



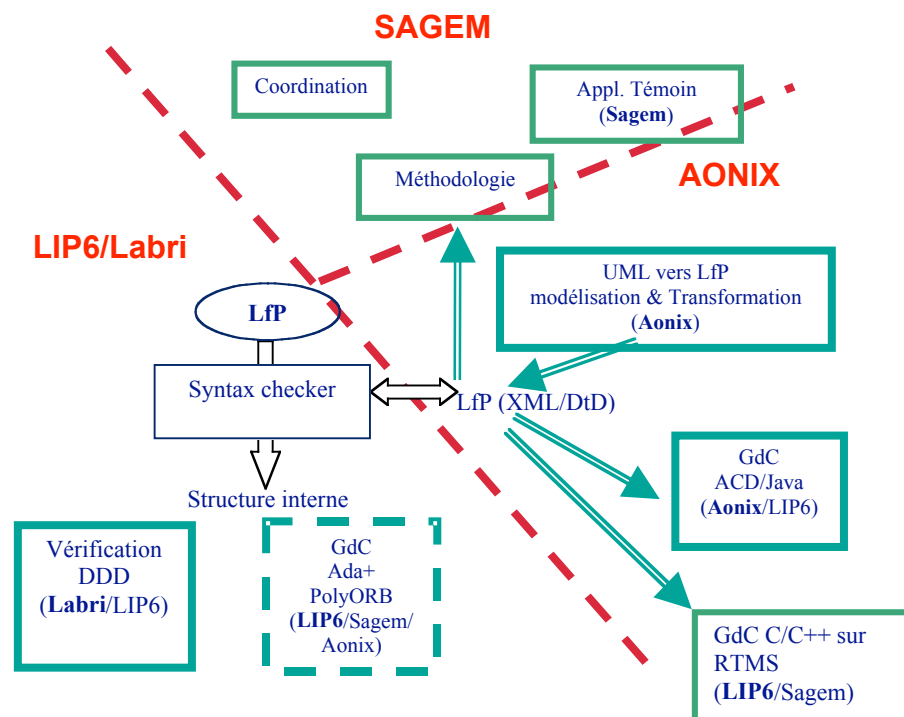
## *Compte-rendu des réunions du 10 juillet et 15 septembre 2003*

### 1 Ordre du jour

1. Préciser les «frontières» entre les actions des différents partenaires
2. Lister les technologies propriétaires impliquées
3. Revoir précisément les lots
4. Listes des fournitures
5. Définir l'échéancier de démarrage
6. Définir le type des fournitures
7. Choix techniques

### 2 Avancement des travaux

#### 2.1 Frontières



Le schéma ci-dessus situe les domaines d'intervention des différents partenaires tels qu'ils ont été définis durant la réunion.



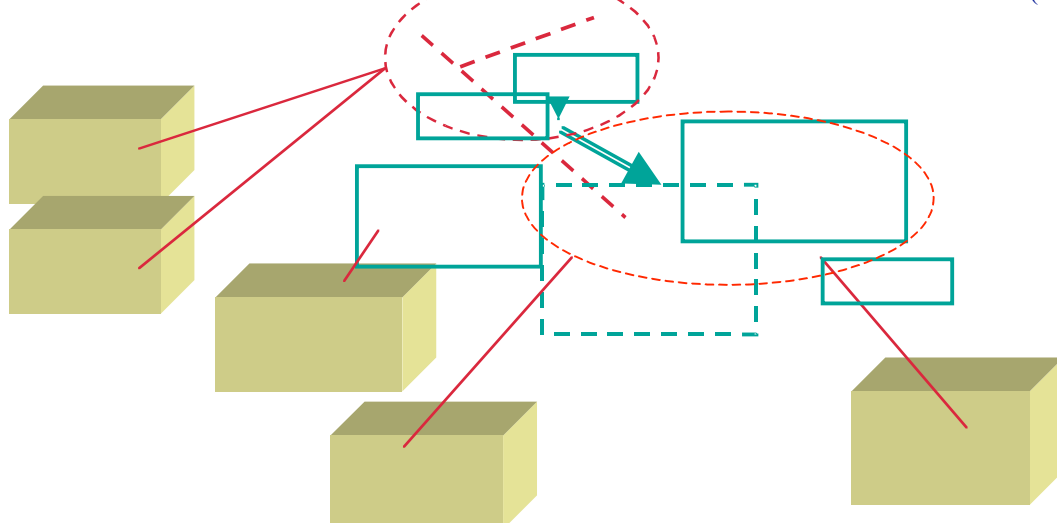
On notera deux langages cibles C++ (nécessaire pour l'application-témoin - Sagem -) et Java (Aonix). Les générateurs s'appuieront sur les travaux de F.Gilliers (thèse en cours) et utiliseront le produit ACD de Aonix.

## 2.2 Technologies propriétaires

AONIX	ACD, STP, RAVEN
SAGEM	Technologie métier (drones), Méthodologie par prototypage : GT9 Objet
LIP6	LfP & Outils, MetaScribe, FrameKit, CPN-AMI
Labri	DDD
Autres	PolyORB (LIP6/ENST), Open Source (GMGPL)

## 2.3 Organisation des sous-projets

SP1Méthodes SP2Modèle SP3Vérification SP4Prototypage SP5Applicationtémoin StructureinterneLfP(XM)



Il s'agit dans les grandes lignes d'une reprise de la proposition.

