

UNIVERSITÉ DE MONTPELLIER
FACULTÉ DES SCIENCES

Licence 3 Informatique

HLIN613 Stage effectué dans le cadre du Cursus Master en Ingénierie

**Création de tutoriels présentant les fonctionnalités de la
plateforme MaDKit**

RAPPORT DE STAGE

LIRMM, équipe SMILE
du 23 mai 2018 au 20 juillet 2018

Étudiante :
Amandine PAILLARD

Date de rédaction :
19 juillet 2018

Tuteur de stage :
Fabien MICHEL

Soutenu le :
1 septembre 2018



Remerciements

Avant d'entamer ce rapport, il y a des personnes qu'il me faut remercier. Sans eux ce stage n'aurait peut être pas eu lieu ou ne se serait pas aussi bien passé.

Parmi les personnes qui ont rendu ce stage possible je voudrais remercier Anne-Élisabeth BAERT pour m'avoir permis de faire un stage dans le domaine de la recherche. Merci également à Hinde BOUZIANE pour m'avoir conseillée et accompagnée dans mes recherches de stage tout au long de l'année.

Au sein de l'équipe dans laquelle j'ai travaillé, je suis bien évidemment reconnaissante envers Fabien MICHEL pour avoir accepté d'être mon tuteur et d'avoir consacré de son temps afin que je puisse découvrir les systèmes multi-agents et prendre en main les outils et technologies que j'allais être amenée à utiliser pendant mon stage. Je remercie également Anne TOULET, Nadira BOUDJANI, Abdelkader GOUAICH et Elcio ABRAHÃO pour leur accueil chaleureux et bienveillant au sein de l'équipe SMILE. Merci aussi à Iago BONNICI pour la bonne humeur et la joie de vivre que tu apportes chaque jour dans le bureau.

De même, je tiens à remercier toutes les personnes que j'ai pu croiser au LIRMM et en particulier Laurie et Gladys pour leur accueil, leur disponibilité ainsi que la gentillesse dont elles font preuve tous les jours.

Enfin, pour des remerciements un peu plus personnels, je pense à Julie et à Thomas pour avoir été là tous les jours et pour nous être soutenus mutuellement dans nos stages respectifs. Je remercie également mes amis qui m'ont fait découvrir le LIRMM avant même que je songe à y faire mon stage, merci à Maëlle, Rémi, Mattéo et Florent. Enfin parmi les personnes qui n'ont pas encore été citées je souhaiterais remercier celles qui m'ont aidée dans la rédaction de ce rapport de part leur correction et leurs remarques -pas toujours pertinentes- : merci Félix, Christiane et Jean-Christophe.

À tous, je vous remercie.

Table des abréviations

CMI : Cursus Master en Ingénierie

LIRMM : Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

SMILE : Système Multi-agents, Interaction, Langage, Evolution

API : Application Programming Interface

MaDKit : Multi-agent Development Kit

CNRS : Centre National de la Recherche Scientifique

MASQ : Multi-Agent System based on Quadrants

AGR : Agent / Group / Role

CGR : Community / Group / Role

Table des matières

1	Introduction	5
1.1	Contexte	5
1.2	Organisme d'accueil : le Laboratoire d'Informatique, de Robotique et de Microélec- tronique de Montpellier	5
1.2.1	Présentation	5
1.2.2	Activité	6
1.2.3	Organisation interne	6
1.3	Équipe SMILE : Système Multi-agent, Interaction, Langage, Évolution	7
1.3.1	Présentation	7
1.3.2	Domaines de recherche	7
2	Présentation de la mission	9
2.1	Enjeu	9
2.2	Présentation des outils et techniques employés	9
2.2.1	Introduction aux systèmes multi-agents	9
2.2.2	MaDKit, une API pour modéliser des systèmes multi-agents	9
2.2.3	Outils de travail employés	10
2.3	Organisation	11
2.3.1	Déroulement de la phase de prise en main	11
2.3.2	Déroulement de la réalisation des tutoriels	11
3	Travail effectué	13
3.1	Tutoriel sur les codes de retour dans MaDKit	13
3.1.1	But	13
3.1.2	Structure	13
3.2	Tutoriel sur les propriétés dans MaDKit	14
3.2.1	But	14
3.2.2	Structure	14
3.3	Tutoriel sur les options de MaDKit	14
3.3.1	But	14
3.3.2	Structure	14
4	Bilans et perspectives	15
4.1	Présentation des résultats	15
4.2	Retour sur les difficultés rencontrées	15
4.3	Conclusion et perspectives	15

