

Intergiciels : abstraction et/ou adaptation

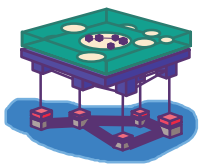
Pierre Crégut (DTL/TAL – Lannion)

Atelier du RIS – 6 juin 2002 - Toulouse

Le présent document contient des informations qui sont la propriété de France Télécom. L'acceptation de ce document par son destinataire implique, de la part de ce dernier, la reconnaissance du caractère confidentiel de son contenu et l'engagement de n'en faire aucune reproduction, aucune transmission à des tiers, aucune divulgation et aucune utilisation commerciale sans l'accord préalable écrit de France Télécom R&D

(diffusion
interne)

(Nom du fichier) - D1 - (Date)



été

programme vision

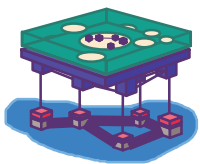
Cadre de la présentation



- ➔ Travaux réalisés par les équipes de FTR&D dans le cadre d'un ensemble de projets de recherche long terme sur les infrastructures logicielles (programme vision e-Mediate)
- ➔ Un cadre commun aux projets : les travaux sur les architectures à composant (modèle Fractal)
- ➔ Réalisations dans le cadre d'un consortium, ObjectWeb, dont le but est la réalisation d'intergiciels (Java) en open-source interagissant dans un cadre commun
 - INRIA
 - Bull
 - France Télécom R&D
- ➔ Présentation = un ensemble de problèmes ouverts + recherche de solutions dans le cadre d'e-Mediate.

France Télécom R&D

(diffusion
interne)



Plan



été

programme . vision

Spécificité des systèmes de télécommunication

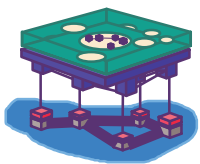
Rôle des intergiciels

Eclater les intergiciels

Spécifier les exigences non fonctionnelles

Composants pour les E.N.F.

(diffusion
interne)



été

programme vision

Spécificité des télécoms



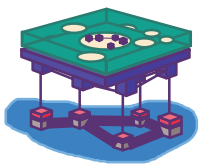
→ Réseau ubiquitaire mais pas uniforme

- Accès hauts débits avec garantie de service (ATM)
- Haut débit discontinu : satellites
- réseaux sans vrai QoS (IP)
- réseaux mobiles avec connectivité variable – réseaux ad hoc
- multiplicité des normes (ex: tél. mobile)
- etc.

→ Diversité des plate-formes et équipements

- Evolution continue du système : on ne jette pas un réseau
- présence de nombreux équipements non contrôlés par l'opérateur (terminaux clients + serveurs...)
- Accélération avec la déréglementation/concurrence

Un réseau est un système ouvert très dynamique



été

programme vision

Diversité des exigences

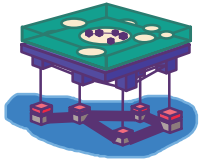


- Diversité des clients = diversité des exigences sur les services
 - Des domaines demandant un masquage total des fautes
 - Fail safe : intégrité des bases de données
 - Forte disponibilité : télévision, visiophonie (multimedia temps réel)
 - Best effort.

- Expression complexe de la disponibilité
 - Exigence moins radicale que le secteur du transport par ex.
 - Compromis avec le coût, les performances
 - Qualité de service, disponibilité *difficiles à caractériser* (surtout par le client)

- Rôle particulier de la sécurité
 - Influence croisée entre sécurité et sûreté
 - Caractère particulier de l'expression de la sécurité

(diffusion
interne)



été

programme .vision

Rôle des intergiciels



→ Rôle fonctionnel :

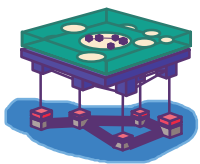
- une couche intermédiaire entre les systèmes (OS) et les applicatifs fournissant aux applications un accès à des ressources via une **abstraction uniforme** (données, communications) indépendante de l'implémentation
 - simplification de la programmation
 - homogénéité : **éviter n service x m systèmes**

→ Respect de contraintes non fonctionnelles :

- sécurité, disponibilité, performance
- De nombreux **services complémentaires** (de persistance, d'authentification, de réplication, etc.) ou ajouts internes (temps réel)
- Souvent vissés dans l'intergiciel

→ Rôles multiples (3 principaux identifiés)

(diffusion
interne)



