|  |
| --- |
|  |
| Système Multi-Agents I4.0 |
| Rapport – Travail de Bachelor  2016-2017 |
| Professeur : Hubert Droz |
| **Vincent Chaperon – INF3dlm-b** |
| **27/01/2017** |
|  |

**Abstract**

Ce rapport présente le développement d’une application implémentant un « Système Multi-Agents » permettant d’explorer l’applicabilité de ce concept à l’organisation des unités de production dans une usine. Ce projet vise à soutenir les industriels de l’arc jurassien à faire face à la concurrence mondiale dans le domaine du « Smart and Micro Manufactoring » et de l’Industrie 4.0.

En informatique, un système multi-agents est un système composé d’agents autonomes interagissant entre eux dans un certain environnement.

This report presents the development of an application implementing a "Multi-Agents System" to explore the applicability of this concept to the organization of production units in a plant. This project aims to support industrialists in the Jura region to compete globally in the field of "Smart and Micro Manufacturing" and Industry 4.0.

In computer science, a multi-agent system is a system composed of autonomous agents interacting in a certain environment.

**Table des matières**

[1 Introduction 2](#_Toc486505941)

[2 But du projet 3](#_Toc486505942)

[3 Réalisation du projet 3](#_Toc486505943)

[3.1 Méthodologie de développement 3](#_Toc486505944)

[3.2 Structure du programme 3](#_Toc486505945)

[3.2.1 Schéma général des classes 3](#_Toc486505946)

[4 Difficultés rencontrées 3](#_Toc486505947)

[5 Guide développeur 4](#_Toc486505948)

[6 Guide utilisateur 4](#_Toc486505949)

[7 Conclusion 4](#_Toc486505950)

[8 Bibliographie 5](#_Toc486505951)

[9 Annexes 7](#_Toc486505952)

# Introduction

Dans le cadre du travail de Bachelor de la troisième année de la haute école ARC ingénierie, il m’était demandé d’explorer l’applicabilité du concept de « Système multi-agents » pour organiser des unités de productions dans une usine. Ce travail est la suite du travail d’automne réalisé précédemment qui visait à simuler en programmation une chaîne de production simplifiée implémentant différents agents. Il s’agit d’implémenter un « Système multi-agents » se connectant et utilisant de vraies machines.

Ce projet vise à soutenir les industriels de l’arc jurassien à faire face à la concurrence mondiale dans le domaine du « Smart and Micro Manufactoring » et de l’Industrie 4.0.

Ce rapport présentera les grandes lignes du développement de ce programme, notamment les parties essentielles au fonctionnement des agents, la connexion aux diverses machines selon certains protocoles ainsi que les différentes difficultés rencontrées lors de l’élaboration de cet implémentation. Il sera également composé d’un guide développeur ainsi que d’un guide utilisateur permettant de reprendre aisément le projet afin de le faire évoluer.

Le langage de programmation choisi est le Java pour les différents agents, celui-ci étant lié au framework JADE permettant le développement d’un « Système Multi-Agents » et l’IDE choisi est Eclipse. Le langage de programmation du robot est le Deko et la plateforme de développement utilisée est Synapxis. Enfin le langage de programmation de la presse est le C/C++ et l’IDE est Visual Studio.

# But du projet

L’objectif de ce projet est d’explorer l’applicabilité du concept de « Système Multi-Agents » pour organiser des unités de productions dans une usine. Il s’agit de simuler en programmation une chaîne de production simplifiée en implémentant divers agents. Ces différents agents pouvant s’adapter aux modifications d’un environnement et pouvant interagir entre eux selon des schémas relationnels spécifiques, ils permettent de gérer au mieux les machines de production.

Ce projet vise à soutenir les industriels de l’arc jurassien face à la concurrence mondiale dans le domaine du « Smart and Micro Manufactoring » et de l’Industrie 4.0.

Afin de vérifié l’applicabilité du concept de « Système Multi-Agents » dans ce domaine, la simulation en programmation s’est effectuée sur la base de deux scenarii dans le domaine de la Plasturgie.

