



HAL
open science

Human-inspired algorithms for designing new control system in the context of factory of the future

Tsegay Tesfay Mezgebe

► **To cite this version:**

Tsegay Tesfay Mezgebe. Human-inspired algorithms for designing new control system in the context of factory of the future. Automatic. Université de Lorraine, 2020. English. NNT : 2020LORR0006 . tel-02510677

HAL Id: tel-02510677

<https://hal.univ-lorraine.fr/tel-02510677>

Submitted on 18 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



ALGORITHMES INSPIRÉS DES SOCIALES POUR CONCEVOIR DES NOUVEAUX SYSTÈMES DE PILOTAGE DANS LE CONTEXTE DE L'USINE DU FUTUR

THÈSE DE DOCTORAT

Soumise par Tsegay Tesfay MEZGEBE

Octobre, 2019

École doctorale IAEM Lorraine
Centre de recherche en Automatique de Nancy (CRAM)

Thèse pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ DE LORRAINE, Nancy, FRANCE
(Spécialité Automatique)

Composition du jury

Rapporteurs

- Pr. Jonathan GAUDREAU, Université Laval
- Pr. Damien TRENTSAUX, Université Polytechnique Hauts de France

Examineurs

- Dr. Olivier CARDIN, Université de Nantes
- Pr. Lhassane IDOUMGAR, Université de Haute-Alsace
- Dr. Fouzia OUNAR McF, Aix-Marseille Université
- Pr. Hind BRIL EL-HAOUZI, Université de Lorraine (Directrice de thèse)
- Dr. Guillaume DEMESURE, Université de Lorraine (Co-encadrant)

Invité

- Pr. Andre THOMAS, Université de Lorraine
 - Dr. Remi PANNEQUIN, (Ingénieur de recherche), Université de Lorraine
-



HUMAN-INSPIRED ALGORITHMS FOR DESIGNING NEW CONTROL SYSTEM IN THE CONTEXT OF FACTORY OF THE FUTURE

DOCTORAL THESIS

Submitted by Tsegay Tesfay MEZGEBE

October, 2019

School of IAEM Lorraine

Center of Research in Automatic of Nancy (*CRAN*)

A thesis submitted in fulfillment of the requirements for the degree of Doctor from
University of LORRAINE, Nancy, FRANCE
(Specialty in Automatic engineering)

Members of the jury

Reviewers

- Professor Jonathan GAUDREAULT, Université Laval
- Professor Damien TRENTESAUX, Université Polytechnique Hauts de France

Examiners

- Dr. Olivier CARDIN, (HDR), Université de Nantes
- Professor Lhassane IDOUMGAR, Université de Haute-Alsace
- Dr. Fouzia OUNAR, Aix-Marseille Université
- Professor Hind BRIL EL-HAOUZI, University of Lorraine, (Director of thesis)
- Dr. Guillaume DEMESURE, University of Lorraine, (Co-supervisor)

Invited

- Professor Andre THOMAS, University of Lorraine
 - Dr. Remi PANNEQUIN (Research engineer), University of Lorraine
-

DECLARATION

I declare that this dissertation is my genuine work and all sources and materials used for this work have been duly acknowledged. This dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor from the University of LORRAINE, Nancy, FRANCE. I gravely declare that this dissertation is not submitted to any other institution anywhere for the award of any academic program. Requests for permission for quotation from or reproduction of this manuscript in whole or in part may be granted by the author.



Signature

January, 2020

Date

ACKNOWLEDGEMENTS

God! I thank you for blessing me, my family, and for the strength you gave me in my day to day activities. I would like to acknowledge several people who in different ways contributed to this work. Without their contributions, the work would not be possible. First and foremost, I would like to express my highest gratitude to my thesis director Professor Hind BRIL EL-HAOUZI for her enormous encouragement as well as her time and patience to reply all my questions and clarify all my confusions. I am also beholden to thank Dr. Guillaume DEMESURE and Dr. Remi PANNEQUIN for their day to day engagement and discussions to model and test experiments. I would also like to thank Professor Andre THOMAS, my former thesis director, for giving me an opportunity to study in the University of Lorraine and Dr. Gebremeskel Kahsay GEBREMARIAM for his inspiring support while I was in Ethiopia.

My sincere gratitude goes to Professor Jonathan GAUDREULT and Professor Damien TRENTESAUX for their thorough review of my dissertation manuscript and Dr. Olivier CARDIN, Professor Lhassane IDOUMGAR, and Dr. Fouzia OUNAR for examining my thesis and defense. Ethiopian Ministry of Science and higher education and Embassy of France in Ethiopia are also part of my acknowledgement for their financial support during my stay in France. I would also like to thank my friends and colleagues for making the study years memorable and unforgettable. Last but not least, I would like to express my warmest thank to my wife, children, and parents for their unfailing support and endless love.

FORWARD

After I got my masters' degree from Mekelle University (Ethiopia) department of Industrial engineering, I started to work as lecturer and assistant professor in the same University. Parallel to this, I have been thinking to pursue my doctoral study abroad and I personally was ambitious to had societal experiences outside my country. Meanwhile, in the framework of its staff development plan, Mekelle University was motivating its academic staffs to pursue their doctoral study either within the country or else. At this moment, I heard about commencement of a sandwich doctoral program between the Government of Ethiopia and the Government of France to strengthen their economic and political relations. I was excited on this opportunity in order to satisfy my personal ambition and to enhance my capacity on research and development. Accordingly, I made competition throughout the country and finally selected to pursue my study in one of the French Universities; which finally joined the Esteemed University of Lorraine.

During my study, I not only deepen and broaden my expertise but also I have been able to dynamically improve my social relations with international students and researchers. I have been able to establish a research networks during my participation in international and national conferences, workshops, and seminars. Meanwhile, this opportunity has made me to hone additional skills such as improving the French language, playing some sports like bowling, golf, etc.

After completing my doctoral study, I am excited to go back to my country in order to significantly serve the community through projects, researches, teaching, and consultancy services.

ABSTRACT

The use of traditionally centralized control system does not able to meet the rapidly changing customer expectations, high product varieties, and shorter product life-cycles. In particular, the emergence of Cyber Physical System (CPS) which can be seen as interacting networks of physical and computational components has provided the foundation for many new factories' infrastructures and improved the quality of products and processes. This Cyber Physical System has dramatically impacted the centrally predictive control system in responding to perturbation(s) in the current dynamic market characteristics. Urgent change for example is one of the common perturbations and has significant perturbing ability to a central predictive control system. Accordingly, it is now accepted that using agent-based control system improves the reactivity to treat these perturbation(s) until they are no longer limiting factors.

In this study, the use of human-inspired interaction approach (by means of negotiation and consensus-based decision-making algorithms) is explored to design and propose a new control system. It has taken advantages from Industry 4.0 assets and encompassed heterogeneous and intelligent entities (mainly the product entities and resource entities) and discrete event systems. Each entity could have different capability (evolution, learning, etc.) and the whole physical and control system may lead emerging behaviors to dynamically adapt the perturbation(s). Hence, every intelligent entity decides when to broadcast its current state to neighbor entities and the controlling decision depends on the behavior of this state.

The negotiation and consensus-based decision-making algorithms were initially formulated and modeled by networking all the contributing and heterogeneous entities considering two real industrial shop floors. Then after, simulation and application implementation tests on the basis of full-sized academic platform called TRACILOGIS platform have been conducted to verify and validate these decision-making algorithms. This has been done with expectations that the applicability of these algorithms will be more adaptable to set best priority-based product sequencing and rescheduling than another decision-making approach called pure reactive control approach. Accordingly, the experimental results have shown that the negotiation and consensus among the decisional entities have significantly minimized the impact of perturbation(s) on a production process launched on the TRACILOGIS platform. Meanwhile, using these two decision-making algorithms has conveyed better global performance (e.g., minimized makespan) over the pure reactive control approach.

Keywords Control system, Multi-agent System, Negotiation, Consensus, Entity state, Simulation

RÉSUMÉ

Le pilotage centralisé traditionnel de système manufacturier n'est plus capable de répondre aux besoins du marché en termes d'attentes des clients, de la variété et du cycle de vie des produits. En particulier, l'émergence des systèmes Cyber-Physiques (CPS), vus comme des réseaux de composants physiques et informatiques en interaction, a impacté le pilotage central prédictif pour contrer les perturbations dans les actuelles caractéristiques dynamiques du marché. Par exemple, la modification urgente des besoins de l'entreprise est l'une des perturbations les plus communes qui impacte le planning central d'une manière significative. A l'heure actuelle, le pilotage distribué, pouvant être basé sur les systèmes multi-agent, permet d'améliorer la réactivité pour répondre à différents types de perturbations afin que celles-ci ne soient plus des facteurs contraignants le pilotage.

Les travaux de cette thèse explorent la conception de nouveaux algorithmes de pilotage en se basant sur les approches d'interactions sociales, telles que la négociation ou la théorie du consensus). Ces algorithmes intègrent les avantages des actifs liés à l'industrie du futur et englobent différentes entités intelligentes et hétérogènes, ainsi que des systèmes à événements discrets. Chaque agent peut avoir des capacités différentes, en termes d'évolution, d'apprentissage ou autre, permettant d'une manière globale d'obtenir des comportements émergents qui s'adaptent dynamiquement aux perturbations rencontrées. Ainsi, chaque agent intelligent décide quand il transmet son état actuel à son voisinage et la décision locale dépend du comportement de l'état des agents.

Afin de vérifier l'efficacité des algorithmes proposés (négociation et consensus), ceux-ci ont été modélisés et formulés en mettant en réseau toutes les entités intelligentes, le tout basé sur deux ateliers industriels. Les algorithmes ont ensuite été testés en simulation et d'une manière expérimentale sur la cellule flexible de production du plateau technique TRACILOGIS. Des comparaisons ont été faites avec l'algorithme réactif actuellement implémenté sur cette plateforme, afin de montrer l'applicabilité des algorithmes dans l'obtention de ré-ordonnancement et re-séquencement de produits par priorité. Par conséquent, les résultats expérimentaux réalisés sur TRACILOGIS ont montré que les algorithmes de négociation et de consensus ont minimisé l'impact des perturbations en obtenant de meilleures performances globales qu'auparavant.

Mots-clés Système de contrôle, Système multi-agent, Négociation, Consensus, État de l'entité, Simulation

LIST OF ACRONYMS

ACL	Agent Communication Language
ADACOR	ADaptive holonic COntrol Architecture
AGV	Automated Guided Vehicles
APICS	American Production and Inventory Control Society
CoMM	Consensus algorithm for Multi-agent Manufacturing system
CRAN	Centre de Recherche en Automatique de Nancy
CPS	Cyber Physical System
ERP	Enterprise Resource Planning
FIFO	First In First Out
FIPA	Foundation for Intelligent Physical Agent
HCA/S	Hybrid Control Architecture/System
HOLOMAS	HOLONic and Multi-Agent Systems for manufacturing
ICT	Information Communication Technology
IMS ²	Intelligent Manufacturing Systems and Services
IoT	Internet of Things
ISET	The Engineering of Eco-Technical Systems
JADE	Java Agent DEvelopment framework
KID	Knowledge, Information, Data
KPI	Key Performance Indicator
MES	Manufacturing Execution Systems
MPS	Master Production Schedule
NRC	Negotiation-based Reactive Control approach
PLC	Production Line Controller
PRC	Pure Reactive Control approach
PROSA	Product, Resource, Order, Staff Architecture
RFID	Radio Frequency Identification
SOHOMA	Service Orientation in HOLONic and Multi-Agent manufacturing
Topose	TracilOgis Platform Agent SystEm
UML	Unified Modeling Language
WIP	Work-in-Progress

LIST OF FIGURES

Fig. I-1 Cadre de l'industrie 4.0.....	1
Fig. I-2 Un processus de fabrication et son environnement.....	2
Fig. 1-1 Different decision levels within manufacturing planning and control	11
Fig. 1-2 Evolution of Manufacturing System Paradigms	11
Fig. 1-3 Screen shot example of shop floor	13
Fig. 1-4 Throughput time of a manufacturing process	14
Fig.1-5 Dynamic scheduling algorithms	15
Fig. 1-6 Structural control architectures and relationship between decision entities	16
Fig. 2-1 Some examples of Control Architectures	25
Fig. 2-2 Coordination mechanisms (a) bio-inspired mechanism (b) Physical-inspired mechanism (c) Human/social-inspired mechanism	28
Fig. 2-3 Three sub-classes of hybrid control architectures	30
Fig. 2-4 PROSA reference architecture	33
Fig. 2-5 ADACOR Holon	34
Fig. 2-6 Society of cooperative multi-agent.....	37
Fig. 3-1 Cybernetic action loop to identify and to diagnosis perturbation(s)	46
Fig. 3-2 The REcursive hybriD Control Architecture.....	47
Fig. 3-3 Flow chart of a scaled-down shop floor.....	54
Fig. 3-4 A contract-net type negotiation procedure.....	59
Fig. 3-5 The proposed Negotiation control architecture.....	60
Fig. 3-6 UML sequence diagram for the proposed multi-agent negotiation.....	63
Fig. 3-7 UML class diagram meta-model for the developed multi-agent system negotiation	64
Fig. 3-8 Flow chart of a scaled-down shop floor with its perturbation	65
Fig. 3-9 MAS based cooperative decision architecture to satisfy the rush order	67
Fig. 3-10 Algorithmic procedure to reach consensus among multi-agent	68
Fig. 3-11 Priority-based communication topology among different entities	72
Fig. 3-12 System rescheduling to satisfy the rush order	74
Fig. 3-13 Meta-model for the proposed interaction approaches.....	76
Fig. 4-1 General description of the TRACILOGIS platform	80
Fig. 4-2 Layout of the TRACILOGIS test-bed platform	82
Fig. 4-3 The platform's implementation architectures	83
Fig. 4-4 The pyramid of control architecture of the platform	84
Fig. 4-5 Intelligent components and assembled product types of each manufacturing order.	85
Fig. 4-6 Analysis of product lateness in both control approaches.....	88
Fig. 4-7 Analysis of machine utilization rate in both control approaches	88

Fig. 4-8 Makespan analysis in both control approaches	89
Fig. 4-9 Probability density function for both control approach	90
Fig. 4-10 Scatterplot of makespan versus utilization rate in both control approaches	91
Fig. 4-11 Process graph of the TRACILOGIS platform	92
Fig. 4-12 Deterministic schedule (MPS) for 12 manufacturing orders	93
Fig. 4-13 Product types of each manufacturing order	93
Fig. 4-14 The leader-Follower communication topology among the product agents.....	95
Fig. 4-15 Pseudo code for the three groups of decision entities.....	96
Fig. 4-16 Role of hybrid decision to satisfy the rush order.....	97
Fig. 4-17 Time based evolution of state in the consensus-based decision-making algorithm	98
Fig. 4-18 Time based evolution of state in the myopic decision-making algorithm	99
Fig. 4-19 State disagreement ($ x_i, x_j $) level for the two decision-making algorithms.....	100
Fig. 4-20 Batch lateness for both control approaches.....	101
Fig. 4-21 Mean completion time of all manufacturing orders	102
Fig. 4-22 Sensitivity analysis between performance indicators	103
Fig. 4-23 UML topase sequence diagram	105
Fig. 4-24 Gantt chart for real completion time and sequence of each product	105
Fig. 4-25 Precedence relationship of real production start date and production end time of each product type	106
Fig. 4-26 Gantt chart for the real completion time of each product	107
Fig. A~1 Evolutionary approach of Industry 4.0.....	134
Fig. A~2 Drivers, Requirements, and Characterization of Industry 4.0	135
Fig. A~3 Reference Architecture Model for Industry 4.0.....	136
Fig. B~1 Conceptual view of CPS.....	137
Fig. B~2 Cyber Physical Analytics Platform with Self Learning Capabilities	139
Fig. C~1 Viable System Model for system adaptability.....	141
Fig. D~1 Flow chart of RFID networks	142
Fig. D~2 Intelligent components and assembled product type	142
Fig. D~3 UML of product's main behavior	143
Fig. D~4 The UML of creating product's route	144
Fig. D~5 The Java software structure	145
Fig. D~6 FIPA standard Agent Communication Langague, ACL	145
Fig. D~7 The cooperative interactions among agents (a) Subscription: message sent as soon as the event takes place (b) Interrogation: response based on last event occurred (c) Request for action by PLC only.....	146
Fig. E~1 The RFID HP network technology.....	147
Fig. F~1 The seven lean wastes.....	149

Fig. G~1 The Multi-agent Systems	153
Fig. G~2 Agent control loop	153
Fig. H~1 MAS interaction topology	156
Fig. H~2 The FIPA Contract Net Protocol	157
Fig. H~3 Peer-to-Peer Protocol	158
Fig. H~4 Broadcasting protocol	158
Fig. I~1 A Holarchy system	160
Fig. I~2 Linkage between business drivers and control system.....	161
Fig. I~3 Agent-based architecture of holons.....	162
Fig. I~4 Head and base of holons	163
Fig. I~5 Holonic architecture of hybrid system	163
Fig. J~1 A Python pseudo code for <i>update state</i> of the product agents within the consensus algorithm	164
Fig. J~2 A controlling Python pseudo code of <i>loop request</i> for the consensus algorithm ...	165
Fig. J~3 Completion times of each product of each manufacturing order for the consensus algorithm	166
Fig. J~4 Screenshot of topase configuration and communication among agents in one of the implementation experiment.....	167
Fig. J~5 UML class diagram (a) for topase-ontology of all agents (b) for topase product agent messaging ontology	169

LIST OF TABLES

Table I-1 Deux contextes industriels.....	5
Table I-2 Logiciels et ressources utilisés pour tester les approches de contrôle proposées.....	6
Table 1-1 Manufacturing paradigms and their indicators (Source: El Haouzi, 2017)	12
Table 1-2 Sources and impact of perturbation(s) (adapted from Saadat and Owliya, 2008) .	15
Table 2-1 Comparison and characterization of different control architectures and paradigms	26
Table 2-2 The different control system views and the interaction and coordination mechanisms among decision entities	28
Table 2-3 Application domains of MAS in manufacturing systems	36
Table 4-1 Processing time of each product type P_i in each machine m	86
Table 4-2 Computed intention (in seconds) of each product type	87
Table 4-3 Processing time of each product i in each machine m	94
Table 4-4 Estimated completion time versus real completion time for each product.....	106

TABLE OF CONTENTS

DECLARATION	i
ACKNOWLEDGEMENTS	ii
FORWARD	iii
ABSTRACT	iv
RÉSUMÉ.....	v
LIST OF ACRONYMS.....	vi
LIST OF FIGURES	vii
LIST OF TABLES.....	x
TABLE OF CONTENTS	xi
INTRODUCTION GÉNÉRALE	1
I. Introduction et Motivation	1
II. Objectif de recherche.....	3
III. Méthodologie de recherche et contexte ISET	4
IV. Contribution de l'étude.....	7
V. Structure de la thèse.....	8
CHAPTER ONE.....	10
MANUFACTURING AND ITS WORKING ENVIRONMENT	10
1.1 Introduction	10
1.2 Manufacturing systems	10
1.2.1 Shop floor	13
1.2.2 Throughput time.....	13
1.2.3 Manufacturing perturbation(s)	14
1.2.4 Scheduling	15
1.3 Control Systems.....	16
1.3.1 Centralized/hierarchical control system	16
1.3.2 The heterarchical control system	18
1.3.3 Hybrid control system (HCS)	19
1.4 Multi-agent manufacturing Systems (MAS)	21
1.5 Holonic Manufacturing System (HMS).....	21
1.6 Synthesis and Discussion	22
CHAPTER TWO	24
STATE-OF-THE-ART ON CONTROL ARCHITECTURES AND INTERACTION	
MECHANISMS.....	24
2.1 Introduction	24
2.2 General framework on control Architectures and Interaction mechanisms.....	24

2.2.1 Survey on Holonic Manufacturing System (HMS) reference architectures.....	32
2.2.2 Survey on Multi-agent System (MAS) paradigm	36
2.3 Synthesis and Discussion	41
CHAPTER THREE.....	44
THE CONTRIBUTION: HUMAN-INSPIRED INTERACTION APPROACH TO DEAL WITH PERTURBATION(S) IN MULTI-AGENT MANUFACTURING SYSTEM	44
3.1 Introduction	44
3.2 Implementation architecture: a Recursive hybrID Control Architecture (REDCA)	45
3.2.1 The decision entities	47
3.2.2 Structural organization of the control architecture.....	49
3.3 Applicability of the proposed interaction approach	50
3.3.1 Industrial context-I: A Negotiation-based reactive control approach (NRC) to deal with material build-up problem	54
3.3.2 Industrial context-II: a Consensus-based decision-making algorithm (CoMM) to deal with rush order.....	65
3.4 Synthesis and Discussion	75
CHAPTER FOUR.....	79
EXPERIEMENTAL RESULTS AND DISCUSSIONS: A TRACILOGIS platform case study	79
4.1 Introduction	79
4.2 General description of the TRACILOGIS platform	80
4.2.1 Implementation architecture of the platform	83
4.3 Simulation test results and discussion from the Negotiation-based control approach ..	85
4.3.1 Simulation configuration of the TRACILOGIS platform	85
4.3.2 Analysis and manipulation of the simulated results.....	87
4.4 Simulation test results and discussion from the Consensus algorithm	91
4.4.1 Simulation configuration of the platform	91
4.3.2 Analysis and manipulation of the simulated results.....	97
4.5 Real applicability implementation of the proposed control approaches.....	104
4.6 Summary and discussion.....	107
GENERAL CONCLUSION AND FURTHER WORKS	110
Further work.....	114
REFERENCES	117
APPENDICES	134
Appendix A Industry 4.0: Evolution and Requirements	134
Appendix B Cyber Physical Systems (CPS).....	137
Appendix C Viable System Model (VSM).....	140

Appendix D TRACILOGIS test-bed platform.....	142
Appendix E Radio Frequency Identification (RFID)	147
Appendix F Lean Manufacturing	149
Appendix G Multi-agent Systems (MAS)	152
Appendix H Interaction protocols	156
Appendix I Holonic Manufacturing System (HMS)	159
Appendix J MatLab and/or Python pseudo codes, Java codes, UML diagrams, and results for the proposed algorithms	164

INTRODUCTION GÉNÉRALE

I. Introduction et Motivation

L'un des plus gros problèmes pour les industries manufacturières actuelles gérant de multiples ressources au sein de leur chaîne d'approvisionnement est le marché mondialement volatil. Le cadre de l'Industrie 4.0¹, montré dans Fig. I-1, exhorte également ces industries à faire face à la variabilité² dynamique et les progrès technologiques en matière d'information et de communication pour une gestion efficace des connaissances requises pour la gamme de produits actuellement en augmentation et pour la nécessité de mettre à jour et de reconfigurer régulièrement leur système. Cette variabilité est rendue possible, notamment, par les avancées technologiques à la fois dans l'information et dans la communication qui conduisent à l'émergence de nouveaux paradigmes basés sur des objets connectés tels que le Cyber Physical System (CPS). Voir Appendix A et Appendix B pour des descriptions détaillées d'Industrie 4.0 et Cyber Physical Systems respectivement.

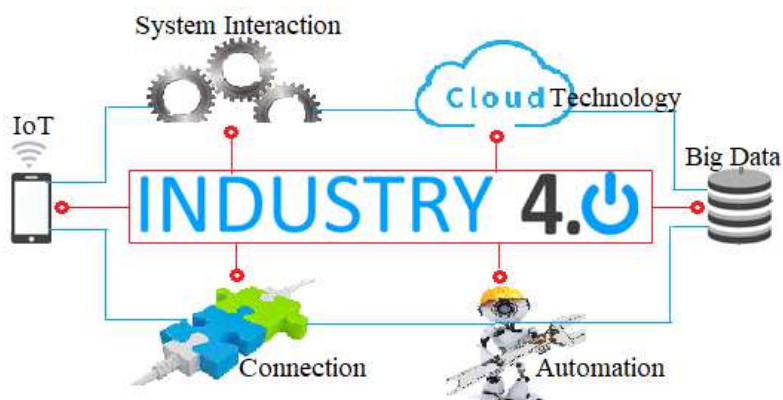


Fig. I-1 Cadre de l'industrie 4.0 (adapted from <https://www.neosoft.ca/fr/industries/industrie-4-0-iot>)

D'autre part, ces industries manufacturières sont difficiles avec la survenue fréquente d'événements inattendus appelés perturbation(s)³ dans ce rapport. Si la ou les perturbations survenues ne sont pas contrôlées à leur origine, leur impact peut être aussi important que conduire à des activités sans valeur ajoutée présentées dans Fig. I-2(a). Par exemple, lorsque le processus de fabrication d'un atelier est affecté par une défaillance de l'équipement, l'écart entre

¹ *L'industrie 4.0* (une expression des savants allemands) est également utilisée avec *l'usine du futur* (une expression des savants français)

² Variabilité telle que le contrôle des délais, les dates d'échéance, l'inventaire des travaux en cours, la variété des produits, etc. (Geissbauer et al., 2016; Koch et al., 2014)

³ Perturbation(s), utilisé de manière anonyme avec un événement inattendu, dans le système de fabrication est l'un des problèmes importants qui se rencontrent fréquemment lors de l'exécution des systèmes de fabrication (Saad et Gindy, 1998). Il représente des changements et des perturbations prévisibles et imprévisibles provenant à la fois des environnements externes et internes; voir la sous-section 1.2.3 pour le détail.

le plan de processus et la production exécutée devient plus large et, partant, la durée de fabrication étendue. Makespan est une différence de temps entre la production commencée et la production terminée pour une séquence d'emplois ou de tâches (Skorin-Kapov et Vakharia, 1993). Si un atelier (il peut s'agir d'un magasin de travaux, d'un atelier de flux ou d'un magasin ouvert) ne peut pas réduire cette durée, les commandes des clients ne seront pas expédiées à l'heure de livraison prévue, ce qui peut entraîner une augmentation des coûts et une réduction qualité. Ainsi, afin de créer un alignement de bout en bout entre la prise de décision de fabrication et les objectifs stratégiques d'une organisation, la minimisation de la durée en fournissant une planification efficace des tâches aux ressources du système de fabrication est considérablement requise. Pour réaliser cette exigence et se déployer dans le cadre de l'industrie 4.0, les usines et les instituts de recherche supérieurs doivent toujours rechercher des systèmes de contrôle qui mettent en réseau des composants physiques et informatiques; voir Fig. I-2(b).

