THALES

Réalisation d'une bibliothèque de déploiement de systèmes multiagents

Stage de fin d'études – EPITA 2005

Benjamin DEVEZE
Spécialisation SCIA

EPITA

Plan de la présentation

I. Présentation de l'entreprise

II. Sujet du stage et Existant

III. Les Réalisations

IV. Conclusion

Questions

Présentation de l'Entreprise

Présentation de ThalesGroup

Domaines

- Aéronautique
- Défense
- Sécurité

Chiffres

- *60 000* Collaborateurs dans *50* pays
- C.A. 2004 : *10,3* Milliards d'euros

Organisation

- Systèmes Aériens
- Systèmes Terre et Interarmées
- Naval
- Sécurité
- Service
- Aéronautique

Présentation du Service

But

Appliquer les techniques de l'intelligence artificielle aux domaines de l'entreprise

Equipe

- Environ 10 personnes
- Stagiaires, étudiants en thèse

Axes de recherche

- Représentation des connaissances
- Diagnostic automatique
- Planification
- Arithmétique des intervalles
- Programmation logique et à contraintes
- Raisonnement en temps contraint
- *Systèmes multiagents*

➔ Sujet et Existant ➔

Les SMA dans le service

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 erreurs prise en compte à la source

Présentation d'ALBA

Description

- Bibliothèque de déploiement d'agents écrits en Prolog

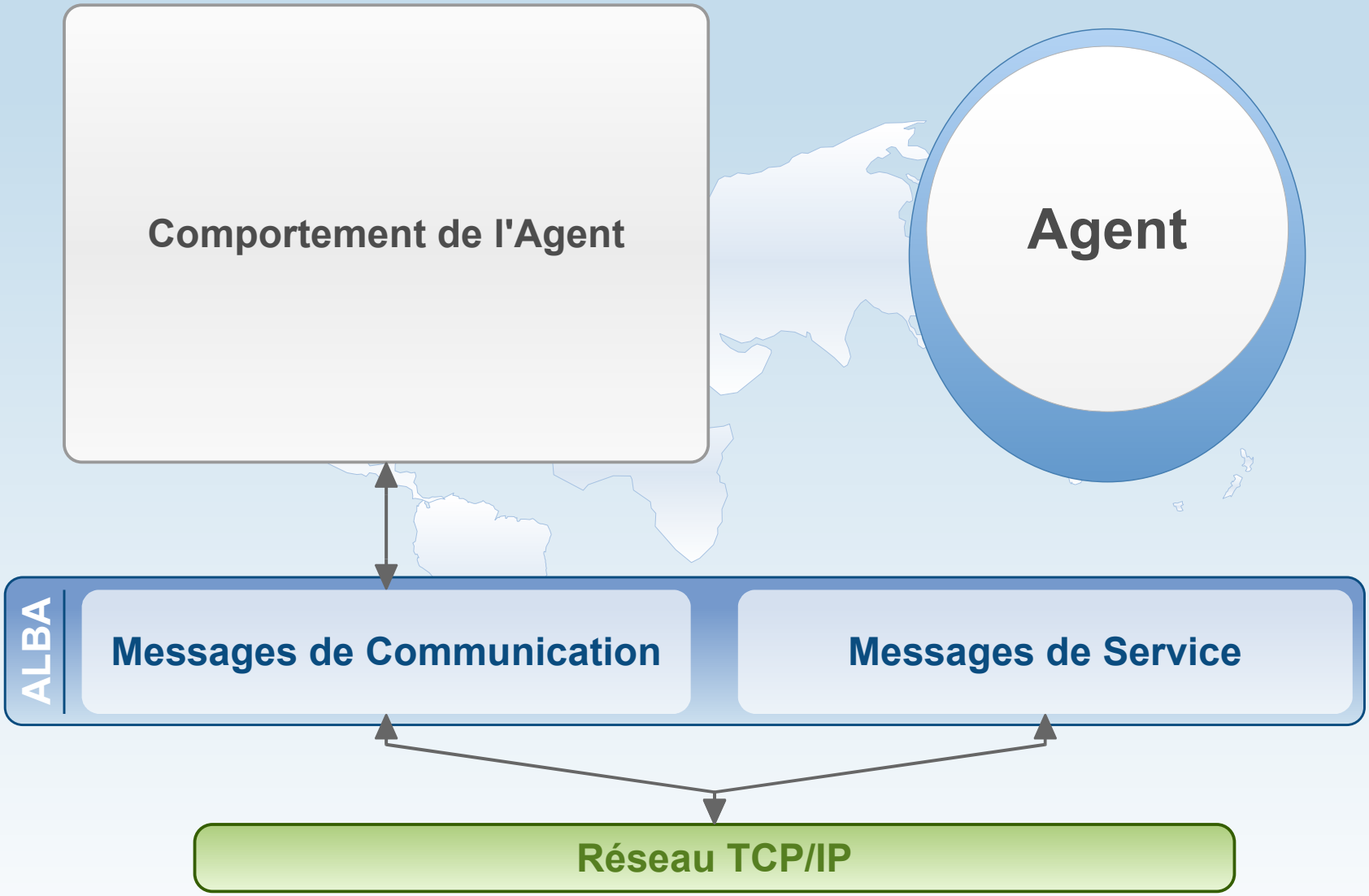
Fonctionnalités

- Gestion efficace des communications
- Création locale et distante d'agents
- Migration des agents
- Gestion d'agents hétérogènes

