

Processus formel à l'aide du raffinement pour le passage d'une idée de protocole à un protocole pour des agents

Marc-Philippe Huget

Université de Savoie

LSE – ESIA – LISTIC

Marc-Philippe.Huget@univ-savoie.fr

Plan de la présentation

- Contexte
- Quid du raffinement
- Raffinement dans le cadre des protocoles d'interaction
- Application
- Conclusion et perspectives

Contexte

- Systèmes dynamiques pour deux types de « public »
 - Les agents mobiles
 - Les utilisateurs
- Notion d'agents kleenex : on les utilise une fois et après on les jette

Contexte : les agents mobiles

- Agents poids lourd / Agents poids plume pour une tâche donnée
- Exemple : Agent créé pour négocier l'accès à une ressource
- L'agent n'a que les connaissances nécessaires pour effectuer la tâche

Contexte : les utilisateurs

- Proposer des agents au service des utilisateurs qui répondent à un besoin donné
- Exemple : acheter un CD sur l'Internet, obtenir des informations administratives pour un déménagement
- Qualifier un agent en fonction du besoin

Idée générale

- Fortement emprunte de la programmation orientée interaction
- Faciliter une tâche ardue : l'intégration des protocoles dans les agents
- Partir du protocole et mettre l'agent autour/dedans plutôt que l'inverse

Raffinement

- Notion décrite en premier par Wirth pour la conception de programmes
- Idée assez simple : concevoir un programme en effectuant de petites modifications entre deux compilations, plutôt que considérer de grandes modifications

Raffinement

- Raffinement = type particulier de processus de développement
- Approche formelle + raffinement = conception incrémentale
- Approches formelles considérées : CCS, Z, B pour n'en citer que quelques-uns
- Apport réel : possibilité de valider le modèle défini à chaque étape du raffinement = test de régression linéaire

Raffinement

- Deux types de raffinement :
 - Décomposition ou raffinement horizontal
 - Transformation ou raffinement vertical

Raffinement horizontal

- Ne change pas le niveau d'abstraction
- Ajoute de nouvelles fonctionnalités à un système
- Exemple 1 : ajout de la caractéristique fiabilité dans un protocole
- Exemple 2 : ajout d'un message supplémentaire dans un protocole
- Différents niveaux de granularité pour la décomposition

Raffinement vertical

- Change le niveau d'abstraction
- Souvent utilisé pour aller d'un Platform Independent Model à un Platform Specific Model
- Exemple 1 : spécialisation d'un protocole pour des messages au format FIPA
- Exemple 2 : spécialisation de la notion d'entiers à celle d'int pour le langage C
- Typiquement utilisé pour produire du code

Raffinement dans les protocoles

- Il peut être utilisé à plusieurs niveaux :
 - Définir le protocole
 - L'instancier pour un problème particulier
 - Définir l'agent
 - Générer le code correspondant

Comment appliquer le raffinement ?

- Nécessité d'avoir :
 - Un modèle (formel ou non)
 - Des règles de raffinement
- Ici :
 - Automate à état fini
 - Règle du type : préconditions - actions
- Non considéré ici : logique de réécriture

Application

Conclusion

- De nombreux obstacles se présentent encore devant nous :
 - Exhaustivité des règles de raffinement (comment être sûr que l'on a bien tout)
 - Caractère exclusif de règles (comment s'assurer que l'on n'a pas deux règles qui s'opposent)
 - Niveau d'abstraction du formalisme (comment concilier un point de vue formel avec des concepts multi-agents)
 - Génération de code (comment réussir à générer du code)

Conclusion

- Lié aux travaux de thèse de Selma Azaiez sur le processus de raffinement pour la conception de systèmes multi-agents

Merci de votre attention. Questions ?