Middleware for Ubiquitous Programming

Antonelli Paul

RUNES : Reconfigurable Ubiquitous Networked Embedded Systems

Sujet : Runes' Component based middleware services and interactions with other applications

L'ensemble du projet RUNES est basé sur un modèle de composants constituant le noyau (middleware kernel) ainsi que l'interface de programmation (API) des services pouvant êtres associés. Chaque composant permettant de remplir une ou plusieurs fonctionnalités, il est alors possible de connecter différents composants entre eux via des interfaces et réceptacles vus plus en détail dans la section "Architecture des composants".
Ces regroupements de composants, appelés "Component Framework" (CF), sont conçus de manière à remplir une fonctionnalité (ou un domaine de fonctionnalités) bien spécifique et peuvent être construits récursivement (un ou plusieurs CF contenus dans un autre CF...) .
Ils doivent par ailleurs pouvoir être liés à d'autres CFs durant l'exécution ("run-time plug-in") afin de modifier ou d'étendre leurs fonctionnalités et/ou comportements. Un regroupement de CFs dédiés à un ensemble de fonctionnalités d'un même domaine peut alors être considéré comme un "service".

Un certain nombre de services ont donc été implémentés par l'équipe du projet RUNES afin de pouvoir poser les bases nécessaires au développement ainsi qu'au fonctionnement des futures applications distribuées et/ou embarquées.

Voici donc un bref aperçu des principaux services d'ores et déjà existants, développés par l'équipe du projet RUNES :

- Interaction service CF

Ce service permet d'assurer la transmission aux applications de toutes les communications provenant d'autres CFs ou services dans le but de permettre leur interaction. Afin de ne pas êtres limité à une seule méthode de communication (par exemple le publishing/subscribing), la notion de "plug-in interaction paradigm" (PIP) à été développée .
Un PIP est un CF particulier destiné à venir se greffer sur l'"Interaction service" afin d'étendre les possibilités d'interaction de celui-ci. Un certain nombre de PIP génériques et extensibles ont donc été développés afin de couvrir les paradigmes de communication les plus répandus.
Le service d'interaction permet donc de mettre en relation une application possédant ses propres besoins (formulés sous forme de prédicats) avec un PIP, en fonction des informations que celui-ci fournit sur lui-même (nom, but, contraintes d'utilisation, qualité de service).

- Overlay service CF

Ce service traite l'ensemble des besoins liés à la couche réseau du middleware. Il permet de s'abstraire de tout support physique du réseau tout en contrôlant la transmission et le routage des messages, ainsi que l'état du composant. Il est lui aussi basé sur l'utilisation de plug-ins ("plug-in overlay networks") venant se greffer sur ce service en fonction de la configuration du réseau et des besoins.

- Advertising and Discovery service CF

Ce service permet à tout CF venant d'être chargé d'informer l'ensemble du système de son arrivée (advertising). Il permet également à tout CF préalablement chargé de chercher un autre CF disponible depuis peu (discovery). Il est basé sur un ensemble d'interfaces telle que "Advertiser", "Advertisable", "Discovery" ou encore "ComponentListener".

- Logical Mobility service CF

Ce service traite l'ensemble des besoins liés à la configuration et à la modification des applications au sein d'un système distribué en transmettant des unités logiques (LMU), principalement des "ComponantTypes". Le but de ce service est de permettre au système de se reconfigurer dynamiquement en obtenant des données provenant de ses propres éléments. Il devrai également permettre l'intéropérabilité avec des applications et des environnements distants, ce qui n'avait pas été envisagé lors de sa conception.

REFERENCES :

www.cl.cam.ac.uk/~cm542/papers/jwin.pdf
www.ist-runes.org/docs/deliverables/D5_01.pdf
www.ist-runes.org/docs/deliverables/D5_02.pdf
www.ist-runes.org/docs/deliverables/D5_02_01.pdf