Axes directeurs

- Abstraction des tâches de bas niveau, gestion des erreurs
- Simplicité de programmation
- Décentralisation
- Généricité
- Robustesse, tolérance aux fautes
- Portabilité

Communication



Pourquoi Prolog?

- Langage bien adapté aux domaines de l'intelligence artificielle

- Productivité
- Maintenabilité

- Introspection
- Modification dynamique du code

- Reprise du corpus de programmes du service

L'existant : ALBA 1.0

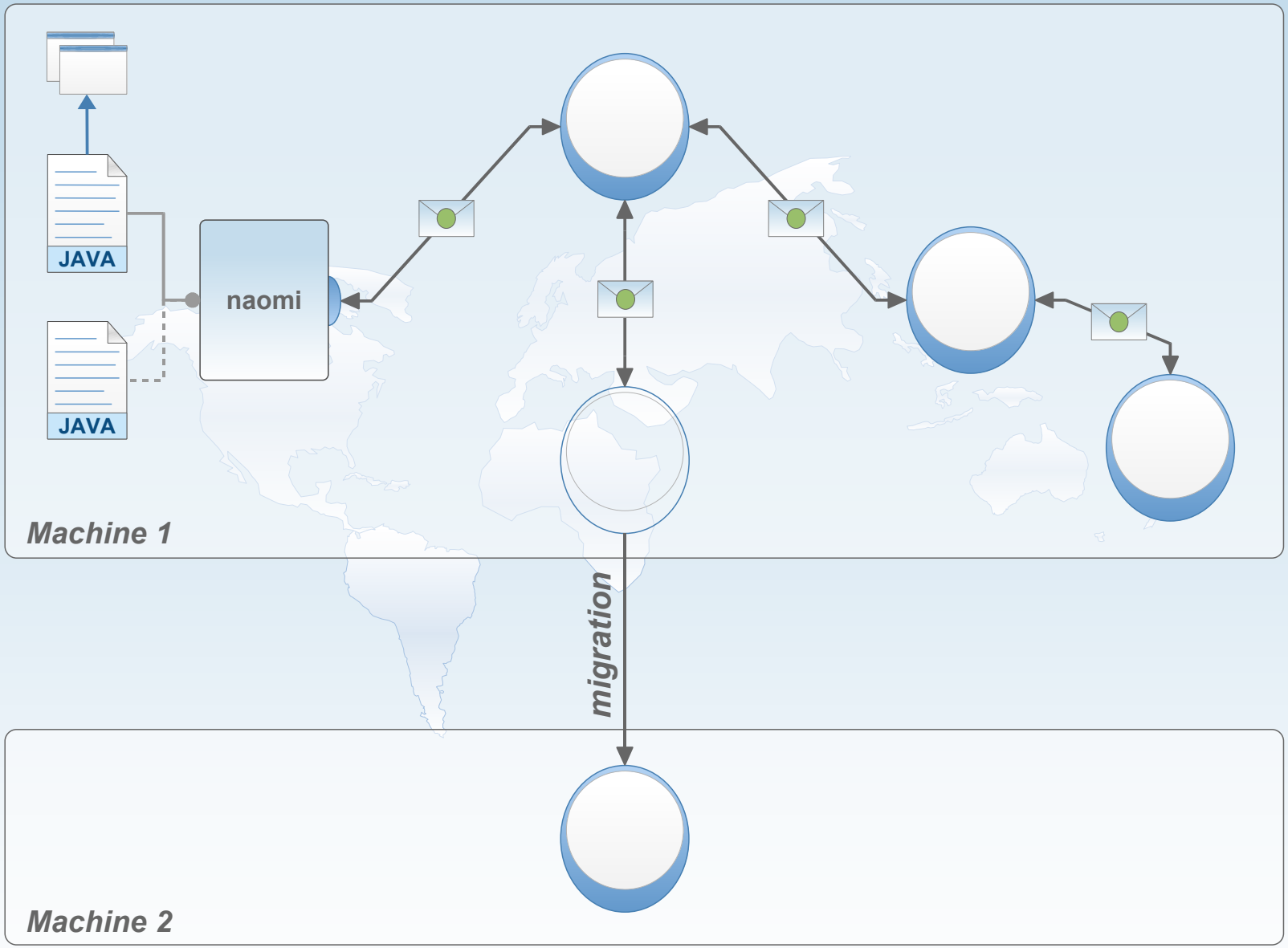
Apports

- Bibliothèque fonctionnelle
- Démonstration pratique que cela fonctionne
- Acquisition d'un bon savoir faire dans le domaine