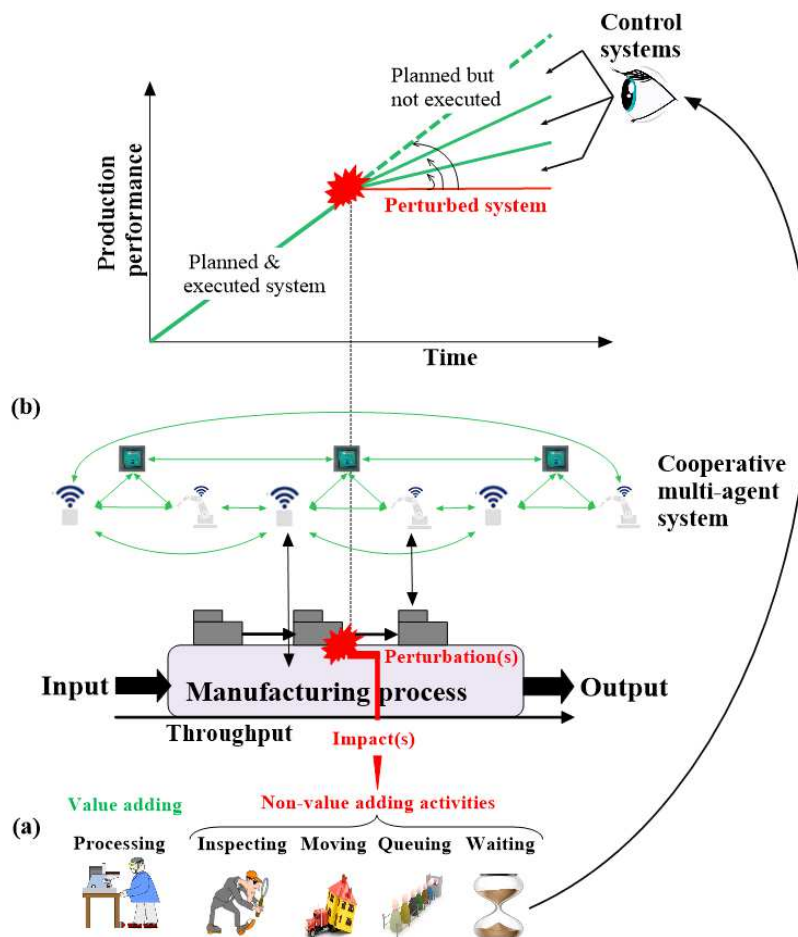


Fig. I-2 Un processus de fabrication et son environnement
(a) Impacts des perturbation(s). **(b)** Recherche de système de contrôle

Cela confirme une réactivité et une stabilité intelligentes au moment des perturbations, une adaptabilité rapide au moment des changements de production, et donc augmente la complexité et le dynamisme d'un système et/ou à l'extérieur du système (Thierry et al., 2008). Dans le cadre de cette recherche, une approche de contrôle prédictif centralisé, un mode de contrôle réactif distribué et un système de contrôle hybride ont été conçus et mis en œuvre en tant que systèmes de contrôle les plus couramment acceptés au cours des décennies précédentes. La section 1.3 explique les différentes vues de ces systèmes de contrôle.

II. Objectif de recherche

Il est clair que l'utilisation d'une approche de commande prédictive centralisée ou d'une approche de commande réactive distribuée dans un système de fabrication donné a montré ses propres avantages et limites. Mais, la coexistence des deux modes de contrôle comme nouvel écosystème amplifie leur contribution en prenant la spécialité autonome des entités décisionnelles dans le mode de contrôle réactif distribué⁴ et mise à jour et interface du rôle des entités dans l'approche de contrôle prédictif centralisé.

L'objectif de cette étude doctorale était d'explorer une nouvelle utilisation de l'approche d'interaction inspirée par l'homme (au moyen de la négociation et d'algorithmes décisionnels fondés sur le consensus) combinée à la simulation⁵ comme support à la décision dans la conception et la gestion d'un nouveau système de contrôle. Il a pris en compte les avantages des actifs de l'industrie 4.0 et la coopération multidimensionnelle entre différentes cybernétiques (produits intelligents, ressources, etc.), qualifiée atouts⁶, et différents niveaux décisionnels. Pendant ce temps, tous entités décisionnelles⁷ et les niveaux de décision au sein du système de fabrication sont conçus pour coopérer et interagir en tant que systèmes multi-agents, Fig. I-2(b), tel que:

- a) Tous les niveaux de décision sont conçus pour être automatisés et exécutables par toutes les entités de décision
 - b) Les gestionnaires et l'expertise développent et poursuivent des collaborations avec leurs parties prenantes internes et externes
- In order to achieve this objective, the study was specifically focused to answer the following questions:

⁴Il utilise la réactivité de CPS qui pourrait prendre en compte tous les événements inattendus (Thomas et al., 2017)

⁵Voir les travaux de Shannon (1998) pour l'art de la simulation et les travaux de Pfeiffer et al. (2008) pour le rôle de la simulation dans l'entreprise numérique

⁶Les atouts qualifiés comprennent la simulation, l'automatisation collaborative, l'auto-organisation et les systèmes, etc.

⁷L'entité de décision, utilisée de manière anonyme avec l'agent, est une unité de construction autonome, intelligente et coopérative (produits, machines, robots, commandes, etc.) au sein du système de fabrication avec la capacité de montrer le comportement de communication physique et informationnel, de prendre des décisions et d'agir pour les décisions (El Haouzi et al., 2009). Il perçoit son environnement à travers des capteurs et agit sur cet environnement à travers des actionneurs.

- 1) Quels inconvénients existent alors qu'un système de fabrication suit une approche de contrôle prédictif centralisé ou un mode de contrôle réactif distribué pour réagir en cas de perturbation (s)?
- 2) Quel est le rôle du système de contrôle hybride dans le renforcement de la coopération entre les entités de décision localement autonomes avec celle des entités mondiales?
- 3) Comment les entités décisionnelles au sein du système de contrôle interagissent et se coordonnent afin d'être réactives et adaptables aux occurrences fréquentes de perturbation?
- 4) Comment *l'état*⁸ a coordination basée sur les entités décisionnelles affecte l'agilité⁹ et stabilité¹⁰ d'un système de fabrication?
- 5) Comment les technologies de traçage aident à identifier/tracer les variabilités de fabrication et à partager ces informations entre les entités décisionnelles?

III. Méthodologie de recherche et contexte ISET

Le contexte de cette étude doctorale était un programme sandwich entre le gouvernement éthiopien et le gouvernement français. Cela prend du temps tous les six mois dans les deux pays. L'objectif était d'améliorer les compétences et les capacités des étudiants des établissements d'enseignement supérieur français compte tenu des contextes industriels comparables des deux pays. Dans ce cadre, deux contextes industriels ont été envisagés dans les usines de bois des deux pays, Table I-1. L'atelier d'imprégnation et de plastification a été tiré de Maichew Particleboard Manufacturing (MPBM) factory en Ethiopie. Il utilise des planches sablées comme matière première principale et des planches laminées comme produits. La lacquering and polishing boutique d'emploi a été prise à partir de ACTA Mobilier Carpentry Factory en France. Il utilise des panneaux stratifiés comme matière première principale et des panneaux laqués de haute qualité (fabriqués en panneaux de fibres à densité moyenne pour les cuisines, salles de bains, bureaux, stands, meubles d'hôtel, etc.) comme produits (Noyel et al., 2013). Ces deux contextes sont réduits à une plate-forme de banc d'essai académique pleine grandeur appelée TRACILOGIS platform¹¹. Comme cette platform est invariante, les contextes industriels réduits peuvent provenir d'autres industries d'assemblage autres que les industries du bois. Il démontre également plusieurs domaines éducatifs liés à l'unité d'automatisation, l'unité de contrôle de la production ou l'unité de fabrication des industries du vêtement et de l'automobile par exemple.

⁸L'État est une variable d'intérêt partagée entre les entités décisionnelles

⁹L'agilité est une capacité à survivre et à prospérer dans un environnement concurrentiel en réagissant rapidement à l'évolution des marchés (Gunasekaran, 1999)

¹⁰Lissage de la production

¹¹TRACILOGIS platform est une plateforme technologique disponible à l'ENSTIB/Campus Fibre et Bois, Epinal, France. Sa description détaillée est présentée dans Appendix D

Table I-1 Deux contextes industriels

S/N	Atelier	Problèmes fréquemment observés	Sources
(1)	Lacquering and polishing shop floor	Perte de temps pour rechercher des pièces	ACTA mobilier
		Changements fréquents de taille et de type de lot	
		De nombreux changements (par exemple, la couleur de la laque qui provoque une reprise)	
(2)	Lamination and impregnation shop floor	Manque de composants WIP (manque de papier d'imprégnation)	MPBM
		Changements externes	
		Goulot d'étranglement des ressources (provoque la création d'un WIP)	
		Produits défectueux	

Compte tenu de la configuration de la plateforme TRACILOGIS, orientée multi-agents sociabilité¹² approche, (Isern et al., 2011), a été utilisé pour modéliser, simuler et mettre en œuvre les algorithmes de prise de décision proposés. Les agents locaux au cours du processus décisionnel coopèrent leur état avec un ou plusieurs agents voisins ou agents globaux configurés avec des ordinateurs numérisés, des interfaces, etc. Cette coopération a pour objectif de créer de la sociabilité (Olfati-Saber et al., 2007), pour le collectif décision, parmi toutes les entités de décision autonomes lors de l'apparition de perturbation (s). Parallèlement, les approches de contrôle proposées ont été initialement modélisées numériquement en mettant en réseau toutes les entités contributrices, puis vérifiées et validées avec simulation d'événements discrets¹³ et tests d'implémentation d'applications sur la base de la plateforme TRACILGOIS respectivement. L'induction de produits intelligents (et / ou l'émergence récente de CPS) et de ressources intelligentes ont également augmenté cette étude pour se concentrer sur l'approche de contrôle de système multi-agents.

Matériaux utilisés en utilisant la plate-forme de banc d'essai TRACILOGIS, cette étude a utilisé différents logiciels de simulation/émulation, des logiciels de mise en œuvre d'applications et d'autres ressources présentées dans Table I-2:

¹²Sociabilité (alternativement utilisée avec l'inspiration humaine): les agents partagent leurs connaissances et demandent des informations aux agents voisins ou à l'environnement pour le processus de prise de décision (McArthur SD et al., 2007; García AP et al., 2010) à Dorri A. et al. (2018). Chaque agent utilise son historique, ses paramètres détectés et les informations d'autres agents pour prédire les actions futures possibles. Ces prédictions permettent aux agents de prendre des mesures efficaces qui atteignent leurs objectifs.

¹³Voir les travaux de Misra (1986) et Sharma (2015) pour le concept de simulation d'événements discrets

Table I-2 Logiciels et ressources utilisés pour tester les approches de contrôle proposées

Logiciels de simulation et de modélisation		Logiciels de test d'applicabilité		Logiciels de manipulation et d'analyse statistique	Technologie de traçabilité et d'identification
Simulator	Programming Language	Simulator	Programming Language		
MatLab	Python, MatLab	JADE	Java	ClickCharts, UML, Minitab, and MS Excel	RFID ¹⁴

Le contexte du département d'ingénierie des systèmes éco-techniques (ISET) du département ISET du CRAN s'intéresse à la modélisation du processus d'évaluation et de prise de décision nécessaire pour contrôler et gérer les systèmes de fabrication complexes (El Haouzi, 2017). Les systèmes étudiés jusqu'à présent sont principalement des applications de systèmes d'événements discrets qui relèvent des domaines de l'industrie, des réseaux de communication et de l'énergie. Le travail du département est divisé en trois projets scientifiques:

- Ingénierie globale de la sécurité de fonctionnement du système (équipe de projet Operation System)
- Ingénierie de la maîtrise des flux physiques ou des énergies du système (équipe projet Ambient Intelligence Systems)
- Ingénierie des réseaux de communication

Dans ce cadre, des recherches liées aux systèmes de contrôle pilotés par produit (PCS), aux systèmes hybrides contrôlés par des produits intelligents, à la modélisation, à l'interopérabilité et à l'évaluation à travers les travaux de thèse de Klein (2008), Herrera (2011), El Haouzi (2008), Pannequin (2007) etc. ont été étudiés. Pannequin (2007) a étudié la mise en œuvre du concept PCS par contrôle hybride et a proposé un outil de modélisation et d'évaluation de ces systèmes. Les deux thèse de Klein (2008) et El Haouzi (2008) avaient démontré, comme objectif principal, la crédibilité et la validité du concept PCS à l'échelle industrielle. El Haouzi (2017) a déclaré *que notre interprétation du PCS nous a amenés à répondre aux problèmes de synchronisation des flux physiques dans un contexte de production en rendant le produit ou un groupe de produits (kanban) actif dans un système de contrôle hybride*. Herrera (2011) a également travaillé sur l'architecture de conduite hybride. La thèse de

¹⁴La RFID est largement acceptée comme une possibilité technologique de traçabilité et d'identification de l'environnement de décision probabiliste pour relier le système physique au système d'information afin de récupérer et de recevoir à distance des données entre les entités. Voir Appendix E

Baïna (2006) et les travaux de Tursi (2007) considèrent le produit instrumenté comme le principal acteur de l'interopérabilité Business to Manufacturing (B2M). En plus du contexte ISET, les sujets des systèmes de fabrication intelligents sont devenus le sujet de chercheurs internationaux tels que SOHOMA, HOLOMAS, IMS² etc.

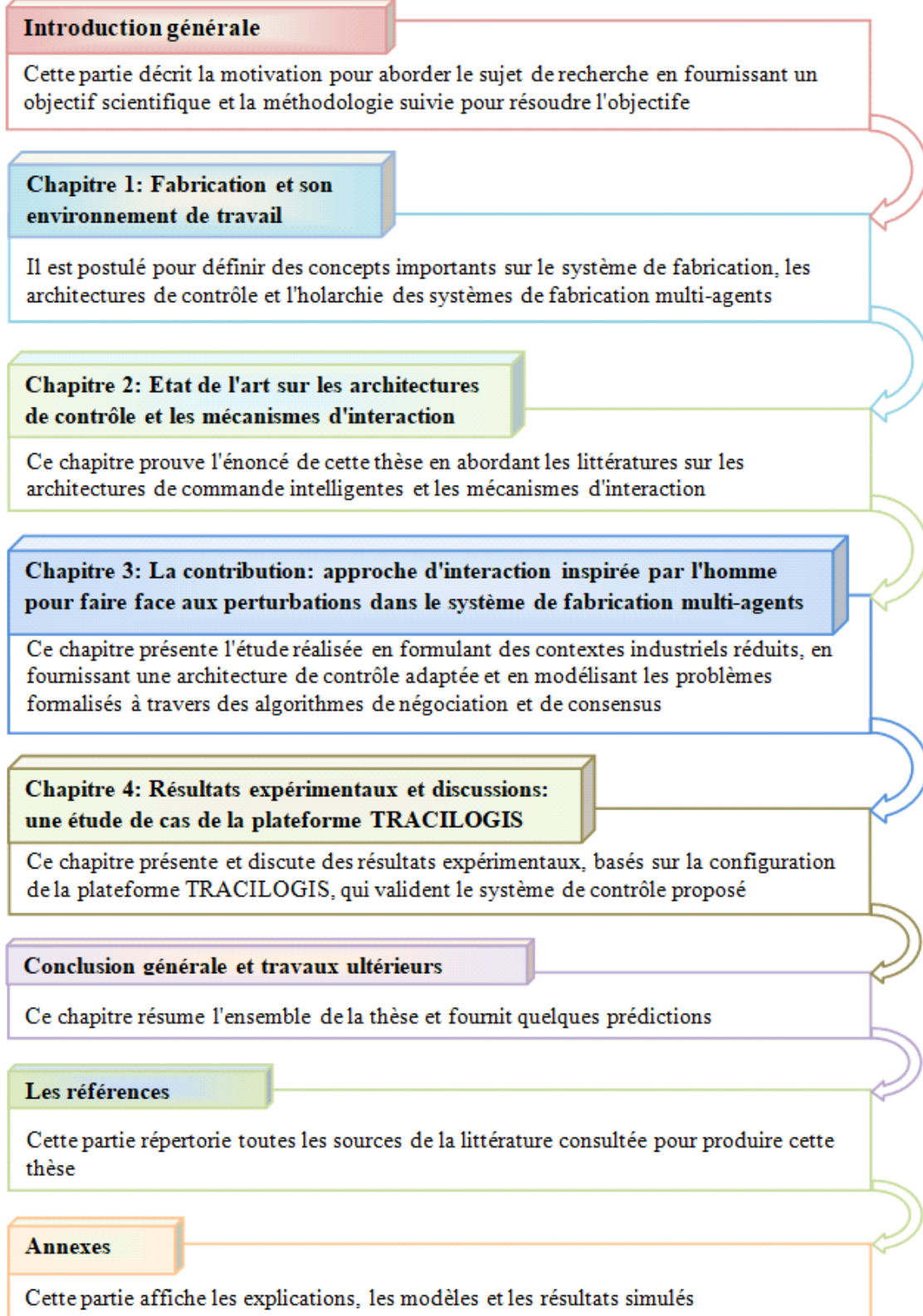
Dans le cadre de ces domaines, cette thèse fait partie du deuxième projet de l'ISET avec un intérêt à utiliser l'approche d'interaction d'inspiration humaine pour concevoir un nouveau système de contrôle. Avant les travaux de recherche en cours, El Haouzi et al. (2013) ont travaillé sur l'ingénierie de modèles de simulation adaptables pour tirer parti des masses de données générées par CPS et Giovannini (2015) sur la formalisation des connaissances dans le cadre de la personnalisation de masse.

IV. Contribution de l'étude

La contribution de cette étude est à la fois théorique et expérimentale. Il a conçu et modélisé une approche d'interaction inspirée par l'homme (au moyen de négociations et d'algorithmes décisionnels fondés sur le consensus) adaptés pour contrôler un système complexe et perturbé dans les dimensions multi-domaines d'un atelier. Il a créé un environnement dans lequel toutes les entités contributrices opèrent, communiquent et interagissent les unes avec les autres de manière productive et aussi proche que l'intelligence humaine. Parallèlement, les tests de simulation et de mise en œuvre d'applicabilité effectués sur la base de la plate-forme de banc d'essai TRACILOGIS ont constitué une autre contribution requise pour montrer l'applicabilité industrielle des approches de contrôle proposées. Il a été proposé avec une importance pour répondre au contexte des exigences individuelles des clients et des produits personnalisés où la charge de travail lissée et la prise de décision en régime permanent deviennent un problème majeur.

V. Structure de la thèse

La structure globale de cette thèse est présentée comme suit



CHAPTER ONE

MANUFACTURING AND ITS WORKING ENVIRONMENT

1.1 Introduction

This chapter is postulated mainly to define some important concepts about manufacturing systems in section 1.2 and control systems followed by their challenges and requirements in section 1.3. It also presents two paradigms, which model and implement the intelligent control systems, namely the multi-agent system in section 1.4 and holonic manufacturing system in section 1.5. Finally, a concluding discussion is presented in section 1.6.

1.2 Manufacturing systems

Manufacturing systems is a recipe of all intermediate manufacturing processes required in the production and integration of product's components. The process begins with product design and materials requirement planning and then after it is modified through manufacturing processes to become the required part. In addition to these processes, manufacturing system incorporates energy, equipment and facilities, labor, market information etc. as an input materials and wastes and scraps as non-desired outputs (Rey G.Z et al., 2014). Meanwhile, manufacturing system is a set of decisional sub-system, information and communication sub-system, and operating sub-system. The decisional sub-system in turn can be decomposed into three levels, Fig. 1-1:

- *Strategical level* is responsible for developing strategies of factories so as to satisfy customer requirements. That is, demand management and resource planning are developed at glance and conveys a production plan to next stage with an annual time (t_a) horizon
- *Tactical level* is master production scheduling and transforming to manufacturing orders and shop floor at weekly time (t_w) horizon
- *Operational level* is execution of manufacturing orders based on current conditions of available resources

At the operational decision level, the evolution of customer expectations and variability and complexity of products have changed the nature of manufacturing paradigms from craft production to mass customization; see Fig. 1-2 and Table 1-1.

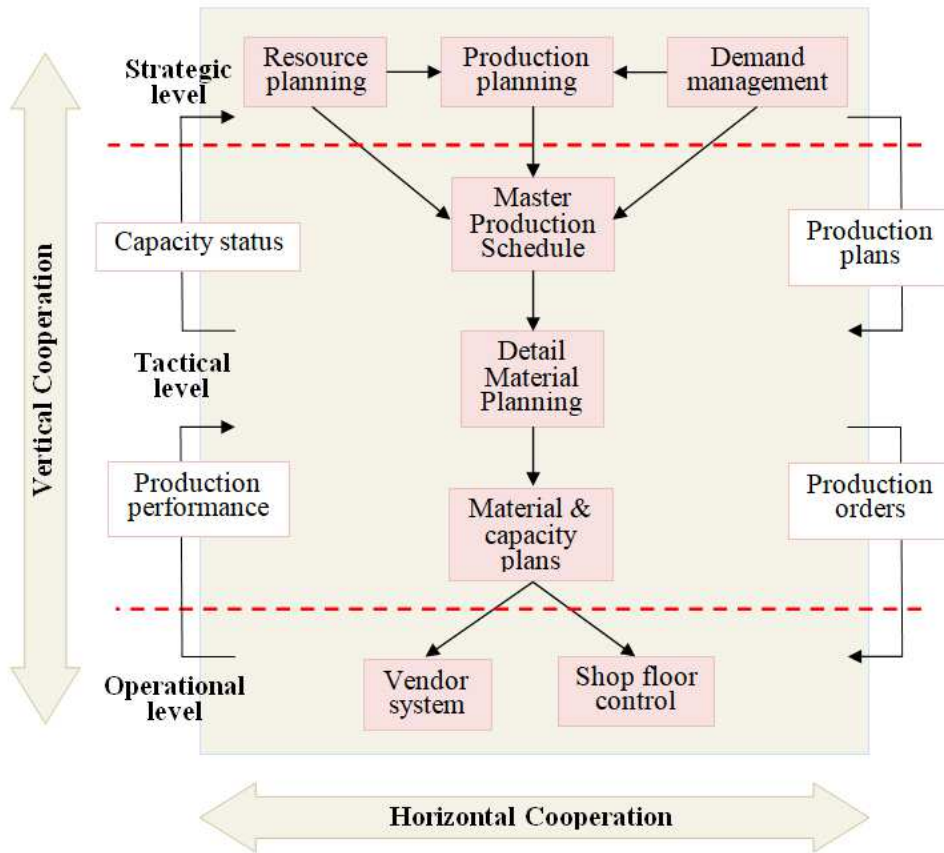


Fig. 1-1 Different decision levels within manufacturing planning and control (Source: Vollmann et al., 1997)

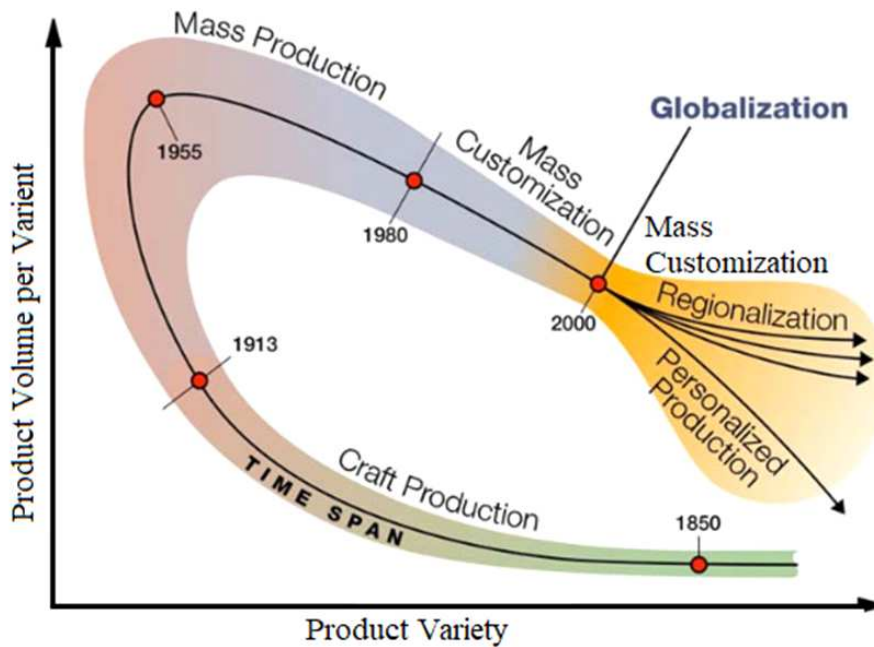


Fig. 1-2 Evolution of Manufacturing System Paradigms (Source: Koren, 2010)

Table 1-1 Manufacturing paradigms and their indicators (Source: El Haouzi, 2017)

Manufacturing paradigms	Indicators		
	Market	Business model	Technology and process enablers
Craft production	Very small volume per product	Pull-sell-design-make-assemble	Electricity and machine tools
Mass production	Steady demand	Push-design-make-assemble-sell	Interchangeable parts and assembly line
Flexible manufacturing	Smaller volume per product	Push-pull-design-make-sell-assemble	Computers
Mass customization and personalization	Globalization, fluctuating demand	Pull-design-sell-make-assemble	Information technology
Sustainable production	Environment	Pull-design for environment-sell-make-assemble	Industry 4.0

From the paradigms presented in Table 1-1 and Fig. 1-2, flexible manufacturing systems (FMS) and reconfigurable manufacturing systems (the mass customization and personalization) have taken due attentions since recent decades. According to APICS¹⁵, FMS is a group of numerically controlled resources and/or machine tools interconnected by a central control system. Flexibility just allows a system to react in case of changes and disturbances. It is presented in Browne et al. (1984) and ElMaraghy (2005) that an FMS is aimed for greater responsiveness to market changes, rapid turn-around, high quality, low inventory costs, and low labor costs. Robots, computers, sensors, actuators, and inspection machines are main components of FMS to create different types of flexibility (Rey et al., 2014). Three from the different types of flexibility are defined as follow to stress component’s flexibility required in this study.

