

Marcel CREMENE Univ. de Cluj Napoca

Michel RIVEILL Univ. de Nice – Sophia Antipolis

Christian MARTEL Univ. de Savoie



# Objectif:

 Nous souhaitons pouvoir modifier l'architecture de services après leur construction - adapter des services.

 les services sont assemblés par des composants réutilisables



- Pourquoi adapter des services
- 2. Un modèle d'architecture pour l'adaptation des services
- 3. Travaux expérimentaux
- 4. Évaluation et perspective



Un service doit rendre service ©, c'est-à-dire est adapté aux besoins et à l'environnement de celui qui l'utilise.

### Contraintes :

- matérielles : PDA, réseau à faible débit
- environnementales : environnement calme, bruyant
- liées à la personnes : déficients visuels, enfants, personnes étrangères

### Ces contraintes peuvent être variables :

- on ne peut pas toutes les prédire a priori
- on ne peut pas construire une nouvelle version du service pour chaque ensemble de contraintes



### Scénario: un service de forum

- Soit un service de forum construit pour :
  - des utilisateurs anglophones
  - accès uniquement par le Web
- Situation 1 : un utilisateur nonanglophone souhaite utiliser ce forum...
- Situation 2 : un utilisateur accède au service avec un téléphone mobile...



# Approches possibles pour adapter des services

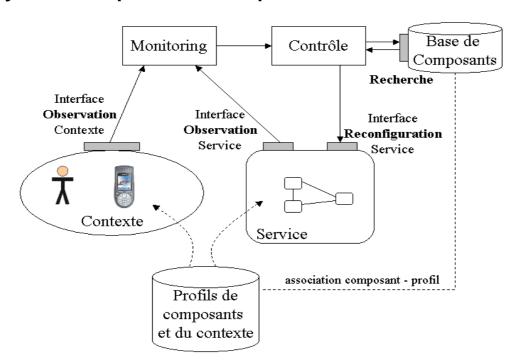
 Construire le service en anticipant les différents contextes possibles (langue, terminaux)

Adapter, dynamiquement, le service à des nouveaux contextes que l'on découvre en cours d'exécution

# 2. Architecture d'un système adaptatif

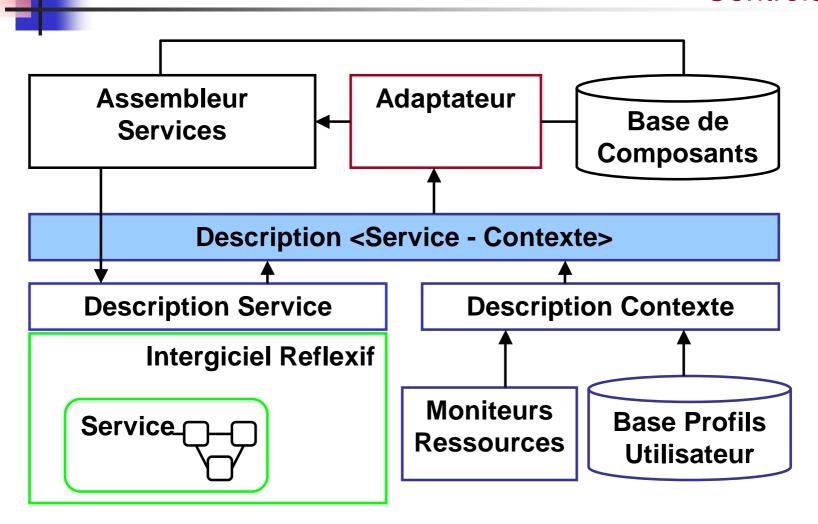
Système adaptatif = a)des *éléments configurables* dynamiquement, b)*monitoring*, c)*contrôle* 

Un système pour l'adaptation de services :