Limitations

- Gestion de la mémoire
- Migration
- Remontée de messages de bas niveaux
- Communications perfectibles

Schéma général de fonctionnement



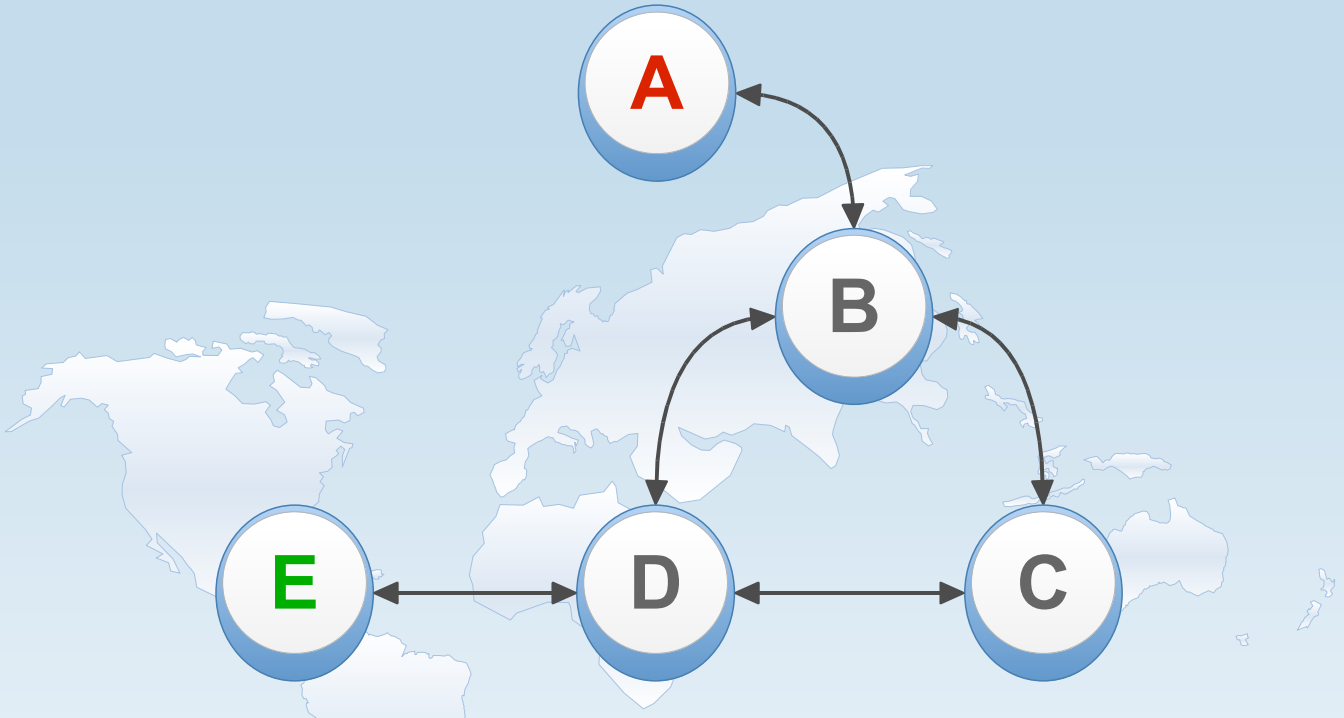
➔ Les Réalisations ➔

Principaux apports d'ALBA 2.0

Apports

- Refonte complète du code
- Rationalisation des communications, séparation des messages de service et des messages entre agents
- Abstraction des routines, gestion des erreurs
- Gestion de la mémoire
- Ajout d'agents externes
- *Recherche d'agents*
- *Migration fonctionnelle*
- *Intégration d'évènements*