Machine flexibility: variety of operations that a machine can perform with reasonable changeover time. This includes sequence flexibility which is an ability to produce a part by alternative machine sequences.

Operation flexibility: property of product that allows alternative manufacturing operation sequences to process same product. It is enhanced by the ability to execute all manufacturing tasks on numerous product designs in small quantities and with faster delivery.

Process flexibility: set of products that can be manufactured without major setup changes.

¹⁵ APICS is a dictionary of Association for Operations Management that contains core terminologies and emerging vocabularies that one needs to speak the same language across his/her supply chain

On the other hand, the main idea behind mass customization is the use of flexible computer-aided manufacturing systems to produce custom output when necessary (Pine and Davis, 1993; Gilmore and Pine, 1997). It combines the low unit cost of mass production processes with the flexibility of individual customization. Meanwhile, this paradigm shows that system operation in response to unexpected production changes and new requirements of market can be changed dynamically in order to adjust capacity and functions of a production.

1.2.1 Shop floor

It is an area of manufacturing systems within a factory that incorporates detailed production activities and work-in-progress including raw materials, semi-processed products, workers, machines and equipment, production time etc., Fig. 1-3. It can be job shop, flow shop, or open shop.



Fig. 1-3 Screen shot example of shop floor (Source: ACTA Mobilier)

1.2.2 Throughput time

According to Moore & Scheinkopf (1998) and Panizzolo & Garengo (2013), throughput time is a measure of the time required for a material, part or sub-assembly to pass through a manufacturing process following the release of an order to a shop floor, Fig. 1-4. It consists of processing time, move time, queue time, waiting time, and inspection time. Processing time is the time period during which work is performed on a product. Inspection time is the time during which quality of the product is confirmed. Move time is the time during which materials or works-in-process are moved from one workstation to another. Queue time is the period of time during which the product awaits transfer to a workstation, undergoes further inspection and subsequent manufacturing processes. The manufacturing process adds value when customers are willing to pay a manufacturer more money for the products than the manufacturer paid its vendors for the materials and services that went into those products.

Bottlenecks control the throughput of all products processed by them. If work centers feeding bottlenecks produce more than the bottleneck can process, excess work-in-process is built up. Therefore, work should be scheduled through the bottleneck at the rate it can process the work. Work centers fed by bottlenecks have their throughput controlled by the bottleneck and their schedules should be determined by that of the bottleneck. Manufacturing lead time is the time period between receiving of an order and shipment of the completed order to its customer. It is the summation of inventory time and throughput time.

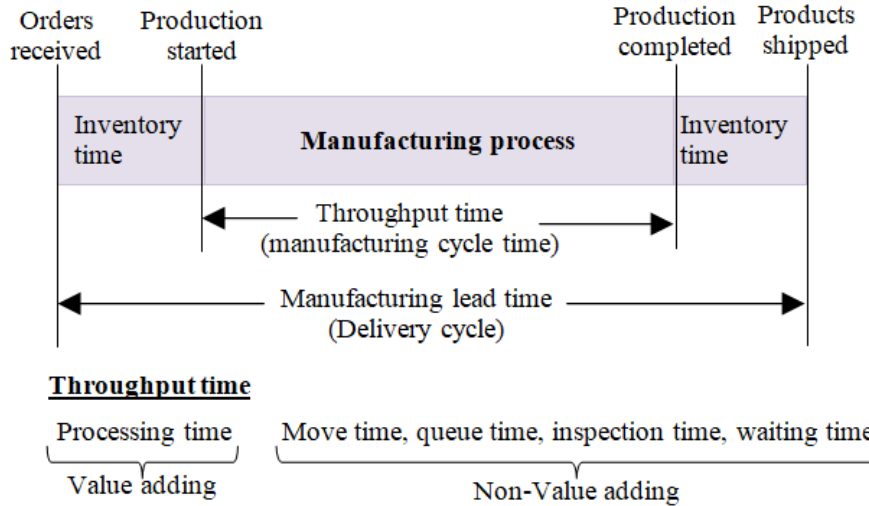


Fig. 1-4 Throughput time of a manufacturing process (adapted from http://www.accountingexplanation.com/throughput_time.htm)

1.2.3 Manufacturing perturbation(s)

The performance of a manufacturing system is mainly measured with its capability to achieve its predefined objectives under worst scenarios (or under the occurrence of disturbances (Labib & Yuniarto, 2005) and urgent changes). Perturbation in manufacturing process is a deviation of system or process from its predicted and normal state caused by outside influence. If this event is not controlled at its origin, it leads to non-value adding activities presented in Fig. 1-4 or lean wastes presented in Appendix F. As it is presented in Park and Tran (2011), this outside influence could be sourced from internal or external environment and Saad and Gindy (1998) have presented some examples under each environment shown in Table 1-2.

Table 1-2 Sources and impact of perturbation(s) (adapted from Saadat and Owliya, 2008)

Sources	Impact on	Examples
Internal sources	Production equipments	Failure of machines
	Material handling	Failure of transportation equipments, conveyors
	Stocks	WIP built-up, raw materials, finished products
	Control systems	Failure of sensors, actuators
External sources	Changes	Priority changes, cancelation of orders, quantity, variety, quality
	Supplies	Supply of wrong parts, delayed delivery time

1.2.4 Scheduling

Before defining a schedule, it is necessary to define process plan since both of them are integrated to each other. A process plan specifies what manufacturing resources and technical operations are needed to produce a product (a job) within a set of machines (Valckenaers and Van Brussel, 2016; Mohammadi and Selim, 2005). Scheduling is assigning of operations of all the jobs on machines/resources while precedence relationships in the process plans are satisfied (Sousa et al., 2007). Two scheduling approaches are most common in manufacturing systems: static scheduling and dynamic scheduling. Unlike static scheduling, dynamic scheduling is the problem of scheduling in the presence of unexpected and real time events (or perturbation(s)) (Ouelhadj and Petrovic, 2009). According to Ouelhadj and Petrovic (2009), most commonly used techniques to solve dynamic scheduling problems are described in Fig.1-5.

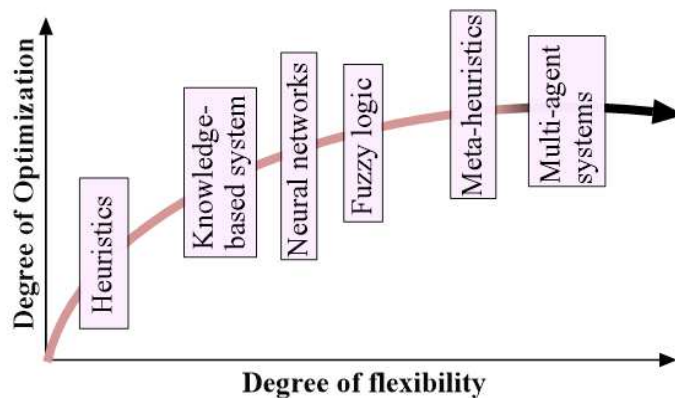


Fig.1-5 Dynamic scheduling algorithms (adapted from Rey G.Z. et al., 2014)

1.3 Control Systems

According to APICS by John H. and Blackstone Jr. (2010), “control system is a system that has as its primary function the collection and analysis of feedback from a given set of functions for the purpose of controlling the functions. Control may be implemented by monitoring or systematically modifying parameters or policies used in those functions, or by preparing control reports that initiate useful action with respect to significant deviations and exceptions”. Control systems can be viewed through different parameters such as behavioral functioning, structural organizations etc. Structurally, Dilts et al. (1991) have been identified four manufacturing control systems, see Fig. 1-6. (a) Central control system; (b) Hierarchical control system; (c) Heterarchical control system; and (d) Hybrid control system.

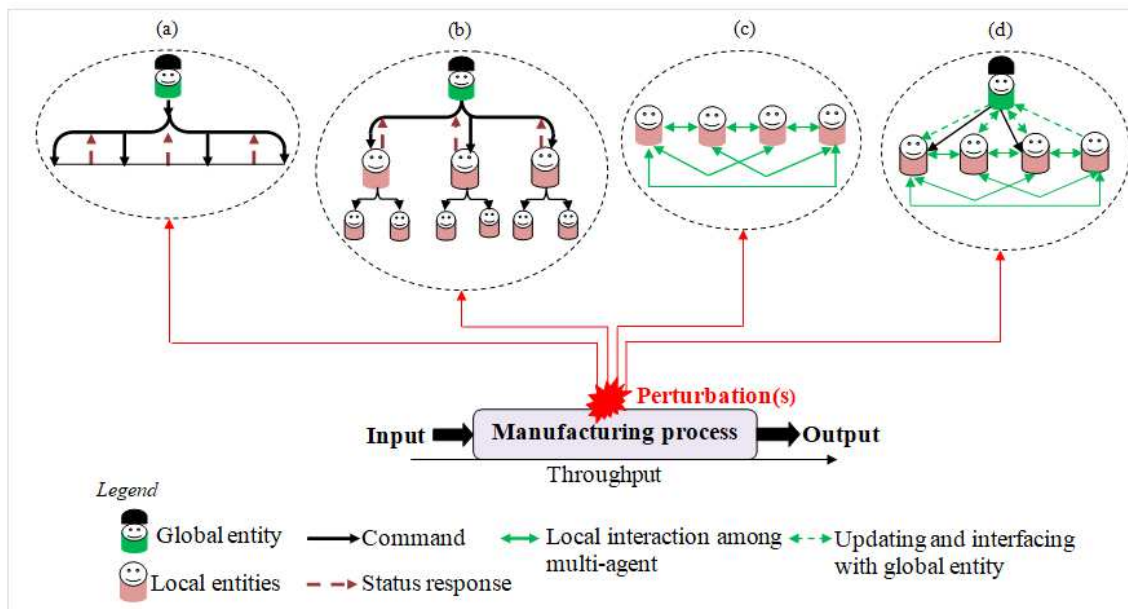


Fig. 1-6 Structural control architectures and relationship between decision entities

1.3.1 Centralized/hierarchical control system

Central control system is known by monitoring all planning and scheduling activities with a single decision entity (or supervisory entity) at the top, Fig. 1-6(a), and it is suited for deterministic environment. It allows global optimization and hence trust in the architecture is very high as the approach is well established, frequently used and implemented for long time. Compared to decentralized or heterarchical control system, it offers strict and secured enforcement to its predictive operation’s plan (Anderson and Bartholdi, 2000; Saharidis et al., 2006). Nonetheless, while manufacturing systems exercise this control system, they started to

face both with poor responsiveness to unexpected disturbances (and changes) and difficulty to make an immediate modification at the time of variability.

These limitations have instantiated to the birth of hierarchical control system. It is much like central control system but differs with its specific characteristics like dividing its control levels into different control functionalities, Fig. 1-6(b), with master-slave relationships (Valckenaers and Van Brussel, 2016). Meanwhile, it is known for its benefit to allow an incremental and gradual implementation of control models using linear programming meta-heuristics such as genetic algorithm. However, the reactivity to disturbances and changes is unsatisfactory as they follow lengthy bottom-up consulting approach. Additionally, it is characterized with long decision loop and it is difficult to modify, maintain, and make any structural changes while a system is in operation.

Challenges of decision entities in centralized/hierarchical control system the lower level decision entities in central/hierarchical control architecture are non-responsive (blind) to unexpected events occurred during the manufacturing execution. Meanwhile, it is presented in Cardin (2017) and Saharidis et al. (2006) that the decision entities are characterized with the following limitations:

- Master-slave relationship as the central decision maker simply directs its command to operators and executers at the operation level
- Decision entity's inability to offer necessary reactivity and flexibility for unpredicted events; simply forwarding these events to upper decision maker
- Decision entity's pitiable adaptability
- They are inefficient to deal with a stochastic environment
- Weak capacity of decision entities to eliminate/minimize non value-added activities encountered at operational level
- Incapability of decision entities to extend their level of effort required to consider future growth and its implementation

The impact of such characteristics of decision entities in central/hierarchical control system is illustrated and narrated by different researchers. For instance, Herrera et al. (2014) have modeled automotive turbochargers manufacturing process context with two job shops (each with different work cells) designed to produce semi-finished products of same order lot. It also includes three decoupling or storing points namely raw material storage, WIP storage, and finished product storage. Based on the five contributing activities of manufacturing

planning and control system in Fig. 1-1, the enterprise resource planning function proposes a weekly centralized master production schedule (MPS). The launching of the manufacturing order is defined taking into account a centralized decision maker and all executing resources. Herrera et al. (2014) hypothesized that if unpredicted event occurs on one of the job shops while executing for the ordered lot, it may lead to difficulty in finding an optimal solution in a very short horizon. One of the solutions might be to split those remaining lots between the other two work cells and the operators will have to ask for an enterprise resource scheduler to reschedule and ERP will go to reschedule it based on the new capacity and resource. Consequently, a lot of working time and resource may be lost during the time needed to:

- Report what has happened to the central decision maker
- Estimate the current and future possible states
- Generate new ideas and choose best of them
- Finally re-launch the new process plan and reschedule the job shops

Until all these things are launched and the new schedule is dispatched to the operators, the whole system will be in a disturbed mode. Thus, Herrera et al. (2014) have concluded that shifting the decision level to distributed reactive control system rather than controlling it centrally would be better though it has its own challenges.

1.3.2 The heterarchical control system

As a response to the limitations observed in the central and/or hierarchical control system, researchers have shifted to develop and implement heterarchical control system characterized by a distributed and flat structure of decision entities, Fig. 1-6(c). It consists of locally distributed and autonomous entities that cooperate with each other to provide reactivity, robustness, and adaptability to ever changing manufacturing environments (Rey G.Z. et al., 2013). The main idea behind this control system is to focus on the operational level in Fig. 1-1 where the unexpected events actually occur. Flexibility, empowerment of employees, survival on its own etc. are the key positive indicators of this control system even though it has high setup cost.

Unlike the global type control system employed by many manufacturing industries in the previous decades, distributed reactive control approach has been revolutionized to realize responsiveness and reactivity of components. Indeed, the fact that distributed entities could cooperate and make decisions without being involved in rigid hierarchies' and fault tolerance sounds promising (Zambrano et al., 2011). However, this paradigm has started to face some important challenges which broadly can be classified as:

- Procedural (or lack of design methodologies and standards) and
- Conceptual (or lack of communication and interoperability protocols) aspects

Challenges of decision entities in heterarchical control system the implementation and usage of central predictive control approach has long history as it is well established and trusted by central managers. This is not usual in the case of distributed reactive control mode even though it proofs high reactivity, adaptability, robustness, and realization of globally optimized plan. A socially myopic¹⁶ behavior of decisional entities prevents distributed reactive control mode not to be as trusted as central predictive control approach. It is stated in (Duffie, 1996; Duffie, 2008; Borangiu, 2015; Leitão, 2009; Ottaway and Burns, 2000) that instead of high degree of autonomy, local goal orientation, and locally contained information of decision entities in this control system, they are characterized with some drawbacks presented as follow:

- Lack of consistency and non-coherency to global decision
- Limited capability to predict process outcomes due to myopic behavior and the absence of centralized coordinator
- Difficult to guarantee a minimum level of operational performance as a result of entity's own goal orientation and making decisions based on local information
- Least commitment of decision entity about other entity's offer around him
- Lack of information that an intelligent entity has over its future and its environment as it does not give much attention to predefined plan etc.

Different researchers have been worked to address such myopic behavior of decision entities. For instance, while Trentesaux et al. (2009) illustrates one of the designing challenges of distributed reactive control mode, they have described that it is hard to understand the way in which entities seek information and the possible relationships with their environment. Rey G.Z. et al. (2014) have also presented that the problem of myopic behaviors of entities can be addressed through negotiation protocols such as contract-net, product driven approach etc.

1.3.3 Hybrid control system (HCS)

In order to alter the challenges presented in central/hierarchical control system and heterarchical control system, the interest of hybrid control system has been increased since the last decade. It couples the high and predictable performance promised by predictive centralised

¹⁶Two types of myopia: social myopia is a condition of decision entities in distributed reactive decision-making are not capable of balancing their local objectives with system's global objective (Rey G.Z. et al., 2014) Temporal myopia is a condition such that decision entities are not long sighted

control mode with the robustness against disturbances based on the reactivity of Cyber Physical System (CPS) (Jimenez et al., 2017; Valckenaers and Van Brussel, 2016). The required goal is attained if supervisor entities have broader view of all the participating local entities and enough vision of executing the system's objectives with specific aims to:

- Control myopia (by command, advising, or updating their subordinates' decisions) (Ouelhadj and Petrovic, 2009; Wong et al., 2006)
- Handling all sorts of interacting issues and conflicts at the local level, command based updating etc.
- Serve as an assistant, capable of processing more information than local entities
- Etc.

Meanwhile, HCS provides stability in the face of disturbances, adaptability and flexibility in the face of change, and efficient use of available resources for final effectiveness. This in turn preserves the stability of a hierarchy while providing the dynamic flexibility of a heterarchy by eliminating their shadowed drawbacks.

These characteristics led HCS to gain due attention by different researchers to solve different manufacturing and scientific problems and mainly to satisfy the framework of Industry 4.0. For instances, El Haouzi (2017), Zimmermann et al. (2017), Herrera et al. (2013), Pannequin (2007), Cardin et al. (2017), Rey G.Z. et al. (2014), Pach et al. (2014a), Pach et al. (2014b), Karanasos et al. (2015) etc. have been explicitly tackled this interest. Cardin et al. (2017), for example, have strengthen the need of this hybrid control system by identifying three future challenges, that could be topic of interest in the coming years, of manufacturing industries:

- Estimation of future performances
- Designing efficient synchronization mechanisms and
- Designing efficient switching strategies

In order to model and implement the intelligent control systems presented earlier, plenty of researchers have been used holonic manufacturing systems and multi-agent manufacturing systems. Both provide an overall production performance and ensure reactivity face to unpredicted events and the following two consecutive sections define and illustrate both systems respectively.

1.4 Multi-agent manufacturing Systems (MAS)

The Industry 4.0 vision has brought a new wave of operating systems combining physical and virtual agents especially the CPS. This indicates that while an agent working by himself is capable of taking autonomous actions, the real benefit of agents can only be harnessed when they work collaboratively with other agents (Dorri A. et al., 2018). Multiple agents have to collaborate as Multi-Agent Systems (MAS) to solve a complex task. MAS is a system consisting of multiple autonomous agents which are situated in an environment (with objects) that the agents can partially perceive and in which they can act and cooperate to achieve system objectives (Michael Wooldridge, 2009; Botti and Giret, 2008). Meanwhile, it is an implementation approach to address challenges (that includes heterogeneous nature of manufacturing systems and their online interactive nature) by offering alternative models and solutions to design systems based on decentralized control functions (Thomas et al., 2017). The messaging and communication media that specifies the information exchange between the agent and all of its neighbors without affecting its autonomy is called interaction protocol. Appendix G and Appendix H presents about multi-agent system and interaction protocol respectively.

1.5 Holonic Manufacturing System (HMS)

HMS in manufacturing context was developed by Suda, (1989) as a response to growing perception that Japanese manufacturing firms lacked competitiveness in a global manufacturing environment. Suda (1989) hypothesized that the cause of this inability to compete was rigid manufacturing practices that did not have the necessary agility and responsiveness in the increasingly volatile markets. Farid (2004), Botti and Giret (2008), Christensen(1994), and Van Brussel et al. (1998) have also supported the hypothetical analysis of Suda (1989) with a concluding remark that the goal of HMS is to attain the benefits that holonic organization provides to living organisms and societies in manufacturing industries. Thus, HMS is a holarchy that integrates the entire range of manufacturing activities (or holons) from order booking through design, production and marketing to realize an agile manufacturing enterprise (Valckenaers and Van Brussel 2016; Thomas et al., 2017). Due to a self-organization ability of holons, HMS combines the high efficiency and predictability of hierarchical control systems with robustness and agility of heterarchical control systems. See Appendix I for detail explanation of HMS.

1.6 Synthesis and Discussion

In the recent decades, the manufacturing industries managing multiple resources within their supply chain are facing with challenges such as globally volatile market, short product life cycle, requirement of highly adaptive and reactive system etc. This happens because the manufacturing system is a vast system that incorporates energy, equipment and facilities, labor, market information etc. as an input materials and wastes and scraps as non-desired outputs (Rey G.Z et al., 2014). In order to create an end-to-end alignment between a manufacturing decision-making and strategic objectives of these industries, minimizing the impact of the challenges by providing efficient planning of tasks to resources within the manufacturing system is significantly required. To realize this requirement and to deploy with the framework of industry 4.0, factories and higher research institutions must always search for intelligent control systems that network physical and computational components. This confirms intelligent reactivity and stability at the time of disturbances, fast adaptability at the time of production changes, and hence enhances complexity and dynamism of a system and/or outside the system (Thierry et al., 2008). Besides, the emergence of cyber physical system, software driven commercial platforms, advancements in information and communication technology etc. have allowed these control systems to show their benefits in reducing the impact of these challenges.

Accordingly, different efforts have been conducted to design and develop different view of control architectures (for instance, central architecture, hierarchical architecture, heterarchical architecture, and hybrid architecture) even though their effectiveness varies from architecture to architecture. Besides, to model and implement these control architectures, plenty of researchers have been used holonic manufacturing systems and multi-agent system paradigms with capability to provide an overall production performance and ensure reactivity face to unpredicted events.

In the framework of these efforts, the recent research community is showing its interest to deeply work on hybrid control system with an implementation media in multi-agent system for many reasons. Cardin et al. (2017), for example, have stated the need of this hybrid architecture by identifying three future challenges that could be topic of interest in the coming years (a) estimation of future performances (b) designing efficient synchronization mechanisms and (c) designing efficient switching strategies. As part of this interest, this doctoral study is commenced to work on the use of human-inspired interaction algorithms to design a new control system with highest capability to react to perturbations occurred at operational decision level.

CHAPTER TWO

STATE-OF-THE-ART ON CONTROL ARCHITECTURES AND INTERACTION MECHANISMS

2.1 Introduction

In order to reduce challenges observed in hierarchical and heterarchical control architectures, another system that takes the advantages of these two control architectures is addressed by many scholars in the previous research decades. This chapter reviews these control architectures with holonic manufacturing system reference architectures in sub-section 2.2.1 and multi-agent manufacturing system paradigms in sub-section 2.2.2. Section 2.3 summarizes and discusses the reviewed control architectures and paradigms in terms of their drawbacks and interaction mechanisms they used.