Chapitre 1

Introduction

1.1 Contexte

Ce stage s'inscrit dans le cadre de ma troisième année de licence informatique en Coursus Master en Ingénierie à la Faculté des Sciences de l'Université de Montpellier. Cette formation préconise que les étudiants aient réalisés une cinquantaine de semaines de stage avant la fin de leur master et ceci non seulement dans le but de leur faire découvrir le monde de travail et de la recherche mais aussi de développer leur culture professionnelle. Pour mon stage j'ai donc décidé de découvrir le monde de la recherche qui m'était encore inconnu. Ainsi j'y ai vu l'occasion de m'approcher d'un milieu vers lequel je ne souhaite -à ce jour- pas m'orienter pour mon futur professionnel. Ce stage a donc été l'opportunité de découvrir un autre milieu que celui de l'entreprise - milieu que je connaissais déjà - et comme l'occasion, peut-être unique, de réaliser cette expérience dans ma vie.

Concernant le choix du laboratoire de recherche dans lequel faire mon stage, le LIRMM m'est vite apparu comme la réponse évidente. Nombreux de mes camarades de promotion avaient déjà réalisé un stage dans ce laboratoire et la majeure partie de notre corps enseignant y travaille. Je me suis donc tournée vers cet organisme et je me suis renseignée sur le travail qu'effectuait les différentes équipes de recherche qui le compose. Lors de ces recherches, l'équipe qui a le plus retenu mon attention est l'équipe SMILE. Cette équipe étudie les systèmes multi-agents et leur application pour divers domaines comme entre autres, les systèmes distribués ou bien les jeux vidéos sérieux ou ludiques. Dans le but de pouvoir travailler avec des systèmes multi-agents, des chercheurs de l'équipe ont créé une API permettant de modéliser ces systèmes.

Ainsi, l'enjeu de mon stage est d'améliorer la documentation existante par l'ajout de plusieurs tutoriels permettant de présenter le fonctionnement de diverses fonctionnalités de cette API.

Ce qui a donc motivé mon choix de faire un stage au sein de l'équipe SMILE du LIRMM est découvrir le monde de la recherche, mais aussi d'en apprendre plus sur les systèmes multi-agents, notion que je n'avais encore jamais étudiée et qui me semblaient être très intéressante.

Dans ce rapport, après avoir présenté le LIRMM, le travail de l'équipe SMILE et quelques notions du paradigme multi-agents, je détaillerai la mission qui m'a été confiée. Je préciserai également les divers outils avec lesquels j'ai pu travailler et conclurai sur les différents apports que ce stage m'a apporté pendant ces deux mois.

1.2 Organisme d'accueil : le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

1.2.1 Présentation

Le LIRMM est un laboratoire de recherche issu à la fois de l'Université de Montpellier et du CNRS. Créé en 1992 et situé sur le campus Saint-Priest aux bâtiments quatre et cinq, il regroupe

les départements informatique, robotique et microélectronique et est donc une unité mixte de recherche. Depuis juillet 2015, il est dirigé par Philippe POIGNET, assisté par Christophe PAUL et Michel RENOVELL.

En février 2016 le laboratoire comptait pas moins de quatre cent quarante-quatre employés dont cent quatorze enseignants-chercheurs, quarante-neuf chercheurs et cent soixante-dix doctorants. Parmi ces employés du LIRMM il ne faut pas oublier les personnes s'occupant de la bonne gestion du laboratoire.

1.2.2 Activité

Les principales activités du LIRMM sont la parution internationale de publications, la création de matériels et logiciels mais aussi la création d'entreprises ou de partenariats industriels.

En plus des activités précédentes, les chercheurs du LIRMM déposent également de nombreux brevets et autres logiciels à l'Agence de Protection des Programmes. Enfin, les membres du laboratoire ont également l'occasion de travailler sur des projets de l'Agence Nationale de Recherche.