été

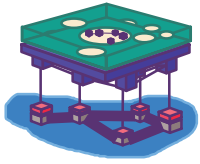
programme . vision

Un exemple



- La plate-forme Continuum (Dang Tran et al – FTR&D/DTL/ASR)
- Plate forme pour mondes virtuels répartis
 - sur un ORB pour les communications (Jonathan)
 - utilisation de la diffusion multicast
 - Utilisation des techniques réactives pour la propagation des évènements (notion d'instant synchrone).
- Abstraction pour les applications développées :
 - Divers jeux en réseaux utilisables à grande échelle
 - Autres utilisations possibles (travail coopératif, etc.)
- Abstraction plus difficile vers le bas :
 - Intégration diffusion / ORB / moteur synchrone

(diffusion
interne)



été

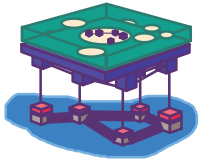
programme vision

Des intergiciels minimaux



- Séparer les différentes fonctionnalités (ex : ORB)
 - Construire l'abstraction du système et donner un cadre architectural commun : langage d'interface avec repr. commune de l'information
 - Assurer la fonction de communication en assurant le transport de l'information
 - Assurer les exigences non fonctionnelles : disponibilité / QoS / sécurité
- Arriver à exprimer (et pas seulement réaliser) les exigences non fonctionnelles dans le cadre architectural (notion de réflexivité)
 - à la conception :
 - Pour choisir les bons composants pour les E.N.F.
 - mais aussi dans les infrastructures à l'exécution :
 - lié au caractère dynamique des environnements de déploiement
 - condition de l'adaptabilité des infrastructures

(diffusion
interne)



été

programme . vision

Un exemple dans les OS

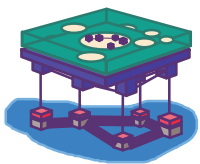


- Evolution des OS : des systèmes monolithiques aux micro-noyaux /
exo-noyaux

- Exemple : Think (J-P Fassino & J-B Stefani)
 - Un cadre architectural pour décrire les systèmes avec un langage
d'assemblage de composants
 - Un outil pour créer de nouveaux assemblages
 - Une déclinaison pour les OS (Kortex)
 - Coût de la composition relativement faible.

- Principe à décliner globalement au niveau des intergiciels

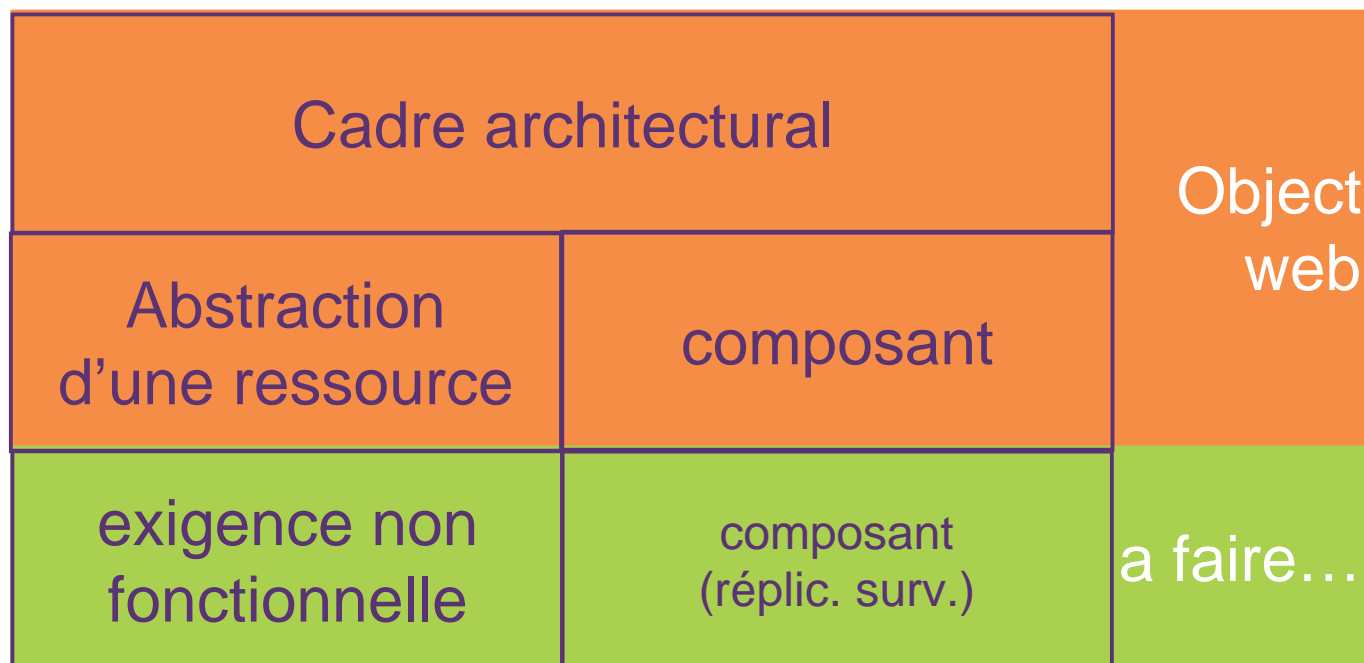
(diffusion
interne)