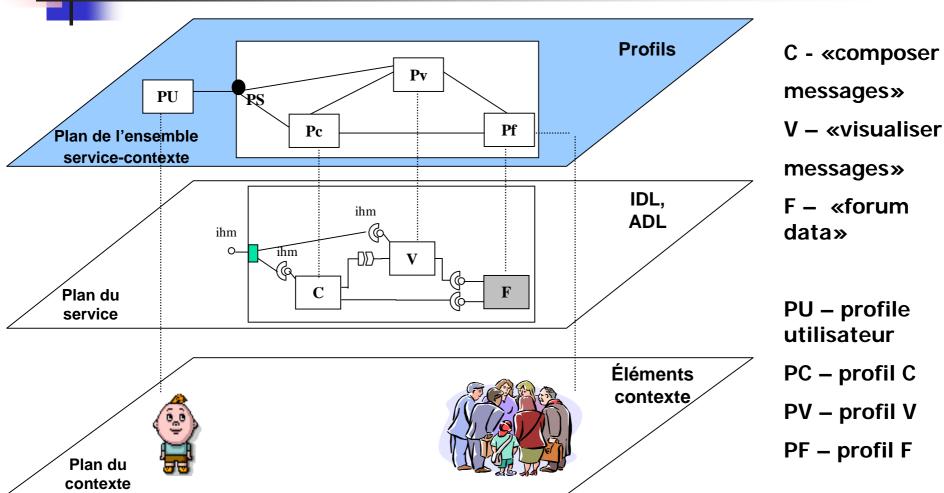
# Architecture proposée

Elém. config. Monitoring Contrôle





### Ensemble service-contexte



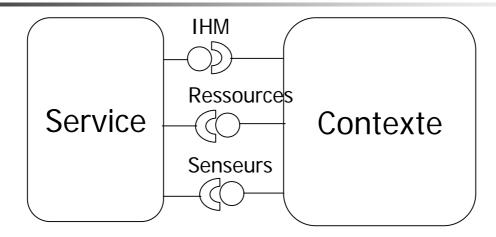


### Profils - contexte

- Contexte hiérarchie de paramètres
  - Utilisateur :
    - Langue
    - Age
  - Terminal:
    - Mémoire disponible
    - Écran
      - Taille
      - Couleurs
  - Environnement :
    - Bruit externe



### Liens service - contexte

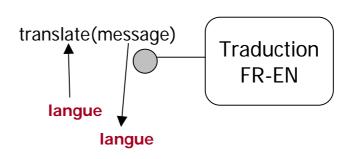


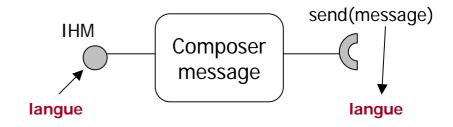
### Points d'association (entrés ou sorties) :

- Composant entier (ex. mémoire)
- Interface métiers ou IHM (ex. langue)
- Interface/méthode/argument ou valeur retournée

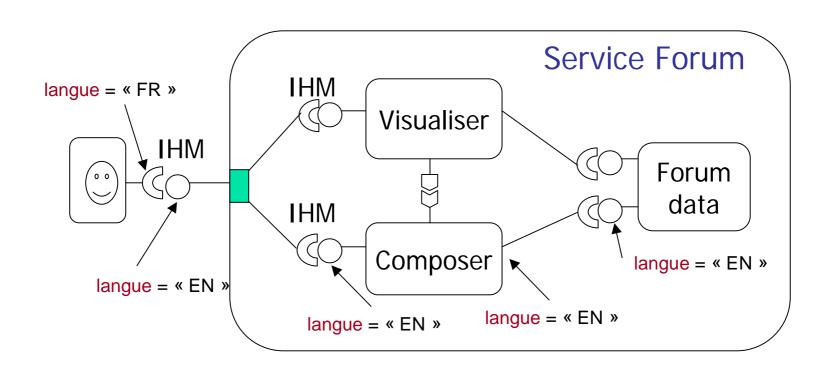
## Profils - composants

```
profile comp_name = « Traduction_FR-EN »>
 <point id = « 1 », type = « input »>
    <interface>Translation</interface>
    <method>translate</method>
    <argument>message</argument>
    <condition>langue = 'FR' </condition>
  </point>
 <point id = « 2 », type = « output »>
    <interface>Translation</interface>
    <method>translate</method>
    <condition>langue = 'EN' </condition>
  </point>
</profile>
```