1.2.3 Organisation interne

Comme il a été vu un peu plus tôt, le LIRMM est composé de trois départements. Chacun de ces départements est divisé en équipe de recherche de la façon suivante.

- Département Informatique : traite des thèmes allant des mathématiques au génie logiciel en passant par la bioinformatique.
 - Équipe ADVANSE (ADVanced Analytics for data SciencE)
 - Équipe ALGCO (Algorithmes, Graphes et Combinatoire)
 - Équipe COCONUT (Agents, Apprentissage, Contraintes)
 - Équipe DALI (Digits, architectures et logiciels informatiques)
 - Équipe ECO (Exact Computing)
 - Équipe ESCAPE (Systèmes complexes, automates et pavages)
 - Équipe FADO (Fuzziness, Alignments, Data and Ontologies)
 - Équipe GraphiK (Graphs for Inferences on Knowledge)
 - Équipe ICAR (Image et Interaction)
 - Équipe MAB (Méthodes et algorithmes pour la bio informatique)
 - Équipe MAORE (Méthodes Algorithmes pour l'Ordonnancement et les Réseaux)
 - Équipe MAREL (Models And Reuse Engineering, Languages)
 - Équipe SMILE (Système Multi-agent, Interaction, Langage, Evolution)
 - Équipe TEXTE (Exploration et exploitation de données textuelles)
 - Équipe ZENITH (Gestion de données scientifiques)
- Département Robotique : effectue des recherches sur des thèmes allant de la conception aux tests des systèmes intégrés et microsystèmes.
 - Équipe CAMIN (Control of Artificial Movement and Intuitive Neuroprosthesis)
 - Équipe DEXTER (Conception et commande de robots pour la manipulation)
 - Équipe EXPLORE (Robotique mobile pour l'exploration de l'environnement)
 - Équipe ICAR (Image et Interaction)
 - Équipe IDH (Interactive Digital Humans)

- Département Microélectronique : étudie divers thèmes tels que la navigation, le traitement des images ou bien la supervision de systèmes dynamiques complexes.
- Équipe ADAC (ADaptive Computing)
- Équipe SmartIES (Smart Integrated Electronic Systems)
- Équipe TEST (Test and dEpendability of microelectronic integrated SysTems)

Mon stage se déroulant au sein de l'équipe SMILE du département Informatique, nous allons voir plus en détail le fonctionnement et le travail de celle-ci.

1.3 Équipe SMILE : Système Multi-agent, Interaction, Langage, Évolution

1.3.1 Présentation

Un système multi-agents est un système dans lequel des agents autonomes interagissent et coordonnent leurs actions dans un but précis. Selon le système à modéliser, les agents sont soit artificiels soit humains mais ce qui est important à comprendre c'est que les agents -qu'ils soient humains ou non- sont inhérents au système et sont des entités autonomes. Bien que l'équipe SMILE étudie les systèmes multi-agents et leurs concepts, les chercheurs étudient également les applications que ces systèmes peuvent avoir. Ces dernières peuvent être catégorisées dans plusieurs domaines comme il est précisé dans la partie suivante.

1.3.2 Domaines de recherche

Tous les domaines suivants concernent ou se basent sur le paradigme multi-agents.

Modèles organisationnels pour les systèmes multi-agents

Les chercheurs développent de nouveaux modèles organisationnels relatifs aux systèmes multi-agents. Parmi les modèles étudiés, on peut citer AGR ou MASQ.

Simulation multi-agents

Un des enjeux de la simulation multi-agents -simulation de plus en plus utilisée de nos jours- est d'améliorer la façon dont les ressources de calcul sont utilisées afin de pouvoir simuler des populations toujours plus grandes dans des environnements virtuels plus grands eux aussi. On appelle ceci le calcul haute performance.

Multi-agents et Web Service

Ici on étudie le web comme un système composé d'une multitude d'agents et on en déduit des propriétés pour des futures architectures ou applications.

Multi-agents et Jeux (sérieux)

Ce thème implique l'application du paradigme multi-agents à l'implémentation de jeux sérieux. Le travail de ce domaine se compose non seulement de recherches mais également du développement de telles applications utilisées dans le domaine de la santé.