2.2 General framework on control Architectures and Interaction mechanisms

The autonomously reactive and centrally predictive behaviors of decision entities in distributed reactive control mode and central predictive control approach respectively are encouraging for control approaches change. As a response to this changing call, coupling these two control approaches was believed to minimize their over shadowing drawbacks. According to Cardin et al. (2017) and Jimenez et al. (2017), for instance, minimizing these drawbacks is highly dependent on the way how these two control approaches could be combined. Holonic manufacturing system and multi-agent system paradigms have been handed-over the coupling procedure in order to stabilize manufacturing execution system at glance and performance of whole system then after. Even though both paradigms have a lot in common, the multi-agent system provides multi-domain functions through its cooperative and competitive agents while the holonic manufacturing system is used only in manufacturing domain through its inherently cooperative holons. This indicates that the search for an effective implementation of holonic reference architectures and multi-agent system paradigms remains a subject of great attractiveness in the community of researchers working on agile-adaptable system. Fig. 2-1 presents some examples of intelligent control architectures and Table 2-1 and Table 2-2 summarizes the characterization of these control architectures and paradigms considering different views and parameters such as agents' and/or holons' interaction modes, characterization of reference models, and ease of portability etc.

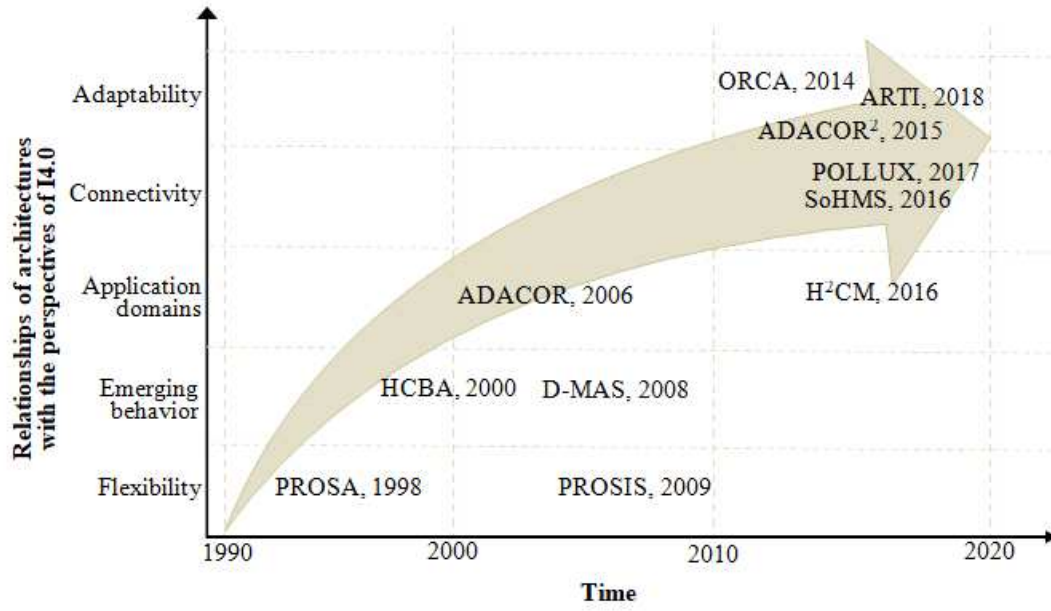


Fig. 2-1 Some examples of Control Architectures (Source: Cardin et al., 2018)

Table 2-1 Comparison and characterization of different control architectures and paradigms (Source: El Haouzi, 2017)

Architectures and paradigms	Relations between agents or holons			Characterization of reference models			Portability		References
	Agents or holons	Interaction & coordination*	Communication mode	Functional aspects	Organizational aspects	Operational aspects	Ease of installation	Ease of adaptability	
PROSA	Product, resource, order, and staff holons	Living organisms-based combination (self-assertive)	Product holon serves as information server to other holons	Staff holon assists basic holons with expert knowledge	Covers aspects of hierarchical and heterarchical controls	Staff holon dynamically stabilize operations in the face of disturbances	Demands advanced control algorithms	Reference architecture for other HMSs	Van Brussel et al. (1998)
HCB: Holonic Component-based approach	Resource and product holons	Both static and dynamic interactions	Black Board (BBS) and Message Broker (MB)	Distributed and bottom-up execution	Distributed control through factory level, cell level, and machine level	Dynamically form a virtual manufacturing controller across the system	Demands advanced computer technology	Resource holon can be replaced easily	Chirn and McFarlane (2000)
ADACOR: Adaptive control architecture	Product, task, operation, and supervisor holons	Pheromone based information propagation	Contract net protocol	Operation holons find their way without supervisor	Rule-based system: developed using Java Expert System	Supervisor holons regulate the activity of the operational holons	Demands FIPA standard JADE	Case-by-case based self-organization and learning	Leitão and Restivo (2006)
Agent-based design of HMS	LPC, CPL, and BBL agents	Behavioral and structural	InteRRaP software	InteRRaP software creates five layers	Combined effect of central and local agents	Bottom-up execution among five control layers	Demands layer-based CIM system	InteRRaP configured PLC	Fisher (1999)

STATE-OF-THE-ART ON CONTROL ARCHITECTURES AND INTERACTION MECHANISMS

MetaMorph : Adaptive Agent-based architecture	Virtual enterprise, distributed intelligent, and concurrent engineering	Mediator- centric federation and Organizational change & learning mechanism	Brokering and recruiting communicatio n	Artificial intelligence based configuration	Multi-agent virtual manufacturing system	Coordination clusters or virtual clusters	Mediators provide system without interfering to low-level decisions	Suitable to very dynamic system but demands MetaMorph- II	Maturana et al. (1999)
Agent-based fractal architecture	Observer, analyzer, resolver, organizer, and reporter agents	MANPro negotiation scheme	Petri-net and UML	IDEF0 model	Hierarchically integrated functional levels with bottom-up reactivity	Basic fractals unit (BFU)	Demands dynamically fractal units and/or agents	Mostly theoretical	Ryu and Jung (2003)
Neuroendocrin e-inspired control approach	Manufacturing cell, control, perception, and decision agents	Biological sensory neurons based	Neuroendocrin e approach	Autonomous & spontaneous behavior	Recursive VSM based relations	Neuro-control and hormone regulation to stimulus sensing	Demands nervous system	Faster as the agents uses bionic neuroendocri ne system	Tang et al. (2011)
Multilevel order decomposition	ERP, MES, and field control agents	PABADIS'PR OMISE architecture	Bill of Operations (BOO)	Bottom-up and horizontal execution among three layers	Automation pyramid structure: ERP – MES – field control	ERP plans & controls production order and supervises resources through three sub-agents: OAS, RSA, & PDR	Simple	Adabtable as ERP is involved in production planning and control	Wunsch and Bratukhin (2007)

Table 2-2 The different control system views and the interaction and coordination mechanisms among decision entities

The different views		Interaction and coordination mechanisms	References
Structural, Fig. 1-6	Central	Supervisory entities are manipulated to coordinate	(Jimenez et al. 2017; Rey et al. 2013; Leitão and Restivo 2006)
	Hierarchical	autonomous entities; overall view of system. It	
	Heterarchical	switches the structure formation between decisional	
	Hybrid	entities	
Behavioral functioning	Proactive	Entities eliminate events before they appear	(El Haouzi, 2017; Herrera et al., 2016 ; Caridi and Cavalieri 2004; Rahwan et al. 2003)
	Reactive	Entities respond to events after their happening	
	Interactive	Agents share equal rights by direct messaging and exchanging their information (Belief, Desire, and Intention) through contract-net	
Coordinational interaction, Fig. 2-2	Bio-inspired	Modification of food searching environment by releasing pheromones: minimizing route	(Pannequin and Thomas 2012; Valckenaers and Van Brussel 2016; Liang and Smith, 2004)
	Physical-inspired	Machines emit attractive potential-fields to attract jobs depending on the services they provide	(Pach et al. 2012; Pach et al. 2014b)
	Human-inspired	Members debate (negotiate) to make law (consensus) as nearly intelligent as human being	(Borisoff and Victor 1989; Zhang et al. 2009)

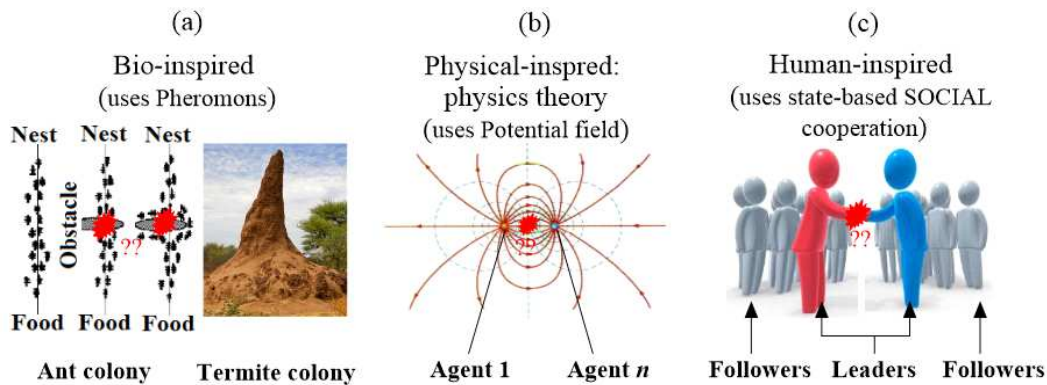


Fig. 2-2 Coordination mechanisms (a) bio-inspired mechanism (b) Physical-inspired mechanism (c) Human/social-inspired mechanism

To design and implement these intelligent control architectures, different technological and managerial coordination mechanisms, listed in the third main row of Table 2-2, have been also evolved in the previous years though their experience of implementation differs (Brennan, 2000). The bio-inspired (also called nature-inspired algorithms in Fister et al., 2013) approach in Fig. 2-2(a), a behavioral metaheuristic schedule generator, has important

place in the research domain of intelligent manufacturing system; in Leitão et al. (2012) and in Ueda (1992) for example. Holonic manufacturing system is one of the most commonly implemented manufacturing systems using this approach. PROSA (Van Brussel, 1998) for example has used virtual ants to explore the intentions of future behavior and guide decisional entities through the decision-making process. ADACOR by Leitão and Restivo (2006) and ADACOR² by Barbosa et al. (2015) are other examples that used evolutionary interaction mechanism. However, this bio-inspired organization does not guarantee an optimal solution as the entities attempt to obtain a best result within specific metric (Jimenez et al., 2017). It is highly dependent on the intensity of odor associated with the biological pheromone (that evaporates with time). It only concentrates on actual works/routing of agents without any negotiation among each other and as a result nimble performance is achieved compared to other organizations.

In the physical-inspired organization, Fig. 2-2(b), it is a machine or resource entity that plays the critical role in controlling the system. Each resource senses intention from products and emits attractive potential fields according to its availability and the services provided to him by the products (Pach et al., 2014b). As the intensity of these fields decrease with distance, products dynamically decide for arriving resource that emits the largest potential field. It is mainly used as key reactive mechanism to handle a transient state of flexible manufacturing system (Pach et al., 2012). Nevertheless, the reduction of potential fields' attraction intensity with distance weakens the required performance.

The third interaction mechanism presented in Fig. 2-2(c) is the human (social)-inspired coordination among the decision entities. Human being has flexible social solidarity and higher level of intelligence than other animals (Zhang et al., 2009). Members of the society once again begin to recognize each other to share a concern in the common welfare and wellbeing of each other. Their communication skill revolve around making sure everyone feels heard and respected while negotiating a mutually beneficial solution that everyone involved can accept (Borisoff and Victor, 1989). Human being also requires minimal supervision, multi-tasking, low energy cost, and understanding of syntax and semantics. This confirms that making decisional entities of manufacturing system to decide as close as to the human intelligence helps to optimize a system better than that of the bionic or potential-field based approaches at minimal supervision. Decision entities in this mechanism follow societal

cooperation and *time-driven*¹⁷ evolution to minimize conflicts as recursive¹⁸ as possible. Furthermore, these decision entities in this mechanism use their dynamic¹⁹ current state to continuously negotiate with their neighbor entities and maintain some societal cooperation at their operational and structural levels (Tonino, 2002; Rahwan et al., 2003). Consequently, multiple variables and parameters determine how these decision entities converge to their final offer called consensus (Olfati-Saber, 2007). Particle Swarm Optimization (PSO) is one of the population-based stochastic optimization approaches that uses social cooperation (Zhang et al. 2009; Tanweer and Sundaram 2014; Sha and Lin 2010; Shi 2001; Kennedy 2006) etc.

Different researchers have also studied hybrid control architectures by classifying into different sub-classes²⁰. For instance, Pach et al. (2014b) have classified hybrid control architectures into four sub-classes considering structural dynamism and control homogeneity. As part of such classification and the scheme of this doctoral study, different hybrid control architectures can be reviewed by classifying into three sub-classes; presented in Fig. 2-3.

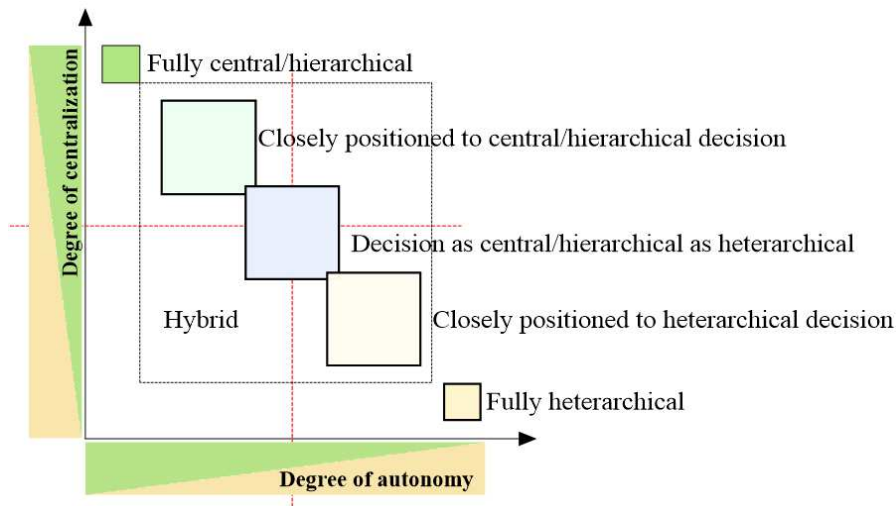


Fig. 2-3 Three sub-classes of hybrid control architectures (adapted from Pach et al., 2014b)

1. *Hybrid control architectures more closely positioned to centralized predictive decision:* the decision changes caused by unexpected events, are performed and authorized by global entities upon the request from local entities. A high level order is given to the local entities for execution (Pach et al., 2014b). Adaptive production control system by Ottaway

¹⁷*Time-driven* is addressed to indicate a current state-based evolution of decision entities with changing time

¹⁸*Recursion* inside an operating unit, same configuration exists with local regulation and local management and all decisional entities decide at their level; Stafford Beer's Viable System Model

¹⁹Decision entities which are not part of the perturbed routing sheet continue their rout as per decision proposed by central supervisors

²⁰Sub-class is a term taken from the work of Pach et al. (2014b)

et al. (2000) and hybrid hierarchical/heterarchical structure by Ou-Yang and Lin (1998) are some of the hybrid control architectures that give much power to global entities to decide.

2. *Hybrid control architectures with decision capability as centralized as distributed reactive*: it present collaborative decision architecture between a central and/or a global entity and the real execution entities. The decision in this sub-class is universal. The global entity controls the system but the control system shifts down to distributed reactive mode if perturbation occurred. Pollux by Jimenez et al. (2017) is the most recent control architecture which falls into this sub-class. The pairwise relationship and their level of interaction between global entities and local entities range from a fully master/slave hierarchical interaction to a fully cooperative heterarchical relationship. According to Jimenez et al. (2017), the global decisional entities have either a *coercive* role to command lists of actions to be performed by the local decision entities, a *limitary* role to propose set of instructions to satisfy the intentions of these local entities, or *permissive* role to delegate its role to the local entities. This presents that the interaction approach in Pollux structurally switches between coercive, limitary, and permissive roles depending on the operating mode on board. Moreover, in PROSA, a central entity creates schedule and all the local entities refine this schedule as per the occurrence of unexpected event during the manufacturing execution system. It is presented in Pach et al. (2014b) that some other control architectures that suit this sub-class include centralized scheduling system and decentralized manufacturing execution system framework by Novas et al. (2013), hybrid push-pull production system by Aqlan et al. (2017) among many others.
3. *Hybrid control architecture closely positioned with distributed reactive decision (it can also be named as semi-heterarchical control architecture)*: once perturbation is occurred, locally impacted decision entities enter to distributed reactive mode without the requirement of global entities. Besides, decision entities which are not part of the perturbation continue their routing sheet according to the initially dispatched central plan. That is, the switching mechanism is not universal; each entity can choose to switch or not to switch in to heterarchical mode (Pach et al. (2014b)). Recent control architecture that suits to this sub-class is ORCA-FMS by Pach et al. (2014b). ADACOR by Leitão (2006), hybrid multi-agent Architecture by Yang et al. (2007), hybrid Production Control structure by Trentesaux et al. (1998), schedule execution in autonomic manufacturing execution systems by Valckenaers, et al. (2007) etc. are some other examples that suit to this sub-class.

2.2.1 Survey on Holonic Manufacturing System (HMS) reference architectures

HMS field in manufacturing context provides reactivity to disturbances, flexibility to changes, and efficient usage of resources. It has got much attention due to its fitness to create synchronized control system during unexpected disturbances (Jarvis et al., 2008) and its fitness to satisfy the requirements of industry 4.0. Three main holonic hybrid control architectures have been studied and elaborated by different researchers in the previous decades: Product-Resource-Order-Staff (PROSA) reference Architecture, ADaptive holonic COntrol aRchitecture (ADACOR), and Holonic component-based architecture (HCBA). See Table 2-1 for the specific interaction mechanisms each of them follow.

PROSA, Fig. 2-4, is considered as reference architecture for other HMS architectures. It identifies types of holons necessary for any manufacturing system, its responsibilities, and interaction structure in which they cooperate. It uses object-oriented programming (OOP) concept to organize its components. According to Van Brussel (1998) and Valckenaers and Van Brussel (2016), PROSA is built from three holons: product, resource, and order holons. Staff holon is added to assist these basic holons with expert knowledge. Product holon stores process and product knowledge needed to insure the correct execution of the product with sufficient quality. Resource holon is an abstraction of the production means, such as machines, conveyors, pallets, raw materials, tool holders, material storage, personnel, floor space etc. (Valckenaers et al., 2007). Besides, a resource holon offers production capacity and functionality to the other holons. An order Holon is responsible for doing the work assigned on time and in the right way (Giret and Botti, 2006). It also interprets a task within a system and hence prepares a work assigned optimally. Meanwhile, it manages the physical product being processed, the product-state model, and all logistic information related to a job (Valckenaers et al., 2007). For example, if perturbation occurred, this holon become autonomous and manage its own schedule using exploration and intention ants (Pach et al., 2014b).

On the other hand, three of the holons cooperate through aggregation and specialization and exchange information and knowledge about the manufacturing system they are engaged-in, see Fig. 2-4. They represent different scenario in its application; process and technological planning, resource allocation, and logistics management respectively. Valckenaers and Van Brussel (2016) and Van Brussel (1998) have added that product and resource holons are engaged in process knowledge that contains information and methods on how to execute a certain process on a certain resource. Product holons and order holons exchange production

knowledge that concentrates on methods and approaches on how to produce a certain product using a given resources. Likewise, resource holons and order holons concentrate on the process execution knowledge. They give due attention to procedures and methods regarding the progress of executing processes on resources and knowledge on how to request the starting of processes on the resources.

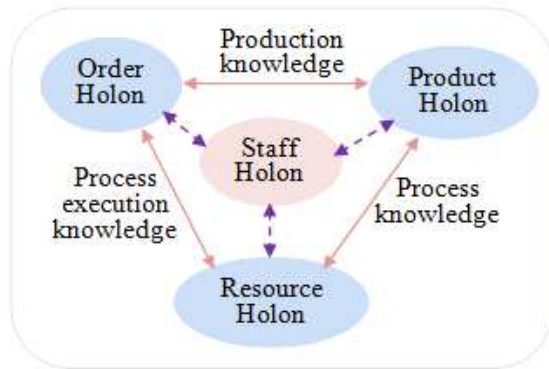
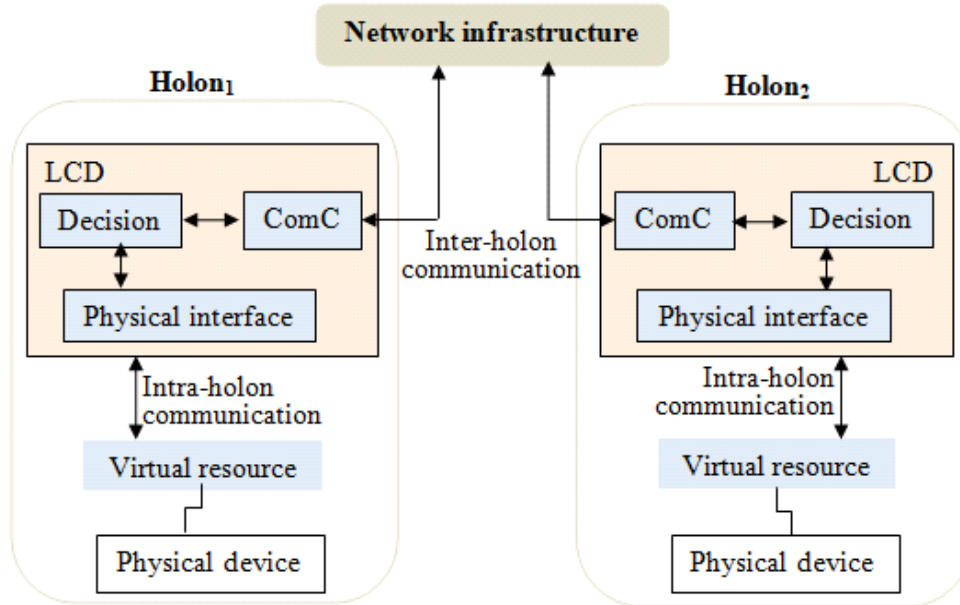


Fig. 2-4 PROSA reference architecture (Source: Van Brussel et al., 1998)

ADACOR (by Leitão and Restivo, 2006), Fig. 2-5, *ADACOR* (by Leitao et al., 2005), and *ADACOR*² (Barbosa et al., 2015) use a pheromone-like propagation mechanism to deal with myopic phenomenon by introducing a supervisory holon. It defines a holonic approach for dynamic adaptation and agility in the face of frequent perturbations within flexible manufacturing systems. It is based on a group of autonomous and cooperative holons in order to represent factory components (either physical resources such as numerical controllers, robots, programmable controllers, etc. or logical entities such as products, orders, etc.). These holons have an autonomy factor that adapts the holon's autonomy according to its environment (Rey G.Z. et al., 2013). A low-autonomy factor implies following supervisory advice and a high-autonomy factor allows self-organization and local problem-solving in case of perturbations. Once perturbation is occurred, the locally autonomous holons switch to heterarchical mode and manage their problem (perturbation) without the involvement of supervisory holon. *ADACOR* groups the holons of a manufacturing system into product holons, task holons, operational holons, and supervisor holons. The first three holons are quite similar with that of PROSA but the supervisory holon introduced in this architecture has specificity over that of the staff holon in PROSA. For instance, it has capability to form group of holarchy and it is in charge of triggering schedules periodically. Each product is represented by one product holon that has all of the product-related knowledge and is responsible for the process planning. To this end, product holon stores information about the

product structure and the process planning to produce it. Every manufacturing order is represented by a task holon, which is responsible for controlling and supervising a production execution. The operational holons represent the physical resources of the factory, such as human workers, robots and machines. They manage the behavior of these resources based on their goals, constraints and capabilities, and try to optimize their agenda.



Legend:

LCD denotes logical control device, ComC denotes communication

Fig. 2-5 ADACOR Holon (Source: Leitao et al., 2006)

HCBA, much like *PROSA*, serves as a template for numerous implementations and applications of holonic systems. Chirn et al. (2000) and Farid (2004) have testified that *HCBA* is derived from the concepts of component-based development (CBD) that provides a guideline for *HCBA*. It is presented in Botti and Giret (2008) and Chirn et al. (2000) that *HCBA* defines two major holons: product and resource holons. The resource holon is an embedded system component that can execute operations such as production, assembly, transportation, and checking. The product holon may contain a physical part such as raw materials, product parts and a controlling part may represent the path controlling a production line, process control, decision-making, and product information. The holonic system is built associating these two types of holons, building nested structures of products and resources. As per *HCBA*, a manufacturing firm can be decomposed into nested business, factory, cell, and machine levels which double as the firms' resource holons and each holon has a component that corresponds to machine, cell, and factory level components in conventionally controlled

Computer Integrated Manufacturing systems (Farid, 2004). Between the cell and machine levels or at the hardware-software boundary, there exists a Black Board System (BBS), a cell-wide system that mirrors the real-time relays of the cell, and each holon is allocated a slice to control its respective hardware resource.

