

Les artifacts de calcul

Une solution aux délibérations longues

C. Dinont, E. Druon, P. Mathieu et P.Taillibert

JFSMA'06

20 octobre 2006



THALES

- 1 Problème
- 2 Artifacts
- 3 Artifacts de calcul
- 4 Discussion
- 5 Conclusion

Contexte

La gestion du temps dans les SMA

- Plusieurs contextes de conversation simultanés
- Signaux d'attention
- Plusieurs contextes d'action (calculs)
- Délais à respecter pour les calculs
- Partage des ressources processeur
 - Contrainte pour la gestion des délais
 - Assurer l'extraversion des agents

Problème

Agents avec de longs calculs

- Algorithmes complexes
 - Algorithmes issus de l'IA
 - Utilisation de plusieurs heuristiques
 - Code hérité
 - Code non modifiable
 - Mais propriétés connues :
 - Algorithme à durée indéterminée,
 - Algorithme à contrat,
 - Algorithme anytime, ...

Problème

Agents avec de longs calculs

- Algorithmes complexes
 - Algorithmes issus de l'IA
 - Utilisation de plusieurs heuristiques
 - Code hérité
 - Code non modifiable
 - Mais propriétés connues :
 - Algorithme à durée indéterminée,
 - Algorithme à contrat,
 - Algorithme anytime, ...

Idée

Encapsuler les algorithmes exécutés par les agents dans des boîtes translucides réutilisables disposant d'un "mode d'emploi"

Problème

Réification du concept d'agent

- But : casser la complexité structurelle des systèmes
- Conserver les propriétés conceptuelles des agents dans les programmes :
 - Autonomie / Proactivité
 - L'autonomie d'un agent est prise en compte dans la programmation des AUTRES agents
 - On n'a pas de modèle précis des autres agents
 - Réactivité / Extraversion
 - Se remettre régulièrement en phase avec l'environnement

Problème

Réification du concept d'agent

- But : casser la complexité structurelle des systèmes
- Conserver les propriétés conceptuelles des agents dans les programmes :
 - Autonomie / Proactivité
 - L'autonomie d'un agent est prise en compte dans la programmation des AUTRES agents
 - On n'a pas de modèle précis des autres agents
 - Réactivité / Extraversion
 - Se remettre régulièrement en phase avec l'environnement

Ne pas reporter la complexité sur la conception de chaque agent

Problème

Où réaliser les calculs longs?



Artifacts



Artifacts (Omicini, Ricci, Viroli)

Théorie de l'activité humaine

- L'utilisation d'outils a toujours accompagné l'évolution de l'espèce humaine
- Le développement de l'intelligence est lié à la disponibilité et au développement des outils

Des outils pour les agents

- Artifacts : des entités utilisées par les agents pour atteindre leurs buts

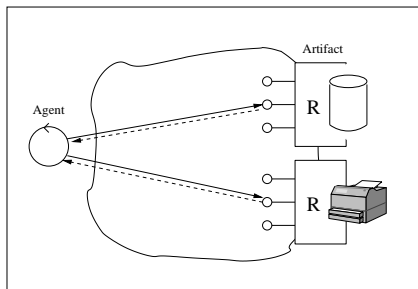
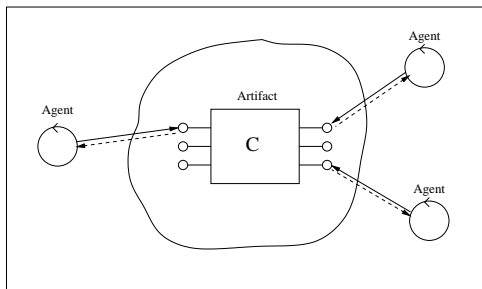
Artifacts (Omicini, Ricci, Viroli)

Définition

Un artifact est caractérisé par :

- son interface d'usage (*actions et perceptions*)
- sa fonction (*description des services rendus*)
- ses attributs (*paramètres et variables internes exhibées*)
- ses instructions opératoires (*mode d'emploi*)

Utilisations possibles des artifacts



Langage de mode d'emploi (Omicini, Ricci, Viroli)

Langage de mode d'emploi

- Basé sur les algèbres de processus
- Deux types d'actes d'interaction : actions et perceptions
- Instructions possibles :

$$I ::= 0 \mid !\alpha \mid ?\pi \mid I;I \mid I + I \mid (I||I) \mid \mathcal{D}(t_1, \dots, t_n)$$

- Définition d'une sémantique opérationnelle
 - Décrit l'évolution de l'instruction opératoire en fonction des événements

Exemple de mode d'emploi

$((!a ; ?fin_a) + (!b ; ?fin_b)) || (!c ; ?fin_c)$

Extensions du langage de mode d'emploi (1/2)