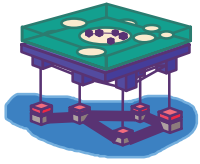
été

programme **vision**

Modèle éclaté des intergiciels



(diffusion interne)



été

programme . vision

Cahier des charges



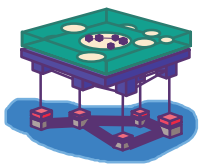
→ La description des E.N.F.

- Doit être intégré au modèle de composant :
- Doit reposer sur un cadre de description relativement homogène (pour éviter la multiplication des formalismes)
- Doit pouvoir être simple et synthétique
- Doivent être composables

→ Les composants les réalisant

- Doivent suivre le modèle global de composants
- Doivent pouvoir coller à la décomposition des E.N.F.
- Doivent être suffisamment indépendants pour permettre une composition simple (interactions).
- Doivent être adaptables :
 - statiquement (nécessité)
 - dynamiquement (utile pour les infrastructures dynamiques)

(diffusion
interne)



été

programme **vision**

Spécifier les E.N.F.



Exemples de travaux sur le thème (directions de recherche) :

→ Travaux de A. Arora (Ohio State University)

- Notion de détecteur / correcteur
 - On peut assurer la TF par des détecteurs (fail safe) et correcteurs (masquante)
 - Tout dispositif de TF peut être modélisé comme un ens. de détecteur/correcteur

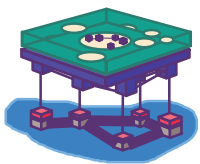
→ Travaux de M. Rodriguez, J-C Fabre et J. Arlat

- Utilisation de la logique temporelle pour spécifier le comportement
- Application comme « wrapper » pour des OS.

→ Travaux à FTR&D

- Spécifier la propagation des erreurs par des diagrammes de séquences (projet RNRT MAGDA)
- Prise en compte du temps :
 - spécifier le temps dans LDS ou UML (INTERVAL, OMEGA)
 - Mélanger les modèles synchrone et asynchrone.

(diffusion
interne)



été

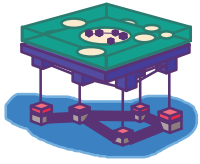
programme vision

Composants pour les E.N.F.



- Intégration de la composition des exigences à l'architecture
 - Notions de composants de composants (modèle récursif de Fractal)
 - Notions d'enveloppes
 - Composant \Rightarrow possibilité de composer les exigences
- Arora (suite) - Approche de la SdF sur des modèles à composants
 - Définir un modèle du système comme une composition de composant
 - Associer à chaque composant des E.N.F simples
 - Surveiller (détecter/corriger)
- Travaux dans e-Mediate
 - Travaux sur le diagnostic (Projets RNRT MAGDA I et II)
 - Description des éléments sous forme de composant ayant un comportement sur les notifications d'erreur.
 - Adaptabilité : composants de persistance ou réplication. Politiques modifiables

(diffusion
interne)



été

programme . vision

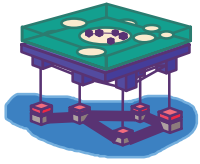
Conclusion



→ Où en sommes nous ?

- Exprimer les exigences
 - un langage
 - un moyen de les faire décrire et valider
- les composer :
 - simple possibilité théorique de composants remplissant les E.N.F. (ou d'enveloppes)
 - problèmes d'interactions entre composants
 - certainement besoin d'abstractions simples.
- Validation :
 - besoin d'une approche formelle
 - taille des systèmes
 - diversité des acteurs
 - preuve ou outils automatiques : pas mûre

(diffusion
interne)



été

programme vision

Conclusion (2)



- Intergiciels actuels inadaptés
- On a au mieux des réponses partielles au problème de l'expression des E.N.F.
- Besoins d'un cadre pour les refléter dans le modèle.
- Même dans un cadre statique la notion de composant d'E.N.F. n'est pas simple.
- L'évolution dynamique des E.N.F. des systèmes est un problème assez ouvert même si quelques travaux (par ex. changement des politiques de réplication).

(diffusion
interne)