Pujo et al. (2009) has also offered other holonic control architecture called PROSIS (product, resource, order and simulation Isoarchic structure). The PROSIS concept inherits from the PROSA basic concepts (Product Holon, Resource Holon and Order Holon). Product Holon consists of a material product (the physical part) and an informational product (the immaterial part), called I_Product. This I_Product contains the manufacturing process data of the product, but also its state model and all information concerning its traceability. There, thus, are as many PH as manufactured products or as products in the Work In Progress (WIP). This is a major difference with PROSA. This unit identification requires the deployment of ad hoc technologies, a good example of which being RFID. The ID Tag is fixed on the manufactured product. Order Holon represents a task in the production system: a work order concerns a set of PH. It is thus closely related to the concepts of batch, WIP and lead times. A product order is associated to each task. Concretely, the order sheet or the container, the pallet or the case is equipped with an ID Tag. This allows synchronization with the I_Order, which ensures, during completion of the work, the respect of the lead times as well as the taking into account of economic factors (size of the batch, WIP quantities, minimization of the production changes, split batches, etc.). Resource Holon remains conceptually similar to the definition of PROSA. The Staff Holon is not used in PROSIS because this type of holon is not needed in an isoarchic context. It is replaced by a Simulation Holon having a totally different objective. The Simulation Holon is added to show and evaluate the prospects for evolution of the production system with time. Indeed, the main difficulty with a self-organized decision system is the lack of visibility on the future activities, due to the absence of planning and scheduling.

To sum-up, the holonic reference architectures have been studied by the holonic community and clearly showed that they are capable to minimize impact of perturbations on a holonic system. However, the setup for holonic structure requires high effort and hence development of holonic control applications is sophisticated. This is because modeling of complex dynamic systems is costly and incurs significant processing overhead due to the demand for powerful modeling platforms and high complexity (Dorri A. et al., 2018). Meanwhile, the application domain of holonic reference architectures is limited to manufacturing systems only. Hence, the flexibility, autonomy and scalability afforded by

agents made agent-based modeling a low-cost and low resource solution for modeling complex systems (Dorri A. et al., 2018). Subsequently, the developments in multi-agent communication and sensing have evolved significant interest and research activity in the area of cooperative MAS.

2.2.2 Survey on Multi-agent System (MAS) paradigm

In the intelligent manufacturing system, it is well accepted that MAS based control system is a good way to deal with disturbances and decision changes. Thanks to built-in capabilities of smart decision-making entities, it provides right answer to exciting needs and requirements by the framework of industry 4.0. Meanwhile, this multi-agent system provides software packages for the implementation of HMS through cooperative algorithm such that each agent's beliefs are brought to negotiation until all neighboring agents reaches at their final offer (Farid, 2004). Agents in such control system are characterized by their autonomy in which decisions at the time of necessity are made by these agents. This helps to create robust system with respect to time-evolving communication. Without being exhaustive, some intelligent manufacturing systems that used MAS paradigm could be cited: (Leitão, 2009; Isern et al., 2011; Xiong and Fu, 2018; Wong et al., 2006; Jennings et al., 2001) etc. In all these citations, it is clearly presented that the applicability of MAS is characterized by their large scale in terms of number of agents, dynamic nature, and complex functionality. Table 2-3 summarises some application domains of multi-agent system in manufacturing systems.

Table 2-3 Application domains of MAS in manufacturing systems

Specific applicability	Agents in each application	References
Order release	Order agent	
Design	Feature agents	
Master production scheduling	Planning agents	(Verstraete et al., 2008; Holvoet et al., 2009; Caridi and Cavalieri, 2004)
Material requirement planning	Planning agents	
Scheduling	Scheduling agents	
Controlling & monitoring	Controlling and monitoring agents	

The following two sub-sections present some literature surveys related to cooperative and consensus-based multi-agent decisions in the context of networking and intelligent manufacturing systems.

2.2.2.1 Survey related to cooperative decisions

MAS consists of society of agents, distributed inside manufacturing cells, that could potentially cooperate with each other and their outer environment in order to perceive, reason, and converge to a suitable solution (Isern et al., 2011; Wooldridge, 2009; Botti and Giret, 2008). Within the context of this cooperation, Weiss (1999) has pointed out knowability, reactivity, proactivity²¹, adaptability²², and sociability as basic characteristics of multi-agent to create an environment that provides an infrastructure specifying communication and interaction protocols. Keeping these characteristics, in all but the most trivial of non-deterministic environments, these decision entities during their execution are exposed to possibility of failure (e.g., unexpected breakdown of machine agent in manufacturing systems). Due to this non-deterministic nature, Wooldridge (2009) has raised two questions that should always come when one needs to implement adaptive agents, Fig. 2-6:

- a) How one can build agents that are capable of independent and autonomous action in order to successfully carry out tasks delegated to them? Micro level design solution
- b) How one can build agents that are capable of interacting with other agents particularly when the other agents cannot be assumed to share the same interests/goals? Macro level sociability solution,

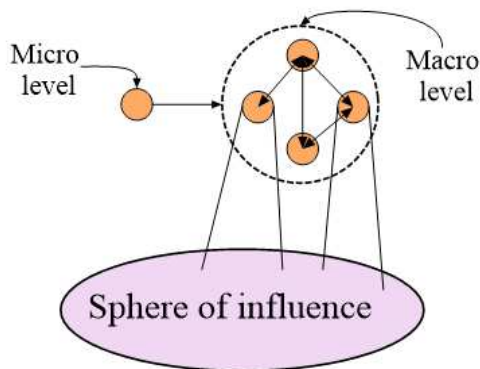


Fig. 2-6 Society of cooperative multi-agent (adapted from Wooldridge, 2009)

Different authors have been worked to answer and validate these questions. For instance, Isern et al. (2011) have presented that even though agents are perceived as autonomous entities, they are also members of a society. These agents have to construct

²¹ Proactivity indicates that decision-making relies to a larger extent on its own perception than to prior knowledge given to it at design time

²² Adaptability is capability to adapt to dynamically changing environment in order to maintain its pre-scheduled role and achieve its goal

societal relationship to provide a common domain through which they can act and communicate and have to exchange information with other agents and maintain some relationships at organizational level. Consequently, the mere presence of multiple agents makes the environment appear dynamic from the point of view of each agent, with the control system they follow, typically distributed reactive control (Vlassis 2003). Meanwhile, Holvoet and Valckenaers (2009) have stated that the applicability of MAS is characterized by their large scale in terms of number of agents & physical distribution, their very dynamic nature, and their complex functional & non-functional requirements. Caridi and Cavalieri (2004) have also presented that optimal global performance could be achieved if several decision-making agents, distributed inside manufacturing cells, cooperate and interact for common offer. To realize these applicability characteristics, as it is explained earlier, different coordination mechanisms such as the bio-inspired organization, physical-inspired control organization, and human-inspired (through negotiation between intelligent entities (Tonino 2002; Rahwan et al. 2003; Kraus 1997)) could always take appropriate attention as they bring amplified benefits.

From these mechanisms, negotiation among the multi-agent has taken viable attention as it enables for a group of agents to achieve their objectives (scheduling a plan in real manufacturing system for example) at mutual agreement. This is proved by different researchers mainly in the communication domain. For instance, Dimopoulos and Moraitis (2006) and Isern et al. (2011) have stated that individual agents can have different goals; varied levels of rationality, heterogeneous capabilities etc. and hence can generate and execute their plans independently. However, as they operate in the same environment, conflicts may arise and hence they need to coordinate their course of action. Wooldridge (2009) and Rey G.Z. et al. (2014) have also indicated that negotiation among agents is foreseen to host a robust-predictive-reactive scheduling²³ and also to tackle myopia.

Besides, Tonino et al. (2002) have investigated different automated agent negotiation approaches including game-theoretic, heuristic-based, and argumentation-based approaches. Three of them emphasize the importance of exchanging information between agents in order to mutually influence their behaviors. The game-theoretic approach helps to determine an optimal strategy by analyzing the interaction of agents as a game rule between identical and self-interested participants (Nagarajan and Sošić 2008; Rosenschein and Zlotkin 1994).

²³Robust predictive-reactive scheduling takes a focus on generating schedules to minimize the impacts of disruptions on the performance. A usual solution is to consider not only schedule efficiency, but mainly deviation from the original schedule; termed as stability (Vlk M. & Bartak R., 2015)

However, unbounded computational resources can be taken as limitation of this approach. To overcome this limitation, a heuristic approach (Aydogan et al., 2013; Kraus, 2001) has come with the principle of *produce good enough rather than optimal outcomes*. Irrespective of its advantage, this approach is also known for its sub-optimal outcome as it does not examine the full space of possible outcomes (Jennings et al., 2001). Subsequently, argumentation-based negotiation approach (Monteserin and Amandi, 2011; Rehwan et al., 2003; Sierra et al., 1997) has evolved to overcome the knowledge limitations of agents in game-theoretic and heuristic negotiation approaches.

As it has been surveyed by Rahwan et al. (2003), argumentation-based negotiation approach allows agents to exchange additional information or to argue about their beliefs & other's mental attitudes during the negotiation process. Agents accept, reject, or critique an offer proposed by other agents until they agree on this offer. Meanwhile, Monteserin and Amandi (2011) have presented that when agents negotiate, the arguments uttered to persuade the opponent are not the result of an isolated analysis, but of an integral view of the problem that they want to agree about. Before the negotiation starts, they have in mind what arguments they can utter, what opponents they can persuade, which negotiation can finish successfully and which cannot. Hence, argumentation-based negotiation approach has been gained increasing popularity for its potential ability to overcome the limitations of other conventional approaches.

MAS negotiation has been also specifically used by different researchers to solve the problem of scheduling in manufacturing systems. For example, it is presented in Madureira et al. (2014) that negotiation in scheduling is generally used to improve quality of final solutions. Madureira et al. (2014) worked on negotiation mechanism to deal with resolution of scheduling in real manufacturing system. They proposed this mechanism considering set of resource agents and a coordination mechanism combining a single solution obtained by the resource agents into a global solution. The works of Zatter et al. (2010) and Adhau et al. (2012) have also proposed negotiation mechanisms for integrating process planning and scheduling and for distributed multi-project scheduling respectively. In Xiong and Fu (2018), a new immune multi-agent scheduling system has been developed to solve a flexible job shop scheduling problem. But its applicability was based on immune system such that a body is protected from foreign antigens by immune response. If the immune system is unexpectedly failed, there will not be chance to recover and solve the scheduling problem. Hence, as continuation to the works conducted on negotiation among multi-agent for improving

performance of a manufacturing system, proposing a negotiation-based control approach considering smart product agents and resource agents was one of the contributions of this study.

However, while autonomous decision entities execute according to multistage negotiation, they may decide alone by rejecting the negotiated offer with their neighbor entities. At this instant, it is necessary to reach a consensus on the course to be taken (Ren and Beard, 2008). Even though there are few scholars who worked on the role of consensus to solve manufacturing systems' problems, most consensus-based literatures concentrate to solve the problems of networking and the following sub-section presents literatures on this domain.

2.2.2.2 Survey related to consensus in cooperative decisions

Literatures on MAS have revealed that negotiation is good element of networked agents to reach at their final offer called consensus. Consensus has become applicable in mathematics, physics, control theory etc. and its applicability is revealed in synchronization of collaborative decision support systems (Cao et al., 2012). These applicability's include rendezvous in the space (Dimarogonas and Kyriakopoulos, 2007; Sinha and Ghose, 2006), flocking (Olfati-Saber, 2006; Lee and Spong, 2007), decentralized scheduling in networking domain (Moore and Lucarelli, 2007), coupled oscillators (Su et al., 2009) among many others. In all these application domains, agents not only update their own state but also the state of neighbors. This helps to design an update law and hence to converge to a common value based on the designed update law.

Consequently, different researchers have been worked to design different control laws and protocols. Dimarogonas and Kyriakopoulos (2007) for example, have proposed discontinuous and time-invariant non-holonomic control law so as to examine stability of a system. They described convergence of designed multi-agent system relies on connectivity of a communication graph that represents an inter-agent communication topology²⁴. Lee and Spong (2007) on their flocking study have proposed a provably-stable flocking control law. It was proposed considering a stable flocking of multiple inertial agents (keeping their shape and velocity). It ensured that internal group formation is exponentially stabilized to a desired shape while all agents' velocities converge to a centroid velocity with time-invariant evolution. Moore and Lucarelli (2007) have also proposed consensus variables with nearest-

²⁴Topology refers to location and relations of agents (Dorri A. et al., 2018)

neighbor communication topology to solve a decentralized adaptive scheduling of tasks or mission timing problems. On the other hand, Ren and Beard (2008) have hypothesized that to reach consensus among group of agents, there must be a shared variable of interest called informational state. Wang and Shao (2015) have strengthened the role of informational state by developing consensus protocol where state of neighbor agent is continuously controlled by state of initial agent. All the above reviews indicate that there has been a great extent of concern on consensus algorithm to solve conflicting problems among own goal-oriented agents.

To sum-up this sub-section, an agent in multi-agent system maximizes its benefit on the expense of its neighbor agents. Hence, many researchers have given due attention to the applicability of sociability-based multi-agent systems for controlling such communication problems in the field of networking sciences, sociology, anthropology, philosophy, economics etc. However, its applicability for manufacturing control problems was highly comprehended with challenges such as difficulty to guarantee minimal level of information and operational performance, difficulty of cooperation, lack of commercial platforms (Jimenez et al., 2017; Rey G.Z. et al., 2014) etc. Meanwhile, several challenges within the manufacturing systems such as design of decision model, design of interaction protocols, integration of technological solutions remains yet unsolved. Hence, exploring on MAS-based control algorithms that help manufacturing systems' components cooperate for their common goal is still a demanding research area. To the best of author(s) knowledge, a consensus algorithm from control theory, for example, has rarely been adapted to decision-making algorithm or not yet implemented in any of the manufacturing systems.

2.3 Synthesis and Discussion

Plenty of researchers have given owed concentration on different intelligent control architectures to provide efficient production performance and ensure reactivity face to unpredicted events. However, several issues left unsolved such as autonomy of decisional entities and legal aspects to represent sociability challenges, design of interaction mechanisms and protocols, development of decision models and implementable control architectures, issue of introduction and implementation of technological solutions such as Radio Frequency IDentifications (RFID), type and frequency of perturbations addressed to explain some of them. To emphasize these challenges, Cardin et al. (2017) in their state of the artwork, for example have pointed out three confusing choices by decision entities to reschedule a perturbed discrete manufacturing system: (a) do nothing (b) decide

autonomously and (c) decide by shrinking towards central scheduler. In their state of the art work, Cardin et al. (2017) have also presented that the three main challenges that still need to be investigated in the next few research years are:

- Challenge-I: Estimation of future performances
- Challenge-II: Design of efficient synchronization and/or coordination mechanisms
- Challenge-III: Design of efficient switching strategies integrated into a hybrid control architecture

In the framework of investigating these challenges, different types of coordination mechanisms have been evolved and used so far, Fig. 2-2. However, the drawbacks noticed in the bio-inspired coordination approach and in the physical-inspired organization approach have led this study to choose and to explore the use of human-inspired coordination approach as a support for decision-making in designing and implementing a recursive hybrid control system. Meanwhile, this interaction approach has been chosen over the other interaction mechanisms for the following two reasons:

- This human intelligence has not been studied as much as using the intelligence of other animals (for example ant, termite, bee etc.) to solve the problems of manufacturing systems. Most of the previous studies have worked by integrating the structural interaction approach either with the bio-inspired coordination mechanism or with the physical-inspired one.
- This study believed that the concept *human-inspired* may have better psychological acceptability by workers of industrial enterprises and has better closeness to the legacy of Enterprise's ERP, MPS etc. Meanwhile, the human intelligence closely fits to the requirement of effective management of information and knowledge for the rising products' variety and implementation of continuous improvements induced to regularly update and reconfigure a system

In the context of implementing this human-inspired interaction approach, it is structurally integrated with agent²⁵-based recursive hybrid control architecture (called REDCA²⁶) that falls within the third sub-class in Fig. 2-3.

²⁵In addition to the reviewed research domains, multi-agent system is chosen due to the configuration of the experimental platform, called TRACILOGIS platform, used in this study. See Appendix D for the characteristics of this platform

²⁶Chapter four explains REDCA and its applicability to deal with perturbation(s) in multi-agent manufacturing system.

CHAPTER THREE

THE CONTRIBUTION: HUMAN-INSPIRED INTERACTION APPROACH TO DEAL WITH PERTURBATION(S) IN MULTI-AGENT MANUFACTURING SYSTEM

3.1 Introduction

In the intelligent manufacturing control domain, manufacturing systems are considered as complex systems consisting of many autonomous²⁷ entities with various communication and interaction skills. It is presented in El Haouzi (2017) that such interaction and adaptation are required for two main reasons:

- ❖ The variety of products is always rising and requires effective management of information and knowledge
- ❖ The implementation of continuous improvement processes induces a need to regularly update and reconfigure the tool of production

These requirements could be achieved if the cooperation of different flows (physical and informational) and control paradigms (characterized with agile, resilient²⁸, and one-piece-flow (Feld, 2000) manufacturing system) are well designed and implemented.

As part of satisfying these requirements, this chapter is aimed to present the contribution of human-inspired coordination approach by integrating with structural architecture called recursive hybrid control architecture. The implementation architecture with its executability explanation is presented in section 3.2. In order to validate the applicability of the proposed coordination mechanism, two industrial contexts have considered with their decision-making algorithms presented in sub-section 3.3.1 and sub-section 3.3.2. The validation includes formalization of these two control problems and execution explanations of contributing smart entities to solve the specified problems. Finally, section 3.4 presents the summary and discussion of the proposed interaction approach relative to other approaches.

²⁷Autonomy refers to the capability of an agent to create and control the execution of its own plans and/or strategy (Valckenaers and Van Brussel, 2016; Wooldridge, 2009).

²⁸Holling (1973) defined resilience as a measure of persistence of systems and of their ability to absorb change and disturbance and still maintain the relationships between populations or state variables

3.2 Implementation architecture: a Recursive hybrid Control Architecture (REDCA)

Even though there are different contributing decision parameters, the performance of a manufacturing system mainly depends on the accuracy of scheduling. For instance, Rey G.Z. et al. (2013) have specified that online control and monitoring of manufacturing activities (such as priority-based sequencing, scheduling etc.) is one of the tasks in complex-adaptable systems. Such complex tasks could be solved and/or minimized through three integrated hierarchical levels: strategical level, tactical level, and operational level, see Fig. 1-1. At operational level, an execution process continues with daily and/or hourly time horizons ensuring dispatching, controlling, and disposition activities. While executing based on such hierarchical control system, disturbance and changes coming from external and/or internal environments may occur and could interrupt the required physical and informational flows. In order to reduce the impact of these perturbations, another solution with capability to amplify its control functions both centrally and locally must be designed and developed.

In the framework of this effort, REDCA (structurally semi-heterarchical and behaviorally reactive) is developed to implement the proposed interaction approaches in this study. It is developed with its suitability to create reactive, sociable, cooperative, and adaptable manufacturing systems at the time of perturbations, Fig. 3-1. It is an extended version of VSM²⁹ based work proposed by Herrera et al. (2011) by taking the recursiveness and legacy system of the work. However, REDCA differs from Herrera et al. (2011)'s model by the level of implementability and capacity of adjusting perturbation(s). To adjust a perturbation(s), the different aggregation levels in Herrera et al. (2011)'s model were limited to either waiting until final decision is distributed by a central entity or splitting work-in-progress products to neighbor machines which are working for same lot. This indicates that the model was not able to clearly show a layer-based contribution of each aggregation level. Moreover, it is product agent who plays all the roles in optimizing the system while the other agents (for example machine agents) have contributing role. Consequently, REDCA is developed to remedy these limitations by developing a layer-based control and execution system. It follows similar control functions as VSM with self-adaptability in perturbed environment and indefinite repeatability (called recursion) in order to satisfy control requirements at very local decision levels.

²⁹VSM, an abbreviation for Stafford Beer's Viable System Model, allows proposing an organizational model based on the structure of human nervous system focusing on lower level activities taking into account system components (products, machines, conveyors, operators etc.) that perform assigned tasks. See Fig. C-1 in Appendix C.

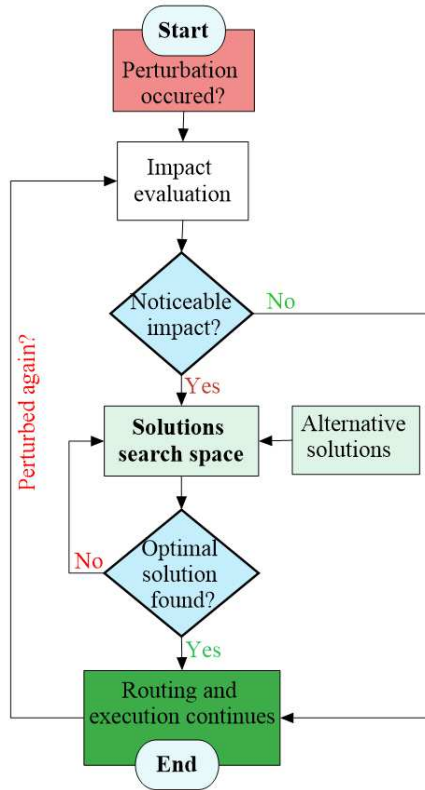


Fig. 3-1 Cybernetic action loop to identify and to diagnosis perturbation(s)

Meanwhile, REDCA is designed considering the concept of dynamic hybrid control system introduced in Pollux by Jimenez et al. (2017). Pollux used a layer-based control system organized into coordination layer, operation layer, and physical layer. Subsequently, REDCA has considered a wider operational specialty of blue-collar layer and an updating and interfacing³⁰ role of white-collar layer; see the structural architecture in Fig. 3-2(A). The operation layer and physical layer of Pollux fall into the blue-collar layer in REDCA. Hence, REDCA is designed to satisfy the requirements (over Pollux and other control architectures) of scalability, dynamic role assignment of decision entities, better closeness to ERP, and better level of acceptability by workers of industrial enterprises. The concepts of white-collar and blue-collar, in this report, are used to represent the role of global and local decision levels in the previously proposed control architectures respectively. Both terms came into common use in the 20th century to mean these who perform office/administrative level and clerical work within manufacturing industry respectively (Wroblewski M.T., 2019). Furthermore, the work in white-collar is unstructured and non-routine whereas the work in blue-collar (also named as working class) is manual and/or operational with dirty hands.

³⁰The role of decision entities in the white-collar layer is not as significant as that of decision entities in blue-collar layer

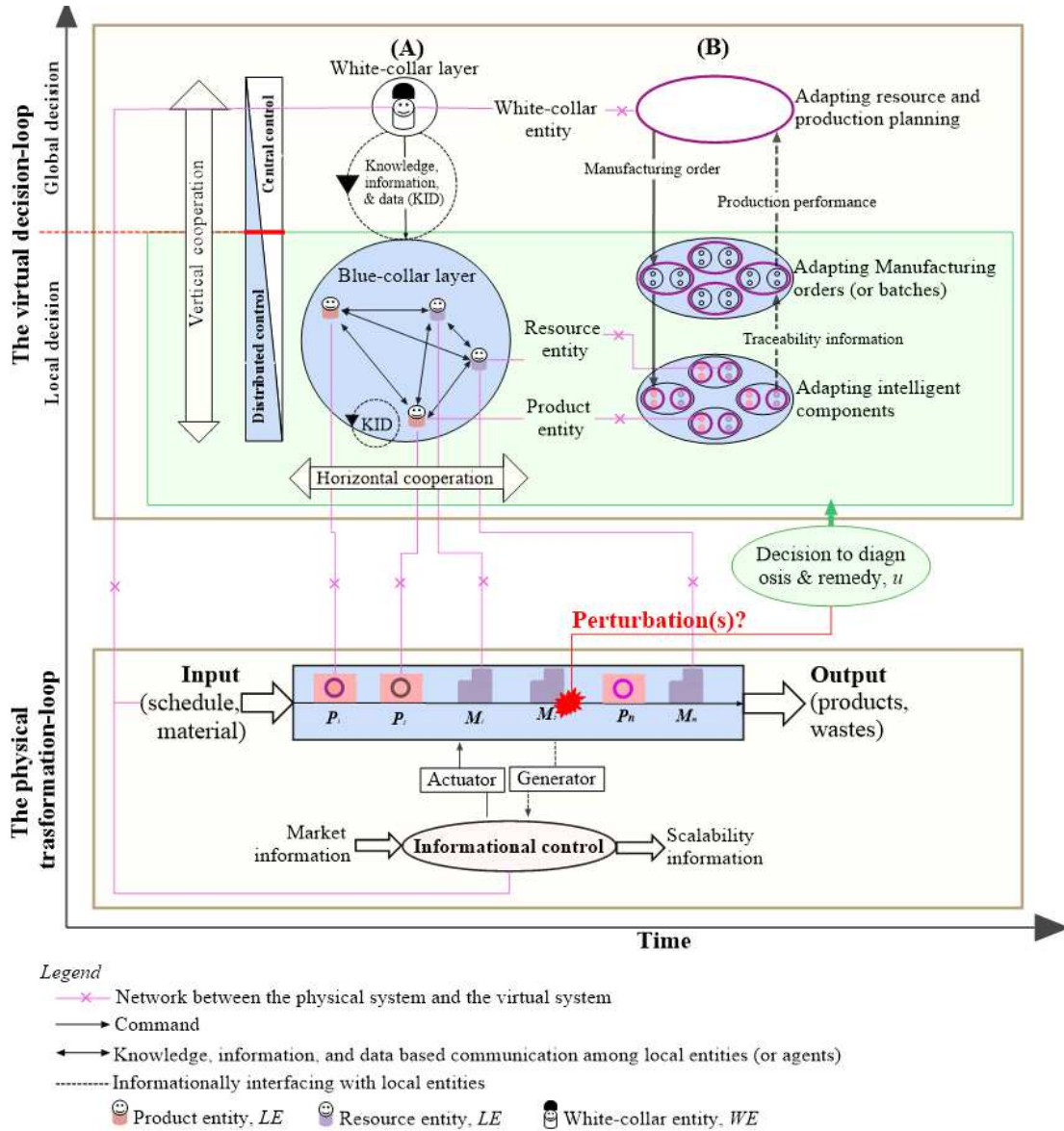


Fig. 3-2 The REcursive hybrID Control Architecture

(A) Organization of the control architecture (B) Aggregated decision levels

3.2.1 The decision entities

All the execution components within a manufacturing system are installed to perform the required actions proposed by decisional entities Jimenez et al. (2017). In REDCA, two groups of decision entities namely local decisional entities (LE) and white-collar decisional entities (WE) are introduced to perform cooperative participations for their common goal and/or objective. In order to make the required decision, all the decision entities consider different parameters, variables, communication protocols etc.