Recherche d'agents

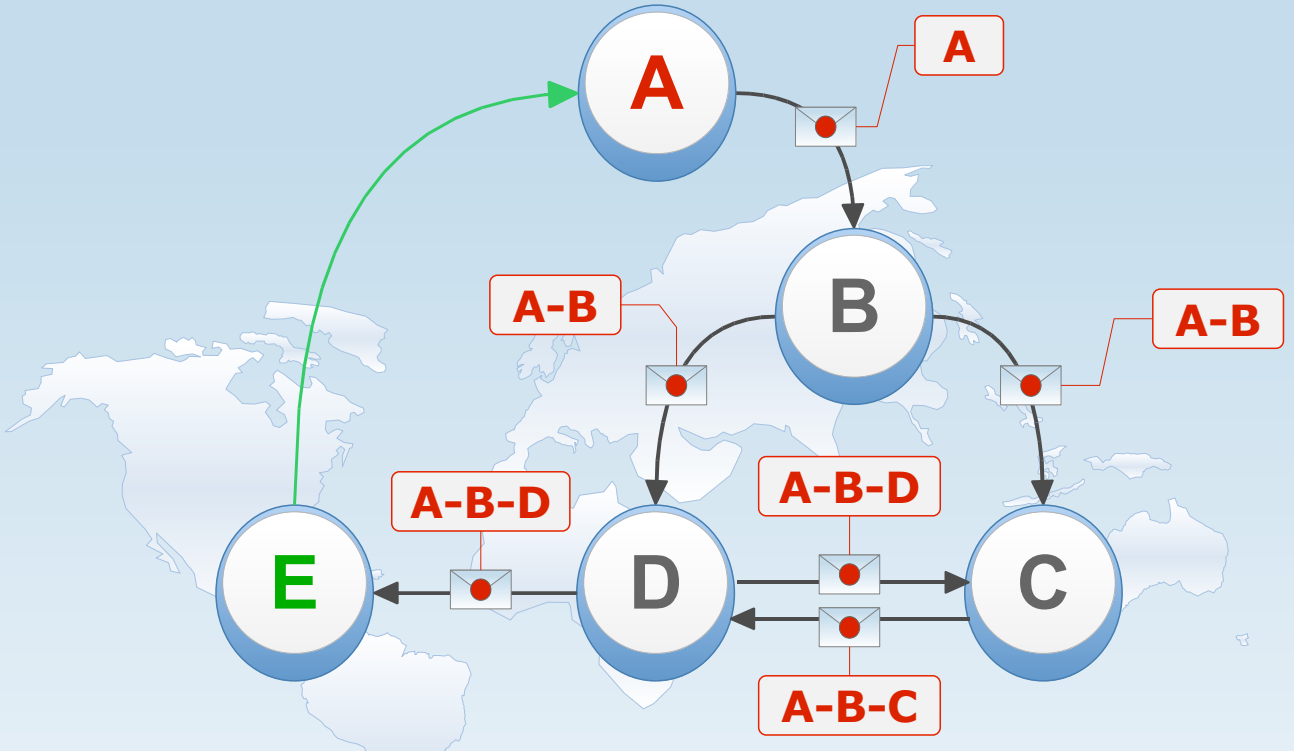


Problématique

- Analogie avec la recherche dans un graphe

- Pas de bouclage
- Limitation du nombre de messages émis
- Mise en contact de deux agents connexes

