

Reflective Middleware Solutions for Context-Aware Applications

Licia Carpa

Wolfgang Eimmerich

Cecilia Mascolo

BDIRA Mezri

mezri.bdira@cpe.fr

Les auteurs



Licia Carpa
Professeur dans le
département
informatique de
l'University College
London



Wolfgang Eimmerich
Professeur d'informatique
distribuées et le directeur
du département
informatique de
l'University College
London



Cecilia Mascolo
Maître de conférence
au département
informatique de
l'University College
London

L'article (septembre 2001)

- Peut-on parler de Middleware pour les réseaux à nœuds mobiles?
- La réflexivité et Metadata pour le Middleware.
- Le contexte d'exécution des applications et le Middleware.
- Principes du Middleware réflexif
- Discussion et remarques

Plan

Préambule

Principe du Middleware réflexif

Conception du modèle

Critiques

Conclusion

Préambule

Principe

Conception

Critiques

Conclusion

Préambule

- Rappel



Middleware



- Middleware désigne, dans le cadre de l'information répartie, toutes les couches logicielles qui permettent à des applications de communiquer à distance.

Préambule

Principe

Conception

Critiques

Conclusion

Préambule



- Permettre à un programme de s'exécuter sur **plusieurs machines** reliées par un réseau.
- Permettre à **différents types de machines** de communiquer à distance.

Préambule

Principe

Conception

Critiques

Conclusion

Préambule

- Peut-on parler de Middleware dans les réseaux à nœuds mobiles ?
- Le Middleware tel qu'il est implémenté pour les réseaux fixes ne sont pas adaptés pour les réseaux à nœuds mobiles.

Pourquoi ?

Préambule

Principe

Conception

Critiques

Conclusion

Préambule

- La perte de connectivité avec le réseau.
 - Une bande passante en perpétuel changement.
 - Un environnement d'exécution qui change.
- ➔ Le Middleware doit s'adapter au contexte d'exécution de l'application.

Préambule

Principe

Conception

Critiques

Conclusion

Principe du Middleware réflexif

C'est quoi le contexte?

- C'est tout élément qui peut **influer** l'exécution de l'application.
- Le Middleware doit prendre conscience de ce qui se passe dans l'environnement d'exécution pour bien fonctionner.

Principe du Middleware réflexif

- La conscience peut être divisée en deux entités la **conscience du noeud** et la **conscience de l'environnement**.
- **La conscience du noeud** (locale): désigne toute information sur la machine sur laquelle l'application s'exécute.

Principe du Middleware réflexif

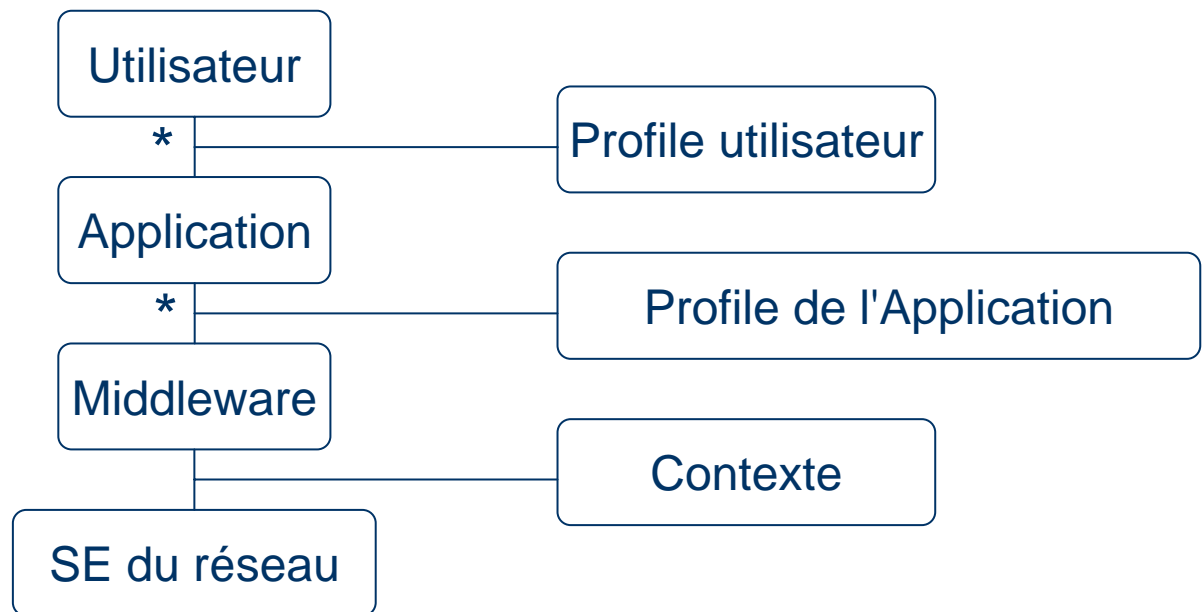
- Exemple: taille de la mémoire, mémoire disponible, état de la batterie, puissance du microprocesseur Etc.
- **conscience de l'environnement**: tout ce qui est extérieur à notre nœud. Toute information sur l'état du réseaux.