White-collar entities engaged to prepare production and resource plan through Human-machine interfacing, Enterprise resource planning etc. Once demand management and resource plan are developed at ERP level, a manufacturing order (*MO*), through MPS, is dispatched using these entities to resources for executing the required operations. These entities are also aimed to declare external changes such as rush order such that if this external change is requested after an online-execution, the locally impacted entities shift their behavior to this white-collar entity in order to declare this change. This proves that white-collar entities decided both in offline and online sub-systems.

On the other hand, the layer-based organization of REDCA differs from that of Pollux on the role of these white-collar entities. The decision entities in coordinating layer of Pollux, which are parallel to the white-collar entities in REDCA, are responsible for global production optimization by hosting a predictive decision-making technique to guide the achievements of goals while the white-collar entities in REDCA are used for updating and interfacing role only. That is, once perturbation is occurred, the global decision entity in Pollux has equal decision level with the local entities where as in REDCA, the white-collar entities becomes out of the decision-loop after the perturbation is occurred even though there is dynamic role assignment of decision entities within the blue-collar layer.

Local entities dedicated to perform the required execution ordered by white-collar entities and to rule a decision-making process at the time of local perturbation occurrence. Each local entity has its own recursive control kernel³¹ and this recursiveness continues until the last operation of a product is completed in a given time horizon (Herrera et al. 2011; Bae S. 2019). After perturbation occurred, the decision-making process begins by sensing the perturbed area through sensors (RFID technology for example) and it uses knowledge, information, and data of the affected decision entities. Then after, they cooperate and diagnose the unexpected event occurred, either through changing their behavior to white-collar entity (the first recursion within the distributed reactive control mode) or acting as local entities. This behavioral change of locally impacted decisional entities in to white-collar entity makes REDCA dynamic. This continues until the unexpected event is no longer a limiting factor. Likewise, the affected local entities do not look to be helped either from entities in the white-collar layer or local entities which are not part of the unexpected event. Product entities and resource entities constitute this local entity with strong horizontal cooperation between them.

³¹Kernel is local regulation and local management

Product entities (P_i) constructed with different product types for different manufacturing orders, are the leading elements within the entire decision loop shown in Fig. 3-2. The aggregated decision levels in Fig. 3-2(B) are instantiated using knowledge, information, and data provided by these entities. Depending on the required communication with, they organize a production process for their different operations by considering their current state and broadcasting their intention to neighbor product entities or available resource entities. They have informational part to assure they are intelligent enough; the first cyber-physical system. This informational part, which has memory on which data and operational requirements are stored, is in charge of transmission of manufacturing orders to resources, reasoning capabilities, and validation of key performance indicators. Moreover, it is a required function to create and manage other neighbor entities. Likewise, it is this part that ensures the product entity to compute its optimal completion time considering available constraints. It also accepts/refuses a configuration change using its environmental knowledge and asks/accepts/refuses to swap its configuration with neighbor product entities.

Resource entities (R_r) configured with hardware parts (machines, conveyer, routers, sensors, production controllers etc.) and control parts in order to provide recursive services for product entities. The hardware part executes operations requested by product entities. The control part monitors the hardware operations by hosting attributes such as adaptive capacity and manages tasks assigned to its hardware part. It is shown in the physical transformation-loop of Fig. 3-2 that the sensors and actuators follow commands received from product entities (at this instance, the behavior of the product entity is acting as white-collar entity) to perform the required process. Once the white-collar entities pass MPS to these entities for executing, they return inform their availability to execute the required operation. This confirms their assistance to product entities in coordinating the entire system.

3.2.2 Structural organization of the control architecture

The structural arrangement of REDCA is organized into two layers namely white-collar layer and blue-collar layer.

White-collar layer is an offline sub-system to coordinate a system towards global performance by preparing production and resource plan through mixed integer programming. Once demand management and resource planning are developed at ERP level, a manufacturing order (MO) is dispatched to a shop floor by white-collar entities (WE) in this layer at periodical time. The manufacturing orders in Fig. 3-2(B) have set of attributes such as

order number, due date, quantity etc. and each manufacturing order has its own product classifications defined by its product's current state. Meanwhile, this layer informationally interface with the blue-collar layer while the local entities in the blue-collar layer generate their executed output; the production performance declaration in Fig. 3-2(B).

Blue-collar layer is an online sub-system for operational execution of the manufacturing orders dispatched by white-collar layer and for controlling the available executing resources. It is represented by local entities (*LE*) responsible for production performance declaration, traceability information, and agile reactivity to unexpected events occurred in addition to the execution control. Decision entities in this layer have dynamic behavior to assign a role among each other (for example local entities may change their behavior in to white-collar entities within this layer until the required perturbation is addressed) and this continues until the last operation of a product is completed in a given time horizon. However, after the perturbed sub-system is adjusted, the entities in this layer generate the improved system to the white-collar layer for their informational purpose only.

3.3 Applicability of the proposed interaction approach

The human-inspired interaction approach is proposed to solve the problem of MAS coordination through negotiation and consensus algorithms. In order to verify and validate this coordination problem, two scaled-down versions of industrial contexts from real Chip wood and Carpentry factories are chosen and formulated in this section. Sub-section 3.3.1 presents the first coordination problem using negotiation algorithm and sub-section 3.3.2 illustrates the second coordination problem using consensus algorithm. In both problems, it is considered that n manufacturing orders MO_1, MO_2, \dots, MO_n each with their own product types P_i and due date of each manufacturing order dd_1, dd_2, \dots, dd_n are received by a shop floor. Meanwhile, product lateness (earliness or tardiness), machine utilization rate, informational state, waiting time, queuing time, and makespan are used as key production performance indicators to formalize the coordination problems.

For convenience of description; indexes, variables, and parameters used in both control problems are listed as follow.

Indexes and notations

i, j	Product index
m	Machine index
r	Resource index
o	Manufacturing order or batch index
k	Operation index
n	Number of parts index (e.g., number of decision entities, machines, manufacturing orders, etc.)
N	Set of parts (e.g., neighbor agents/entities) with $n \in N$
P_i	product i to be processed on set of machines M
i_k	k^{th} operation of product i
MO_o	o^{th} manufacturing order to be processed on set of machines M
\widetilde{MO}_o	Rush order (manufacturing order after it is rushed)
la	Leader/last (or p_leader) agent of manufacturing order MO_o
\widetilde{la}	Leader/last (or p_leader) agent of rush order
Φ_s	Decision node (router) within a system ³²
CR	Critical ratio index
t_e	Time where event-based disturbance or decision change happens

Variables

L_i	Lateness of product i
$Tard_i$	Tardiness of product i
C_i	Estimated completion time of product i
C_o	Estimated completion time of manufacturing order o
W	Waiting time of product i within a system
U	Utilization rate of machine m
C_{\max}	Makespan

³²System(S) defines an entire working environment (including machines, conveyors, routers etc.) of a shop floor. It considers that all decision components are interconnected and interdependent

Parameters

R	Set of resources within a system
M	Set of machines within a system
M_{ki}	Set of machines which perform operations k of product i
MO	Set of manufacturing orders
K_{io}	Set of operations of product i of manufacturing order o
$v_{k(m)}$	Intention ³³ of product i on machine m
C_i^*	Lower bound or optimal completion time of product i
C_o^*	Optimal completion time of manufacturing order o
tt_{km}	Shortest travel time of product i to machine m
a_{km}	Arrival time of product i to machine m
λ	Arrival rate of product i to machine m (e.g., speed of product holder such as conveyor)
q_{km}	Estimated queuing time of product i in a queue before its operation k on machine m
q_n	Number of products in a queue (queue size)
p_{km}	Processing time of product i on machine m
e_{km}	Release/exit time of product i from machine m
ε	Exit/release rate of product i from machine m
w_{im}	Estimated waiting time of product i after its processing on machine m where $w_{im} \in W$
CR_i	Priority of product i on neighbor products j , $\forall j \in MO_o$
ct_i	Current time of product i to estimate its completion time C_i
lt_i	Lead time remained a product i normally takes to completion. It includes setup, processing, travel, and queuing times.
A_m^*	Actual working time used by machine m
A_m	Maximum available time of machine m
s_{im}	Estimated setup time of product i on machine m for every switch of n products

³³Intention is a plan, which a product entity is committed on, that contains list of actions to be performed on upcoming machine (Verstraete et al., 2008)

dd_i	Due date of leader product i of manufacturing order o
Δdd_i	Change of due date of leader product i of manufacturing order o
dd_o	Due date of manufacturing order o
ℓ_{ki}^d	Dynamic looping (buffering) time of product i
ℓ_{ki}^s	Static looping (buffering) time of product i
η_{ki}	Number of looping on buffering zone of product i
x_i	Informational/convergence state of agent i at time t
x_j	Informational/convergence state of neighbor agent at time t , where $j \in N_i$
$u(t)$	Time evolved control action required to be used by an agent
γ	Variability of interest (or observed error) among neighbor product agents
d	A common value that all competitive product agents/entities i, j needs to converge
\mathbb{R}_+	Non-negative rational number
\mathbb{Z}_+	Non-negative integer number
α	A value that indicates the availability of a machine m for processing operation k of product i
β_1	Threshold to measure the supervisee level of product entities by machine entity
β_2	Threshold to measure a time evolved control action $u(t)$
y_{im}	Binary variable to indicate availability of machines

3.3.1 Industrial context-I: A Negotiation-based reactive control approach (NRC) to deal with material build-up problem

It is presented in Fig. I-2, if unexpected system disturbance that significantly impact a master production schedule (MPS) has occurred, it generates non-value adding activities such as queuing and moving times. Heuristic decision is expected right after such disturbance to deal with variabilities and hence to save the planned master schedule. In this case, the big issue is to make decisional entities defining best priority-based routing and/or sequencing in order to behave in a sense that the whole system stays globally near optimal. Empowering decisional entities to decide cooperatively and making all key performance indicators under the control of these entities would be best option. In order to substantiate such decision, a physical system (which is a scaled-down version of lacquering and polishing shop floor from carpentry³⁴ factory) with four chronological activities namely cutting (resizing), drilling, sanding, and coating is considered, Fig. 3-3.

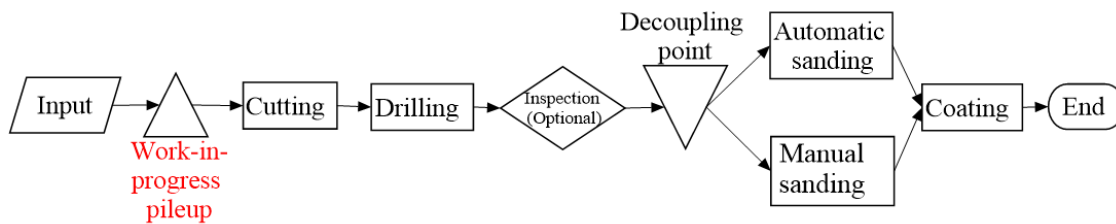


Fig. 3-3 Flow chart of a scaled-down shop floor

For experimental convenience, these activities are modeled using the TRACILOGIS platform (which is composed of four intelligent machines to execute different activities). Meanwhile, it is considered that a planning department of the factory prepares a weekly predictive schedule for each manufacturing order the shop floor received. If unexpected event/s is not noticed, this predictive schedule realizes the full completion of all orders with best rewards. However, it is taken that the physical system is suffering by work-in-progress (WIP) build-up, caused by non-optimal sequencing, between a machine and its upper stream. This event can happen either at the beginning, centre, or at the end of a launched system; in this context it is considered to happen at the beginning and at the center. Accepting this WIP build-up could trade-off throughput and machine utilization if an immediate action is not steadily taken. This demands an efficient production system with a pull³⁵ system. In the

³⁴See the first serial number in Table I-1 for the detail of this shop floor

³⁵According to APPICS, a Pull System is a production or service process which is designed to deliver WIP goods or services as they are required by a customer (or within the production process, when required by the next workspace).

traditional lean manufacturing system, the pull system was usually implemented with the aid of kanban card system. However, this kanban card system uses physical kanban cards attached with the WIP materials or containers that hold the WIP materials. This makes the kanban card system a one directional information flow which does not give empowerment to all components in upper stream and downstream workspaces. Thus, as stated earlier, empowering the decision components in upper stream and downstream workspaces to decide smartly and cooperatively in bi-directional communication approach would be best option. How? The following paragraphs answer this question by developing a negotiation algorithm.

Formalizing the Industrial problem after a central schedule (MPS) is dispatched by white-collar entities to a shop floor, Fig. 3-2(a), two groups of decision-making entities (or agents) namely product entities and resource entities are used to formalize the control problem. Meanwhile, product lateness, machine utilization rate, and makespan as key production performance indicators are used to formalize this problem. The two groups of decisional entities are engaged in processing a knowledge that contains information and data on how to execute a certain process on a certain machine.

Explicitly, product entities (or tasks for convenience) are central and active elements within the decision loop and they store process and knowledge needed to ensure the correct execution within a system. This cognitive ability allows them to make the required decision easier. Meanwhile, each product is in charge of transferring customer orders (and/or manufacturing orders) to machines and needs to keep its due date by completing its route as early as possible. Hence, they frequently compute their completion time using Eq. (3-1) with an objective to minimize at least their waiting time while they are in queue. This gives the completion time of each manufacturing order presented in Eq. (3-2). Subsequently, these two models help the products to evaluate the impact of unexpected event, if occurred, on MPS by computing makespan using the model specified in Eq. (3-3) subjected to constraints 3-3(a) to 3-3(g). See Gupta and Boyd (2008) and Mabin, V. (2010) for theory of constraints

$$C_i = ct_i + \sum_{k=1}^n \sum_{m \in M} y_{im} \bullet (s_{im} + p_{km} + tt_{km} + q_{km}), \quad \forall i \in MO_o \quad (3-1)$$

This gives $L_i = C_i - dd_i, \quad \forall i \in MO_o$

$$C_o = \max_{i=1}^n (C_i) \quad (3-2)$$

$$C_{\max} = \max_{o \in MO} (C_o) \quad (3-3)$$

With an objective to minimize C_{max}

Subjected to

$$(1) C_{ik} \geq C_{i(k-1)} + p_{km}, \quad \forall i \in MO_o, m \in M, k \in K_{io} \quad (3-3(a))$$

$$(2) q_n \leq (\lambda^* q_{km}), \quad \forall m \in M, k \in K_{io} \quad (3-3(b))$$

$$(3) Tard_i \leq \max(L_i, o), \quad \forall i \in MO_o \quad (3-3(c))$$

$$(4) \sum_{o \in MO} o_m = 1 \Leftrightarrow \sum_{i \in MO_o} i_m = 1, \quad \forall m \in M \quad (3-3(d))$$

$$(5) \{C_i, C_{max}, L_i, U\} \geq 0 \quad (3-3(e))$$

$$(6) y_{im} \in \{0,1\} \quad (3-3(f))$$

$$(7) \sum_{m \in M_{ki}} y_{im} = 1, \quad \forall k \in K_{io} \& i \in MO_o \quad (3-3(g))$$

Constraint (3-3(a)) indicates precedence constraint such that in order a product i to be processed in next machine m , it must first be processed in machine $m-1$. Constraint (3-3(b)) states the effect of high arrival rate of products on buffer size that each product entity has to reduce based on the supervision of the machine that they are approaching. Constraint (3-3(c)) presents no product tardiness if its completion time ends before its due date. Constraint (3-3(d)) ensures only one job and/or product is executed on one machine at a time. Constraint (3-3(e)) presents the positivity of variables used as key performance indicators. Constraint (3-3(f)) presents a binary variable which is set to 1 if product i is processed on machine $m \in M_{ki}$, 0 otherwise. Finally, constraint (3-3(g)) presents the requirement of m machines to perform k operations of product i .

On the other hand, the machine entity/entities offer production capacity and functionality in order to execute intentions broadcasted by product entities. They control the timing and type of execution process through their informational part. During the execution process, machines must be fully utilized in order to return their investment cost before its depreciation time. This requires understanding the interdependence of the machine's investment cost and its utilizability and hence such dependency results in trade-offs between investment and usability costs (Vander Veen and Jordan, 1989). Thus, part allocation, production cycles, and setup times of the machine must be managed so that the required utilization decision would always be higher. This hypothesis triggers the machine entities to calculate their utilization rate (U) for every event occurred using the model presented in Eq. (3-4).

$$U = \frac{A_m^*}{A_m} * 100\% \quad (3-4)$$

According to APICS, utilization rate measures how intensively a machine is being used and it is expressed as percentage of the amount of machine's available time required for productive and billable execution process. Meanwhile, this machine entity serves as supervisor (or white-collar entity in Fig. 3-2) to select set of products from set of candidates, based on a defined parameter (critical ratio in this context), approaching to it. This is the first recursion. This helps the machine to keep a defined queue size (or to satisfy constraint 3-3(b) around its workspace).

Designing the negotiation algorithm to create reduced WIP inventory, equalizing the variability of product's arrival rate with service rate of *ready to work machines* is necessary. This requirement can be achieved by making multistage negotiation among decisional entities and by using cooperation protocols that stabilize the specified variability which in turn results best priority-based sequential manufacturing process. This confirms that multi-stage negotiation between intelligent decision entities becomes one of the fundamental issues of research in multi-agent manufacturing systems. A negotiation process aims at modifying local plan of agents through structured exchange of relevant knowledge in order to achieve agreement among the multi-agent. Likewise, effective negotiation and cooperation are considered as one of the real time decision-making functions and optimization mechanisms to improve global performance while keeping their heterarchical control autonomy (Rey G.Z. et al., 2014). Rey G.Z. et al. (2014) have also presented that the problem of myopic behaviors of entities can be addressed through negotiation protocols such as contract-net, product driven approach etc. To achieve these objectives, agents/entities must fulfill the following negotiation characteristics (Jennings et al., 2001):

- *Make proposals*: all entities should be clear about the set of beliefs each entity involved in the negotiation process
- *Counter-proposals*: entities must be able to make contact with potential negotiators and recognize the type of *offer* in progress
- *Accept (commit to provide service) or reject proposals*: agents require both a means to evaluate the relative value to the agent of different offers made during negotiation and a means to evaluate the relative value to other agents of offers the agent may potentially make,

- *Generate arguments in support of their adopted stance on some subjects*: the ability to exchange constraints on acceptable offers and to reason with these constraints may make negotiation more efficient

And to satisfy these characteristics, negotiators could use different negotiation protocols³⁶ and negotiation strategy³⁷. From the different negotiation protocols such as contract-net by Smith (1980), product driven approach by Pannequin et al. (2009), market-like approach by Lin and Solberg (1992), argumentation approach etc., the Contract Net Protocol is a well-known for decentralized task allocation. An agent is suspected of being in a failure state if it is not able to answer the messages which are sent to it and consequently, failures in an agent are detected using a supervisor system introduced in each agent. This system informs the other agents about these failures. According to this premises, Amgoud et al. (2005) for example have proposed a formal negotiation handling protocol between many agents ($i & j \in n, n \geq 2$), see Fig. 3-4. An agent having to discuss several offers, the protocol is supposed to be run as many times as there are non-discussed offers such that a common agreement is still not found.

³⁶Protocol defines a common rules among decision entities with a set of norms that represents constraints for proposals that these entities can do (Madureira et al., 2014)

³⁷Negotiation strategy is a decision-making model (pull or push) and possible actions that negotiators employ in order to achieve their goal in line with negotiation protocol (Governatori et al., 2001)

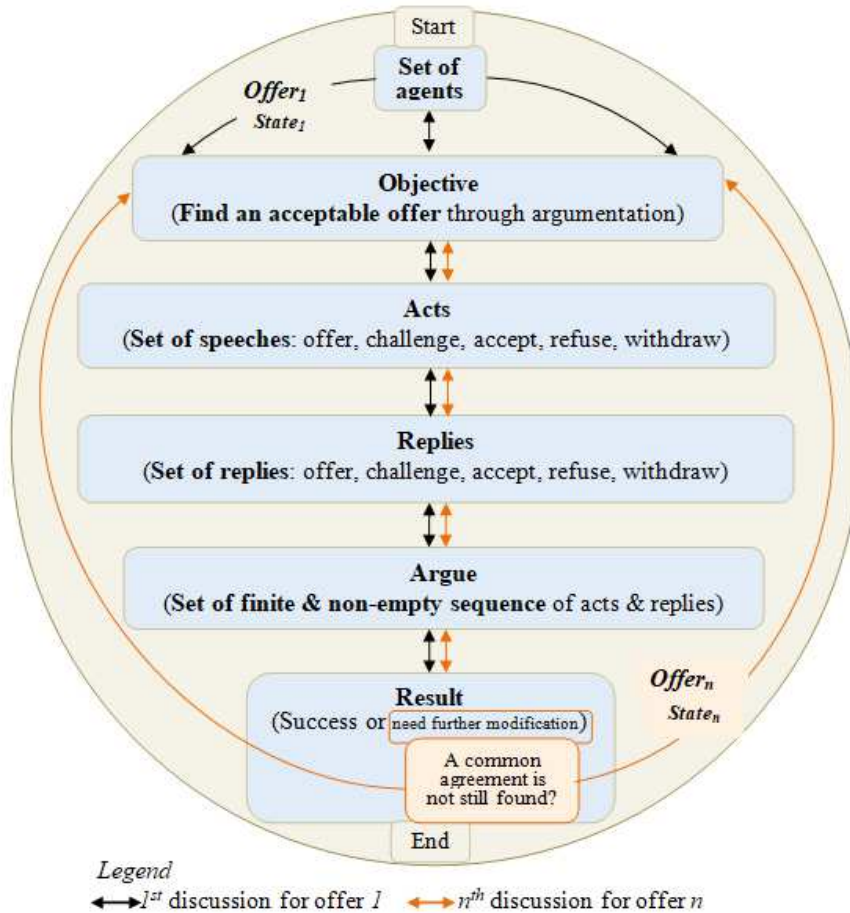


Fig. 3-4 A contract-net type negotiation procedure (adapted from Amgoud et al., 2005)

As part of such negotiation procedure, in order to design negotiation algorithm for the control problem stated earlier, the n manufacturing orders (each representing a specific customer order) are received based on their sequence of arrival $\{(MO_n) \prec, \dots, \prec (MO_2) \prec (MO_1)\}$ such that MO_1 is received first, MO_2 , and MO_n followed. Each manufacturing order is set to have its own completion time $\{C_2 \prec C_1 \prec, \dots, \prec C_o\}$. At this instant, the WIP build-up is made to happen and if the routing sheet starts without optimal operational sequence (that is, if it used *First-In-First-Out* production principle), it will lead to high work-in-progress inventory within the workspace of a bottleneck³⁸ machine and hence higher makespan. Thus, the customer orders received over a specified period of time must be combined into a production plan and dispatched together by their sequence of shortest completion time; $(MO_n) \prec, \dots, \prec (MO_1) \prec (MO_2)$. Products of MO_2 precede products of MO_1 and then products of MO_n follow on the bottleneck machine. This limits a work-in-progress size in order to create smooth and adaptable system with minimal product's queuing time, constraint 3-3(b).

³⁸A machine with highest product's operation processing time

The bottleneck machine is the only machine required to be reschedule in order to create smooth and adaptable system with minimal buffer size, constraint (3-3(b)).

To satisfy the new routing sheet, the system is made to pursue negotiation-based reactive control approach using agent-based interaction protocol. It uses *update status and routing* principle to show that entities continuously interact to optimize the routing sheet. Its control architecture is presented in Fig. 3-5 maintained by explanation of each step revealed on the architecture. Meanwhile, the interaction algorithm is supported with UML meta-models shown in Fig. 3-6 and Fig. 3-7. Know that the UML class diagram presented in Fig. 3-7 has been also used as a basic meta-model for the second context in sub-section 3.3.2. The directory service in the class diagram in Fig. 3-7 acts as a repository for the system providing information about active decision entities.

