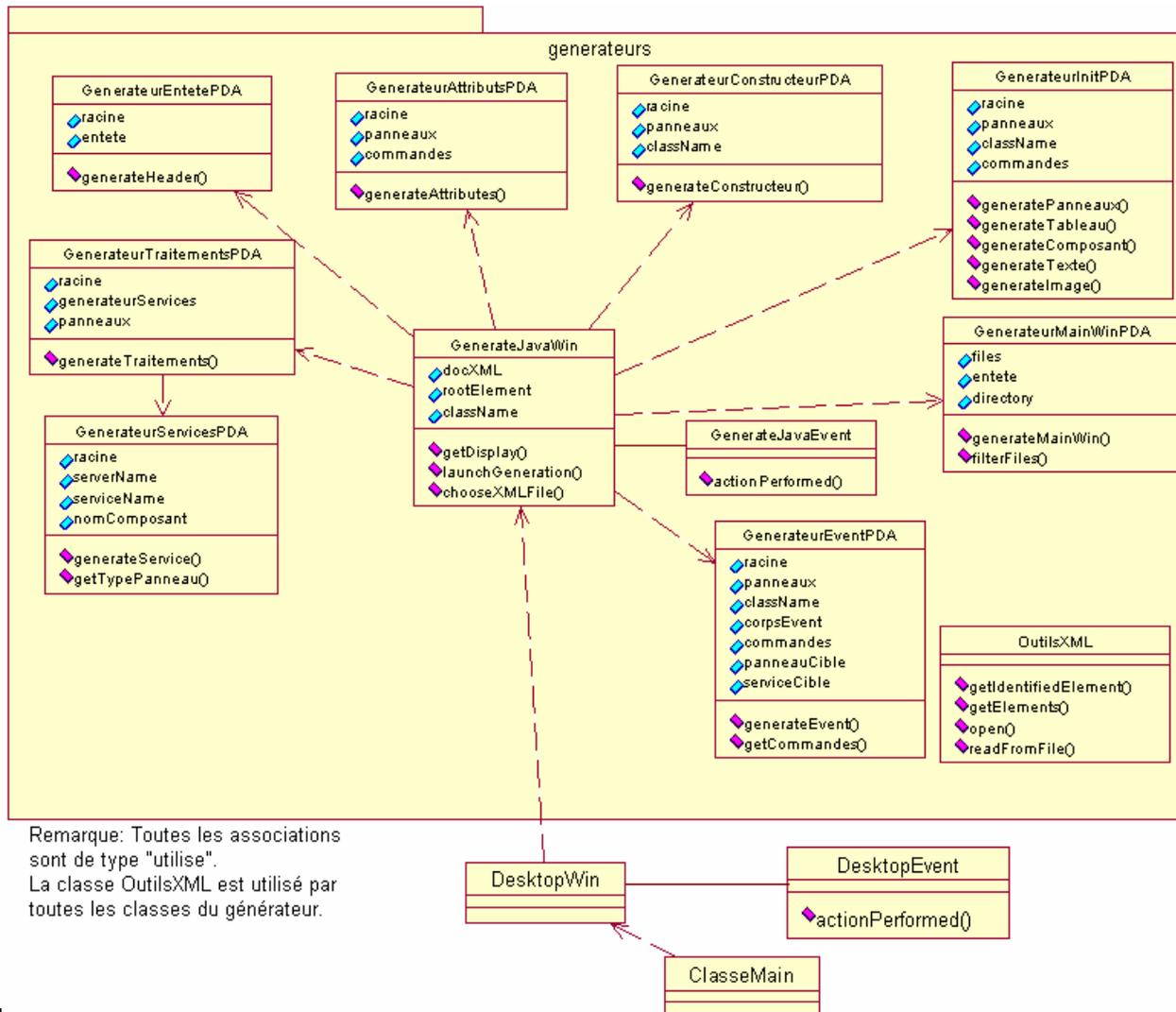


Environnement de génération

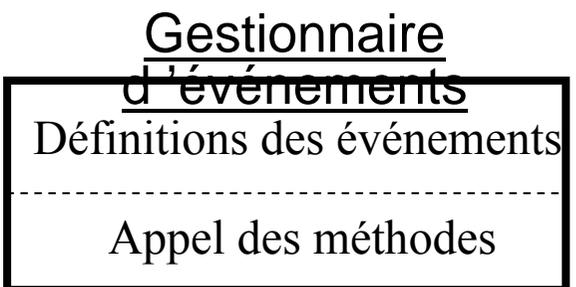
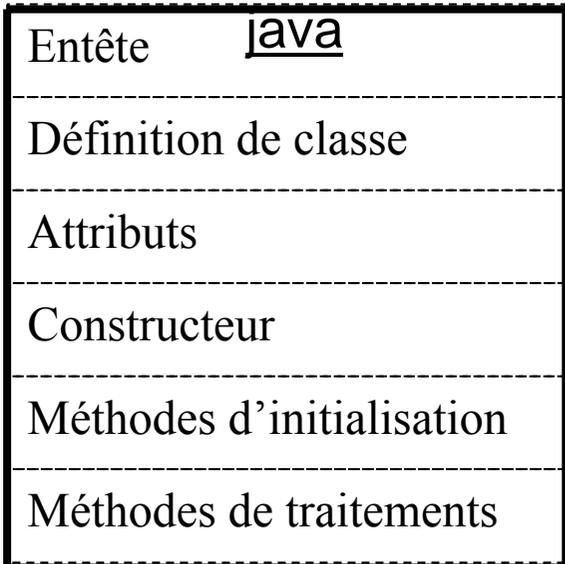
- Interface principale permettant de
 - Choisir un fichier XML
 - Générer pour un type de terminal donné le code source de la fenêtre correspondante et de son gestionnaire d'évènements
 - Compiler le code source généré



