Compte rendu de la réunion du 12.02.2008

Points évoqués

- Construction des diagrammes de séquence des différentes phases, automates des échanges avec FrameKit et automates des états de la session (Urgent).
- Conception du module API.
- Déroulement de la suite du travail.

Rédaction des diagrammes de séquence, automates cami et automates de session (Urgent)

Afin de modéliser les échanges entre Coloane et FrameKit, nous allons diviser ces échanges en différentes phases comme l'authentification, l'ouverture d'une session .... Dans un premier temps, nous identifierons d'abord ces phases, puis nous les étudierons une par une comme suit :

- Construire le diagramme de séquence comprenant les composants COM, API et FK.
- établir la liste des arguments ou paramètres utilisés en input et output , avec leurs destinations(pour qui sont ils destinés ?), et leurs sources (d'ou viennent-ils ?), et éventuellement les erreurs au niveau de cette phase.
- Construire l'automate des échanges avec FrameKit (automate du protocole Cami) correspondant à cette phase.
- Définition des formats des objets échangés entre COM et API avec réflexion sur la responsabilité de chaque classe par rapport à l'objet défini).

A partir de tous ces diagrammes de séquences et automates associés, on construira l'automate d'états de session.

A propos des automates

L'automate d'états de session est explicitement construit, autrement dit, il y aura une structure de donnée représentant cet automate. Il nous permettra de contrôler le comportement de Coloane et de ne pas basculer dans un état instable.

L'automate du protocole cami est quant à lui construit de manière implicite dans le cas où nous utiliserons un parseur (principe des parseurs). Cet automate nous permettra de spécifier exactement les suites de messages autorisées à être échangées entre API et FrameKit, et cela nous aidera à construire les états de l'automate d'états de session.

Conception de API

- Description textuelle (Objectif, contraintes).
- Description à l'aide d'UML (-diagramme de classes- donner les différents composants de notre application et les interfaces)..
- Construction d'un environnement de test (étudier le principe des bouchons, les classes bouchons simulent le comportement d'une classe donnée, ce qui nous aide a tester une partie de l'application sans se soucier d'éventuels autres problèmes).

Déroulement de la suite du travail

Nous commencerons donc par le premier point, à savoir la construction des diagrammes de séquence, automates cami et automates de session pour chaque phase. A l'issue de chque phase, un document sera produit pour être validé par les encadrants.