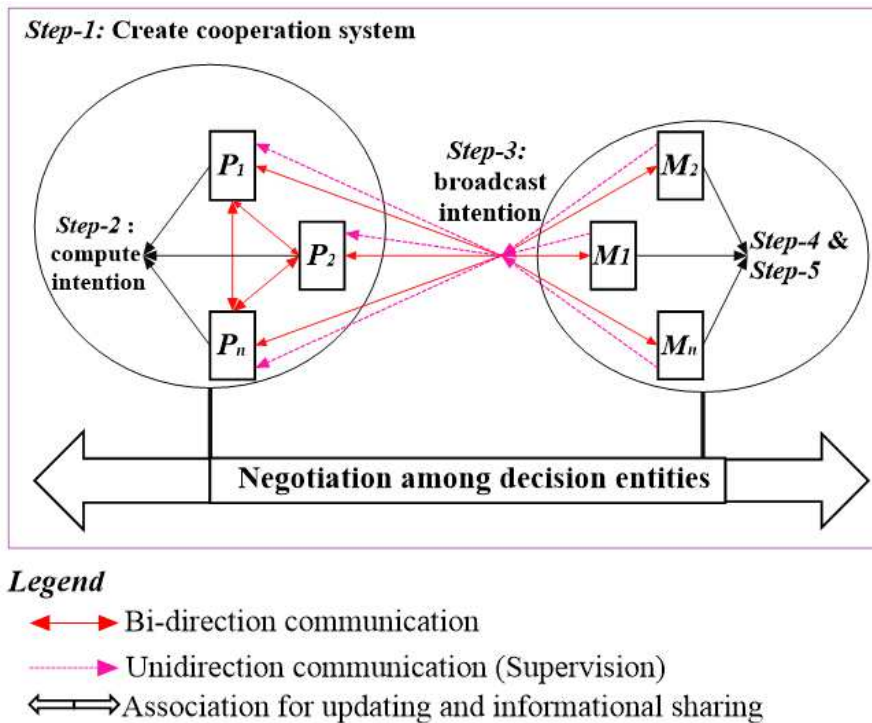


Fig. 3-5 The proposed Negotiation control architecture

Step-1

The local entities (the product and resource entities) create a cooperation system in order to calculate and estimate their current state and hence to set near optimal³⁹ performance. They

³⁹A phrase *near optimal* is used throughout this study to indicate that, as it is presented in Madureira et al. (2014), it is not possible to adopt optimal solutions because the resolution to optimality in an acceptable time for decision-makings is normally intractable and many problems in reality are so dynamic that when a solution is executed, the characteristics of the problem may changes and this could not be the optimal solution for the new problem.

negotiate on execution of operations needed to perform dispatched quantities of products of each manufacturing order. Products to broadcast their intention and machine to select and execute a product or set of products by comparing its utilizability with the products' intention

Step-2

The intention of every product of each manufacturing order is to arrive and process in the next machine as early as possible. Hence, each product computes its intention according to the model presented in Eq. (3-5) such that new product intention is a function of its previous intention and the intention of other neighbor entities.

$$v_{k(m)} = [a_{km} \ p_{km} \ e_{km}], \quad \forall m \in M, \ k \in K_{io} \quad (3-5)$$

Step-3

The computed products' intention is broadcasted into the created system that incorporates a machine that the product/s is approaching and neighbour products $j, \forall j \in MO_o$

Step-4

After each product entity broadcast its intention, machine entities are expected to being fully utilized in order to return the initial cost of the machines within short period of time. Taking this precondition, product prioritization has to be validated (or supervised) by machine entities through calculating their utilization rate U , Eq. (3-4), for every broadcasted intention of the product. As soon as machine entity computes its utilizability, it compares with the intentions broadcasted in order to accept if it does not affect its utilization rate. Otherwise, the machine entity queries products to revisit their intention so as to contest with its utilizability. This strategy helps product entities to balance an input buffer size, q_n , in front of the machine that they are approaching. For example, when last product of MO_l and first product of MO_n meet in decision node Φ_s , the intention of last product of MO_l is to precede the first product of MO_n as the due date of MO_l is less than that of MO_n . If this intention is not satisfied, it delays the entire completion time of all manufacturing orders and increases the non-utilizability of the machine. This step ensures that the intention of product entities is supervised by a bottleneck machine that they are approaching. This is because, machine smartly selects (or pulls in the context of pull production system) product candidates after it knew that the current product in process is released (or exited) from this machine. This directly represents the updating role of machine entity as white-collar entity in Fig. 3-2(a).

Step-5

So as to maximize the machine utilization, balance an input buffer for product's remaining operations, and hence minimize the makespan, decision entities choose a critical ratio (CR).

Critical ratio is mostly used dispatching rule to prioritize products within a shop floor (Wong et al., 2006). A machine entity request products to update their state-based on this critical ratio and product entities calculate their critical ratio (CR_i), for every request sent by the machine, according to the model given in Eq. (3-6).

$$CR_i = \frac{dd_i - ct_i}{lt_i} \quad (3-6)$$

The calculated values of this model also define the supervisee level of product entities by machine entities and this supervisee level is measured using a threshold, β_1

Such that $\left\{ \begin{array}{l} \text{if } \beta_1 \text{ is set to be close to zero (or } CR_i > 1); \text{ product is behind schedule (tardy)} \\ \text{and supervision is not required} \\ \text{if } \beta_1 \text{ is set to large (or } CR_i < 1); \text{ product is ahead of schedule (early)} \\ \text{and supervision is required} \end{array} \right.$

After this, with an intention to select a product (or set of products) with least critical ratio first, machine entity supervises products to arrive based on their critical ratio sequence. This balances the intention of products to complete their route within their sequence of due date and the utilization rate of machine. Meanwhile, after this critical ratio based sequence is launched, if products are still approaching to a machine early (or constraint (3-3(b))) is not satisfied), *step-6* follow.

Step-6

While the routing sheet execute following *step-1* to *step-5*, a machine might be incapacitated to perform according to the predefined negotiation protocol for different reasons (for example, breakdown). At this time, the machine entity (where the unexpected event originated) creates another state (called buffer size, constraint (3-3(b))) to begin another negotiation with the tasks (WIP products) that previously assigned to this machine. Hence, the machine inform the upcoming WIP products to wait somewhere in between their current position and the machine. This leads product entities to wait or buffer in their current location until the buffer size is reduced to a defined limit. The buffering ceases either when the *buffer size become less than the maximum limit* or the buffered product entities knew they are very tardy relative to their estimated completion time. Likewise, during the negotiation, decision entities (e.g., the constraint machine entity) may decide alone by rejecting the negotiated offer (or forgetting the negotiated offer) with their neighbor entities (e.g., product entities). At this time, product entities may obligate to cancel the negotiated offer as their estimated completion time is getting very tardy.

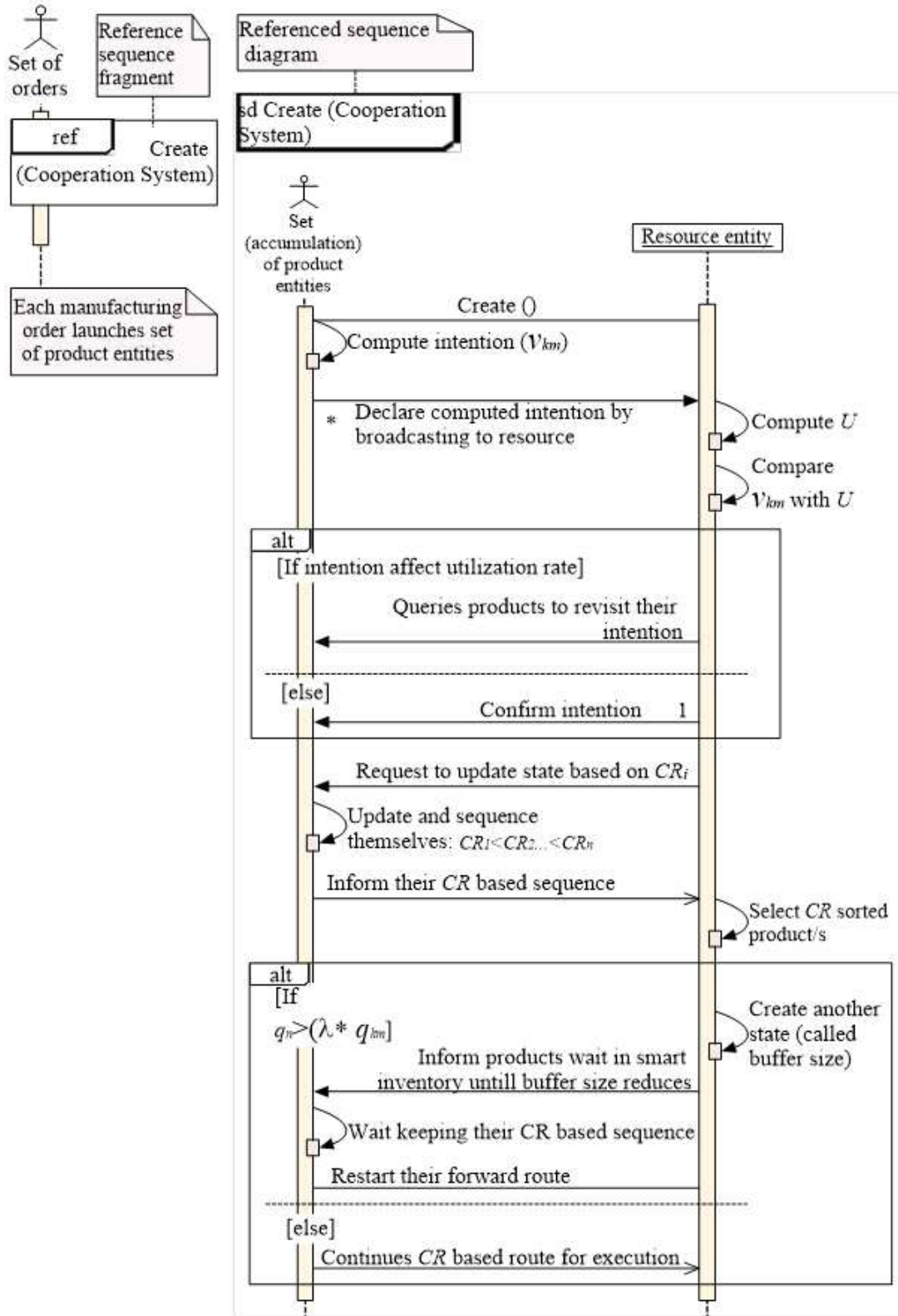


Fig. 3-6 UML sequence diagram for the proposed multi-agent negotiation

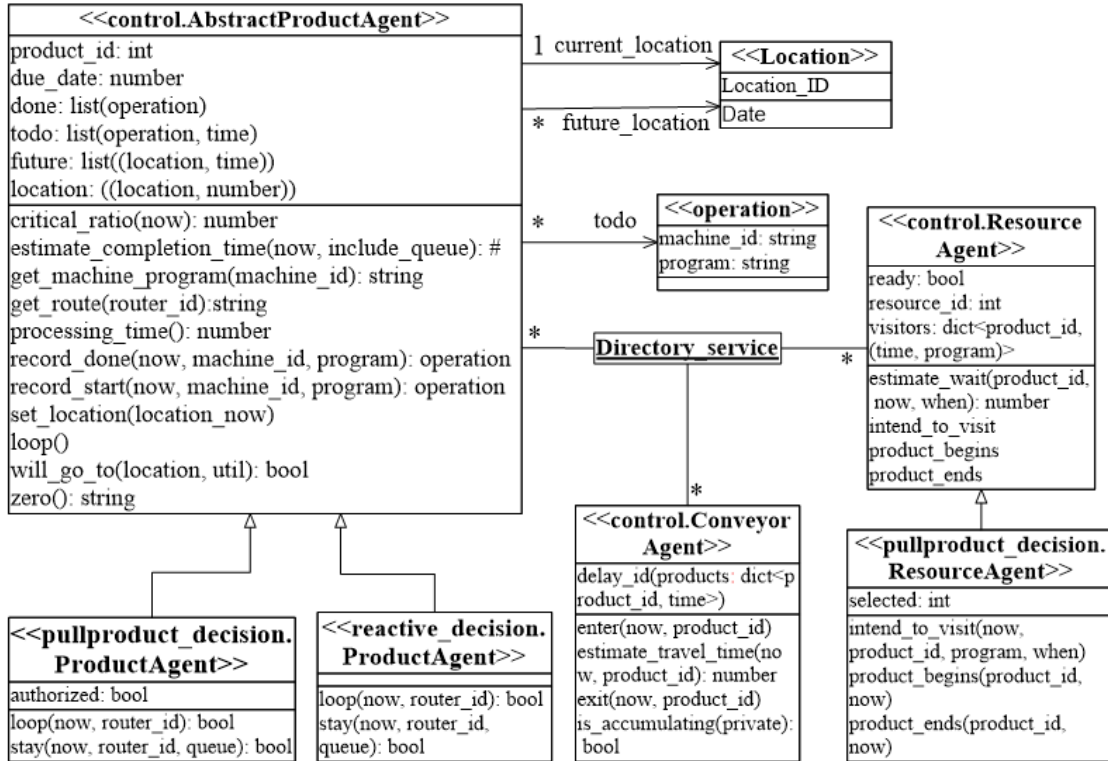
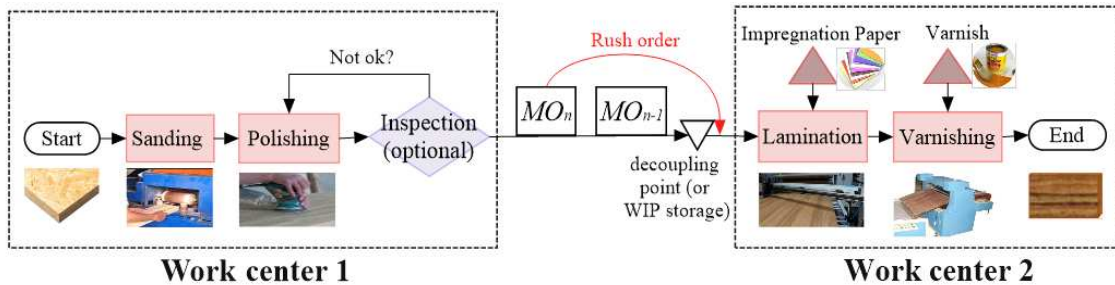


Fig. 3-7 UML class diagram meta-model for the developed multi-agent system negotiation

3.3.2 Industrial context-II: a Consensus-based decision-making algorithm (CoMM) to deal with rush order

Problem statement a make-to-order shop floor, Fig. 3-8, scaled-down version of lamination and impregnation shop floor (extracted from particle board manufacturing factory) is considered to set the problem statement under this context. Sanding of raw board, polishing and inspecting, lamination, and varnishing are its sequential operational activities. Its products are supplied to subsequent job shop or external customers to serve as raw materials for office furnitures, kitchen cabinets, computer desks etc. The shop floor has a capacity to produce different standard products for different customers by changing its setup. Meanwhile, the company’s planning department proposes a timely predictive schedule for each customer order (each with their own product types) the shop floor receives. If perturbation(s) is not noticed, a schedule prepared at the beginning of the execution process realizes the full completion of all orders received with best rewards. However, it most often suffers from many external and internal perturbations and the following two are set as the most common problems (a) Last minute’s rush order request due to high priority raised from external pressure (b) Extra buffer (or higher buffer size) at decoupling point due to the rush order. Rush order is an implicit customer’s priority request for his/her products to be delivered very quickly (Ehteshami et al., 1992; Wang and Chen, 2008). Literatures have treated rush order as a special disturbance that disrupts performance of an entire shop floor (Trzyna et al., 2012). Likewise, the high buffer size, which in turn results higher work-in-progress inventory, encounters because each product of each manufacturing order chooses a destination with shortest completion time scheduled at the central predictive level.



Legend

MO_{n-1} is a preceding manufacturing order and MO_n is its successor manufacturing order

Fig. 3-8 Flow chart of a scaled-down shop floor with its perturbation

Accepting such perturbations could trade-off manufacturing lead time as it leads to non-value adding activities presented in Fig. I-2(a). Thus, immediate to its happening, it must be inserted to current schedule by making optimal rearrangement of all decision entities and resources available within a system. Local and global decision entities have to work cooperatively in order to minimize myopic behavior of locally autonomous entities during this decision change.

Formalizing the industrial problem similar to the first context, two groups of decision entities namely the local entities and white-collar entities are considered to formalize this problem. They cooperate to sustain a Master Production Schedule (MPS) developed centrally and hence to minimize makespan. In order to update the cooperation & communication between these entities, the pyramidal inter-agent interface (or the decision kernel in Fig. 3-2(a)) is used by integrating the local entities with white-collar entities.

Product entities (P_i) they are the decision entities within the execution layer in Fig. 3-2(a), constructed with different product types and quantities for different manufacturing orders. As it is described in section 3.2, the informational part of these entities transmits manufacturing orders to machines by considering their reasoning capabilities. Moreover, they are the drivers of decision changes at the time of perturbation(s) if the decisional system is needed to stay either in a centralized or distributed control mode, Fig. 3-9. This is achieved by calculating and estimating their current state in order to match with the global mission. Consequently, each product entity computes its completion time C_i , Eq. (3-7), in order to minimize non-value adding activities in particular waiting time (W). This gives the completion time of each manufacturing order as computed in Eq. (3-8).

Hence, makespan is obtained from Eq. (3-8) and computed in Eq. (3-9).

$$C_i = ct_i + \sum_{k \in K_{i_o}} \sum_{m \in M_{k_i}} y_{im} \cdot (p_{km} + tt_{km}) + W, \quad \forall i \in MO_o \quad (3-7)$$

$$W = \sum_{k \in K_{i_o}} \sum_{m \in M_{k_i}} y_{im} \cdot (q_{km} + w_{im}), \quad \forall i \in MO_o$$

$$C_o = C_{la} = \max_{i \in MO_o} (C_i), \quad \forall o \in MO \quad (3-8)$$

$$C_{\max} = \max_{o \in MO} (C_o) \quad (3-9)$$

With an objective to minimize C_{\max}

Subjected to

$$(1) \text{ Constraints 3-3(a), 3-3(d), 3-3(f), and 3-3(g)} \quad (3-9(a))$$

$$(2) W_i \leq dd_i - (ct_i + \sum_{k \in K_{io}} (tt_{km} + p_{km})), \quad \forall i \in MO_o, m \in M_{ki} \quad (3-9(b))$$

$$(3) q_{km} \begin{cases} > 0, \text{ if } \lambda > \varepsilon, \quad \forall m \in M_{ki}, k \in K_{io} \\ \equiv 0, \text{ otherwise} \end{cases} \quad (3-9(c))$$

$$(4) w_{im} \begin{cases} > 0, \text{ if } q_{km} > 0, \quad \forall i \in MO_o, m \in M_{ki}, k \in K_{io} \\ \equiv 0, \text{ otherwise} \end{cases} \quad (3-9(d))$$

$$(5) Tard_i = \max(C_i - dd_i, 0) \Rightarrow Tard_o = \max_{o \in MO} (C_o - dd_o, 0) \quad (3-9(e))$$

$$(7) \{x_i, C_i, C_{\max}, W\} \geq 0 \quad (3-9(f))$$

Constraint (3-9(a)) indicates the constraints (3-3(a)), (3-3(d)), (3-3(f)), and (3-3(g)) in the WIP inventory control problem works for this control problem also. Constraints (3-9(b-d)) present a quick decision at decision node Φ_s by product entities to minimize their waiting time after occurrence of the rush order. Constraint (3-9(e)) also presents no product tardiness if its completion time ends before a due date. Constraint (3-9(f)) ensures that all variables used as key performance indicators are non-negative.

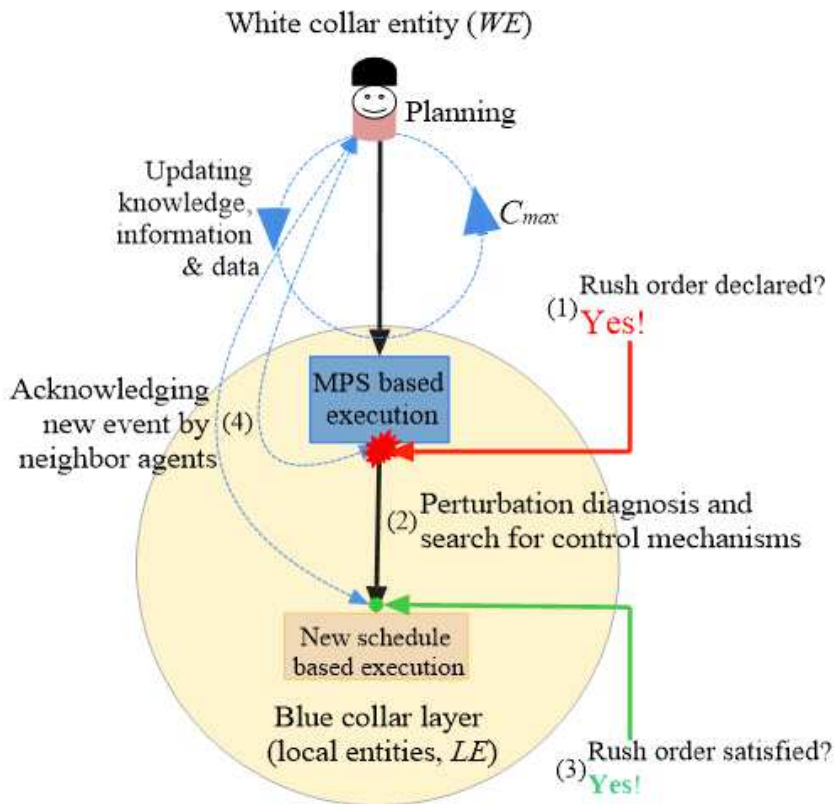


Fig. 3-9 MAS based cooperative decision architecture to satisfy the rush order

Resource entities (R_r) the other physical elements of the blue-collar layer in Fig. 3-2(a). The hardware part executes operations sent by product entities. In case of unpredictable failure of either machine, others have the capability to perform all operations left unprocessed. Meanwhile, each machine is responsible to perform additional operation $k+1$ for a single product of either manufacturing orders. Changeovers of production from one product type to another associates setup times and machines entities are expected to fully utilize their capacity. Taking this hypothesis, machines calculate their utilization rate (U) based on the model presented in Eq. (3-4). Product's prioritization has to be validated by recalculating this utilization rate for their every acceptance. This computation is necessary in order to process products if their arrival rate does not affect the computed utilization rate.

White-collar entities they are elements of the white-collar layer in Fig. 3-2(a) in order to facilitate a system towards global performance through mathematical optimization techniques. A central schedule is dispatched by these entities to be executed by local entities at the blue-collar layer. Meanwhile, the rush order is declared by these entities. In this context, a product entity with urgent priority structurally shifts to white-collar entity to declare the rush order. This is the first recursion. Then after, the product entity turned back to its local behavior in order to negotiate with the preceding product entities that are going to overtake their predetermined priority.

Designing the consensus algorithm as it is presented in Olfati-Saber and Murray (2004), consensus is an algorithmic societal procedure to negotiate among groups of agents i and j through communication edge E_n in order to reach an agreement for the best interest of the whole, see Fig. 3-10.

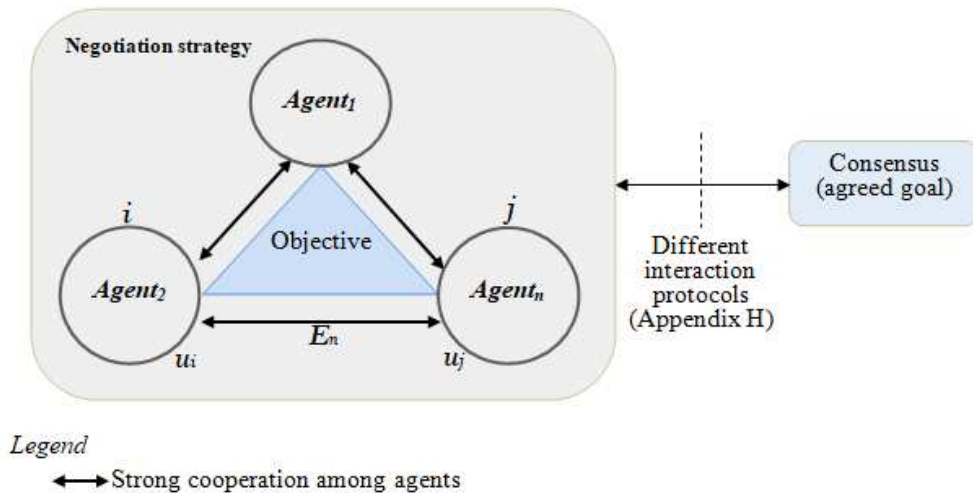


Fig. 3-10 Algorithmic procedure to reach consensus among multi-agent (adapted from Klimeš, 2014)

It pursues certain quantities of interest with or without predefined common orientation of individual agents by respecting the following properties.