Diagramme de classes de l'environnement de génération



Structure d'une fenêtre

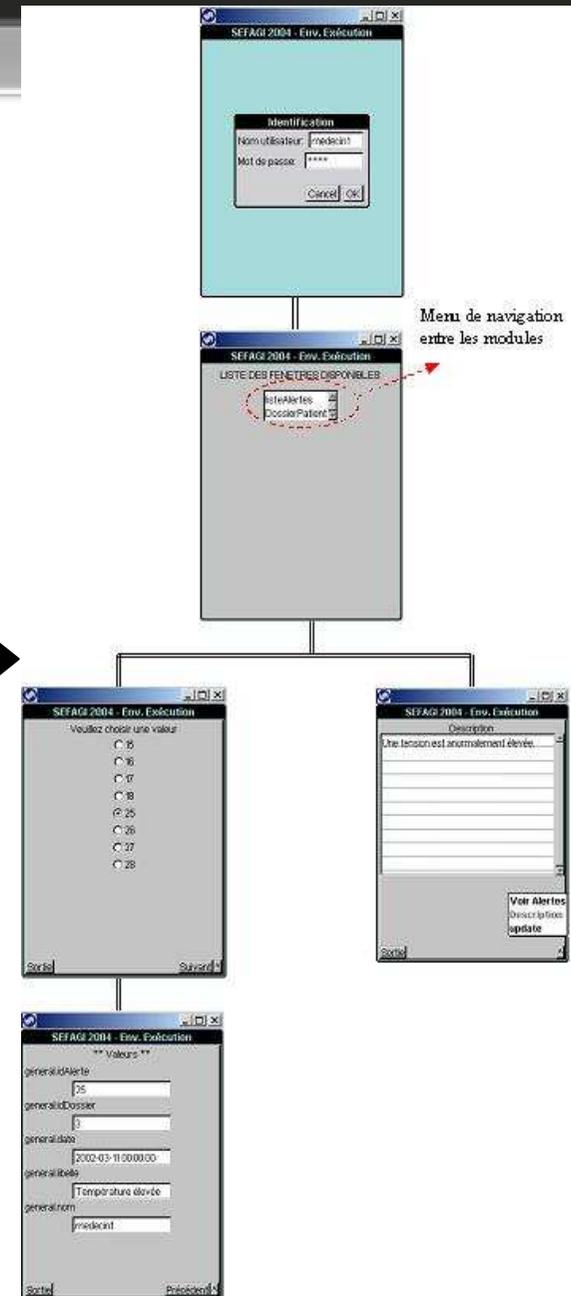


Menu de navigation

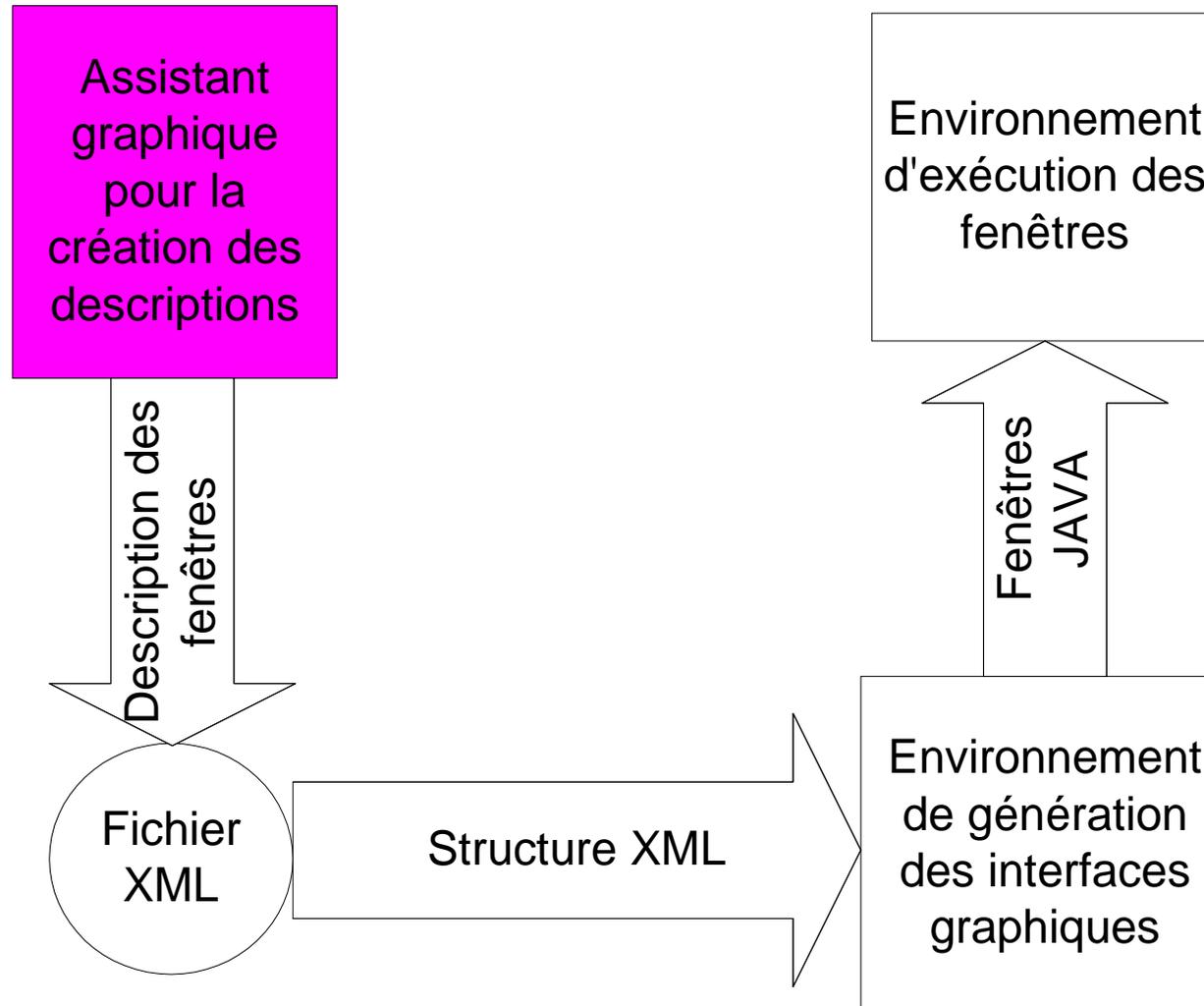
The screenshot shows the SEFAGI 2004 application interface. At the top, there is a navigation menu with three items: 'Quit', 'DossierPatient', and 'listeAlertes'. The 'listeAlertes' menu item is highlighted with a red dashed circle and an arrow pointing to the text 'Menu de navigation'. Below the menu is a table of alerts with the following data:

general.idAlerte	general.idDossier	general.date	general.libelle	general.nom
15	3	2002-02-24 00:00:00	Anomalie Cathéter	medecin1
16	3	2002-02-24 00:00:00	Poids élevé	medecin1
17	3	2002-02-24 00:00:00	Tension élevée	medecin1
18	3	2002-02-24 00:00:00	Anomalie Cathéter	medecin1
25	3	2002-03-11 00:00:00	Température élevée	medecin1
26	3	2002-03-11 00:00:00	Poids élevé	medecin1
27	3	2002-03-11 00:00:00	Tension élevée	medecin1
28	3	2002-03-11 00:00:00	Tension élevée	medecin1

Below the table is a 'Description' field containing the text: 'Une tension est anormalement élevée.' At the bottom, there are 'Commandes' buttons for 'description' and 'update'.

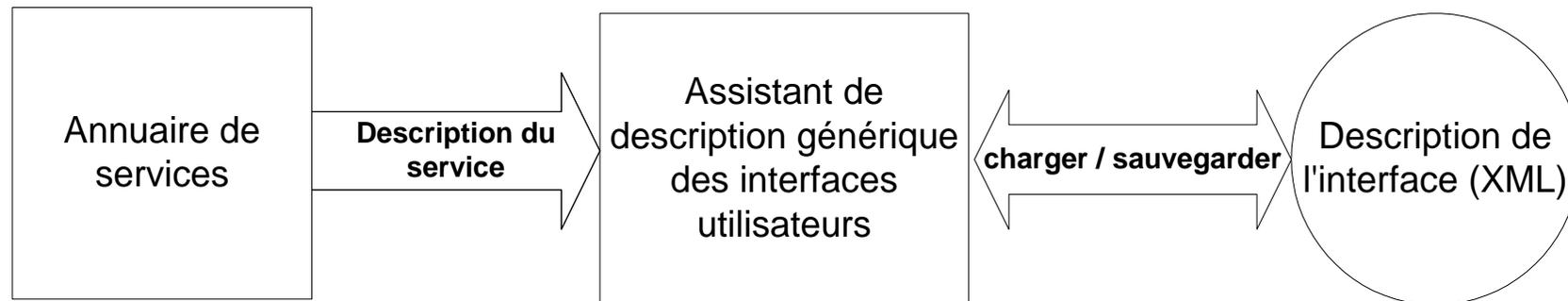


Assistant graphique pour la création des descriptions XML



Assistant graphique pour la création des descriptions XML

- Faciliter la création de la description des fenêtres.



Sicom 2002 - environnement de génération des interfaces graphiques

Assistant graphique pour la génération du fichier XML - DossierPatient

Fichier Gestion des panneaux Gestion des boutons Service Administration Aide

Dossier (Tableau) service1

composant1 type: numérique contenu: dossier.idDossier modifier

composant2 type: texte contenu: dossier.typeDialyse modifier

composant3 type: texte contenu: dossier.modeDialyse modifier

composant4 type: date contenu: dossier.dateCreation modifier

composant5 type: date contenu: dossier.dateModification modifier

composant6 type: texte contenu: dossier.nom modifier

composant7 type: texte contenu: dossier.prenom modifier

composant1 type: date contenu: temperatures.abscisses modifier

composant2 type: numérique contenu: temperatures.ordonnees modifier

composant1 type: texte contenu: images. modifier

Annuaire de services

- calcul
- medical
 - aleres
 - dossier
 - listeDossierDialyse.xml
 - valeursTemperature.xml
 - images
 - videos

nom: sample.ListeDossierDialyse

description: Ce service retourne la liste des paramètres de dialyse associés au dossier identifié par "dossier.idDossier"

Paramètre	nom	type	Paramètre	nom	type
	dossier.idDossier	entrée		general.idDossier	sortie
Valeur	entier	0	Valeur	entier	pas de valeur initiale
	dossier.dateCreation	sortie		dossier.dateModification	sortie
Valeur	date	pas de valeur initiale	Valeur	date	pas de valeur initiale

Annuler Affecter ce service au panneau courant Synchroniser les données avec le serveur

