



## Compte-rendu des réunions du 14 Novembre 2003

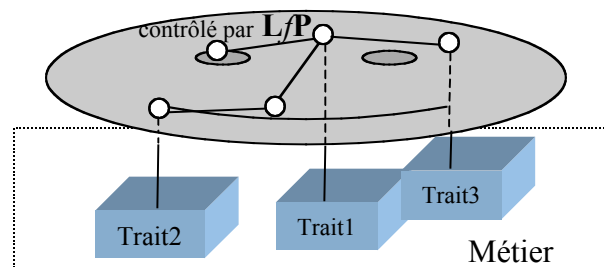
### 1 Ordre du jour

1. Définition des différentes chaînes de transformation
2. Aspects méthodologiques
3. Retours d'Aonix concernant **LfP**
4. Questions techniques diverses
5. Site Web MORSE
6. Rapports d'activité au RNTL
7. Documentation **LfP**

### 2 Avancement des travaux, aspects techniques

#### 2.1 Points abordés

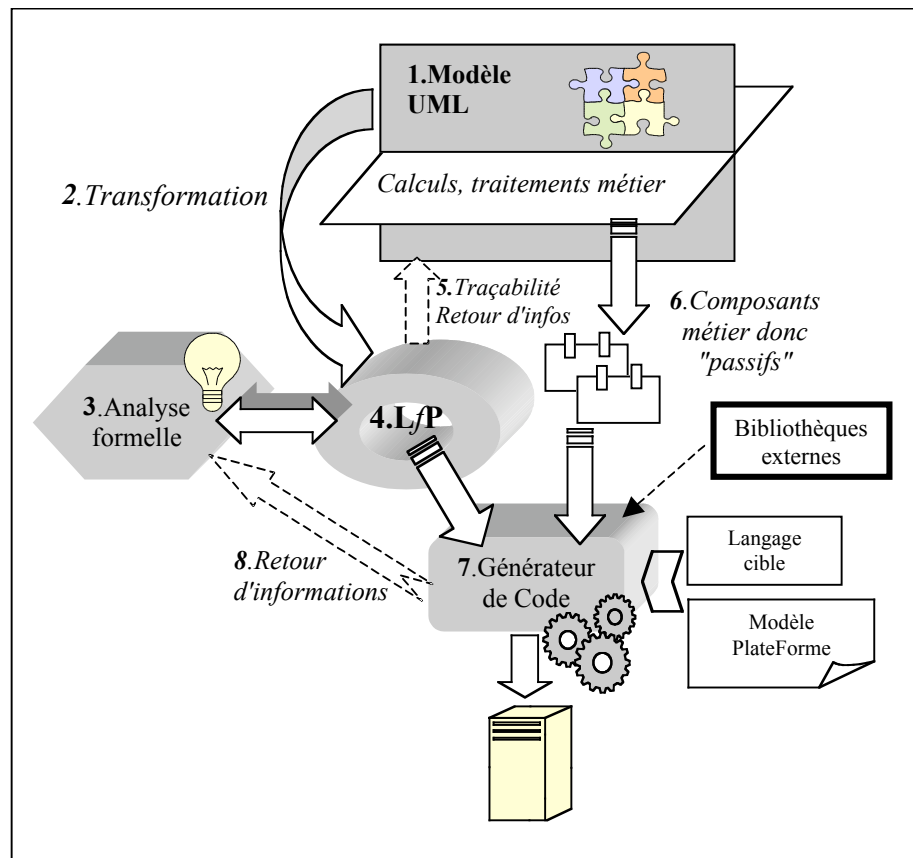
- Tâches actives : il faudra "remonter" des informations de **LfP** vers UML i) afin de remédier au manque de rigueur de UML ii) avant d'assurer une traçabilité forte. Il faudra voir comment on fait cela.
- Tâches passives : elles ne sont peut-être pas nécessaires (dans un premier temps dans **LfP**) sauf si on régit leur utilisation d'une manière très précise en leur associant des règles et des contrats (assertions). Ceci sera donc à prendre en compte au niveau de la méthodologie.
- Il a été rappelé que **LfP** ne s'occupe que du contrôle du système, en conséquence, il ne sera pas concerné par les traitements-métiers (à condition que ces derniers ne gèrent pas de manière cachée le contrôle). Les classes **LfP** ne sont que des classes de supervision.



