



➔ **Programmation d'agents intelligents**
Vers une refonte des fils de raisonnement

Stage de fin d'études – Master IAD 2006

Plan de la présentation

I. Présentation de l'existant

II. Positionnement dans l'état de l'art

III. Analyse du modèle & Extensions

IV. Bilan & Perspectives

Questions

→ Présentation de l'existant ←



But

- ➔ Conférer de l'*intelligence* aux programmes sans accroître leur complexité structurelle

Principe d'autonomie

- ➔ « Un agent A est autonome par rapport à B si B ne peut pas prédire à coup sûr le comportement de A »
- ➔ Dépendance limitée

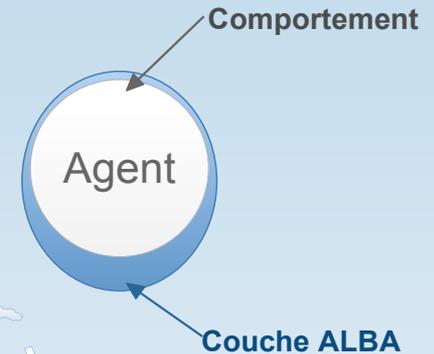
Conséquences

- ➔ Implications sur la programmation des agents
- ➔ Tolérance aux fautes prise en compte à la source

ALBA

- ➔ Bibliothèque pour le déploiement d'agents écrits en Prolog
- ➔ Couche de bas niveau
- ➔ Communication, création d'agents, migration

*Benjamin Devèze, Caroline Chopinaud, Patrick Taillibert.
Alba : a generic library for programming mobile agents with Prolog. ProMAS'06. Hakodate.*



Prolog

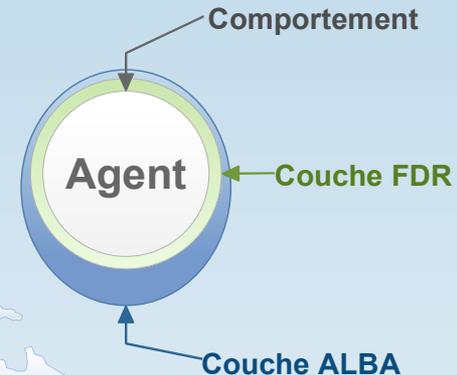
- ➔ Mécanismes bien adaptés pour l'IA (manipulation d'arbres, unification, backtrack, métaprogrammation...)
- ➔ Prototypage
- ➔ Héritage et expérience