Choix du type du panneau

Tableau

Texte

Commande

Graphique-courbes

Graphique-batonnets

Images

Videos

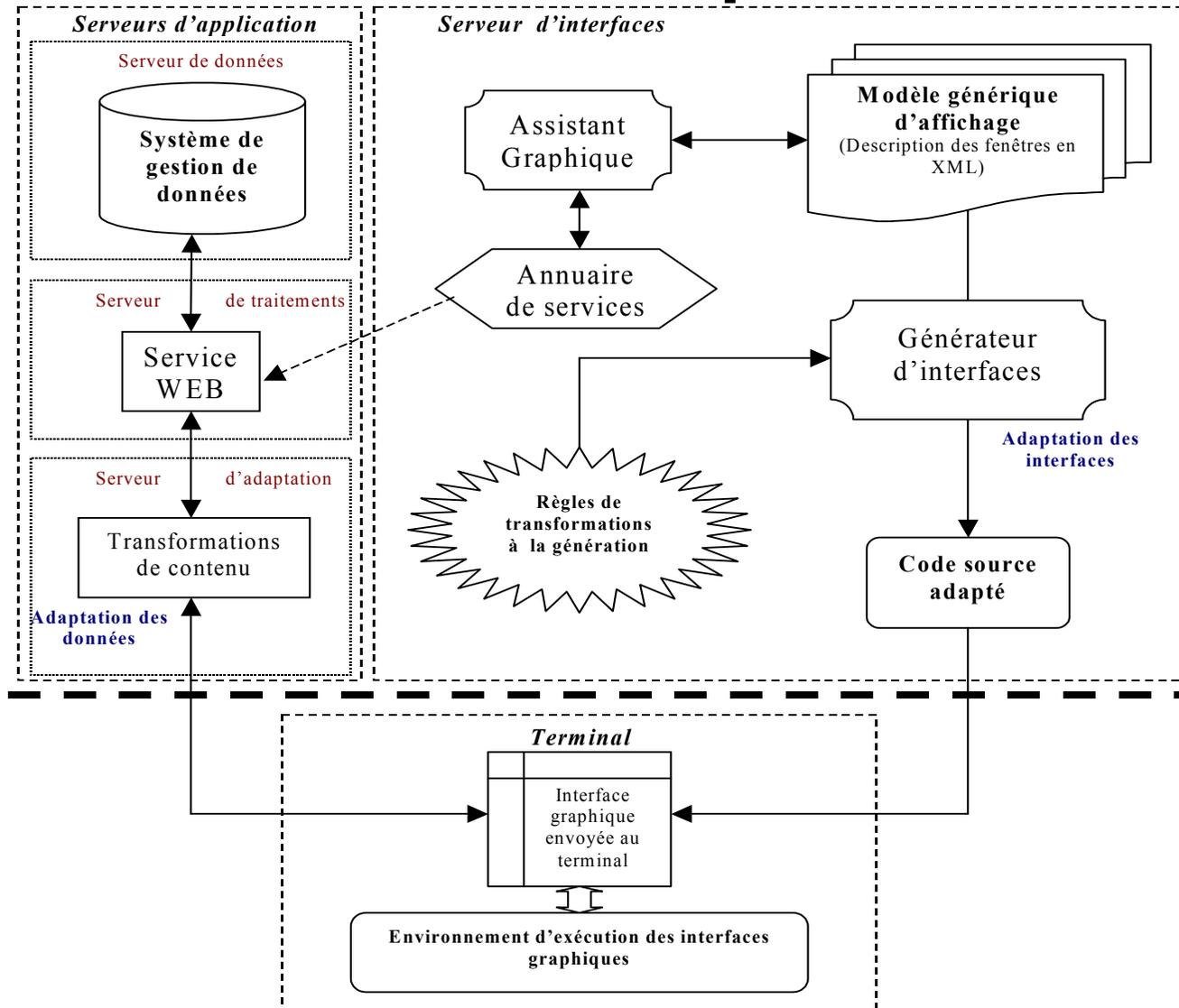
Ajouter Panneau

Annuler

aleres

Numéro Dossier	Date	Libellé
3	2002-02-24 00:00:00	Anomalie Cathéter
3	2002-02-24 00:00:00	Poids élevé
3	2002-02-24 00:00:00	Tension élevée
3	2002-02-24 00:00:00	Anomalie Cathéter
3	2002-03-11 00:00:00	Température élevée
3	2002-03-11 00:00:00	Poids élevé
3	2002-03-11 00:00:00	Tension élevée
3	2002-03-11 00:00:00	Tension élevée
1	2002-02-24 00:00:00	Anomalie Cathéter
1	2002-02-24 00:00:00	Anomalie Cathéter
1	2002-02-24 00:00:00	Anomalie Cathéter
1	2002-02-24 00:00:00	Anomalie Cathéter
1	2002-02-24 00:00:00	Anomalie Cathéter
1	2002-03-11 00:00:00	Anomalie Cathéter

Architecture technique de SEFAGI



Synthèse : types d'adaptation dans Sefagi

- Au cours de la génération (statiques)
 - Transformations de l'affichage (présentation)
 - Correspondance entre les composants du modèle et ceux des langages cibles
 - Transformations de disposition (présentation)
 - Disposition des composants dans les panneaux
 - Disposition des panneaux sur les écrans
 - Transformations de navigation (présentation)
 - Facilités de navigation pour l'utilisateur

Synthèse : types d'adaptation dans Sefagi

- Au cours de l'exécution (dynamiques)
 - Transformations de cohérence (données)
 - Groupage ou dissociation de données.
 - Gestion des points de reprise
 - Transformations de contenu (données)
 - Modification ou substitution d'une ressource par une autre plus adaptée
 - Transformations de transmission (traitements)
 - Protocole de transmission standard
 - Typage standard des données

Conclusion SEFAGI

- Simplicité
 - Avoir des interfaces graphiques spécifiques sans toucher à la programmation
 - Un fichier XML par fenêtre très court par rapport aux autres modèles existants
- Évolutivité
 - Enrichissement de la liste des panneaux
- Portabilité
 - Entièrement développée en JAVA
- Travail restant
 - Un générateur avec règles de transformation externalisées

Adaptation des services

- Les systèmes à services adaptables ont généralement trois parties [Cremene 04]
 - partie modifiable : le service adaptable
 - partie monitoring : évaluation en continu du service et de son contexte
 - partie contrôle : définit les ordres de reconfiguration en fonction d'une logique qui lui est propre
- Deux techniques d'adaptation habituelles
 - Adaptation par entités externes
 - Adaptation par réflexivité

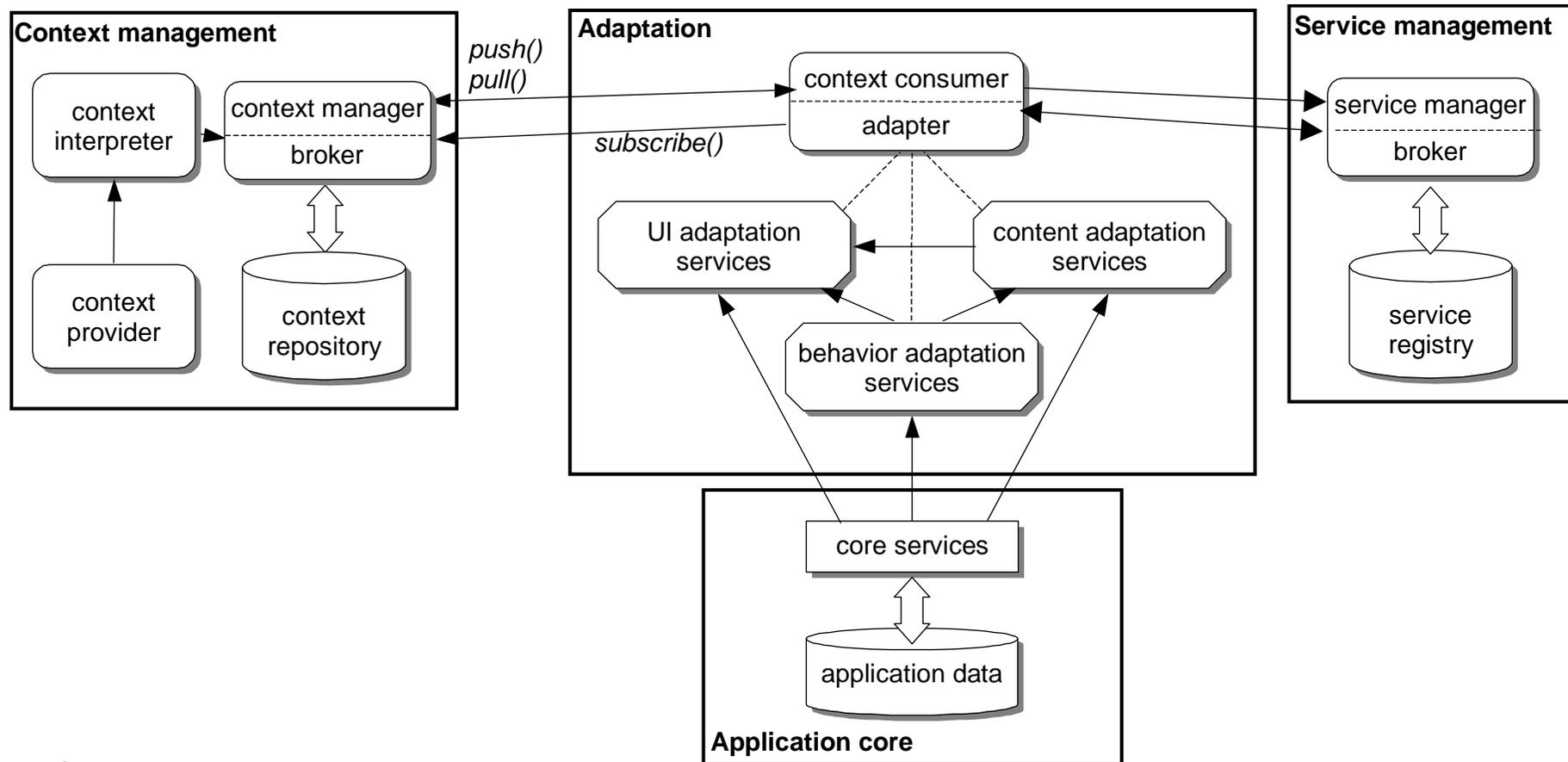
Adaptation de services

- Adaptation par entités externes
 - CAAD : T. Chaari, F. Laforest, A. Celentano design of context-aware applications based on web services Rapport de recherche LIRIS, octobre 2004 - RR-2004-033
- Réflexivité et adaptation par réflexivité
 - RAM : N.M.N. Bouraqadi-Saadani et al. « A reflexive infrastructure for coarse grained strong mobility and its tool-based implementation. » Int. Workshop on experiences with reflexive systems. Sept. 2001
 - DynamicTAO : F. Kon et al. « Monitoring, security and dynamic configuration with the dynamicTAO reflective ORB » Middleware 2000
- Adaptation mixte
 - AeDEN : F. Le Mouel « Environnement adaptatif d'exécution distribuée d'applications dans un contexte mobile » mémoire de thèse de doctorat en informatique, Université Rennes I, 1er décembre 2003

CAAD - Context-aware application design

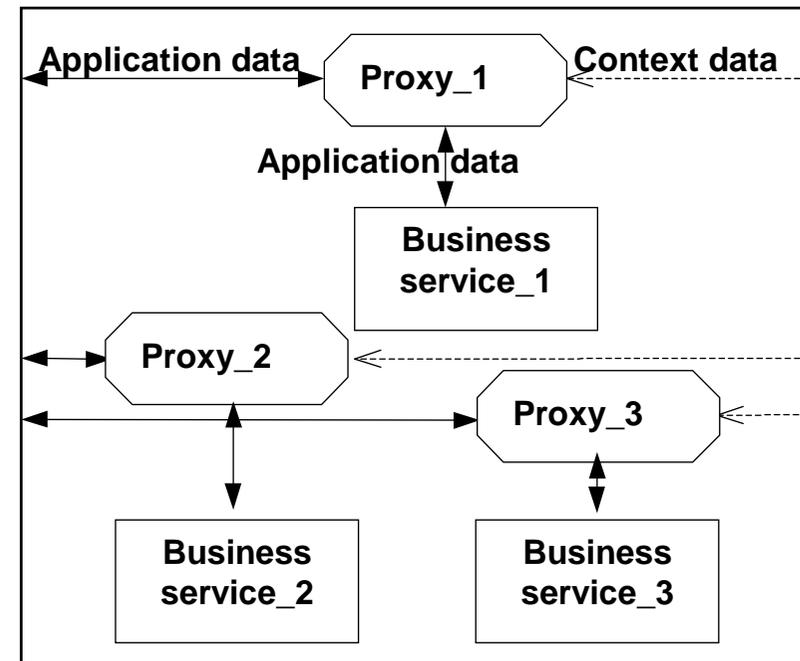
- Encapsuler une application existante avec une couche d'adaptation
 - sans modification de l'application d'origine
 - à l'aide de services web
- Adaptation
 - de l'interface utilisateur
 - des contenus
 - des services