Le premier scenarii consistait à réaliser l’automatisation de la fabrication additive au moyen d’une ligne de fabrication interconnectée pour la personnalisation de pièces de série. Cette chaîne de production comportait trois machines, une presse à injecter, une machine de fabrication additive et un robot à sept axes établissant l’interconnexion entre ces deux précédentes machines.

Le second scenarii consistait à réaliser la fabrication et le montage de boutons d’interrupteur d’éclairage. Les différentes étapes de traitement de ces opérations se déroulaient de manière séparée dans le temps et l’espace. Cette chaîne comportait quatre différents éléments de production, un poste permettant la personnalisation des pièces « en direct », un autre permettant la personnalisation des emballages, un ordinateur pilote rassemblant les données sur le processus et archivant celles-ci dans un « cloud ». Enfin chaque pièce étant porteuse de données, celles-ci communiquent avec les différents postes et déterminent elles-mêmes leur passage dans la chaîne de processus.

# Réalisation du projet

Ce chapitre décrit les différentes étapes de la réalisation du projet. Il décrit l’analyse du projet effectuée, les solutions retenues, la méthodologie de développement suivie ainsi que la structure finale du projet.

## Méthodologie de développement

## Structure du programme

Ce chapitre décrit la structure de mon application.

### Schéma général des classes

# Difficultés rencontrées

Ce chapitre décrit les difficultés rencontrées lors de la réalisation de ce projet.

# Guide développeur

# Guide utilisateur

# Conclusion

# Bibliographie

Documents pdf Synapxis\_xxx

**HE-Arc/IHC/UR-LPR,** Le Locle 2013

Documentation officielle concernant le logiciel Synapxis

Documents se trouvant dans le dossier « doc » du rendu.

Notamment :

* Synapxis\_000\_MainGeneral.pdf

Documentation générale du logiciel Synapxis

* Synapxis\_000\_MainMacro.pdf

Documentation concernant le langage de programmation intégré au logiciel Synapxis

* Synapxis\_DocTraj.pdf

Documentation concernant les cycles d’usinage et l’édition des trajectoires

<http://jade.tilab.com/>

**Telecom Italia Lab**

Documentation officielle de Jade

Notamment :

* <http://jade.tilab.com/doc/programmersguide.pdf>

Guide développeur de Jade

* <http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial-for-beginners.pdf>

Tutoriel de développement d’agents Jade

* <http://jade.tilab.com/doc/api/index.html>

Documentation de l’API Jade

* <http://jade.tilab.com/papers/JADETutorialIEEE/JADETutorial_FIPA.pdf>

Documentation sur la communication entre les agents

* <http://jade.tilab.com/documentation/tutorials-guides/configuring-the-jade-http-mtp/>

Configuration du MTP pour la connexion entre plusieurs plateformes Jade

* <http://jade.tilab.com/doc/tutorials/JADEAdmin/JadePlatformTutorial.html>

Tutoriel sur l’implémentation de plusieurs plateformes Jade

<https://www.developpez.com/>

**Club des développeurs et IT pro**

Plateforme d’échange d’informations et de tutoriels informatiques

Notamment :

* <http://djug.developpez.com/java/jade/behaviours/>

Explications et exemples sur les comportements des agents

* <http://djug.developpez.com/java/jade/communication/>

Explications et exemples sur la communication des agents

<https://infosys.beckhoff.com/>

**Beckhoff Information System**

Documentation officielle de Beckhoff

Notamment :

* <https://infosys.beckhoff.com/english.php?content=../content/1033/tcsample_java/html/tcsample_java_intro.htm>

Exemples d’implémentations et d’utilisations des bibliothèques pour la connexion ADS en Java

* <https://infosys.beckhoff.com/english.php?content=../content/1033/tcsample_vc/html/tcadsdll_api_cpp_sample18.htm>

Exemple d’une implémentation d’un client en C++ permettant la connexion en ADS

* <https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_c/36028797393240971.html>

Liste des codes d’erreur retournés lors de la connexion ADS

* <https://infosys.beckhoff.com/italiano.php?content=../content/1040/TcAdsDll2/html/TcAdsDll_FuncAdsSyncReadWriteReq.htm>

Liste des fonctions de la bibliothèque permettant la connexion en ADS

# Annexes