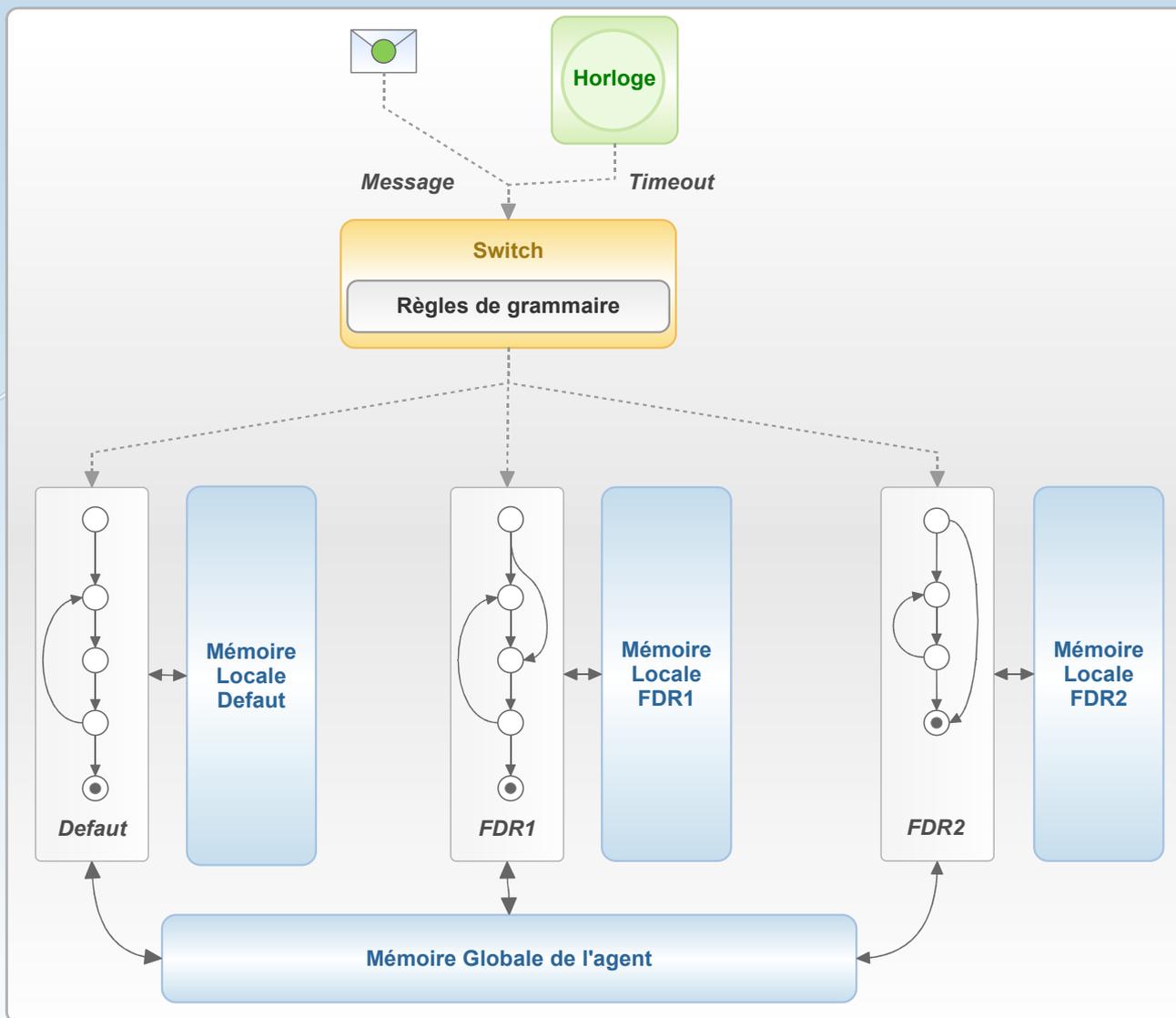
Besoin d'un modèle d'agent

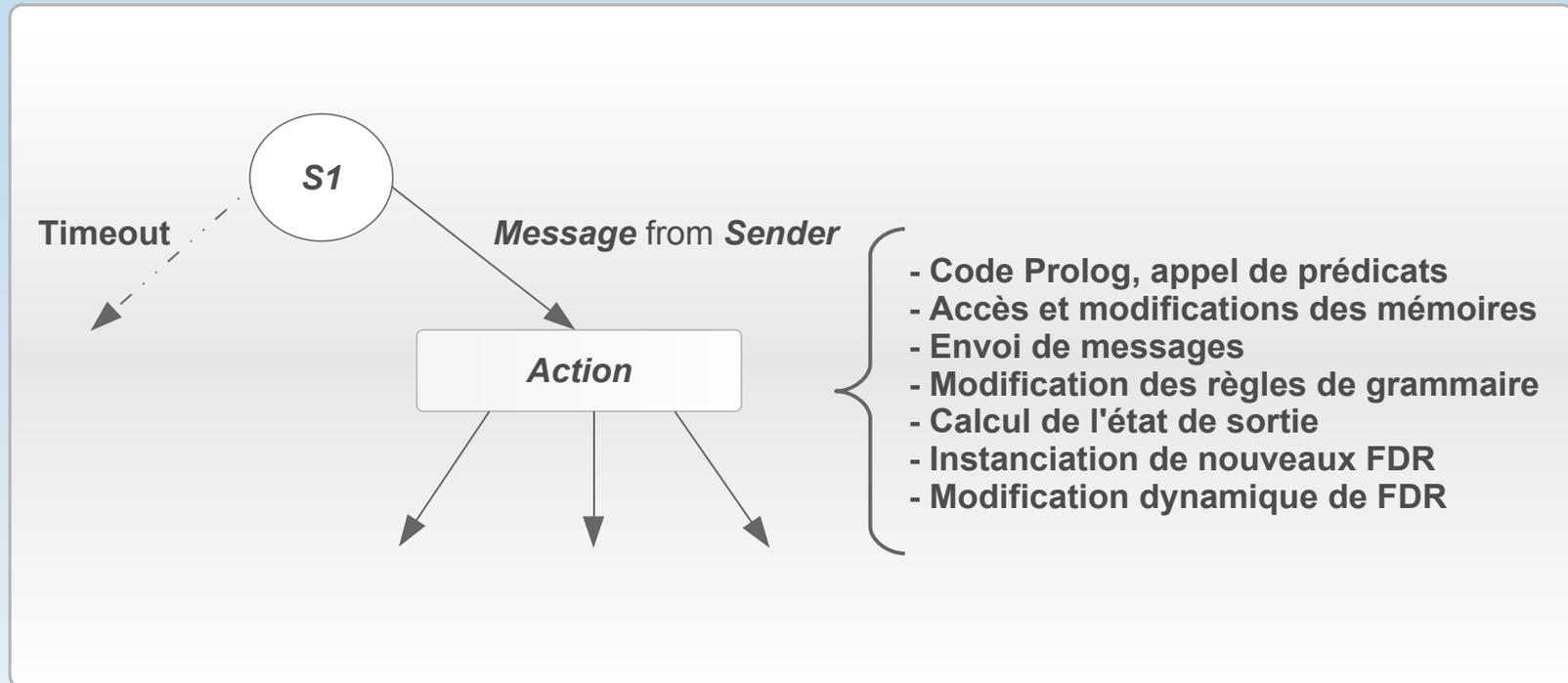
- ➔ Rôle essentiel du filtrage des messages
- ➔ Gestion de plusieurs contextes simultanés