Principe du Middleware réflexif

- Exemple: bande passante, connectivité sur le réseaux, position, services disponibles, Etc.
- Le Middleware doit collecter ces informations, interagir avec l'SE du réseau fondamental, mettre à jours les données du contexte et garantir leurs disponibilités.

Principe du Middleware réflexif

Architecture à adapter pour l'implémentation d'un Middleware



Principe du Middleware réflexif

- L'utilisateur va indiquer, dans son profil, à l'application, comment réagir lors des changements qui peuvent se produire.
- L'ensemble de ces indications constituent la **Metadata** de l'application.

Principe du Middleware réflexif

- On retrouve dans le profile de l'application la configuration qu'elle peut supporter.
- Cette configuration représente la **Metadata** du Middleware.
- Le contexte représente les données du Middleware.

Principe du Middleware réflexif

Mais c'est quoi un Middleware réflexif ?

- On entend dire par réflexivité du Middleware la capacité de permettre à l'application d'accéder, de raisonner et d'en tirer sa propre interprétation.

Comment ça marche ?

Principe du Middleware réflexif

- Le Middleware enregistre la façon avec laquelle il doit se comporter lorsqu'on se retrouve dans un contexte particulier.
- Le Middleware offre ainsi à l'application la possibilité d'être réactive selon le contexte d'exécution dans lequel elle se trouve .

Conception du modèle

- Le Middleware va jouer l'intermédiaire entre l'application et le contexte dans lequel elle s'exécute.

Comment s'adapte le Middleware au contexte d'exécution de l'application ?

Comment se fait l'échange entre le Middleware et l'application ?

Conception du modèle

- Le Middleware va adapter le profile de l'application selon le contexte dans lequel elle s'exécute.
- L'échange se fait par l'intermédiaire d'un échange de fichiers XML entre l'application et le Middleware.

Conception du modèle

- L'application peut demander au Middleware de:
 - Être à l'écoute des changements qui risquent de se produire dans le contexte et adapter ensuite la stratégie adéquate.
- Exemple:

Batterie faible → Déconnexion du réseau
Bande passante faible ↗

Conception

Préambule

Principe

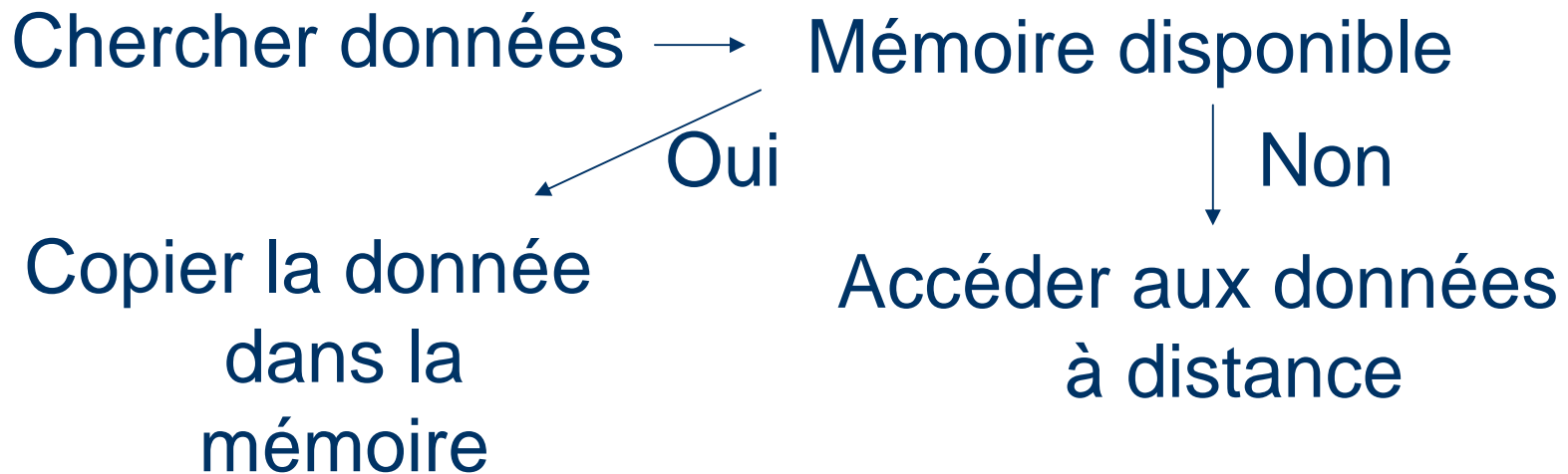
Critiques

Conclusion

Conception du modèle

➤ Recherche ou l'exécution de services.

• Exemple:



Conception du modèle

- Le profile de l'application va déterminer la politique à adapter par le middleware pour accéder à la donnée.
- Le Middleware doit être capable de repérer la disponibilité d'un service sur le réseaux et prendre la bonne décision pour le choix de la source.

Conception du modèle

- Comme l'échange entre le Middleware et l'application se fait sous XML, une grammaire est adoptée pour valider l'échange.
- Dans le cas d'une évolution de profile de l'application une adaptation de la part du Middleware doit être prise en compte.

Conception du modèle

- Ce ci est possible en Java grâce au chargement dynamique des classes.

Que a t on actuellement comme solutions pour les réseaux à nœuds mobiles?

État de l'art

- Tuple Space a été adopté pour des Middlewares de réseaux à nœuds mobiles: Jini/JavaSpaces, Lime et T Spaces.
- Tuple Space ne supporte pas l'arborescence des données. Ce défaut est surpassé grâce l'échange des fichiers XML.

Critiques

- Un article qui va droit au but comme l'OM. En effet, le principe est assez simple à expliquer mais sa mise en pratique n'est pas évoquée.
- L'évolution du Middleware Ok pour Java mais est-elle aussi évidente que ça ?
- Le Middleware et la Qualité de Service dans les réseaux mobiles?

Critiques

- La recherche de service dans les réseaux à nœuds mobiles.
- Plus de détail sur la relation qui existe entre le Middleware et le système d'exécution fondamental du réseau.
- Les autres solutions ne sont pas bien présentées.

Conclusion

- Que se passe t-il si plusieurs applications demande au Middleware d'agir différemment alors qu'ils partagent le même contexte d'exécution?

Conclusion

- Que se passe t'il si l'application se retrouve dans deux situations contradictoire quand des changements reliait aux différents ressources du réseaux se produisent en même temps?

Exemple: « se déconnecter batterie faible » VS
« se connecter bonne bande passante »

Conclusion

- La sécurité reste un problème du fait que n'importe quel nœud peut se connecter au réseau. Le Middleware réflexif peut aggraver la situation du fait que des programmes malicieux peuvent utiliser la réflexivité du Middleware pour récupérer, supprimer des données.

Bibliographies

- Les Middlewares réflexifs:
www.info.fundp.ac.be/~ven/CISma/FILES/2003-zyeden.ps.
- <http://www.cs.ucl.ac.uk/staff/c.mascolo/www/papers.html>. Ensemble d'articles tournant autour du Middleware pour les réseaux à nœuds mobiles.