Recherche naïve d'agents

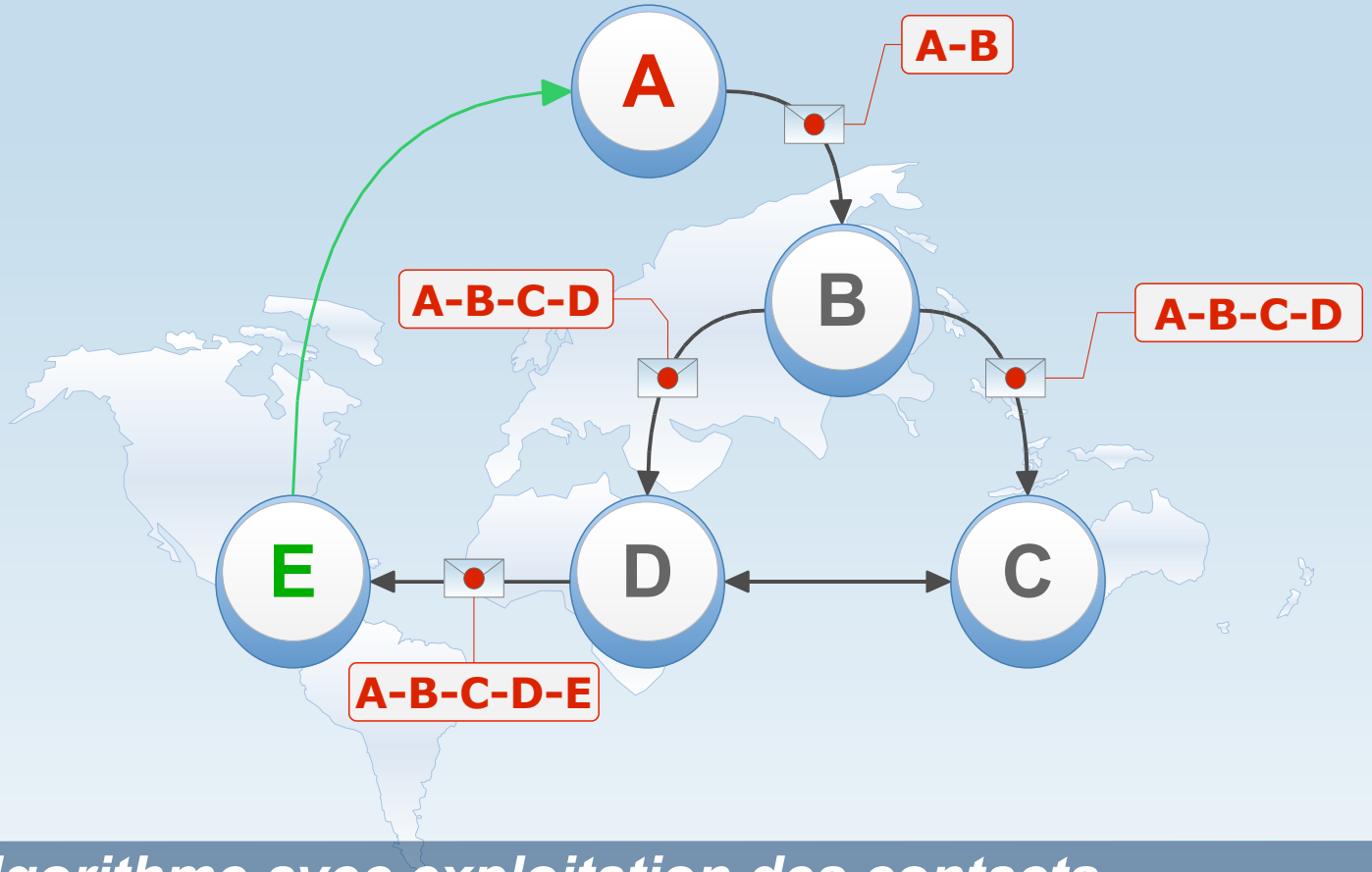


Algorithme naïf

- Pas de bouclage
- Mise en contact de deux agents connexes

- Trop de messages émis (n! au pires cas)

Recherche d'agents - Algorithme



Algorithme avec exploitation des contacts

- Limitation du nombre de messages émis
- Mécanisme de correction d'erreurs

Migration

But

- Faire migrer les agents vers des machines distantes en faisant en sorte que la migration soit transparente pour le reste du système

Intérêts

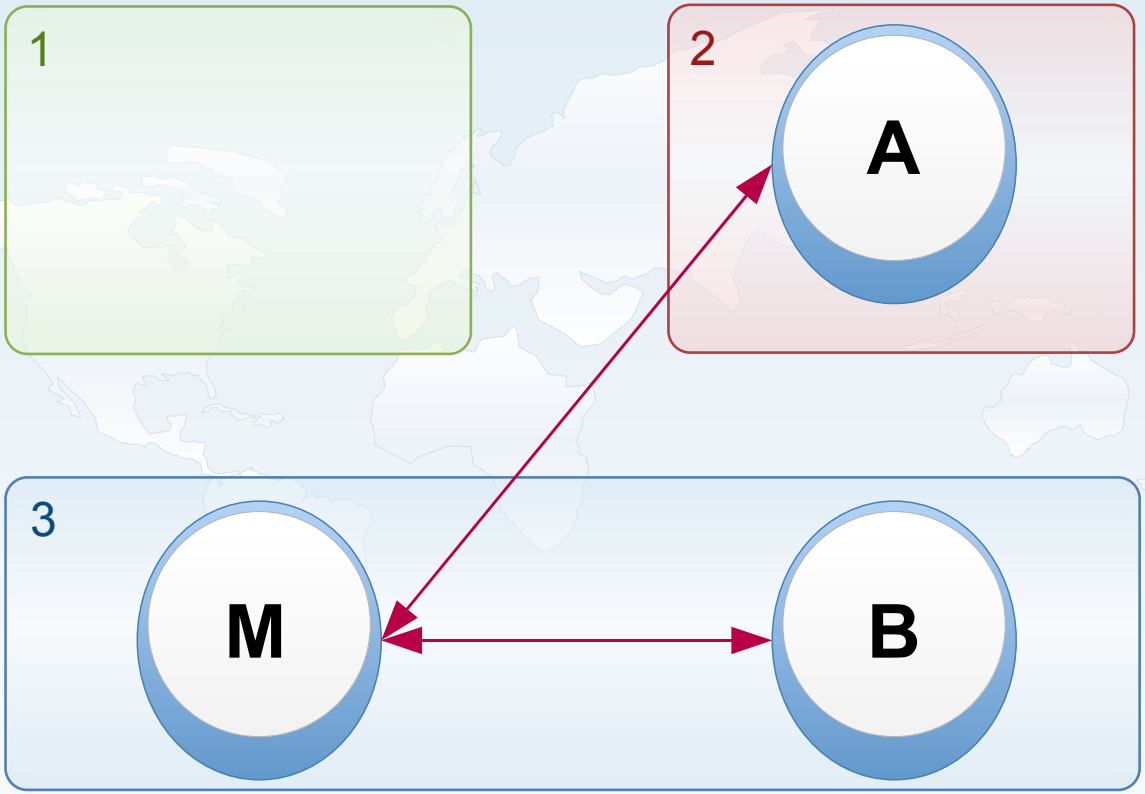
- Déploiement multimachine dynamique
- Répartition des charges de calcul et d'utilisation mémoire