Intégration sémantique de données biomédicales et bio-ontologiques

Création d'index sémantique, et d'autres outils se basant sur des connaissances médicales et ontologiques. La finalité de ces travaux est de permettre de nouvelles découvertes scientifiques grâce à une nouvelle organisation des données.

Maintenant que nous avons vu quels étaient les domaines de travail de l'équipe SMILE, il est temps de s'intéresser d'un peu plus près aux systèmes multi-agents.

Chapitre 2

Présentation de la mission

2.1 Enjeu

Les chercheurs de l'équipe SMILE développent depuis 1997 une API leur permettant de modéliser des systèmes multi-agents. MaDKit est réalisée en Java et se veut adaptable à n'importe quel projet. Afin qu'elle soit la plus facile à prendre en main possible il est capital que MaDKit soit suffisamment documentée. C'est cette idée qui va motiver la réalisation de ce stage.

À ce jour, MaDKit est documentée de diverses façons : son code est accessible sur GitHub, sa Javadoc ainsi que des tutoriels et autres démonstrations sont également disponibles sur son site web (<http://www.madkit.net/madkit/>). De plus, divers rapports et articles la concernant sont mis à disposition. Cependant si la Javadoc de MaDKit est assez complète, nous sommes loin d'avoir chacune de ses caractéristiques expliquées et illustrées. De ce fait enrichir cette documentation est toujours nécessaire.

L'intérêt d'un tutoriel est de pouvoir démontrer par des exemples de différents niveaux de difficulté une notion ou une fonctionnalité propre à MaDKit. Les tutoriels réalisés dans le cadre du stage devront avoir un niveau de difficulté croissant et être accessible au plus grand nombre. C'est pour répondre à cet objectif qu'ils seront en anglais et aussi documenté que possible sans que nous tombions dans des explications redondantes.

2.2 Présentation des outils et techniques employés

Ce stage m'a permis d'aborder des notions que je n'avais encore jamais étudiées. Nous verrons ces dernières dans cette partie.

2.2.1 Introduction aux systèmes multi-agents

//TODO

2.2.2 MaDKit, une API pour modéliser des systèmes multi-agents

Comme il a été dit plus tôt, MaDKit est une librairie en Java permettant de modéliser et de simuler des systèmes multi-agents. Contrairement à d'autres API du genre, MaDKit est basée sur une approche organisationnelle : l'AGR (Agent / Group / Role). La notion capitale de MaDKit est qu'elle ne propose pas un modèle organisationnel spécifique et se veut le plus hétérogène possible.

Existant depuis plus de vingt ans, MaDKit possède plusieurs versions. Depuis 2010 MaDKit en est à sa version 5. Nous pouvons remarquer un changement majeur entre cette version et les

versions 1 à 4. Cette restructuration -plus que simple changement- est dû entre autre au fait que MaDKit v.4 fonctionnait plus comme un plug-in que comme une librairie ; ceci a été réglé avec la version 5. De plus, certaines parties de code constituant le noyau de MaDKit v.4 étaient trop vieilles et la librairie dépendait elle-même de plusieurs autres librairies. Cependant, la version 4 est toujours utilisée mais n'est plus maintenue.

Au total, on peut estimer au minimum le nombre de téléchargements de MaDKit entre 2011 et 2015 à un peu plus de 10 000. Ces téléchargements concernent aussi bien la version 4 que la version 5. Il reste néanmoins difficile d'avoir une idée plus précise du nombre d'utilisateurs régulier de MaDKit.

Plus d'informations concernant MaDKit sont disponibles dans (MICHEL, 2000), (HANACHI et SIBERTIN-BLANC, 2000), (MICHEL, 2010) ou encore (MICHEL, 2015). Des informations supplémentaires peuvent également être consultées sur le site de MaDKit <http://www.madkit.net/madkit/>.

2.2.3 Outils de travail employés

Ce stage a également été l'occasion d'approfondir ma connaissance voir de découvrir de nouveaux outils. L'apprentissage de ces derniers s'est déroulée en deux parties : d'abord leur découverte ou rappel en début de stage mais aussi un apprentissage sur le tas qui a parfois donné lieu à des situations assez coquaces comme il est évoqué dans la partie de ce rapport traitant des difficultés rencontrées dans ce stage.