Métaphores

- ➔ Fils de discussion sur Internet
 - Tri des messages suivant leur objet
 - Gestion simultanée de plusieurs contextes
- ➔ Traitement manuel du courrier
 - Première sélection pour déterminer le contexte
 - Analyse contextuelle approfondie
 - Identification de nouveaux contextes

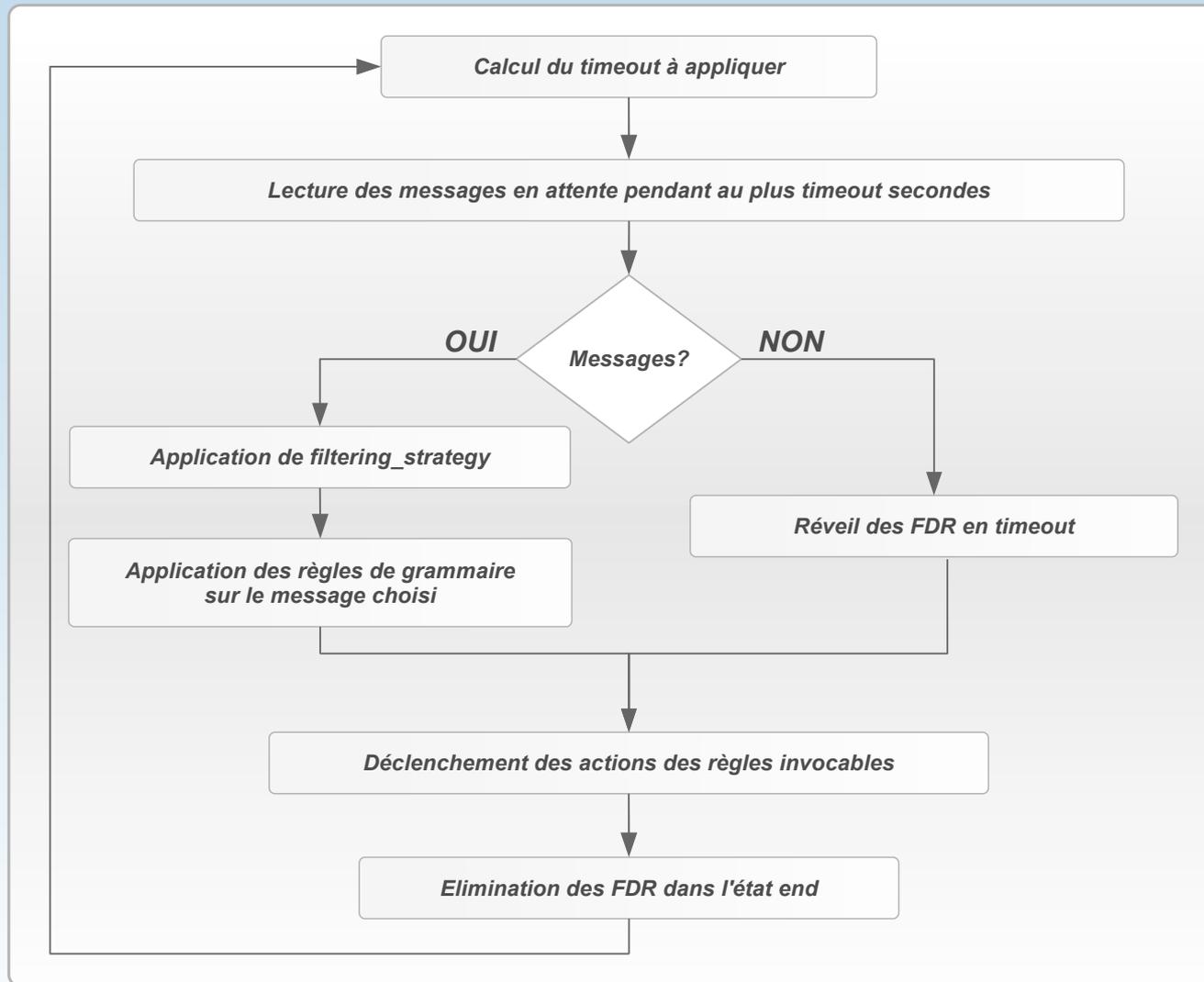


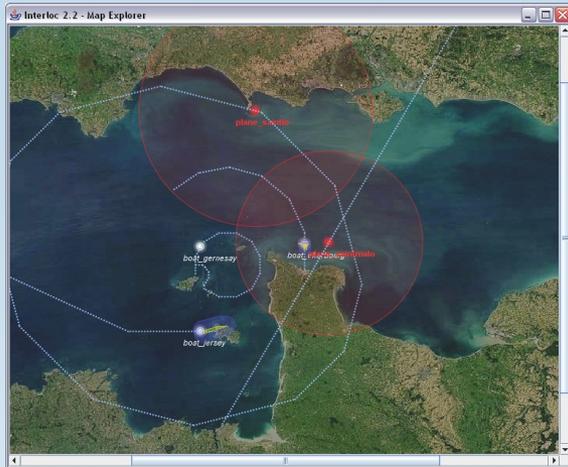




Langage de description

- ➔ *fdr*(Type, State-in, States-out, Message, Sender, Parmfdr, Action) :- filter.
- ➔ *action*(Message, Sender, Parmfdr).





Applications

- ➔ Interloc, localisation passive de cibles marines
- ➔ Synapse, coordination de drones (ContractNet)
- ➔ Aerial, drones qui planifient leurs trajectoires
- ➔ FDR simples et pratiques, isolement des contextes

Sujet du stage

- ➔ Positionnement dans l'état de l'art
- ➔ Analyse et extensions du modèle
- ➔ Implémentations

→ Positionnement dans l'état de l'art ←

Architecture?

- ➔ Décomposition de l'agent en modules
- ➔ Données d'entrée + état courant → actions + nouvel état

Différentes approches

- ➔ Architectures délibératives (HOMER, etc.)
Représentation symbolique du monde
Décision = une déduction logique
- ➔ Architectures réactives (Subsumption, Situated Automata, etc.)
Lien direct entre situation et action
- ➔ Architectures BDI (Belief, Desire, Intentions) (PRS)
Décision = manipulation des croyances, désirs et intentions