Difficultés

- Reprise de l'exécution
- Procédures de rapatriement du code
- Gestion efficace des transferts de données

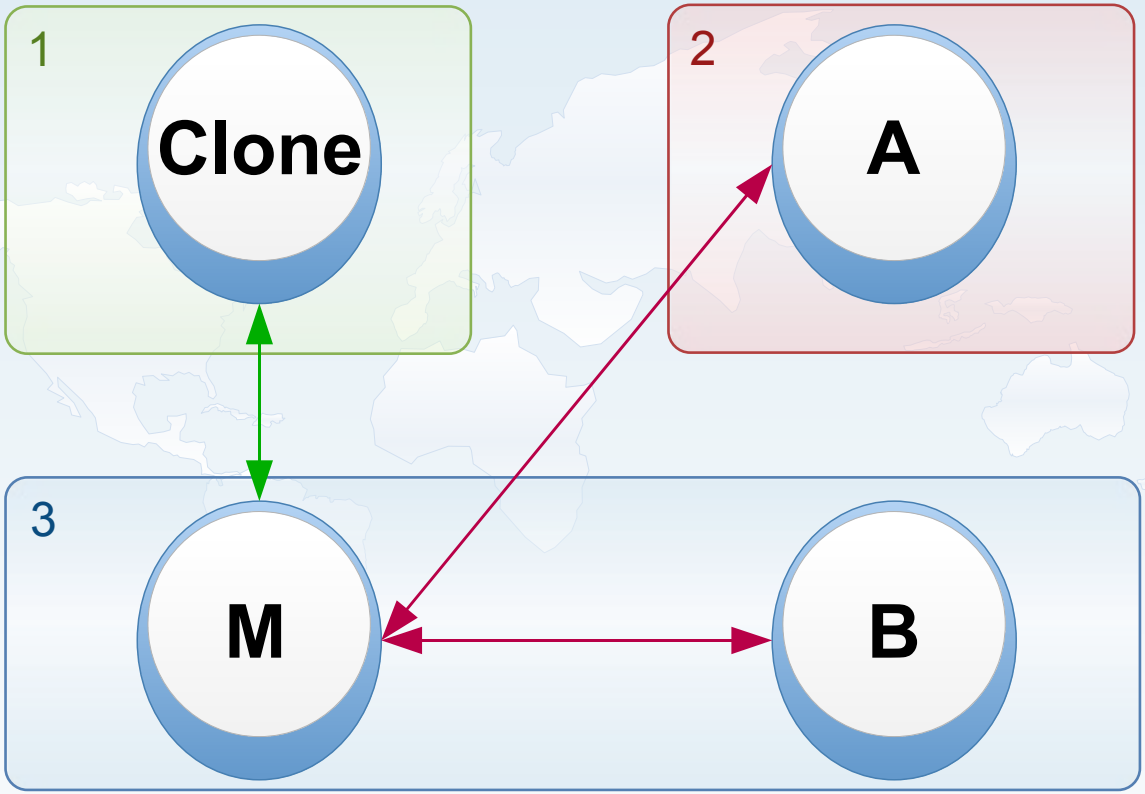
Migration – Algorithme (1)