Git

J'étais déjà familière avec Git avant ce stage. Cependant je suis encore loin de maîtriser toutes les possibilités qu'offrent cet outil. Cette expérience m'a permis d'en voir une utilisation plus professionnelle et plus complète.

Ant

Si j'avais déjà utilisé Maven par le passé, ce stage a marqué ma première rencontre avec Ant. Comme j'avais déjà réalisé des makefiles pour mes projets en C et C++ et j'avais utilisé l'XML par le passé, l'apprentissage de cet outil ne m'a pas posé plus de problème que cela. Par ailleurs un exemple de fichier de configuration est disponible sur le dépôt git du projet ainsi que sur le site web. On peut considérer que mon utilisation de cet outil s'est limitée à l'ajout de cibles couvrant les nouvelles parties du projet créées par mes soins.

Java Web Start

Avant ce stage, je n'étais absolument pas au fait de cet outil, j'ai donc découvert ce dernier pendant mon séjour au LIRMM. Java Web Start est particulièrement utile et intéressant dans le cadre des tutoriels car il permet de lancer un exécutable Java à partir d'une page Web. Les utilisateurs n'ont donc qu'à appuyer sur un bouton pour voir l'exécution du code qu'ils ont sous les yeux.

2.3 Organisation

Au cours de ces deux mois au LIRMM une certaine routine de travail a pris place. J'ai essayé de suivre cette routine du mieux que j'ai pu. Après une période de découverte de la programmation orientée agents, de la plateforme MaDKit ainsi que des différents outils, j'ai pu me lancer dans la réalisation des tutoriels.

2.3.1 Déroulement de la phase de prise en main

Cette phase a duré un peu plus d'une semaine. Personnellement je trouve que cette période a été la phase la plus dense de mon stage car il m'a fallu voir (revoir parfois) plusieurs choses différentes dans un laps de temps le plus court possible pour commencer à vraiment travailler sur ma mission sans perdre trop de temps.

J'ai donc pris un peu plus d'une semaine pour me familiariser avec la programmation multi-agents. Dans cet objectif, les publications de (MICHEL, 2000), (HANACHI et SIBERTIN-BLANC, 2000) et (MICHEL, 2010) m'ont été particulièrement utiles et instructives.

De la même manière, je me suis intéressée aux articles de (OLIVIER GUTKNECHT et MICHEL, 2000) et (PRADEILLES et HISLER, 2010) afin d'en savoir plus sur MaDKit. Je me suis également servie des renseignements disponibles en ligne sur le site de MaDKit (GROUP, 2018) que ce soit la documentation, la Javadoc ou divers tutoriels et démonstrations disponibles. Concernant ces deux derniers points, j'ai pris un soin particulier à comprendre à la fois la façon dont ils étaient construits (ce qu'ils voulaient démontrer et la granularité de difficulté qui les différenciaient) mais ils m'ont également été utiles pour apprendre, moi aussi, à me servir de MaDKit. Ne pas avoir été familière avec l'API ou même avec la programmation multi-agents m'a permis d'avoir un regard neuf par rapport à la documentation déjà existante et de repérer les points où elle n'était pas très suffisamment claire pour un néophyte. Ceci nous a permis, mon maître de stage et moi d'apporter plus de précision dans la documentation existante mais surtout de repérer quelles notions requieraient un tutoriel afin d'être mieux expliquées. De manière plus générale, le but de cette phase préliminaire était d'obtenir une vision globale tout de même globale afin de me donner les notions nécessaires au bon déroulement de mon stage tout en étant suffisamment technique pour s'intéresser à certains détails de l'API.

Mon travail consistant à enrichir la documentation et à faciliter la prise en main d'une API permettant de modéliser les systèmes multi-agents, il s'est avéré judicieux de comparer les informations mises à disposition pour MaDKit avec celles d'autres API ayant le même objectif. Sur les conseils de mon maître de stage je me suis donc renseignée sur les API : Jade, Jason, JaCaMo et SARL. Leur description et comparaison est disponible en annexe 4.3.