La vérification et la génération de code ne concernent que le contrôle. Il faudra donc extraire ces deux parties (contrôle et métier) à partir du modèle UML. Voir comment le faire automatiquement. Comment distinguer ces parties ?



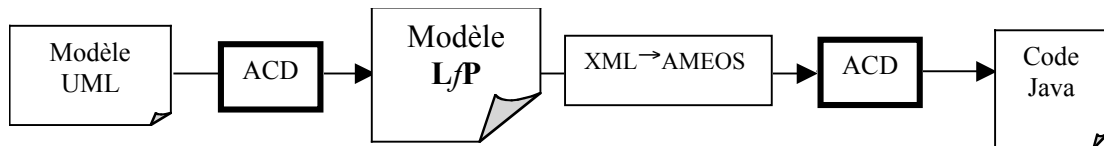
- La question concernant l'utilisation de ROOM reste posée.
- L'exportation du modèle **L/P** se fera (au moins pour ce qui concerne LIP6) en XML et non en XMI.
- Quelques points relatifs à la méthodologie ont été précisés.



1. Un modèle en UML est élaboré à la main, il comprend implicitement une partie "contrôle" et une partie "métier".
2. La partie "contrôle" est transformée automatiquement en un modèle **L/P** (4)
3. Ce modèle est formellement vérifié (une transformation vers des réseaux de Petri est par exemple utilisée).
4. On rentre ensuite dans une étape d'élicitation ( $3 \leftrightarrow 4$ ) → 1 où le modèle **L/P** est raffiné.
5. Ceci implique une mise à jour du modèle UML de base (avec aussi éventuellement un raffinement de la partie "métier").
6. La partie métier est transformée en composants métier (à la main ou en utilisant un outil spécialisé). Ces composants sont en principe passifs vis-à-vis de **L/P**. Si ce n'est pas le cas alors il faudra annoter le modèle **L/P** et en conséquence rentrer de nouveau dans la boucle d'élicitation précédente.
7. Ensuite, une fois le modèle **L/P** achevé alors le générateur de code est activé.



8. La génération de code peut faire partie d'une boucle plus large pouvant éventuellement remonter jusqu'en 4 ou même jusqu'en 1 (mais ce n'est peut-être pas souhaitable).
- Aonix a présenté deux voies pour la génération de code, l'une (solution **B**) envisage une génération de code par ACD directement à partir du modèle UML. L'autre (solution **A**) passe par **LfP** :



*Solution A – AMEOS est le format d'entrée de l'outil ACD-*

La solution **B** suppose que soient remontées vers UML (à partir du modèle **LfP**) toutes les informations nécessaires à la génération de code. D'un commun accord cette solution n'a pas été acceptée car il apparaît plus simple de partir du modèle **LfP** d'autant que ce dernier a été vérifié (cette opération serait très difficile voire impossible en UML). Néanmoins pour des raisons de traçabilité, la question des retours d'information vers UML reste posée.

- La chaîne de transformation UML → **LfP** puis le générateur de code seront insérées dans l'IDE Eclipse.
- Aonix reconnaît quelques difficultés concernant la génération de code C++ (pourtant nécessaire pour l'application-témoin).
- Aonix élaborera des profils UML permettant la définition (en UML) de **LfP**.

### 3 Divers

Un site Web MORSE sera développé par F.Kordon

Il faut définir un modèle de format de documents MORSE

F.Gilliers doit écrire un papier concernant la sémantique de **LfP**

Il a été décidé de structurer les rapports RNTL de la façon suivante :

1. Avancement des tâches
2. Points durs identifiés
3. Actions associés pour réduire les risques
4. Fournitures (documentation, logiciel,...)

Chaque rapport devra comporter 2 à 3 pages.

### 4 Points à aborder lors de la prochaine réunion (18/12/2003)

- Rapport RNTL
- La sémantique **LfP**,



- Point sur la transformation  $UML \rightarrow LfP$ ,
- ROOM (informations relatives au déploiement)
- La méthodologie MORSE