

17ème Journées Francophones des Systèmes Multi-Agents
du 19 au 21 octobre 2009, à Lyon



17ème Journées Francophones des Systèmes Multi-Agents

LA SIMULATION CONJOINTE LOGICIELLE/MATERIELLE COMME PARTIE DU CYCLE DE DEVELOPPEMENT DU SMA EMBARQUE

L'outil MASH: MultiAgent Software/Hardware simulator

Jean-Paul JAMONT, Michel OCCELLO

{jean-paul.jamont,michel.occello}@iut-valence.fr



PLAN

INTRODUCTION

I. Conception traditionnelle de SMA
embarqués

II. Notre approche avec l'outil MASH

III. Une petite démonstration

CONCLUSION

Introduction : Contexte

- Nos travaux précédents sur la conception/réalisation de systèmes multi-agents embarqués
 - La méthode DIAMOND
 - Le modèle MWAC
 - Applications:
 - Systèmes d'instrumentation
 - Géolocalisation
 - Robotique collective
 - Des outils...

3

Introduction : Conception de eMAS

- Construire un SMA embarqué ne consiste pas simplement à déployer le code d'un agent sur une plateforme
- Les agents d'un SMA embarqué sont contraints.
 - Energie *Mode sommeil/réveil...*
 - Mémoire *Démo avec agent embarqué disposant de 16ko (pile=256b)*
 - Puissance CPU *Démo avec agent embarqué disposant d'un proc cadencé à 4MHz*
- ⇒ L'aspect « embarqué » impacte les modèles, les architectures et les implémentations des agents
- ⇒ La dégradation des modèles/architectures pour les dimensionner aux ressources disponibles et autres contraintes insèrent des déviations dans le fonctionnement global du système

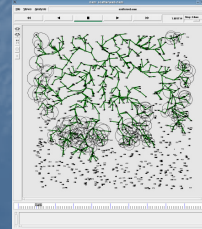
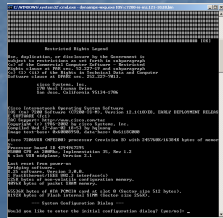
4

Conception traditionnelle de SMA embarqués

Trois approches

On peut distinguer 3 familles d'approches:

- Approches **classiques** (\approx code and fix)
- Approches basées sur la **simulation**
- Approches basées sur l'**émulation**



Conception traditionnelle de SMA embarqués

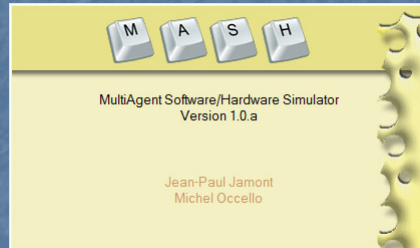
Travaux connexes

- Travaux de D. Weyns, K. Schelfhout, T. Holvoet, T. Lefever [AAMAS2005]
=> Etablir un **environnement virtuel à partir d'observations physiques** pour faire de la **planification** d'actions.
- Travaux de F. Hassaine, R. Moulton, C. Fink [SAC2009]
=> **Avoir des simulations réalistes**

Notre approche basée sur la simulation logicielle matérielle

Le simulateur MASH

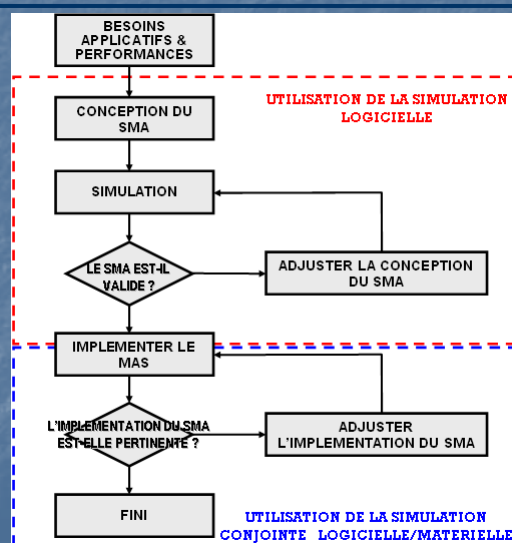
MASH: MultiAgent Software/Hardware simulator