- ➔ Architectures en couches (TouringMachines, INTERRAP)
Décision = plusieurs couches logicielles raisonnant à des niveaux d'abstraction différents



AgencTalk

- ➔ Langage de description de protocoles de coordination
- ➔ Notion de *script*
- ➔ Héritage entre scripts

Kuwabara et al. AgencTalk : Coordination protocol description for multiagent systems. 1995.

COOL

- ➔ Représentation explicite des connaissances portant sur les processus d'interaction
- ➔ Notion de *classe de conversation*
- ➔ Dépendance entre conversations

Barbuceanu et al. A language for describing coordination in multiagent systems. 1995.

Des principes communs mais des différences...

- ➔ Flexibilité des règles de grammaire
- ➔ Flexibilité du langage de description et utilisation de Prolog
- ➔ Objectif : pousser plus loin les principes de COOL et AgenTalk

De l'usage des connaissances procédurales

- ➔ AgenTalk, COOL : point de vue social
- ➔ PRS : aspect individuel
- ➔ Objectif : combiner les deux

→ Analyse du modèle & Extensions ←

Respect des temps et de la réactivité de l'agent

- ➔ Problème de la durée des actions!
- ➔ Retard dans les réveils, agents ne traitant plus les messages...
- ➔ Délégation des calculs « longs » à des artifacts de calcul
Dinont et al. Les artifacts de calcul – une solution aux délibérations longues. 2006.