### 2.3.1 SP1 : Méthodologie de développement de systèmes répartis embarqués asynchrones

SAGEM	33% : compétences en certification
AONIX	53% : compétences en méthodologie objet
LIP6	10% : compétences en prototypage & méthodes formelles
LaBRI	5% : compétences en méthodes formelles

- Définir la méthodologie MORSE  
*On part d'une méthodologie éprouvée (cycle en Y : GT9 de Sagem)  
Cadence l'ensemble du projet*
- Définir les relations entre UML et le langage PIVOT  
*Règles de transformation UML -> Langage Pivot (Utilisation de profils UML)*
- Expliciter et formaliser le matériel de qualification nécessaire  
*Vers des exigences de certification*

### 2.3.2 SP2 : Langage pivot et expression de propriétés sur le modèle

SAGEM	9% : compétences en certification
-------	-----------------------------------



AONIX	14% : compétences UML
LIP6	50% : compétences en prototypage & spécifications formelles
LaBRI	27% : compétences en méthodes formelles & vérification

- Définition du langage Pivot  
*Vérification formelle semi-automatisée*  
*Informations pour le déploiement*  
*Caractéristiques d'un ADL*
- Expression des propriétés  
*Invariants, formule de logique temporelle*  
*Formules «prêtes à l'emploi» pour aider les ingénieurs*
- Choix des propriétés à vérifier  
*Via la notion d'observateurs ou «d'obligation de preuves»*
- Retour sur la méthodologie  
*Raffinement en fonction des résultats des autres sous-projets*

### 2.3.3 SP3 : Vérification formelle de propriétés

SAGEM	8% : observation vis-à-vis des besoins du domaine d'application
AONIX	2% : observation vis à vis de la méthodologie (retour sur la méthode)
LIP6	45% : compétences en prototypage & spécifications formelles
LaBRI	45% : compétences en méthodes formelles & vérification

- À l'aide de réseaux de Petri de haut niveau  
*Simplifications en fonction de la propriété considérée*  
*Prise en compte de la structure du modèle*  
*Utilisation des techniques de programmation linéaire*
- À l'aide de Diagrammes de Décision de Données (DDD)  
*Représentation du modèle et des propriétés en DDD*  
*Vérification symbolique du modèle*  
*Expression du résultat dans les termes du modèle pivot*
- Développement des outils de vérification dans l'atelier MORSE  
*Outiller les techniques de validation*
- Retour sur la méthodologie

### 2.3.4 SP4 : Synthèse automatique de programmes

SAGEM	37% : compétences en systèmes embarqués
AONIX	33% : compétences en exécutifs embarqués et en génération de code
LIP6	26% : compétences en système répartis & génération de code
LaBRI	3% : compétences en sémantique opérationnelle

- Règles de synthèse de programme  
*Conformité DO-178B (traçabilité)*  
*Ré-utilisation*  
*Générer du code auto-évaluant (pour optimisation)*
- Intégration du prototype dans un environnement d'exécution  
*Virtualisation de l'exécutif cible (pour le déploiement)*  
*Définition d'une Application Program Interface*
- Implémentation de l'outil de synthèse automatique
- Retour sur la méthodologie

### 2.3.5 SP5 : AGL MORSE et application témoin

SAGEM	69% : compétences en systèmes de drone
-------	--



---

AONIX	15% : compétences en méthodes de développement
LIP6	9% : compétences en spécification formelle
LaBRI	7% : compétences en vérification formelle

- Interface type des outils pour l'intégration dans l'AGL pilote  
*Mécanismes d'invocation des outils (XMI/Unix/FrameKit)*
- Intégration des outils  
*Intégration des outils produits au sein de l'AGL MORSE (Eclipse?)*  
*Utilisation de la plate-forme FrameKit (LIP6)*
- Définition de l'application témoin  
*Sagem est en train de refondre le logiciel des drones*  
*Reprise d'une partie de l'interface porteur/stations au sol*
- Réalisation de l'étude témoin avec l'AGL MORSE
- Mesures et comparaisons entre:  
*La démarche classique*  
*La démarche MORSE*



## 2.4 Type de fournitures

	Rapports Publiques	Rapports restreints	Doc +logiciel
<b>Sous-projet 1: méthodes</b>			
1.1 - Méthodologie MORSE			
1.2 - def. Transfo. vers mod. pivot			
1.3 - formali. Matériel. Qualif.			
<b>Sous-projet 2: modèle</b>			
2.1 - def. langage pivot			
2.2 - expr. propr. Attendues			
2.3 - propr. & processus de vérif.			
2.4 - retour sur la méthodologie			
<b>Sous-projet 3: vérification</b>			
3.1 - vérif. Via RdP			
3.2 - vérif. Via DDD			
3.3 - outillage			
3.4 - retour sur la méthodologie			
<b>Sous-projet 4: Prototypage</b>			
4.1 - def règles de traduction			
4.2 - prototype & env. d'exec			
4.3 - outillage			
4.4 - retour sur la méthodologie			
<b>Sous-projet 5: témoin</b>			
5.1 - règles outil AGL			
5.2 - intégrer AGL MORSE			
5.3 - def. étude témoin			
5.4 - dev. étude Témoin			
5.5 - mesures			

## 2.5 Divers

Quelques choix techniques effectués :

- Utilisation de l'environnement Eclipse (Java, C++)
- Génération vers Java, C++ et sur PolyORB

Le document de F.Gilliers définissant la syntaxe **L<sub>f</sub>P** a été remis aux participants.

Question à trancher prochainement : L<sub>f</sub>P peut-il est défini par un profil UML ?