

Offre de stage

Inria Nancy - Grand Est (Loria)
Équipe MAIA

Durée : 1 mois (gratifié)

1 Encadrants

Vincent CHEVRIER
Maître de Conférence UL
vincent.chevrier@loria.fr
Bureau C025
03 83 59 20 75

Julien VAUBOURG
Doctorant Inria
julien.vaubourg@inria.fr
Bureau C123
03 54 95 84 14

2 Titre du stage

Évolution de la couche de communication d'agents C++ pour la plateforme de multi-simulation AA4MM.

3 Description du stage

AA4MM [1] est un meta-modèle pour la multi-simulation qui s'appuie sur le paradigme *Agents & Artefacts* [2], une extension du paradigme des systèmes multi-agents. Il a la particularité de permettre de concevoir des simulations distribuées, en faisant notamment communiquer des simulateurs utilisant des formalismes hétérogènes, écrits dans des langages divers et s'exécutant sur des systèmes d'exploitation variés. Un simulateur intègre une multi-simulation AA4MM par le biais d'un agent, et ces derniers s'échangent des informations via des artefacts de couplage.

Afin de s'abstraire des différences liées à la représentation des données entre les diverses implémentations d'agents au sein d'une même simulation distribuée, nous avons choisi de faire transiter les informations en utilisant le format de données textuel JSON. Les échanges sur le réseau transitent au niveau des artefacts de couplage grâce à un middleware de communication qui implémente le standard DDS¹ défini par l'OMG (OpenSplice DDS Community²).

La version C++ de la plateforme utilise actuellement la bibliothèque JSON Spirit³ pour sérialiser les objets C++ qui contiennent les informations à transmettre aux agents distants (objets de données AA4MM). Le principal problème rencontré avec cette solution est que la bibliothèque doit être compilée manuellement, pour la rendre résistante aux threads (cf. le bug Debian⁴ ouvert par nos soins pour demander une version compilée avec cette option, disponible dans les paquets). Puisque la version C++ de AA4MM est liée à la bibliothèque Boost⁵ et que celle-ci propose également des fonctionnalités de sérialisation⁶ JSON, nous souhaiterions l'utiliser pour remplacer JSON Spirit et ainsi nous affranchir de cette dépendance problématique.

Le travail consistera donc à réimplémenter les fonctionnalités de sérialisation et désérialisation d'objets de données AA4MM, à l'aide de la bibliothèque Boost et son parseur JSON. Étant donnée l'absence de possibilités d'introspection sur les objets en C++, ceux qui seront à sérialiser disposeront obligatoirement des fonctions *toJSON* et *loadFromJSON*. Le code nécessaire pour écrire ces fonctions pour une classe donnée devra être le plus minimal et simple possible. Les opérations de sérialisation et désérialisation devront pouvoir se faire de façon récursive et en intégrant aussi bien des types natifs que des types `std::vector` ou `std::map`.

1. <http://portals.omg.org/dds/>

2. <http://www.primstech.com/opensplice/opensplice-dds-community>

3. <http://www.codeproject.com/Articles/20027/JSON-Spirit-A-C-JSON-Parser-Generator-Implemented>

4. <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=746385>

5. <http://www.boost.org/users/>

6. http://www.boost.org/doc/libs/1_54_0/doc/html/property_tree.html

La simplicité d'utilisation de la nouvelle implémentation devra être testée en procédant à la migration des exemples existants. Selon l'avancée du travail et le niveau du ou de la stagiaire, des travaux complémentaires pourront être proposés pour optimiser les communications gérées par OpenSplice ou implémenter de nouveaux exemples d'échanges avec la version Java de la plateforme.

4 Profil du candidat

Connaissance du C++, rigueur.

Références

- [1] J. Siebert, "Approche multi-agent pour la multi-modélisation et le couplage de simulations. application à l'étude des influences entre le fonctionnement des réseaux ambiants et le comportement de leurs utilisateurs." Ph.D. dissertation, Université Henri Poincaré - Nancy I, Sep. 2011. [Online]. Available : <http://tel.archives-ouvertes.fr/tel-00642034> 1
- [2] A. Ricci, M. Viroli, and A. Omicini, "Give agents their artifacts : The A&A approach for engineering working environments in MAS," in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '07. New York, NY, USA : ACM, 2007, p. 150 :1–150 :3. [Online]. Available : <http://doi.acm.org/10.1145/1329125.1329308> 1