CAAD : architecture



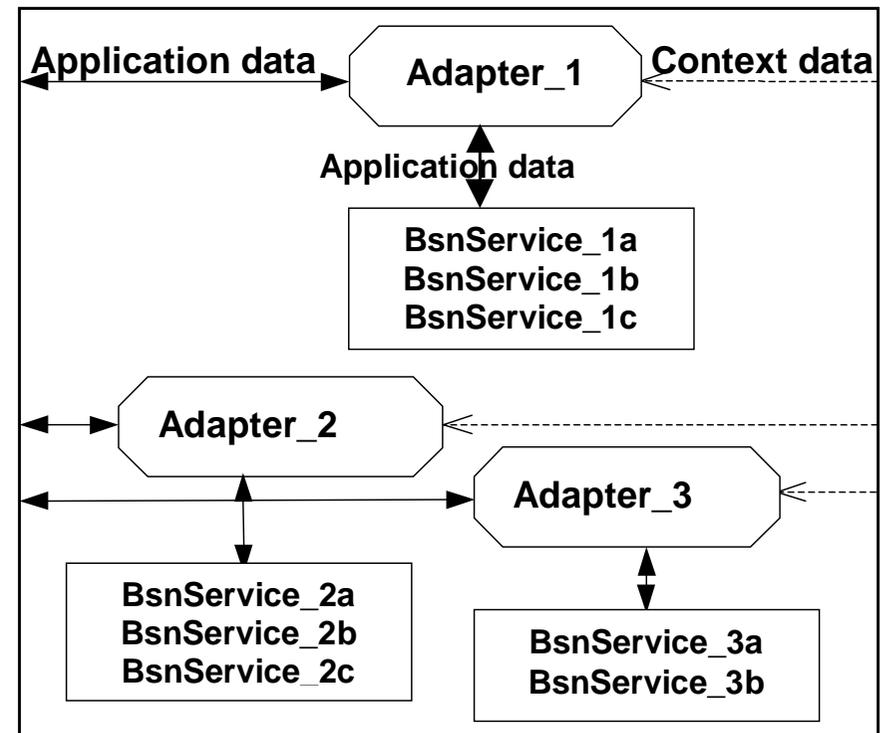
CAAD : adaptation par le DP proxy

- Un service gère les adaptations des entrées et sorties du service legacy
- Adaptation dédiée principalement aux adaptations de contenu



CAAD : adaptation par le DP strategy+

- Un service applique une politique de sélection d'une implantation d'un service parmi l'ensemble des services disponibles pour la fonctionnalité
- peut aussi effectuer lui-même une partie de l'adaptation (adaptation des entrées/sorties, pré- ou post-actions)



Adaptation dans les systèmes réflexifs

- Réflexivité
 - Capacité d'un système à se représenter, s'observer et agir sur lui-même
 - niveau méta décrivant les composants du système
- Introspection
 - donne la possibilité au système de connaître son état interne, lui permettant de raisonner et prendre des décisions
- Intercession
 - permet au système d'adapter son fonctionnement en modifiant son propre comportement

Adaptation dans les systèmes réflexifs

- Sujets d'adaptation
 - entités : méthodes, objets, composants
 - liaisons entre entités de base ou entre entités de base et entités méta
 - ensembles d'entités
- Moment d'adaptation
 - compilation : génération de code en fonction des méta-entités
 - chargement : altération du code compilé lors de son chargement ou modification de l'ensemble d'entités
 - exécution : accès dynamique au niveau méta, par l'intermédiaire de proxies, ou par le support d'exécution

RAM : système réflexif

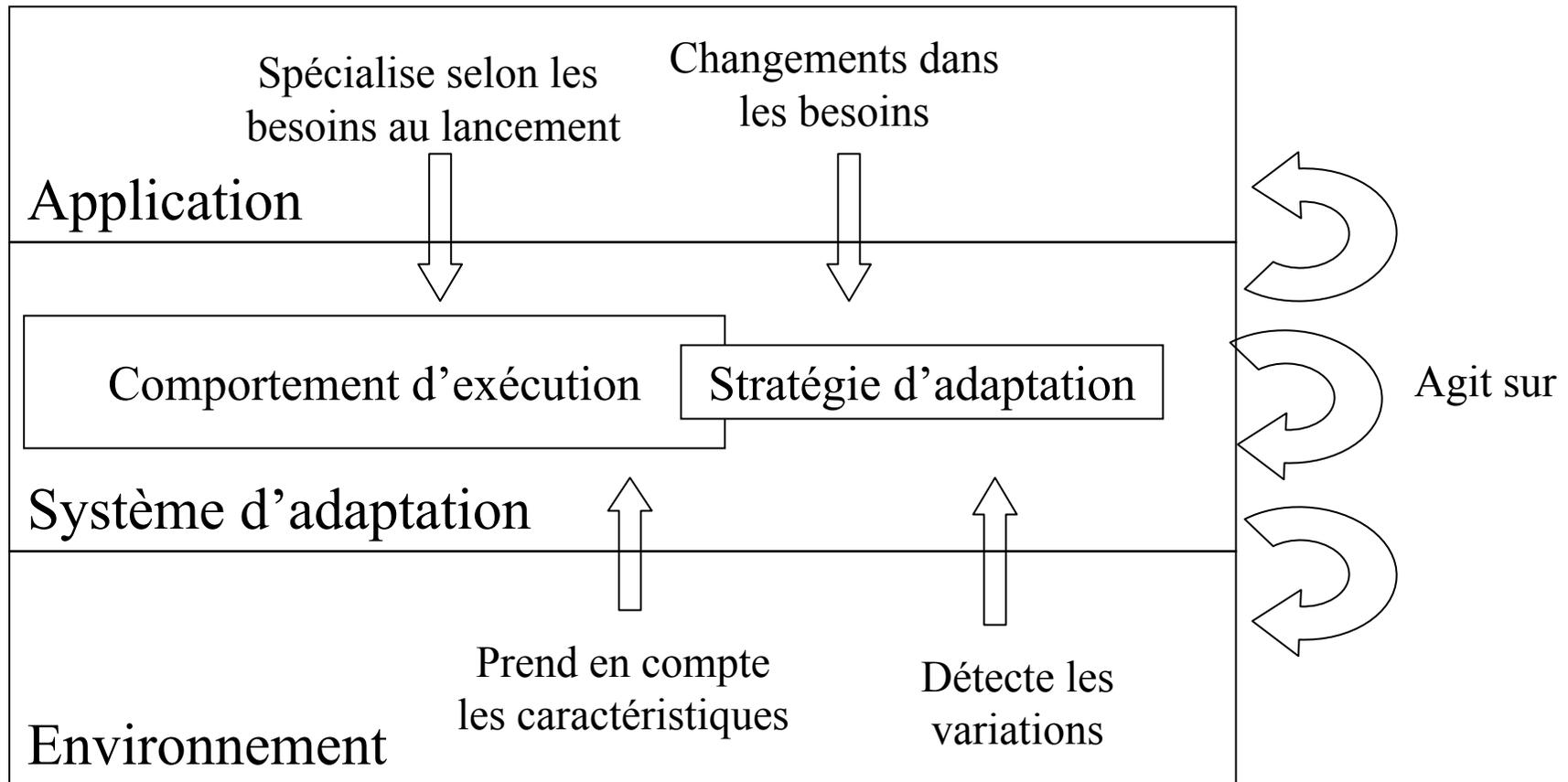
- [Bouraçadi et al 01] Reflexion for Adaptable Mobility
- Mobilité du code = aspect non fonctionnel, donc au niveau méta
- cluster = unité de mobilité de code, comportant
 - un ensemble d'objets applicatifs
 - une méta façade fournissant l'accès aux objets de description des politiques du cluster (migration, file d'attente de messages...)
 - références instanciées : références locales ou distantes à d'autres clusters (gérées par la méta-façade)
 - table de références : répertorie tous les objets instanciés, assure l'unicité des références au sein du cluster

DynamicTAO : middleware réflexif

- [Kon et al. 2000] ORB réflexif se basant sur CORBA
- ensemble de Component Configurators
 - chaque composant contient un Domain Configurator qui maintient des références entre le middleware et les composants dont il dépend (Servant Configurators)
 - le middleware contient un TAO Configurator qui maintient les stratégies du middleware (concurrency, scheduling...)
 - contient des points d'interception qui permettent de dériver les appels faits par les composants vers les implantations des stratégies
 - les implantations des stratégies peuvent être changées dynamiquement (chargement dynamique d'implantations)

AeDEN - Adaptation d'entités logicielles

- [Le Mouel 2003]