7

Notre approche basée sur la simulation logicielle matérielle

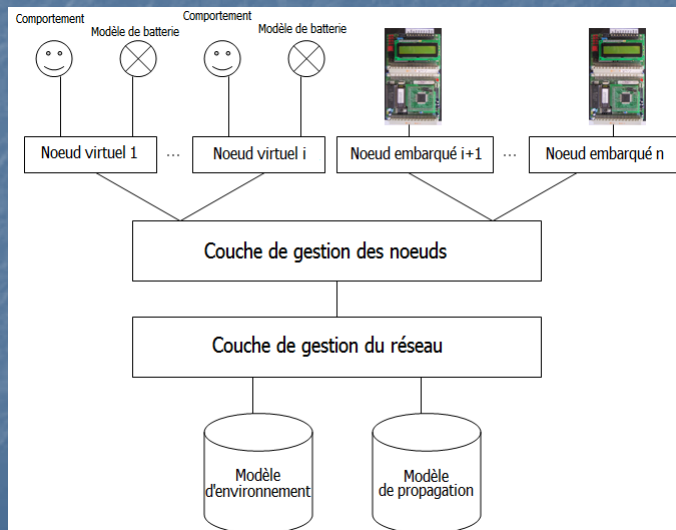
Le cycle de vie de l'outil MASH



8

Notre approche basée sur la simulation logicielle matérielle

Architecture simplifiée de MASH



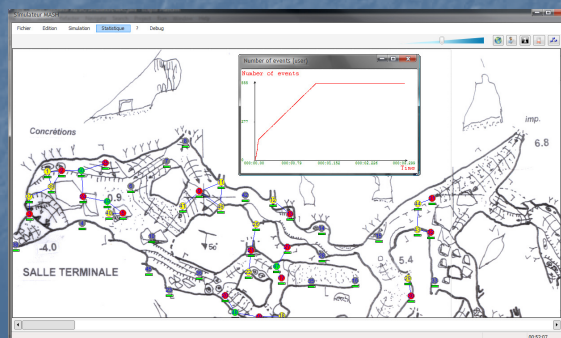
9

Notre approche basée sur la simulation logicielle matérielle

Les différents types de simulation

Simulation du SMA purement logiciel

- + permet une **évaluation facile** du système
- + permet de **simplifier la conception** de **processus décisionnels**
- La **qualité des simulations** dépend de la **qualité des différents modèles** mis en jeu
- On dispose de **beaucoup de mémoire** et de **CPU puissants**



10

Notre approche basée sur la simulation logicielle matérielle

Les différents types de simulation

Simulation du SMA hybride logiciel/matériel

- + Permet de tester les agents embarqués/le SMA embarqué à faible coût
- + Fournit un support durant la phase de débogage
- Gestion du temps

Simulation du SMA purement matériel

- + Le code fonctionne sur les plateformes cibles mais les agents interagissent via l'environnement simulé
- + Il est possible de debugger l'agent embarqué avec un lien série, un lien USB...
- + Le simulateur peut être utilisé comme un outil de visualisation et d'analyse de résultats/traces
- + Comme en simulation logicielle, on peut jouer des scénarios
- Pour avoir des simulations réalistes, il faut de nombreuses plateformes matérielles



11

Conclusion

MASH : a MultiAgent Software Hardware Simulation

- L' utilisation des différentes simulations MASH permet de:
 - Concevoir un SMA virtuel,
 - D'accompagner l'embarquement du code,
 - Tester le SMA embarqué et le debugger,
 - Evaluer les performances de ce système.
- => Concevoir une société hybride d'agents logiciels/matériels

12

Démonstration

- Disponible au téléchargement début d'année prochaine
- Il est réellement facile à utiliser
 - Langage Java , pas de temps simulé
 - Hériter une classe Agent/Object
 - Implémenter les services nécessités LocationInterface,ObjectMotionInterface...
- Peut-être utilisé pour plusieurs domaines d'application:
 - Application du monde réel (ambient/ubiquitous computing,réseaux,robotique collective)
 - Applications « traditionnelles » des SMA
 - Utilisation comme outil pédagogique
 - Enseignement des réseaux
 - Système multi-agent, informatique distribuée