C'est également pendant cette période que j'ai étudié les outils de travail cités un peu plus tôt dans ce rapport.

2.3.2 Déroulement de la réalisation des tutoriels

Dans cette partie nous pouvons distinguer deux niveaux d'organisation distincts : celui de l'agencement entre les tutoriels et celui propre à la façon de travailler qui a été commune aux différents tutoriels. Commençons par l'organisation la plus générale avant d'entrer dans le détail.

Coordination des tutoriels

Tout au long de mon stage j'ai pu réaliser trois tutoriels : sur les codes de retour de MaDKit, ses propriétés et enfin sur ses options. Entre le moment où un thème de tutoriel se dessine et la fin de la réalisation de celui-ci, il est arrivé que le sujet principal du tutoriel change légèrement ou que sa structure soit grandement modifiée. J'ai pu me rendre compte de ceci pour chacun des travaux que j'ai réalisés.

Après avoir étudié différentes API de modélisation de systèmes multi-agents en plus de MaDKit, je me suis rendu compte que les autres API, contrairement à la notre, présentaient assez souvent la façon dont les exceptions sont gérées. Mon idée de base était donc de faire un tutoriel sur la gestion des exceptions dans MaDKit. Au fur et à mesure de l'avancement du tutoriel, il s'est avéré qu'étant en faite en Java la plupart des exceptions que l'on rencontre dans MaDKit était identique aux exceptions Java ce qui n'est pas des plus pertinents à soulever. En revanche il existe une catégorie d'exceptions qui elles, sont propres à MaDKit et qui sont mises en relief par des codes-retour. Ainsi ce tutoriel, s'est beaucoup plus étendu sur la présentation des différents codes retour de MaDKit que sur les exceptions en elles-même.

Si j'ai trouvé l'idée du premier tutoriel en regardant la documentation existante pour d'autres API, le sujet du second m'est apparu en me servant de la documentation de MaDKit. La notion de propriétés et d'options de MaDKit me semblait faiblement documentée et j'ai voulu en faire un tutoriel sans -à ce moment- saisir toutes les nuances qu'offraient ce thème. Un tutoriel expliquant cela m'a donc semblé évidemment. Ce n'est qu'en commençant à travailler que la nécessité de séparer les propriétés et les notions de MaDKit s'est présentée. De ce fait les deux derniers tutoriels ont été réalisés conjointement.

Flux de travail

Tout le code de MaDKit ou des différents tutoriels et démonstrations étant libre, il est disponible sur GitHub : <https://github.com/fmichel/MaDKit-tutorials>. Cela a donc été l'outil de travail privilégié pour la réalisation des tutoriels. Cependant, son utilisation nécessite tout de même une certaine organisation :

- **Fork du projet originel** (<https://github.com/amapai/MaDKit-tutorials>).
- Travail sur un tutoriel dans une **branche** à part du dépôt local.
- **Commits** au besoin.
- **Push** sur le dépôt distant quand une fonctionnalité est complète.
- Ouverture d'une **pull request** une fois le tutoriel fini.

Cela va sans dire que dès la pull request effective, un merge du projet originel au projet forké est réalisé. De même le fork du projet ne se fait qu'une seule fois. De plus, à partir du moment où la branche d'un nouveau tutoriel est créé et la validation d'une pull-request, plusieurs réunions avec mon maître de stage ont eu lieu, selon le besoin.

Chapitre 3

Travail effectué

Dans cette partie je présenterai les trois tutoriels que j'ai réalisé lors de mon stage.

3.1 Tutoriel sur les codes de retour dans MaDKit

// Partie à finir

3.1.1 But

3.1.2 Structure

Ce tutoriel suit l'architecture suivante :

- Erreurs propres à Java
 - CrashingAgent :
- Erreurs propres à MaDKit
 - CrashingAgent :
 - LaunchingNewCrashingAgent :
 - AgentLaunchingNewCrashingAgent :
- Introduction aux codes de retour
 - SuccessfulAgent :
- Codes de retour concernant le lancement des agents
 - NotYetLaunched :
 - AlreadyLaunched :
 - AgentCrash :
 - TimeOut :
 - AlreadyKilled :
- Codes de retour concernant le CGR
 - NotCommunity :
 - AlreadyGroup :
 - NotGroup :
 - NotInGroup :
 - NotRole :
 - RoleAlreadyHandled :

- RoleNotHandled :
- Codes de retour concernant la communication entre agents
 - CantReply :
 - NoRecipientFound :
 - InvalidAgentAddress :
- Utilisation concrète des codes de retour
 - SendingMessageWithReturnCode : on reprend différent code de retour déjà étudié auparavant et on montre comment gérer différents cas de figures.