De l'usage des timeout

- ➔ Gestion des *timeouts* comme des messages
- ➔ Mise en place d'un service de réveil dans les FDR
 - Envoi de signaux internes
 - Gestion de temps absolu et relatif



Problématique

- ➔ Les contextes ne sont pas toujours indépendants
- ➔ Besoin de faire communiquer différents contextes!

Solutions possibles

- ➔ Utilisation de la mémoire globale de l'agent
- ➔ Envoi de messages à soi-même

Signaux Internes

- ➔ `send_signal(Signal, FDR_Destinataire)`



Liste des évènements

- ➔ Réception de messages
 - ➔ *msg(Message, Sender)*
Communication entre agents, actes de langage
Interactions avec les artifacts
- ➔ Réception de signaux internes
 - ➔ *signal(Signal, FDR_ID)*
- ➔ timeout
- ➔ Evènements mémoire

Limitation des FDR

- ➔ Proactivité extrêmement limitée
- ➔ Phases d'attente de messages entre les *timeouts*
- ➔ Comment introduire de la proactivité?

Une autre conception des FDR

- ➔ FDR assimilables à des plans
- ➔ Bibliothèque de plans

Quelques approches

- ➔ Ajout d'une condition d'invocabilité
- ➔ Précondition – Postcondition avec instanciation et composition des plans par chaînage arrière
- ➔ Annotation des plans

Un FDR de gestion des buts

- ➔ Un FDR dédié à la gestion des buts
- ➔ Manipulations en mémoire globale *Goal(Goal, Deadline)*
- ➔ Sélection des buts
- ➔ Sélection des FDR satisfaisant ces buts qui sont invoqués et deviennent des intentions
- ➔ Gestion de l'arrêt des FDR instanciés pour satisfaire des buts

Discussion

- ➔ Analogie avec un modèle BDI
- ➔ Flexibilité dans la gestion des buts
- ➔ Modification dynamique du comportement par les stratégies de filtrage des événements



- Code Prolog
- Appels encapsulés à ALBA
 - * Envois de messages aux agents
 - * Invocation d'artifacts
 - * Création d'agents
 - * Migration
- Appels relatifs aux FDR
 - * Emission de signaux internes
 - * Modification et tests sur les mémoires
 - * Instanciation de FDR
 - * Modification des règles de grammaire
 - * Modification des bibliothèques de FDR
 - * Modification dynamique des FDR instanciés

fdr(Type, State-in, State-out, Event, Parmfdr, Action) :- Filter.

Intérêts

- ➔ Facilite l'analyse des transitions et possibilité d'introspection
- ➔ Facilite la construction itérative de FDR
- ➔ Simplifie la partie action

Diverses propositions

- ➔ Amélioration de la stratégie de filtrage
- ➔ Support de la mobilité
- ➔ Héritage statique
- ➔ Héritage dynamique
- ➔ Composant indépendant
- ➔ Visualisation

→ Bilan & Perspectives ←

Bilan personnel

- ➔ Formation à la recherche
 - Publication
 - Présentation et semaine de conférence
 - Autonomie dans le travail
- ➔ Travail préliminaire à une thèse

Bilan pour le service

- ➔ Positionnement dans l'état de l'art
- ➔ Etat de l'art sur les architectures et langages
- ➔ Analyse et étude du modèle
- ➔ Propositions d'extensions
- ➔ Implémentations

Pour la fin du stage

- ➔ Version longue de l'article sur ALBA (LNCS)
- ➔ Poursuivre et finaliser l'implémentation de certaines propositions dans le modèle actuel :
 - Gestion des signaux
 - Encapsulation de toutes les fonctionnalités d'ALBA
 - Structuration de la mémoire

Thèse CIFRE à THALES?

- ➔ Développer, formaliser et implémenter les idées proposées
- ➔ Conception d'un langage de haut niveau pour décrire le comportement des agents cognitifs
- ➔ Mise en place de l'outillage nécessaire à la mise en oeuvre des systèmes en vraie grandeur
- ➔ Problème du compromis entre aspects cognitifs et aspects réactifs

→ Questions ←