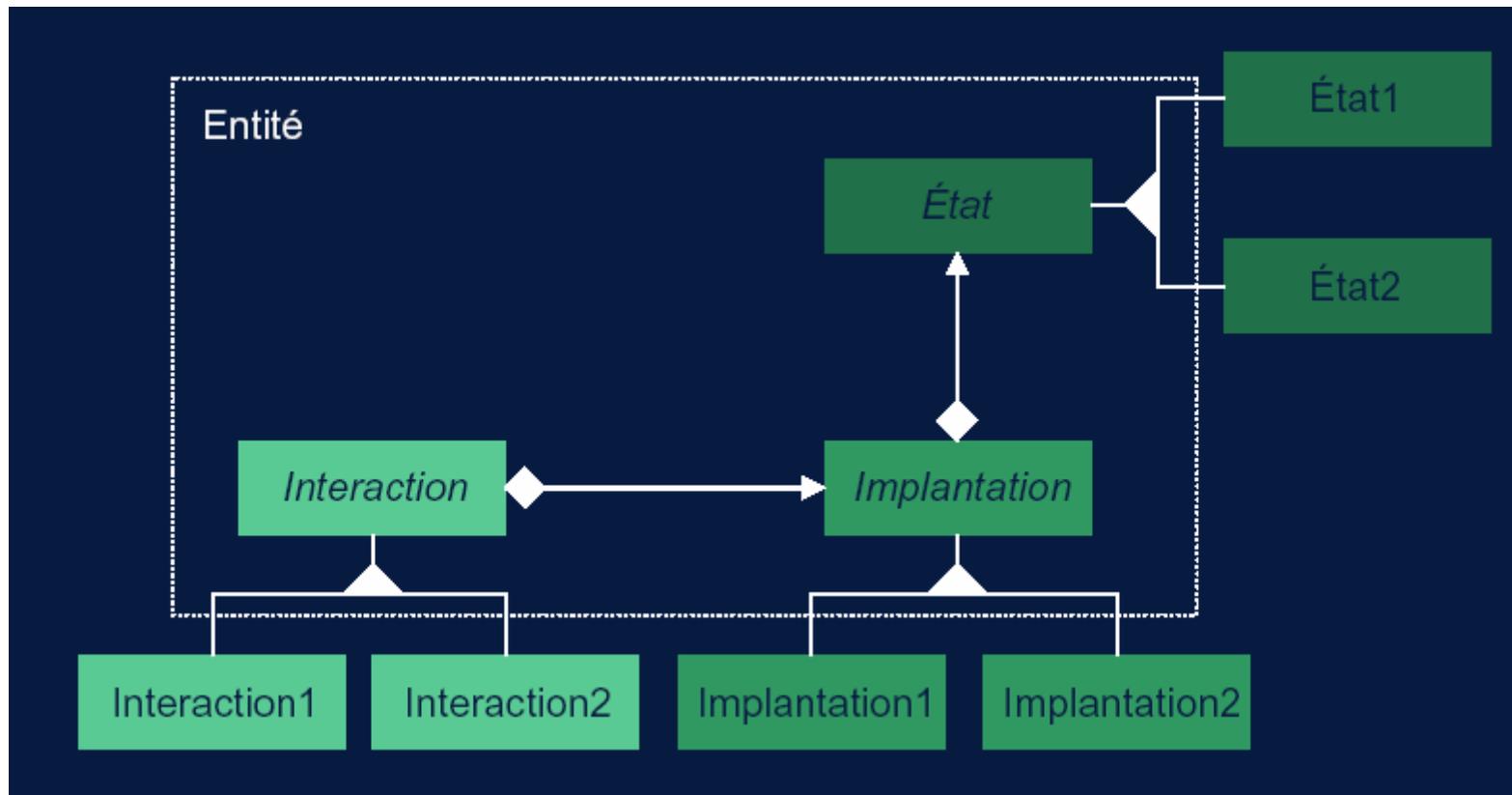


AeDEN - entité

- Entité = unité logicielle de conception
 - abstraite et spécialisable
 - Une fonctionnalité \Leftrightarrow une entité
 - un service \Leftrightarrow une entité spécialisée
- 3 aspects
 - AInteraction (communication avec les autres entités)
 - AImplementation (métier : traitements attendus)
 - AState (état courant interne de la partie métier)
- A chaque aspect correspond un ensemble d'implantations possibles

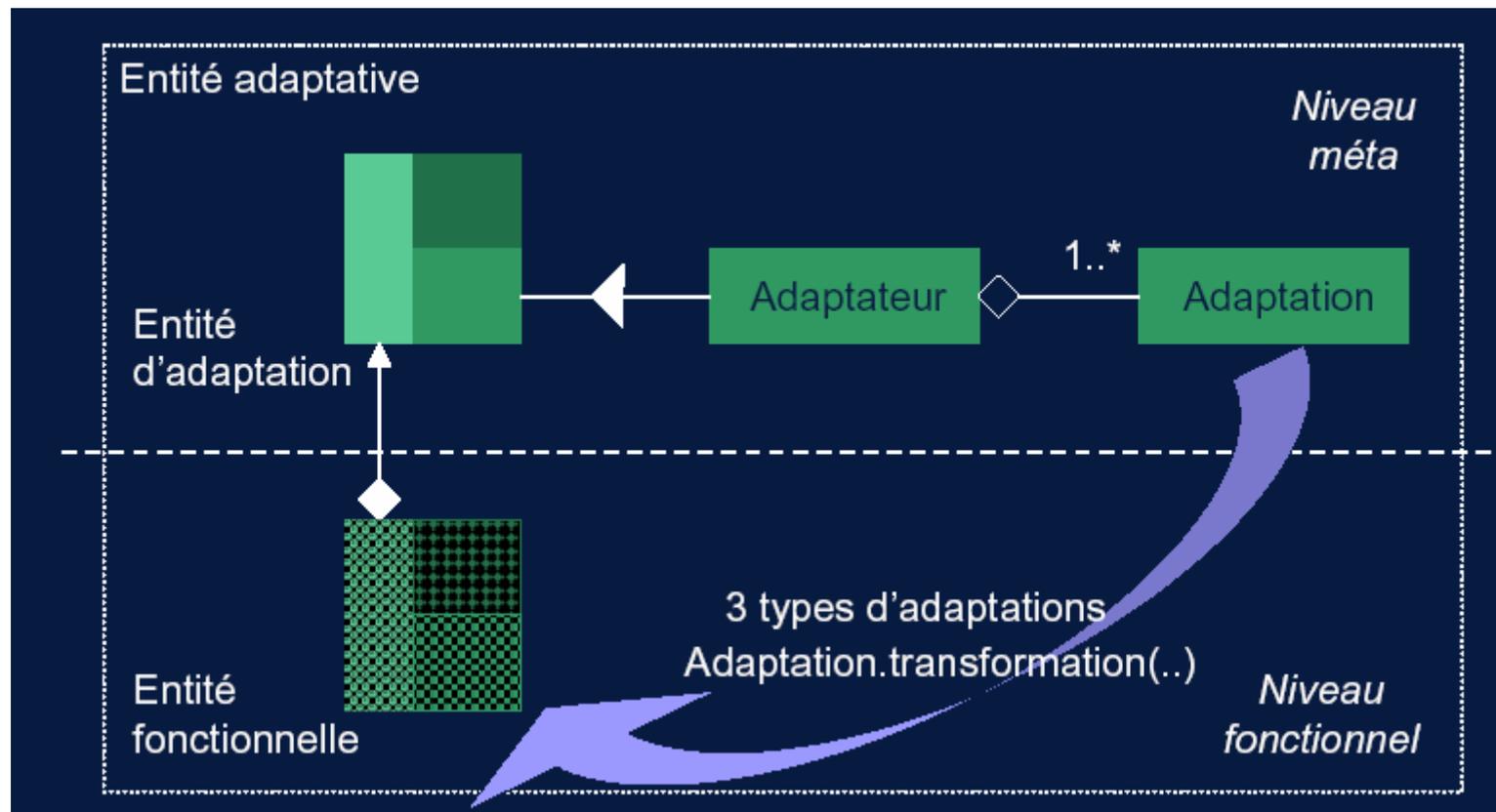
AeDEN - entité

- Entité abstraite + spécialisations possibles



AeDEN - entité adaptative

- Entité + entité d'adaptation
 - adaptations par introspection puis intercession



AeDEN - introspection et intercession

- Chaque entité et chaque entité adaptative implante les méthodes suivantes
 - getInteraction(), setInteraction()
 - getImplementation(), setImplementation()
 - getState(), setState()
- Chaque entité adaptative implante les méthodes
 - getFunctionalInteraction(), setFunctionalInteraction()
 - getFunctionalImplementation(), setFunctionalImplementation()
 - getFunctionalState(), setFunctionalState()

AeDEN - exemple d'adaptation

- Adaptation d'implantation

```
class ImplementationAdaptation extends StateAdaptation { // Adaptation d'implantation
// héritage de la déclaration de variable
// protected EntityAdapter entityAdapter;
...
// héritage de la déclaration de méthode
// public AState stateTransformation(AState st, String[] args);
...
public AImplementation implementationTransformation(AImplementation im, AState st, String[] args) {
// transformation d'une implantation
AImplementation tim; // transformed implementation
...
// Création d'une nouvelle instance d'implantation
tim = new Implementation(im, st);
// Ici, modifications de l'implantation tim
// Puis on prévient, par exemple, les entités ayant des dépendances fonctionnelles
...
return(tim);
}
```