3.2 Tutoriel sur les propriétés dans MaDKit

3.2.1 But

3.2.2 Structure

3.3 Tutoriel sur les options de MaDKit

3.3.1 But

3.3.2 Structure

Chapitre 4

Bilans et perspectives

4.1 Présentation des résultats

Les tutoriels sont désormais en ligne et disponible à cette adresse : <http://www.madkit.net/madkit/tutorials/>.

//TODO

4.2 Retour sur les difficultés rencontrées

Ce stage m'a également permis de faire face à certains défis.

Le premier a été d'avoir choisi un stage dans un domaine de l'informatique que je n'avais pas encore étudié : la programmation multi-agent. Cette difficulté a largement été compensée avec le fait que la mission de mon stage est accessible. Ainsi j'ai donc pu pleinement profiter de cette occasion pour en apprendre plus sur cette notion avec un peu d'avance sur mes études. Par ailleurs, le fait d'étudier les systèmes multi-agents était clairement une des motivations qui m'ont poussée à choisir ce stage plutôt qu'un autre. De plus, découvrir les systèmes multi-agents en même temps que l'élaboration des tutoriels s'est avéré assez utile car cela m'a permis de conserver un regard neuf sur mon travail et m'a aidée dans l'élaboration des tutoriels.

Ce séjour au LIRMM m'a également permis d'étudier des outils que je n'avais encore jamais utilisés. Cela a mené à une période d'apprentissage où j'ai appris beaucoup de choses dans un temps qui se devait être le plus court possible. La découverte de ces outils couplée à celle de la programmation multi-agents m'a demandé d'assimiler beaucoup de choses à la fois mais avec l'usage, j'ai pu me familiariser avec ces technologies sans trop de problème.

Paradoxalement la plus grande difficulté que j'ai pu rencontrer dans mon stage n'est pas venue d'un outil avec lequel je n'avais jamais travaillé mais de GitHub. J'ai ainsi pu découvrir de façon un peu brutale la commande **git reset --hard**. Ceci m'a fait perdre un peu de temps dans l'élaboration des deux derniers tutoriels mais surtout j'ai pu me pencher plus sérieusement sur git et étudier une utilisation plus professionnelle de cet outil.

4.3 Conclusion et perspectives

//AFINIR

Ce stage a été enrichissant et j'ai beaucoup appris que ce soit techniquement ou humainement.

J'ai pu découvrir de nouvelles façons de travailler, mais aussi enrichir ma connaissance de certains outils

Mon travail est d'ores et déjà accessible et peut être consulté par n'importe quel utilisateur ou personne s'intéressant à MaDKit. De cette façon il est utile à toute une communauté et contribue à enrichir la documentation de cette librairie. Néanmoins ce travail ne fait que s'inscrire dans une longue lignée de tutoriels - existants ou futurs - qui permettront une meilleure accessibilité ainsi qu'une prise en main plus facile de MaDKit.