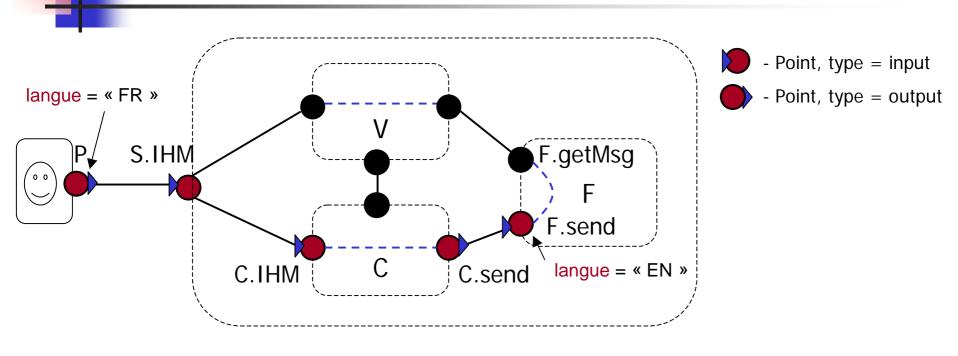


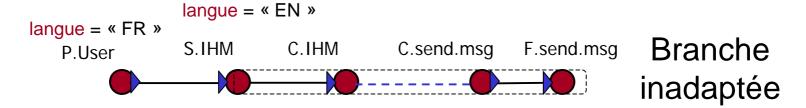


## Composition de profils



# Graphe de propagation du paramètre langue







L'opération de composition dépende de la nature du paramètre :

Langue	: =, attribution
Mémoire	+, adition
Taille écran	+, addition
Bande passante	+, adition ou
	Max, maximum
Bruit	+, addition



## Compatibilité de profils

 Les opérateur qui vérifient la compatibilité des profils dépendent de la nature du paramètre

Langue	C.langue = S.langue, égalité
Mémoire	C.mem > S.mem
Taille écran	C.taille > S.taille
Bande passante	C.bande > disponible
Bruit	C.bruit < S.bruit



## Algorithme d'adaptation

- Vérification : axiomes d'adaptation dans le graphe (1 graphe pour chaque paramètre)
- Recherche de solutions : pour chaque branche inadapté, chaque nœud (interposition)
- Application de solutions : mettre l'utilisateur dans la boucle, techniques d'adaptation



# 4. Travaux expérimentaux

- Configuration dynamique : ISL-Noah, Java Reflect, composants CCM, ADL
- Monitoring et description :
  - Contexte : BD profils utilisateur, détecteur langue, [CC/PP]
  - Service : reflexivité, [XML profils]
- Contrôle : algorithme de recherche dans un graphe, application -> ISL
- Service : Java, XML-RPC, HTTP/HTML