AeDEN - exemple d'adaptation

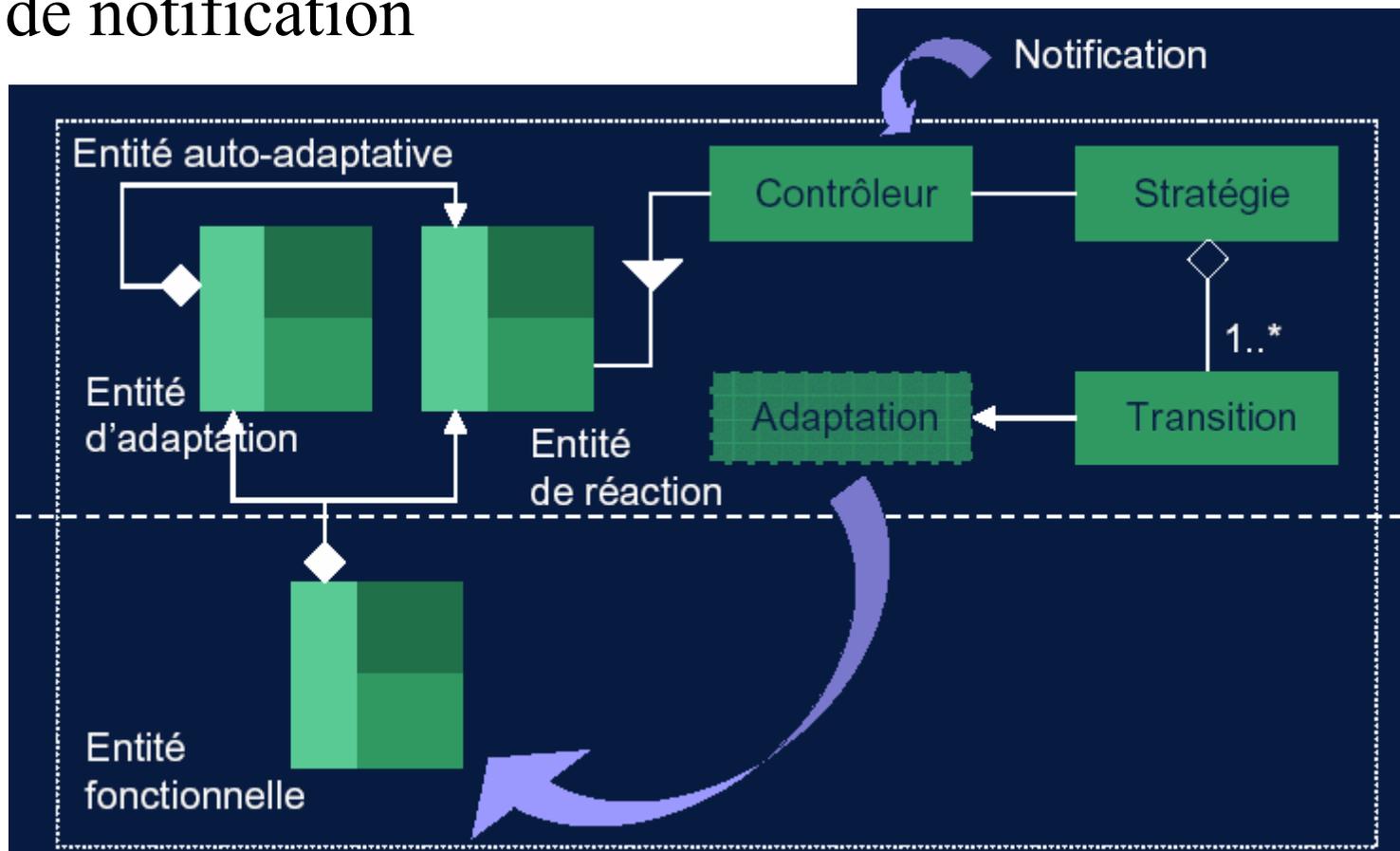
```

public Event applyAdaptation(String[] args) { // méthode d'application de l'adaptation
    AState cfs; // current functional state
    AState tfs; // transformed functional state
    AImplementation cfi; // current functional implementation
    AImplementation tfi; // transformed functional implementation
    ...
    // récupération des éléments de la structure de l'entité fonctionnelle
    cfs = entityAdapter.getFunctionalState();
    cfi = entityAdapter.getFunctionalImplementation();
    ...
    // transformations de ces éléments
    tfs = stateTransformation(cfs, args);
    tfi = implementationTransformation(cfi, tfs, args); // on reprend l'état transformé
    ...
    return(entityAdapter.setFunctionalImplementation(tfi));
}
...
}

```

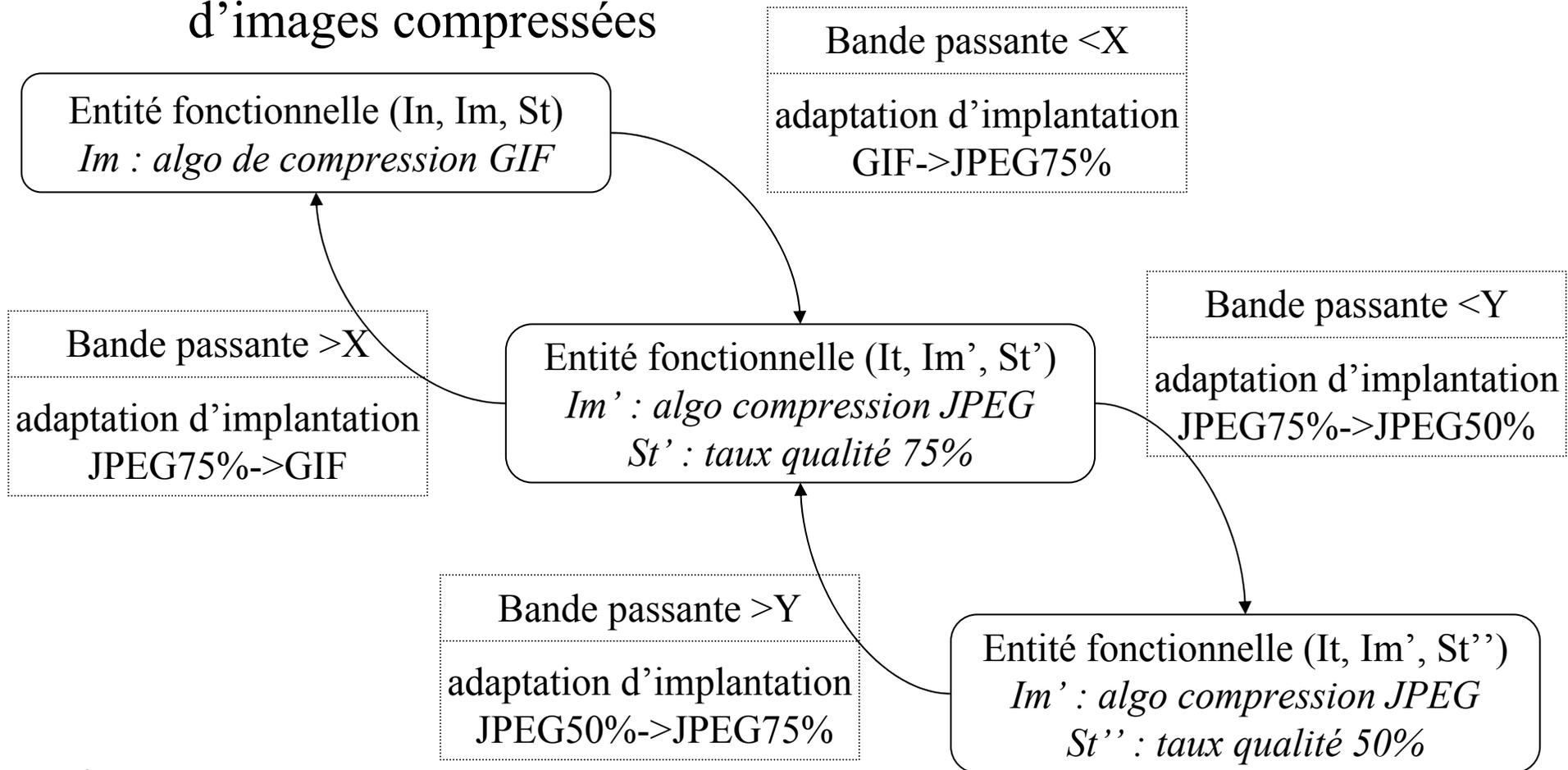
AeDEN - entité adaptative et réactive

- Entité adaptation + entité de réaction liée à un service de notification



AeDEN - exemple de stratégie

- Stratégie d'adaptation pour un service de transmissions d'images compressées



Conclusion sur l'adaptabilité

- Des besoins généraux [Le Mouel 2003]
 - généricité : utilisation par plusieurs types d'applications
 - modularité : découpage et découplage
 - prise en compte du contexte
 - évolutivité : intégrer de nouvelles technologies et de nouvelles fonctionnalités
 - dynamicité : réaction aux changements sans arrêt du système
 - efficacité : performance et stabilité