Bibliographie

- GROUP, MaDKit Development. *MaDKit*. 2018. URL : <http://www.madkit.net/madkit/>.
- HANACHI, C. et C. SIBERTIN-BLANC. *Introduction aux systèmes multi-agents*. 2000.
- MICHEL, Fabien. *Approches environnement-centrées pour la simulation de systèmes multi-agents*. 2015. URL : http://www.lirmm.fr/~fmichel/publi/pdfs/fmichel_HDR.pdf.
- « Programmation situationnelle : programmation visuelle de comportements agents pour non informaticiens ». In : *Systèmes Multi-agents, Défis Sociétaux - JFSMA 10 - Dix-huitièmes journées francophones sur les systèmes multi-agents* (oct. 2010).
- « Une approche méthodologique pour la conception et l'analyse de simulateur multi-agents ». In : *Cinquièmes rencontres des Jeunes Chercheurs en Intelligence Artificielle, RJCIA'00, Lyon, France* (sept. 2000).
- OLIVIER GUTKNECHT, Jacques Ferber et Fabien MICHEL. « MaDKit : une expérience d'architecture de plateforme multi-agents générique ». In : *Systèmes multi-agents : Méthodologie, technologie et expériences - JFIADSMA 00 - huitième journées francophones d'Intelligence Artificielle et systèmes multi-agents, Saint-Jean-la-Vêtre, Loire, France* (oct. 2000).
- PRADEILLES, Vincent et Gaëlle HISLER. *Rédaction de tutoriels expliquant le fonctionnement d'un système multi-agents*. 2010.

Glossaire

Systèmes multi-agents

API (Application Programming Interface) : interface de programmation.

MaDKit (Multi-agents Development Kit) :

Foundation for Intelligent Physical Agents :

AgentSpeak :

plung-in

Table des figures

Annexe A - Comparaison de différents outils de modélisation de système multi-agents

Le but de ce document est d'analyser des moyens d'implémenter des systèmes multi-agents existants dans le but de voir les différences qui peuvent exister entre la façon dont est présentée MaDKit.

Jade

JADE tire son nom de Java Agent DEvelopment Framework. Comme son nom l'indique, il s'agit d'un logiciel intermédiaire qui permet de développer des applications multi-agents respectant les spécifications de FIPA. Le but de cet outil est de mettre à disposition des modèles d'agents et de services afin de simplifier le développement de tels systèmes.

La documentation de Jade est assez complète et peut compter sur des guides ainsi que sur une grande quantité de tutoriels explicitant diverses notions. Le seul bémol que j'ai trouvé à leur présentation est le fait que certains des tutoriels sont en PDF. Ceci rend leur compréhension un peu moins intuitive car on ne peut pas voir concrètement ce que fait le code utilisé. Parmi les guides et tutoriels existant on peut citer :

- le guide du programmeur Jade
- le guide de l'administrateur Jade
- tutoriel sur la configuration de Jade
- tutoriel sur la sécurité au sein de Jade
- tutoriel sur les protocoles de transport de message avec Jade
- tutoriel sur l'utilisation de Jade avec des appareils mobiles
- ...

Jade met également à disposition divers outils tels que Jade RMA, Jade Dummy Agent, Jade Sniffer... Pour plus d'informations : <http://jade.tilab.com/>.

Jason

Jason est un plug-in sous Eclipse permettant d'interpréter une version "améliorée" d'AgentSpeak. AgentSpeak étant un langage de programmation orienté agent. Concernant leur documentation, ils proposent des exemples et de démonstrations qui sont assez similaires aux démonstrations et tutoriels de MaDKit. Parmi leur exemple d'utilisation, on peut compter sur un Gold Miners,

ou un robot domestique. Les démonstrations disponibles couvrent les notions de communication entre agents, leur création, la gestion d'erreurs (...).

En revanche, concernant la documentation à proprement parlé - et non plus des présentations de cas d'utilisation - la Javadoc ainsi qu'un vieux manuel sont accessibles en ligne. Un livre est vendu.

Pour plus d'informations : <http://jason.sourceforge.net/wp/>.

JaCaMo

JaCaMo est un framework fusionnant trois API existantes : **J**Ason pour les agents autonomes, **C**Arthago pour l'environnement et **M**Oise pour l'organisation entre les différents agents.

Ils disposent de quelques tutoriels sur comment télécharger, configurer ou commencer un projet avec JaCaMo. Concernant les démonstrations, on retrouve un gold miners et un hello world. Pour plus d'informations : <http://jacamo.sourceforge.net/>.

SARL

SARL est un langage de programmation orienté agent. Sa syntaxe est intuitive et se base sur les Capacités et les Talents des agents ; le but étant de se concentrer plus sur notre système multi-agent que sur la façon de l'implémenter.

Globalement, la documentation de SARL est assez conséquente mais on regrette toute fois l'absence de démonstrations concrètes. Pour plus d'informations : <http://www.sarl.io/index.html>.