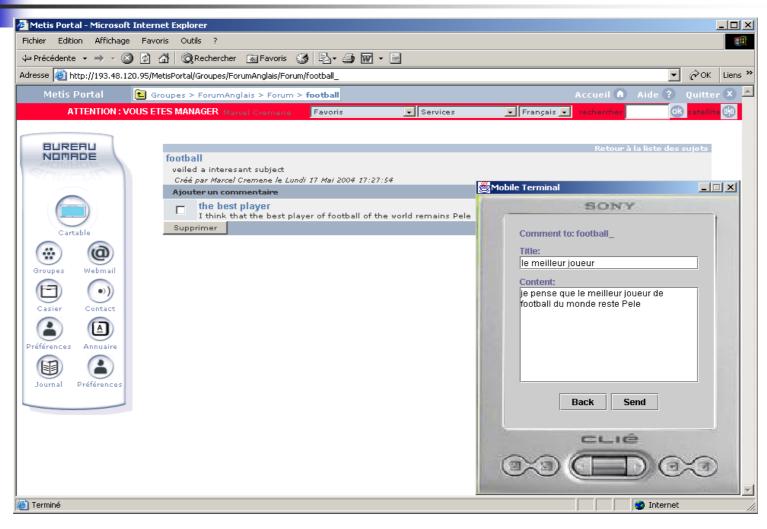
# ...captures d'écran



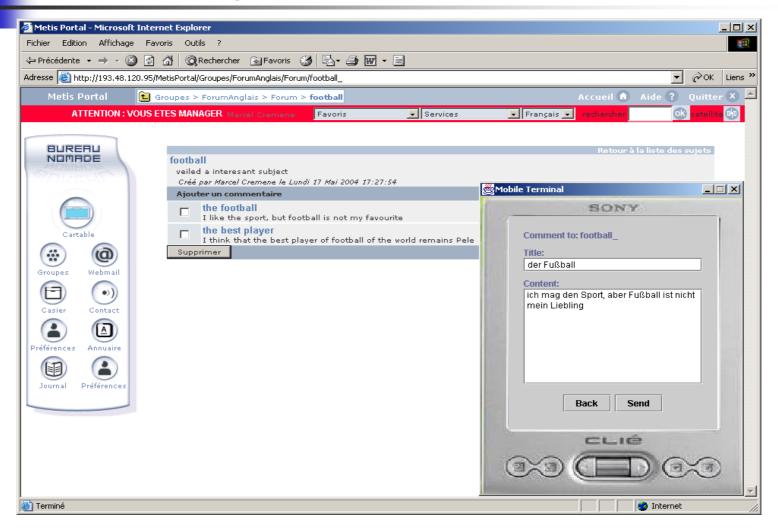




# ...prototype



# ...prototype





### Contributions

- Architecture générale du système
- Partie modifiable : intégration de solutions existantes (réflexivité)
- Monitoring et description : modèle ensemble service-contexte, profils, composition de profils
- Contrôle : alternative aux règles particulières -> d'adaptation non-anticipée
- Prototype forum



## Désavantages

- Complexité
- La composition et les axiomes dépendent de la nature de paramètres
- Une nomenclature commune (ou ontologie) doit être respectée pour les noms de paramètres
- Les constructeurs de composants doivent préciser aussi leur profils
- Testé uniquement pour solution de type interposition de composants



- Architecture capable d'adapter des services à des nouveaux contextes
- L'ensemble service-contexte est « compréhensible » aussi par la machine et pas uniquement par un programmeur
- Un pas vers la composition autonome et non-anticipée basée sur la sémantique de l'ensemble service-contexte



### Perspectives

- Tester plus de scénarios
- Proposer un framework générique
- Profils plus complexes
- Ajout dynamiquement de nouveaux paramètres aux profiles



- Aksit M., Choukair Z., Dynamic, Adaptive and Reconfigurable Systems
   Overview and Prospective Vision, (ICDCSW'03) Providence, Rhode Island,
   USA, May 19 22, 2003
- Cervantes H. Vers un Modèle A Composants Orienté Services pour supporter la Disponibilité Dynamique. Thèse de Doctorat, Université Joseph Fourier, mars 2004
- Blay-Fornarino M., Ensellem, D., Occello A., Pinna-Dery A-M., Riveill M., Fierstone J., Nano O., and Chabert G. *Un service d'interactions : principes et implémentation*. INRIA, Journée composants, Grenoble, France, 17 et 18 octobre 2002
- Keeney J., Cahill V., A Policy Driven, Context Aware, Dynamic Adaptation Framework, Proceeding of the 4<sup>th</sup> International Workshop on Policies for Distributed Systems and Networks, IEEE, 2003
- OMG Specification, CORBA Components, version 3.0, June 2002, <u>http://www.omg.org/technology/documents/formal/components.htm</u>
- W3C Specification, Web Services Description Language (WSDL), March 2001, <a href="http://www.w3.org/TR/wsdl">http://www.w3.org/TR/wsdl</a>
- Noah Documentation, v 2.0 , <a href="http://noah.essi.fr/download.php">http://noah.essi.fr/download.php</a>