Etape 1



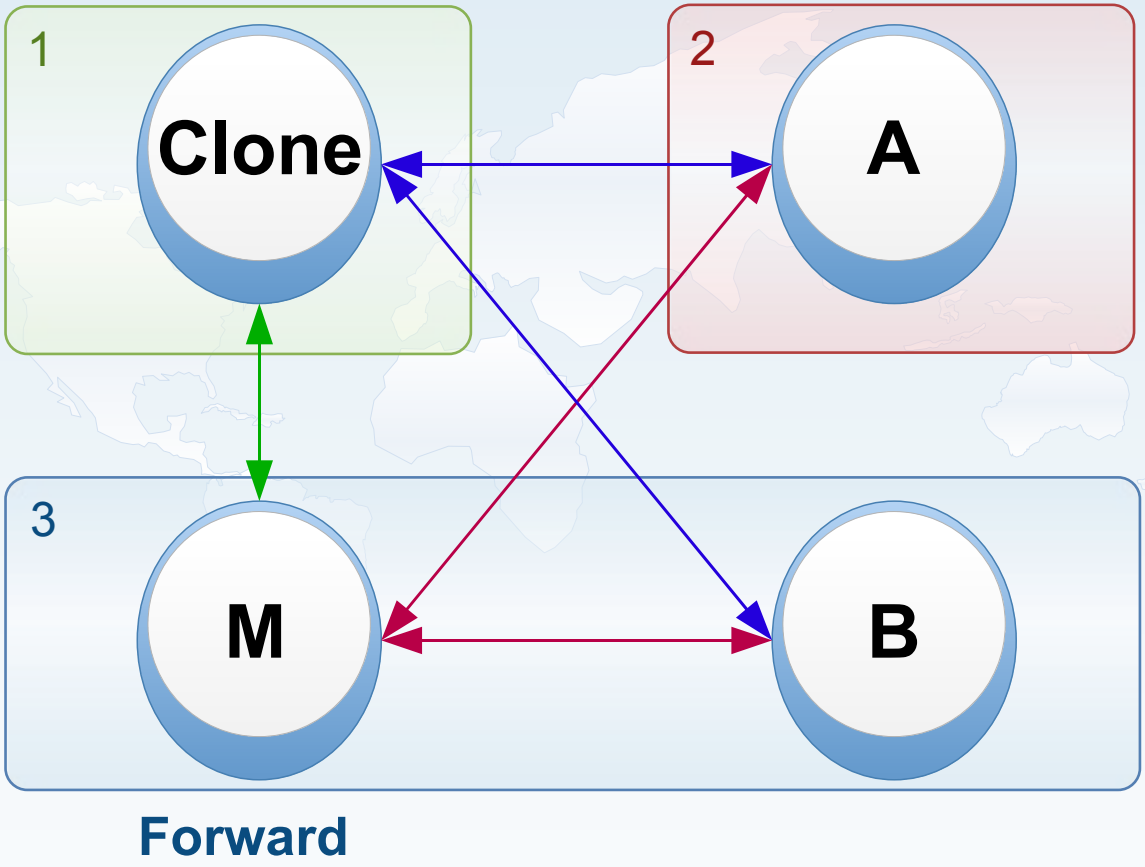
Migration – Algorithme (2)

Etape 2



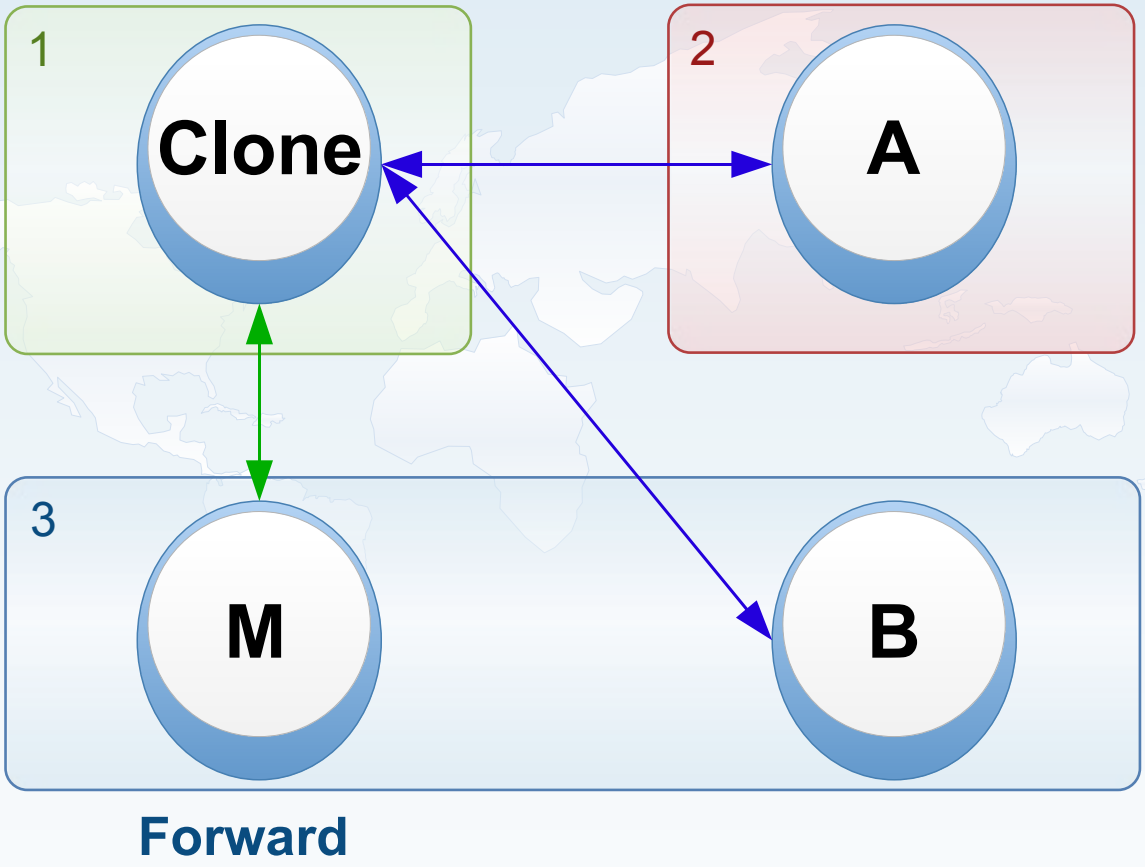
Migration – Algorithme (3)