Gestion du temps

- **Instruction de timeout** : $T(n)$ - *Viroli et al. AAMAS'04*
 - Obligation de réaliser une action avant n tops d'horloge
 - Garantie de recevoir une perception avant n tops d'horloge
- **Instruction d'attente** : $W(n)$ - *Dinont et al. AAMAS'06*
 - Attendre n tops d'horloge avant d'exécuter une action
 - Délai minimal avant de recevoir une perception

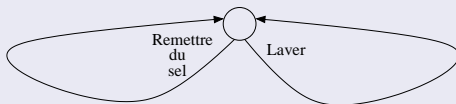
Récupération après expiration d'un timeout

- **Nouvelle instruction de timeout** : $T(n, I)$ - *P.-E. Marson '06*
 - On ne tombe pas dans l'état d'erreur
 - L'instruction opératoire I est exécutée si le timeout expire

Extensions du langage de mode d'emploi (2/2)

Gestion de conditions sur l'historique

- But : Permettre des règles comme :
"Au bout de 10 lavages, il faut rajouter du sel"
- Fonctionnement :
 - Mémorisation de l'historique des actions et des perceptions
 - Utilisation d'expressions rationnelles pour inhiber ou réactiver des branches de l'instruction opératoire



```
`a_remplir_sel;p_fin_replissage;$'' ->
  {activate(laver);deactivate(remettre_du_sel)}
`a_remplir_sel;p_fin_replissage;(a_laver;p_fin_lavage;){10}$'' ->
  {deactivate(laver);activate(remettre_du_sel)}
```



Artifacts de calcul



Artifacts de calcul

Les artifacts de calcul...

- encapsulent l'exécution d'un algorithme
- permettent aux agents de rester extravertis en externalisant leurs calculs longs

Interface d'usage minimale

- Actions :
 - `start_computation(DATA)`
 - `stop_computation`
 - `pause_computation, restart_computation`
- Perceptions :
 - `finished(RESULT)`
 - `error(ERROR_INFO)`
 - `paused, restarted`

Mode d'emploi des artifacts de calcul

Algorithme a contrat

```
contract_algorithm :=  
  !start_computation(INPUT_DATA, CONTRACT) ;  
  ( ( (?finished(RESULT) + ?error(ERROR_INFO))  
      + (!stop_computation ; ?computation_stopped)  
      + (T(CONTRACT) ; ?end_contract ; contract))  
    || pause_restart) ;  
contract_algorithm
```

```
contract := !continue(CONTRACT) ;  
  ( (?finished(RESULT) + ?error(ERROR_INFO))  
    + (!stop_computation ; ?computation_stopped)  
    + (T(CONTRACT) ; ?end_contract ; contract)
```



Discussion



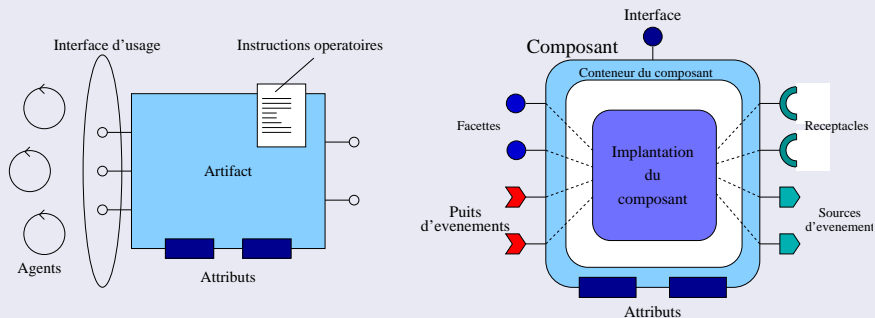
Discussion

Différences agents / artifacts

Agent	Artifact
Autonome	Pas autonome
Extraverti	Introverti
Dirigé par des buts	Orienté service
Actes de langages	Actions et perceptions

Discussion

Différences artifacts / composants



Discussion

Le langage de mode d'emploi

- ✓ Simple
- ✓ Sémantique opérationnelle définie
- ✓ Extensible
- ✗ Trop rigide :
 - Impossibilité de décrire des conseils
- ✗ Possibilités de raisonnement limitées

→ Conclusion - Perspectives ←

Conclusion

Buts

- Des agents avec de longs calculs
- Conserver autonomie et extraversion

Solution

- Des outils de calculs pour les agents
- Réutilisation / extension du concept d'artifact
 - Langage de mode d'emploi basé sur les algèbres de processus
- Simplification du codage des agents
- Réutilisabilité des artifacts dans différents contextes
- Evaluation sur une application de Thales [JFSMA'06, AAMAS'06]

Perspectives

Extensions

- Découverte dynamique de nouveaux artifacts
- Langage de mode d'emploi plus flexible
- Raisonnement sur les durées d'exécution des algorithmes
- Utiliser les artifacts dans d'autres contextes où les actions s'inscrivent dans la durée
 - Utilisation d'un appareil électro-ménager
 - Utilisation d'un véhicule
 - Utilisation d'un téléphone portable
- Utilisation conjointe de plusieurs artifacts
 - Téléphone portable en voiture



Merci
Questions ?