Synthèse : relations entre les niveaux

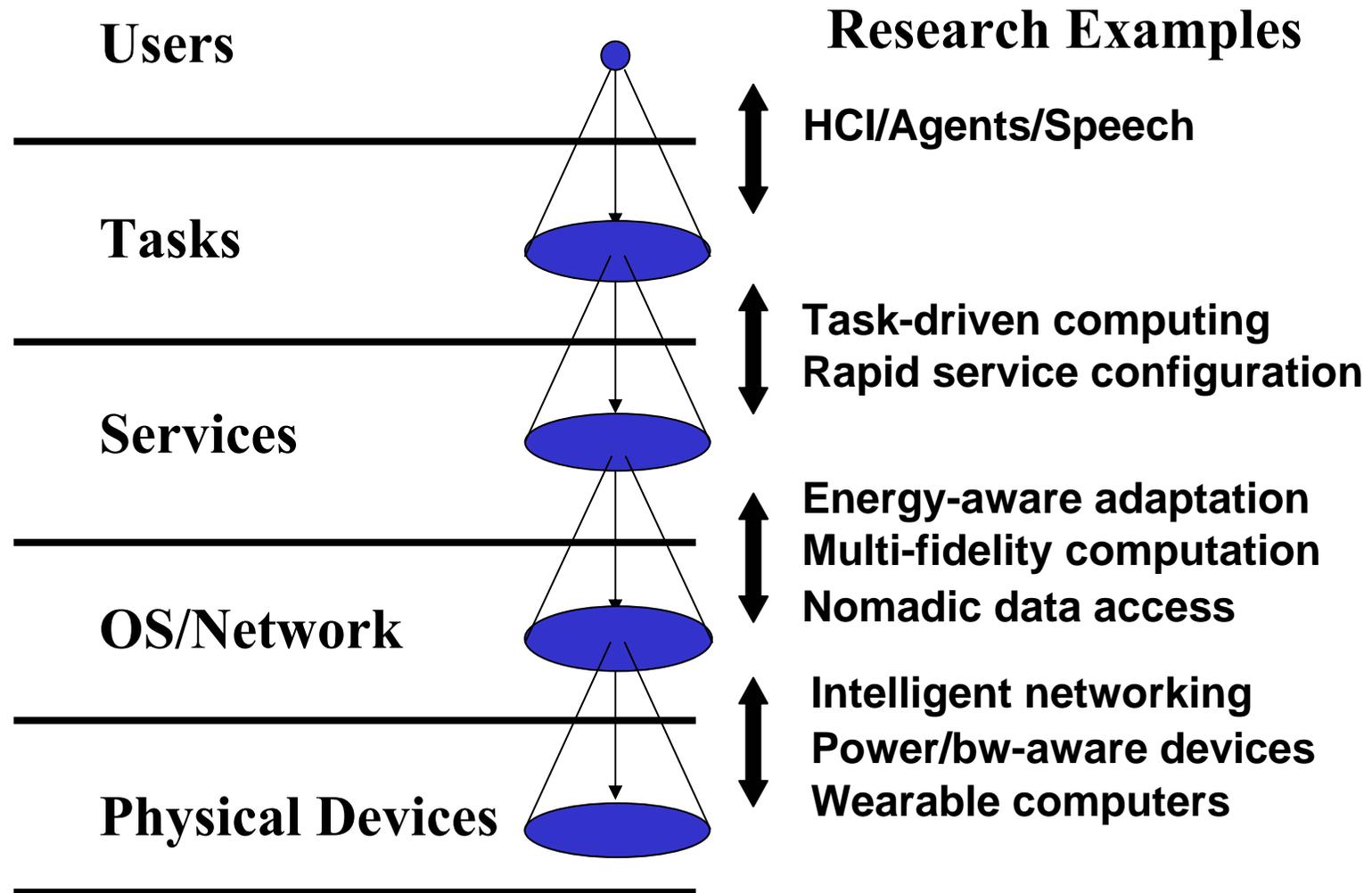
- 3 niveaux de recherche : matériel et système, réseaux, système d'information
 - Pour une adaptation complète et cohérente,
 - Associer des informations provenant des trois niveaux
 - Utiliser des techniques d'adaptation « traversant » les niveaux
- => une communication inter-niveaux des informations de contexte et des services d'adaptation de chaque niveau
- => L'utilisation d'un découpage en niveaux est-elle toujours intéressante ou apporte-t-elle plus de problèmes qu'elle n'en résout?

Exemples de projets

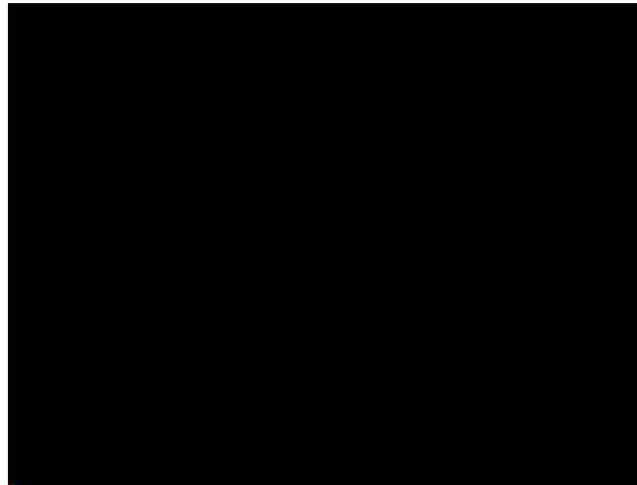
AURA Distraction-free Ubiquitous Computing

- M. Satyanarayanan, Carnegie Mellon University
 - Aura's goal is to provide each user with an invisible halo of computing and information services that persists regardless of location. Meeting this goal will require effort at every level: from the hardware and network layers, through the operating system and middleware, to the user interface and applications.
 - Project Aura will design, implement, deploy, and evaluate a large-scale system demonstrating the concept of a “personal information aura” that spans wearable, handheld, desktop and infrastructure computers.
- <http://www-2.cs.cmu.edu/~aura>

Aura - Mots-clés [Satya 2000]