Etape 3



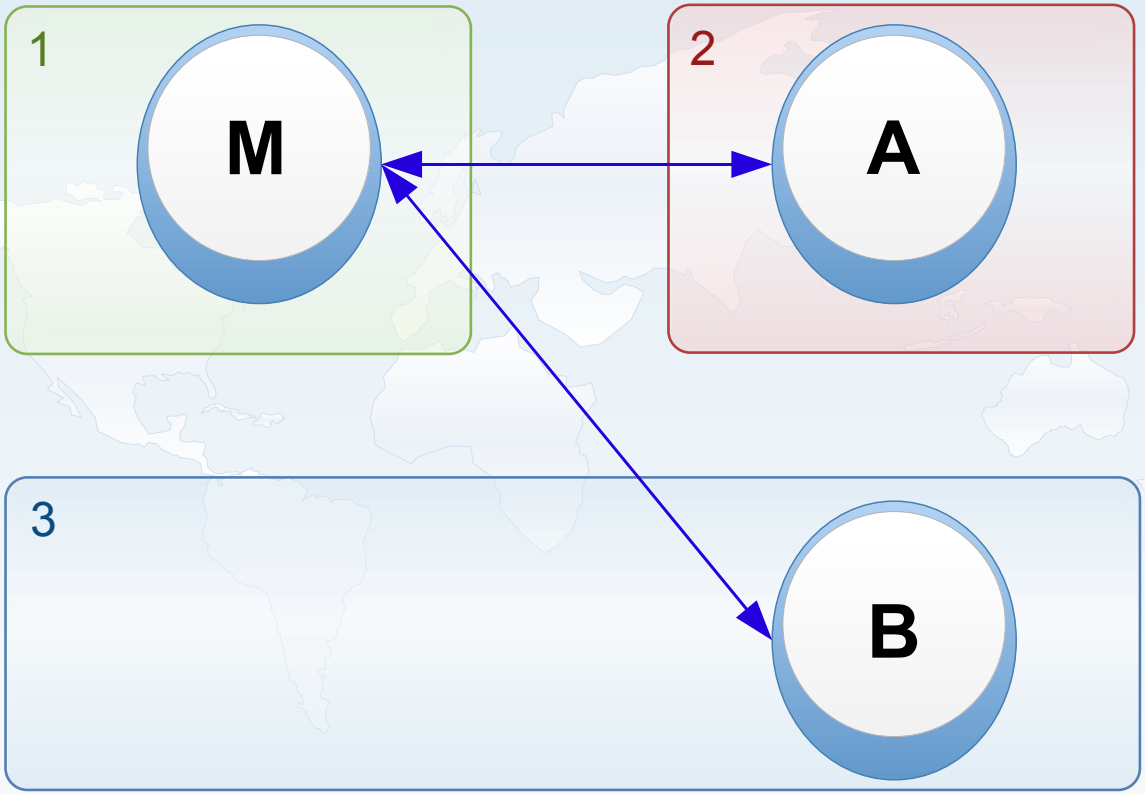
Migration – Algorithme (4)

Etape 4



Migration – Algorithme (5)

Etape 5



Bibliothèque d'évènements

Fonctionnalités

- Définition d'évènements
- Ajout et suppression de *handlers* associés à ces évènements

Principe

- Modification dynamique du code des prédicats

Applications

- Dans ALBA :
  - Evènements : *OnMessageRead, OnMessageSent ...*
  - Ajout de *plugins* sans toucher au code d'ALBA
  - Outil de débogage
  - Instrumentalisation du code permettant de vérifier la cohérence du système
- Généricité, utilisable dans tous les développements Prolog

Description

- Outil permettant de générer de la documentation HTML à partir de code source Prolog dûment commenté

Motivations

- Générations de stagiaires
- Code pas toujours documenté
- Recherche rapide de prédicats dans de nombreux fichiers

➔ Conclusion ➔

Conclusion

Apports au service

- Bibliothèque fonctionnelle pour le déploiement des SMA
- Bibliothèque d'évènements, PrologDoc

Apports Humains

- Vision de la recherche et du développement dans l'Entreprise

Apports Techniques

- Amélioration des connaissances en matière de SMA
- Expérience du développement dans un contexte distribué
- Apprentissage du langage Prolog

Perspectives

- Prolongation : *Coopération aérienne multiplateforme*
- Master Recherche en IA?

➔ Questions ➔