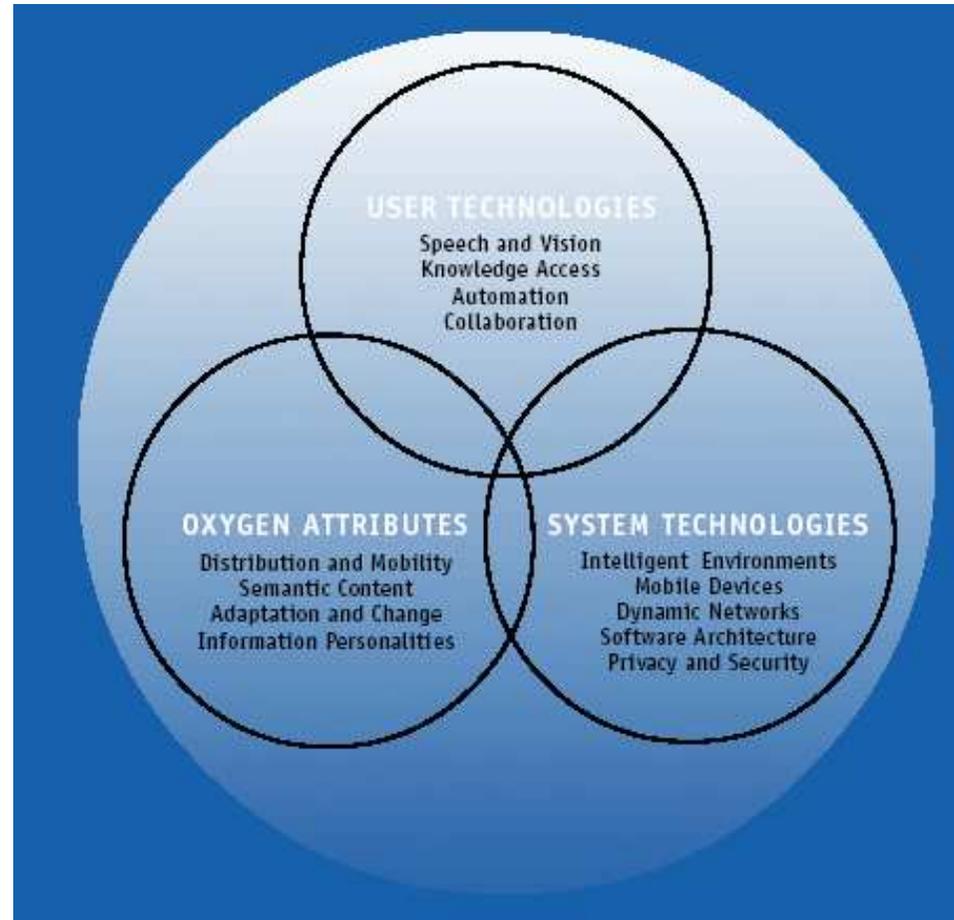


Aura - Démo



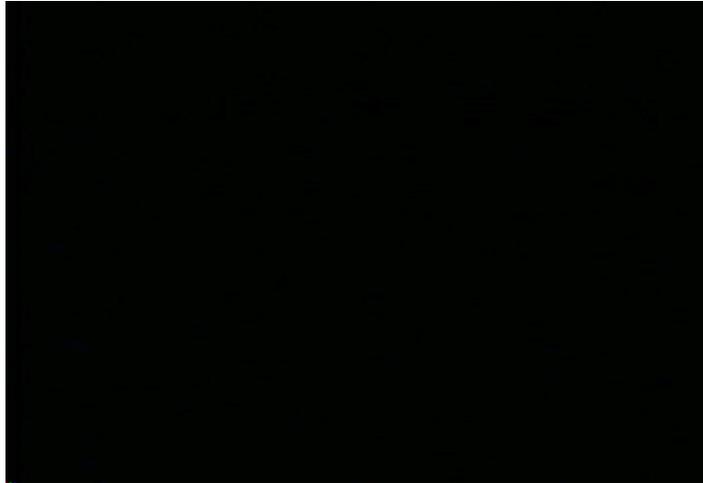
Oxygen

- MIT



– <http://www.oxygen.lcs.mit.edu/>

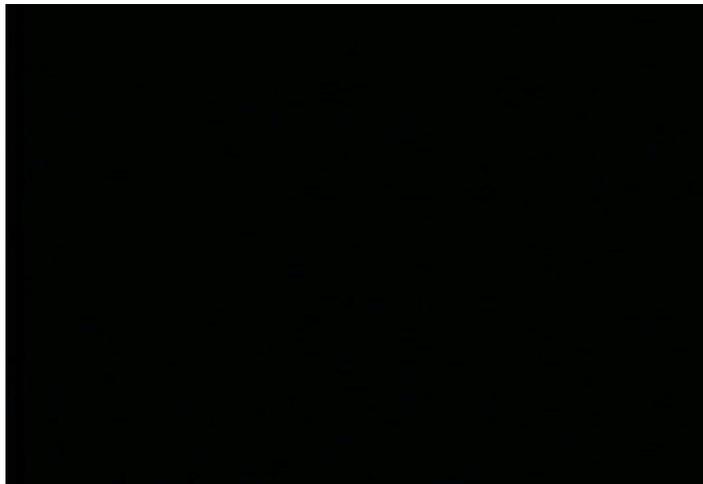
Oxygen - quelques démos



Visitor
guide



Multilingual
speech



Software
proxy



Intelligent Meeting
Room - collaboration

Sentient Computing project

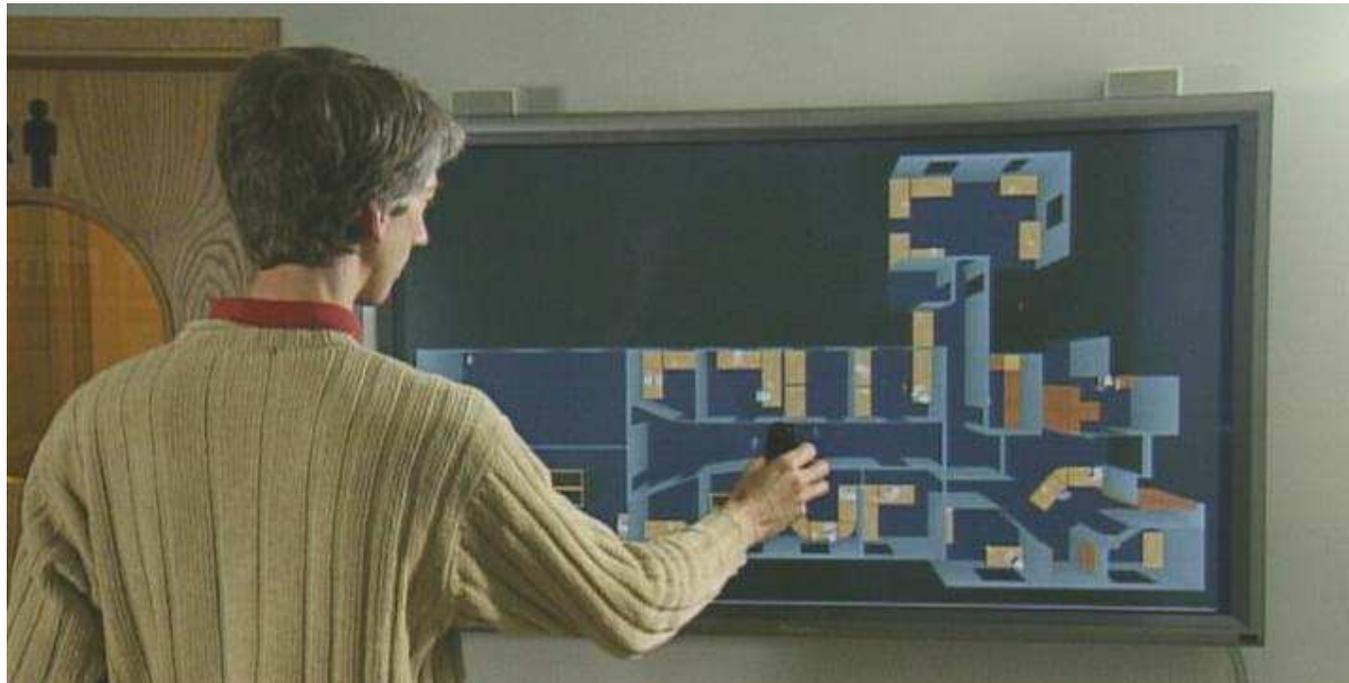
- AT&T Laboratories Cambridge
 - <http://www.uk.research.att.com/spirit/>
- Sentient Computing : using sensors and resource status data to maintain a model of the world which is shared between users and applications
- Bats
 - small (8cm long) devices, with a unique id, an ultrasound transmitter and a radio transceiver, 2 buttons and a beeper
 - located by a central controller, and the world model stores the correspondence between bats and their owners, applying algorithms to the bat location data to determine the location of the person or object which owns it.

Sentient Computing project

- Objets CORBA
 - données de localisation et d'état des ressources
 - tout objet du monde réel (personnes, ordinateurs, téléphones..)
- Chaque objet fournit état + API de l'objet réel
- Les objets gèrent eux-mêmes les transactions, les sessions, la distribution des événements...

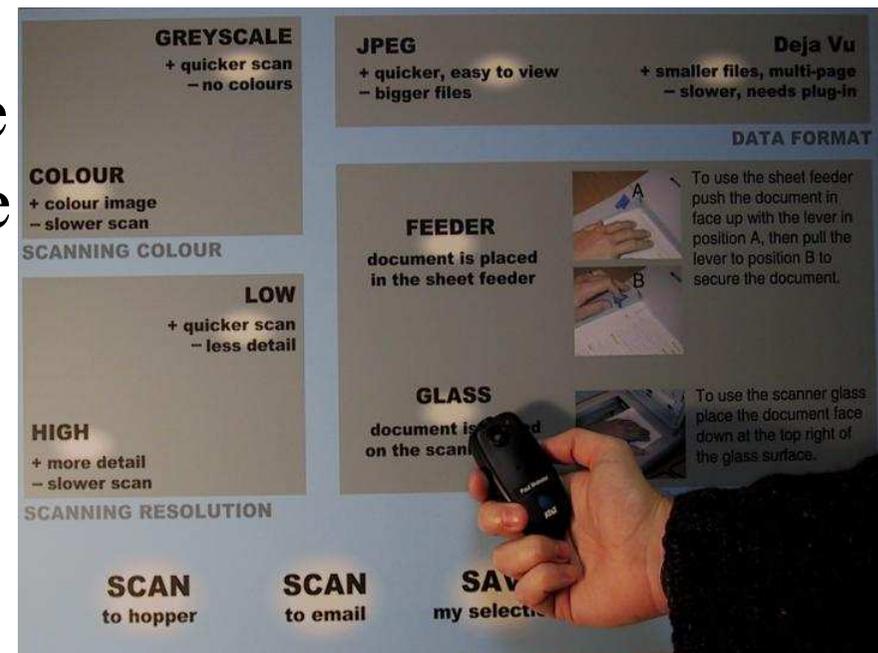
Sentient - Applications

- A browser allows a user to see what's going on anywhere in the building
- Phone someone by pointing at his representation on the 3D model



Sentient - Applications

- Follow-Me : simply teleport my desktop to another PC using my bat, and then teleport away when I'm finished.
- Smart Posters : create a 'button' anywhere in the environment. a button can be a small space anywhere in a building .To press the button, the user just puts the bat on the label and clicks a button on the bat.



Portolano

- <http://portolano.cs.washington.edu>
 - an infrastructure based on mobile agents that interact with applications and users. Data-centric routing automatically migrates data among applications on the user's behalf. Data thus becomes “smart,” and serves as an interaction mechanism within the environment.
- User Interfaces, new interaction modes
- Network Infrastructure
- Distributed Services

Portolano - Mots-clés

- Connecting the physical world to the world-wide information fabric
 - instrument the environment: sensors, locators, actuators
 - universal plug-and-play at all levels: devices to services
 - optimize for power: computation partitioning, communications optimization
 - intermittent communication: new networking strategies
- Get computers out of the way
 - don't interfere with user's tasks
 - diverse task-specific devices with optimized form-factors
 - wide range of input/output modalities
- Robust, trustworthy services
 - high-productivity software development
 - self-organizing, active middleware, maintenance, monitoring
 - higher-level, meaningful services

Endeavour

- The Endeavour Expedition: Charting the Fluid Information Utility
 - specification, design, and prototype implementation of a planet-scale, self-organizing, and adaptive Information Utility.
 - ability of processing, storage, and data management functionality to arbitrarily and automatically distribute itself among Information Devices and along paths through scalable computing platforms integrated with the network infrastructure, compose itself from pre-existing hardware and software components, satisfy its needs for services while advertising the services it can provide to others, while negotiating interfaces with service providers while adapting its own interfaces to meet "contractual" interfaces with components it services.
- <http://endeavour.cs.berkeley.edu/>

Conclusion

Conclusion

- Autres domaines intéressés par les systèmes pervasifs
 - Interactions homme-machine (interactions multimodales entre autres)
 - Agents logiciels (pro-action)
 - Intelligence artificielle (décision et prévision)
- L'informatique pervasive va avoir un grand impact sur notre vie de tous les jours et sur la société
 - Conséquence économiques, sociales et culturelles?

Perspectives

- Améliorations
 - Context-awareness et adaptabilité
 - Architectures P2P
- Lacunes
 - Outils de test – simulation – validation
 - Pro-action

UE connexes du master

- Système
 - Adaptation et Qualité de Service
 - Objets communicants
- Réseau
 - Conception de systèmes embarqués complexes
 - Réseaux sans fil et réseaux mobiles
 - Conception de serveurs d'applications ouverts (P2P)
- Système d'information
 - SI mobiles
 - (Grilles de données)
 - (SI et mémoire d'entreprise)

Références bibliographiques

- M. Weiser « The computer for the twenty-first century », Scientific American, sept 1991:94-104
- M. Satyanarayanan « Pervasive computing : Visions and challenges », IEEE Personal Communications, aug. 2001:10-17
- D. Saha & A. Mukherjee « Pervasive computing : a paradigm for the 21st century », IEEE Computer journal, march 2003:25-31
- S. Frénot « Cours Middleware », Dpt Télécommunications, INSA de Lyon, 2004
- T. Starner « Power and heat in ubiquitous computing », Summer school on ubiquitous and pervasive computing, 2002
- M. Franklin « Challenges in ubiquitous data management », Informatics : 10 years back, 10 years ahead. LNCS 2000, R. Wilhelm (ed.), Springer Verlag, 2001
- A.K. Dey « Building context-aware applications », presentation at Dagstuhl summer school, august 2003

Références bibliographiques

- D. Salber, A.K.Dey, G.D. Abowd « The context Toolkit : aiding the development of context-enabled applications » ACM CHI 99
- Strang & al. Service interoperability on context level in ubiquitous computing environments. Int conf. On Advances in infrastructure for electronic business, education, science, medicine and mobile technologies on the internet, L'Aquila, Italy, jan. 2003
- M. Cremene et al. « Adaptation dynamique de services », Decor '2004, Grenoble, octobre 2004, pp 53-64
- M. Satyanarayanan "Research Challenges in Project Aura" Keynote address at the Ninth IEEE International Symposium on High Performance Distributed Computing, Pittsburgh, PA, August 2000