- *Agreement*: all the correct processes decide the same value
- *Integrity*: each process decides at most once
- *Validity*: each decided value belongs to the set of proposed values
- *Termination*: all the correct processes eventually decide in finite time

In the context of these properties, Ren and Beard (2008) have hypothesized that to achieve the consensus, there must be a shared variable of interest called an *information state* as well as appropriate procedure for negotiating to reach consensus on the value of that variable called the *consensus algorithm*. This algorithm has proven its applicability in different networking scenarios and its applicability in the area of decentralized and reactive task assignment in manufacturing system is attracting the current research community.

In the context of such applicability, this consensus algorithm is aimed to satisfy a rush order at minimal recovery time and to maintain similar performance of planned schedule after this disruptive rush order is occurred. To provide such adaptability and ensure coherence to a global objective, it is the product entity that has to achieve this urgent priority request. They used a Leader-Follower based communication approach among the many MAS consensus features such as event-triggered, dynamic topology, non-linear etc. Leader-Follower consists of a group of leader entities with some amount of power to facilitate cooperation among a large group of their followers (Espejo Raul and Reyes Alfonso, 2011). They define goals and tasks for their followers on one global goal Dorri A. et al. (2018); the recursion. Meanwhile, in this Leader-Follower MAS feature, product entities with urgent priority not only recalculate their own schedule but stimulate others to do it according to the relevance to solve a global problem.

Preliminary of the consensus algorithm consider the n entities communicating for their common goal in Fig. 3-10. Their communication topology is represented using graph theory $G = (V_n, E_n)$ for n nodes;

Where V_n is the set of vertex and/or decision entities ranging from 1 to N ,

$E_n \subseteq V_n \times V_n$ is the set of communication edges ranging from 1 to $N-1$. All edges are said connected if decision entities $\{i, j\} \in E_n \Leftrightarrow \{j, i\} \in E_n$.

The communication topology shows both strong connection (as each node (entity i) has direct relationship with its neighborhood control (node j) and non-strong connection (as nodes are connected in uni-directional communication).

The graph theory also defines a Laplacian matrix L_n ; $L_n = D_n - A_n$

Where $A_n = \{a_{ij}\} \in R^{n \times n}$ is the adjacency matrix,

$D_n = \{d_i\} \in R^{n \times n}$ is the degree matrix with $d_i = \sum a_{ij}, \forall i \neq j$, and

a_{ij} is the (i,j) entry of the adjacency matrix at time t ,

Such that $\begin{cases} a_{ij} = 1 \Leftrightarrow a_{ji} = 1, \text{ connected, } \forall i \neq j, (i,j) \& (j,i) \in E_n \\ a_{ij} = 0 \Rightarrow a_{ji} = 0, \text{ otherwise; } \forall i = j, (i,j) \& (j,i) \in E_n \end{cases}$

The connectivity among decision entities during their communication is also proved by respecting interdependence of different performance measures that help them improve their adaptability to the rush order, Algorithm 3-1.

Algorithm 3-1: Dependence of entity's' state

n communicating agents (or products as decision entities), with $n \geq 2$ and $i \neq j$, must converge to a common goal by designing state x_i and considering interdependent performance measuring parameters, Eq. (3-10(a)). Likewise, an agent i has to share state of its neighboring agent j and use this state as an estimator of its future behavior for execution on a time varying communication topology, Eq. (3-10(b)).

$$u(t) = f(x_i, x_j), \quad \forall i \& j \in n, i \neq j \quad (3-10(a))$$

$$x_i(t+1) = g(x_i(t), u_i(t), u_j(t)), \quad \forall j \in N_i \quad (3-10(b))$$

This dependence would be attained if all decision entities are able to cooperate with each other and evaluate the impact of rush order on neighboring agents, Algorithm 3-2.

Algorithm 3-2: Pseudo code of the consensus priority protocol

Input: rush order for diagnosis (multistage negotiation)

Set n communicating entities with $n \geq 2$ and $i \neq j$

For $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, n$, **do**

If intention based conflict between i & j in decision node Φ_s ,

 Compute state x_i and x_j & exchange the computed state

 Compute $u(t)$, Eq. (3-11);

 { if β_2 is set to be closer to zero, execute an offer(priority) requested by entity i
 Else float the priority; as $u(t) \rightarrow 0$

End if

End for

Output: executed offer ($C_o = C_{la} = \arg \max_{i \in MO_o} \{C_i\}$, $\forall o \in MO$)

Each entity during the protocol in algorithm 3-2 considers three operational characteristics,

- ❖ The information that entity i has about the probabilistic occurrence of an event (the rush order in our case)
- ❖ State x_i that entity i would wish to achieve at minimal waiting time
- ❖ Intention that entity i has agreed to cooperate with others in order to minimize the impact of an event (rush order) on global performance (makespan in this context)

Hence, a consensus model which guarantees these characteristics and a convergence for collective decision by all entities is given as in Eq. (3-11) (Ren and Beard, 2008).

$$u(t) = -\left(\sum_{j \in N_i} a_{ij} [x_i(t) - x_j(t)] \right), \forall i, j \in n \quad (3-11)$$

In order to make Eq. (3-11) easier during its coding in Matlab or Python, it can be rewritten in matrix form shown in Eq. (3-12)

$$u(t) = -\{L_n(t)x(t)\} \quad (3-12)$$

Where $x = [x_1, x_2, \dots, x_n]^T$ is the information state of each entity i ,

$L_n(t) = [l_{ij}(t)] \in R^{n \times n}$ is non-symmetrical Laplacian matrix, and t is the time where agents act their decision

In Eq. (3-11), each entity i must create its own informational state x_i and then after reach consensus with neighbor entities $j \in N_i$ to approve the requested priority offer. To create the information state, each entity sets an event time as $x_i(0)$ where it initially meets with its

neighbor entities j (Ren & Beard, 2008). This promises that the common value is converged combination of all the initial informational states. After all the n entities have shared their interest, consensus is reached when the variability of interest (observed error, γ) among them approach towards zero, Eq. (3-13).

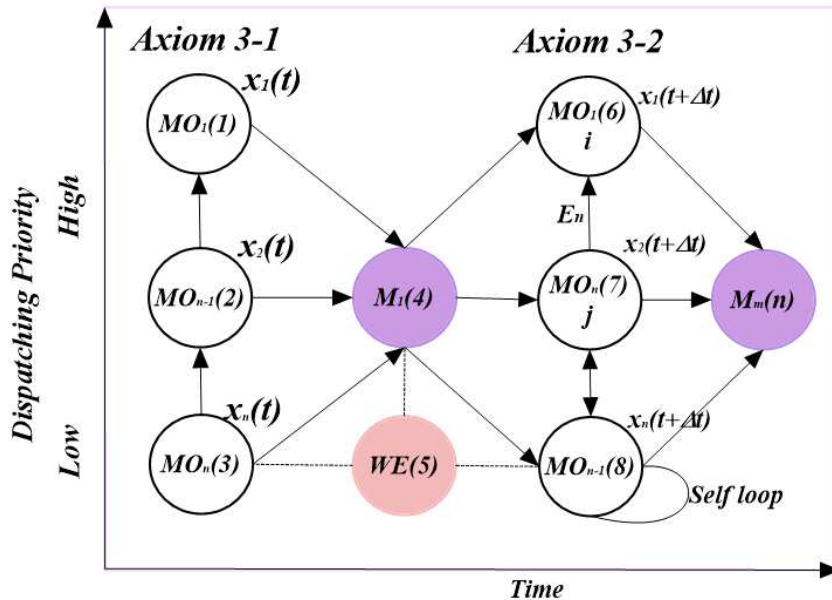
$$\gamma = |x_i(t) - x_j(t)| \rightarrow 0, \forall i, j \in n \quad (3-13)$$

For instance, as time evolves, the product entities presented earlier must calculate their state differences as per the model shown in Eq. (3-13) with a goal to compare with the state of neighbor product entities.

Design and adaptation of the informational state the n manufacturing orders with their communication and execution topology shown in Fig. 3-11 are considered to design and adapt the state x_i . Initially, manufacturing orders entered a system considering shortest processing time first (SPT) dispatch list presented in Axiom 3-1.

Axiom 3-1: Initial dispatch list

$(MO_n) < (MO_{n-1}) < (MO_{n-2}) \dots < (MO_1)$ such that the last operation k of product i of MO_{n-1} has to be completed in next machine m before the last operation k of product i of MO_n is started and same for others.



- Legend
- ↔ Bi-directional communication (strongly connected)
 - Uni directional communication (not strongly connected)
 - Updating and advising role

Fig. 3-11 Priority-based communication topology among different entities

While manufacturing orders route according to this dispatch list, it is taken that MO_n becomes rushed over MO_{n-1} at time $t_e = t + \Delta t$, Axiom 3-2. For convenience, MO_n is written as \widetilde{MO}_o in the upcoming paragraphs.

Axiom 3-2: Modified dispatch list

$(MO_{n-1}) < (MO_n) < (MO_{n-2}) \dots < (MO_1)$ such that the last operation k of product i of MO_n has to be completed in next machine m before the last operation k of product i of MO_{n-1} is started and same for others

In order to satisfy Axiom 3-2, the future impact of the rush order on the cost of waiting time of preceding manufacturing order has to be evaluated. Deciding when a leader agent of \widetilde{MO}_n would broadcast its intention to leader agent of MO_{n-1} sharing same decision node is the required consensus. This is necessary because if the leader agent of MO_{n-1} passes decision node Φ_s before that of the rush order, it increases its completion time without value addition. This happens because the white-collar entity (the product entity with highest priority in this context) would not allow this agent to proceed the next decision node Φ_{s+1} before that of the rush order. This is because, the rush order is declared by these white-collar entities. Consequently, each leader agent of each manufacturing order, only the strongly connected entities, competes to minimize its waiting time and hence minimizes its completion time presented in Eq. (3-7).

From Algorithm 3-1,

$$C_i = f(W) \text{ and } C_i = f(C_j), \forall (i \neq j) \in MO_o, k \in K_{io}, m \in M_{ki}$$

This gives the below two extreme values,

$$C_{la} = \arg \max_{i \in MO_o} \{C_i\}, \forall o \in MO \tag{3-14}$$

$$C_i^* = ct_i + \sum_{i \in MO_o} (tt_{km} + p_{km}) \ \& \ C_i^* < dd_i \tag{3-15}$$

Let,

- g_1 be a gap of this lower bound from the estimated completion time C_i , Fig. 3-12
- g_2 be a gap of this lower bound from the delivery margin dd_i , Fig. 3-12.
- And of course C_i^* be the lower bound, with $C_i^* < dd_i$, of C_i ,

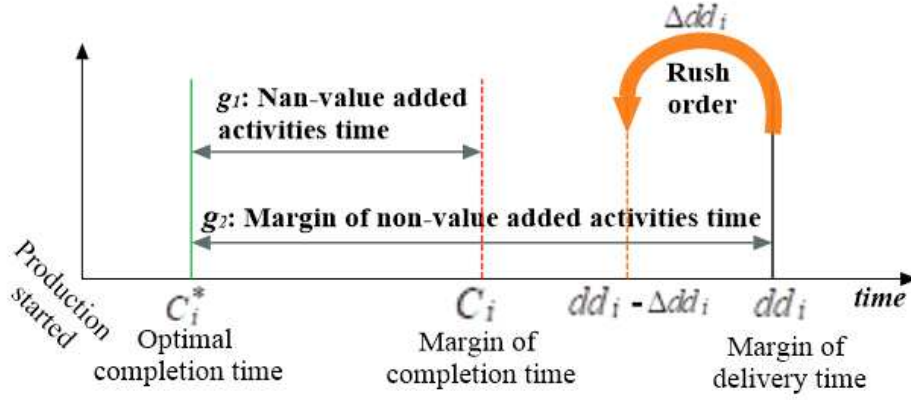


Fig. 3-12 System rescheduling to satisfy the rush order

The rushed order shifts down its initial due date to a new due date $dd_o - \Delta dd_o$. This implies that its completion time reduces at least by the processing time of the preceding manufacturing order in the next machine. On the contrary, the completion time of the preceding manufacturing order increases by the same amount. However, at normal scenario, the target of products of each manufacturing order is to minimize its waiting time (W) in order to achieve its best individual performance.

Thus, consensus allows ensuring a minimal tradeoff between agents' state which they want to improve their own performance. Consequently, the informational state that makes each agent to converge towards a common value (or invariant centroid state) $d \in \mathbb{R}_+ \setminus \{0\}$ so that Eq. (3-13) would be satisfied is presented as:

$$x_i(t) = \frac{g_1}{g_2} = \frac{(C_i - C_i^*)}{(dd_i - C_i^*)}, \forall i \in MO_o \Leftrightarrow x_{la}(t) = \frac{(C_{la} - C_{la}^*)}{(dd_{la} - C_{la}^*)} \quad (3-16)$$

Where $C_{la}^* = C_i^* + \alpha p_{kn}, \forall \alpha \in \mathbb{Z}_+ \setminus \{0\}$

This indicates that the completion time of each product agent of the rush order must close to the lower bound value so as to obtain $x_i \approx 0, \forall i$. To compare the impact of the calculated state, percentage of closeness towards zero could be used with an objective to reward lowest percentage values, section 4.4. For instance, as shown in the first option of Fig. 3-12 (where option 1 is g_1 and option 2 is $g_2 - g_1$), if the completion time could be able to approach towards C^* , the percentage of closeness declines to zero; the required consensus is achieved and a new routing sheet continues.

3.4 Synthesis and Discussion

To reduce the impact of perturbation(s) on manufacturing process, the proposed interaction approach is made to have structural and locally operational interacting features presented in Fig. 3-13. The local entities cooperate, considering their current state, to reconfigure (e.g., reschedule, resequence, recalculation etc.) a perturbed system and hence in order to nearly maintain the initially prepared execution plan. This constitutes a permissive⁴⁰ role of local entities in Jimenez et al. (2017). The interfacing role by the white-collar entities on the other hand is expressed by a concern given to statically launch a deterministic dispatch rules at the white-collar layer and declare an external events (rush order for example) within the blue-collar layer. Unless there was no an upward and downward informational interfacing between these two layers, local entities (mainly the product entities) were to concentrate only on their shortest path (or shortest completion time C_i) to exit from a system rather than concentrating on minimizing makespan. This confirms the modularity and flexibility of the proposed control approach for quickly adapting a perturbed routing sheet.

In the framework of this interaction, the human-inspired interaction approach together its implementation architecture is used to deal with two industrial contexts through two case studies namely negotiation-based reactive control approach and consensus-based control approach. Both control approaches are chosen over the other MAS features as they showed agent-oriented sociability, Isern et al. (2011), to control a multi-agent manufacturing system. In the negotiation-based control approach, product entities as local entities and machine entity that the product entities are approaching as local entity and as white-collar entity sociably negotiated to react and adapt a material build-up problem. The machine entity structurally shifts to supervision level from its local behavior in order to select (pull) products from set of candidates to reduce and/or prevent occurrence of high buffer size for example. This is clearly presented by defining queue size using critical ratio and machine utilization rate as decision-making parameters in the applicability implementation experiment presented in section 4.5. Once, this event is addressed, the machine shifts back to its local behavior to execute the tasks/operations assigned to him.

⁴⁰Coordinating entities delegate full decisional autonomy for local entities

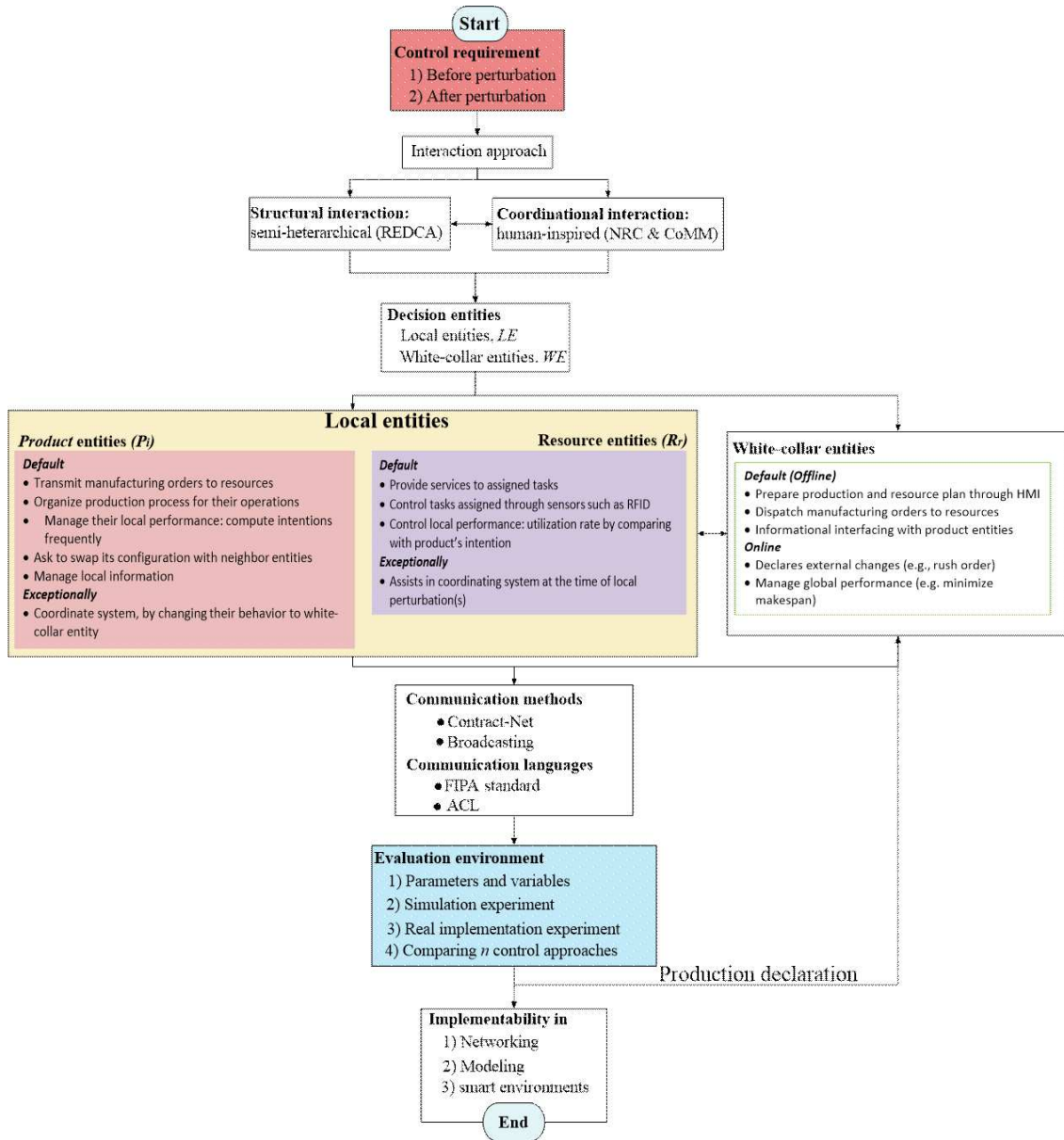


Fig. 3-13 Meta-model for the proposed interaction approaches

However, the local entities, as autonomous decision entities, may decide alone by rejecting the negotiated offer with their neighbor entities. This rejection may occur because the decision entities may negotiated on variables which benefit only for one sided entity rather than negotiating using a common state x_i . At this instant, it is necessary to asymptotically reach a final agreement using the common state x_i for the best interest of the whole; Fig. 3-10. This scenario is specified using leader-follower based consensus algorithm to satisfy a rush order. A product entity with urgent priority structurally shifts to white-collar entity (within the blue-collar layer) to declare the rush order and then after turned back to its

local behavior in order to negotiate with the other preceding product entities. This instance is clearly presented in the simulation experiment presented in section 4.4. The reactive communication methods used to address the perturbations sourced from either internal or external environments avoids aggregation of the decision entities either in the blue-collar layer or white-collar layer reported in Fig. 3-2(a). This verifies the recursive interaction-ability either among the local entities or between the white-collar entity and local entities.

Following the detail illustration of the proposed control architecture and decision-making algorithms in this chapter, the scaled-down case study (using a real implementation platform) is explained in the next chapter to verify and validate the algorithmic procedures instantiated herein.

CHAPTER FOUR

EXPERIEMENTAL RESULTS AND DISCUSSIONS: A TRACILOGIS platform case study

4.1 Introduction

After the constituent elements of the proposed control approaches have been modeled in sub-section 3.3.1 and sub-section 3.3.2, this chapter verifies and validates these control approaches using the TRACILOGIS test-bed platform. It also articulates the performance of the control approaches compared to pure reactive control approach (PRC). Pure reactive is chosen as comparative approach because agents in the TRACILOGIS platform were initially instantiated for this control approach (Zimmermann et al., 2018):

- a) Using Dijkstra algorithm⁴¹, product agents calculate their production range at every move the shortest path to their final destination and hence they simply send their intention to be executed as per their arrival sequence
- b) Product agents have no awareness on state of their neighbors
- c) Machine agents are passive which only wait for instructions from product agents

The proposed control approaches are verified on the basis of simulation study in section 4.3 and section 4.4. Constituent elements are coded using Python, MatLab, Java softwares and Matplot lib & Minitab are used as statistical manipulators based on mean values, maximum values, minimum values, etc. To validate the industrial applicability of the simulated algorithms, JADE based applicability implementation on the TRACILOGIS platform is presented in section 4.5. Finally, the last section summarises the verified and validated results relative to the importance of the proposed control approaches.

⁴¹ Dijkstra's algorithm (or Dijkstra's Shortest Path First algorithm) is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by Dijkstra, E. W. (1959)

4.2 General description of the TRACILOGIS platform

TRACILOGIS test-bed platform, Fig. 4-1, represents a flexible job shop and allows studying different types of identification, traceability, and control systems. Its operating system is composed of an extensive system of networks linking different actors such as:

- Sensors,
- Actuators to automata or even automata to computers,
- RFID (Radio Frequency IDentification) networks,
- PLC (Production Line Controller that manage all of the automaton actions) etc.

It postulates upper-stream work center (the transformation station in Fig. 4-2) and downstream work center (the assembly/disassembly station in Fig. 4-2) connected by buffering work center. Tests using this platform help to know the applicability of control systems to solve real industrial problems. These stations constitute the four automation zones, *Zone-A* to *Zone-D* in Fig. 4-2, each connected with PLC controlled conveyor technology.

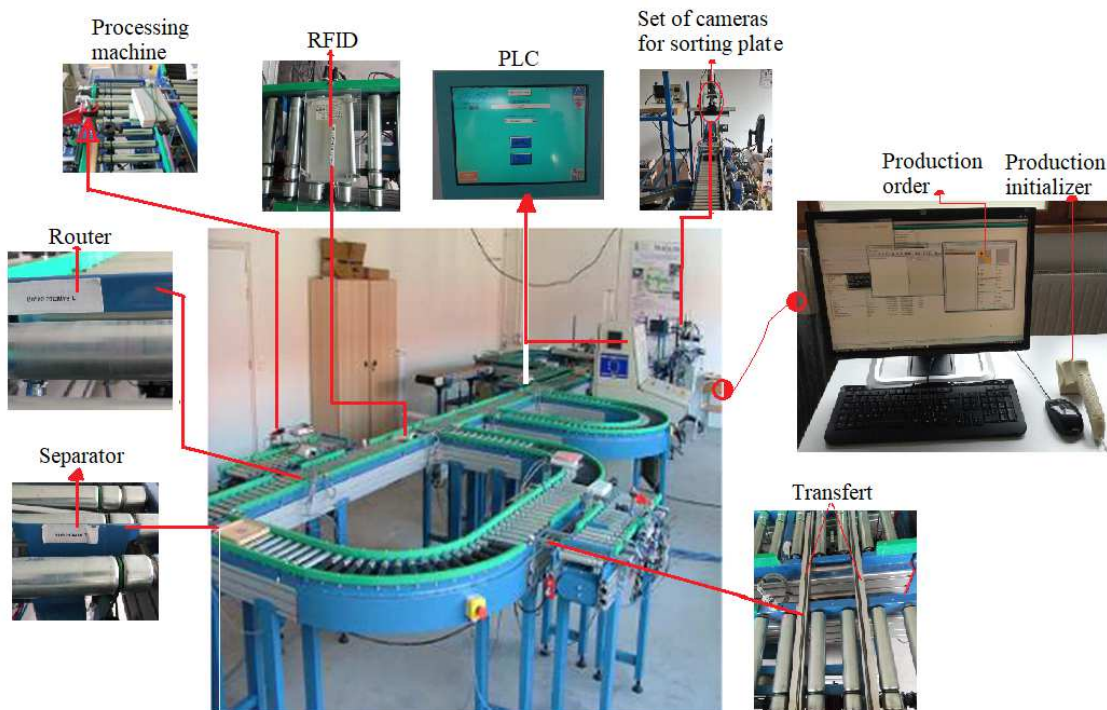


Fig. 4-1 General description of the TRACILOGIS platform

All zones are controlled by Siemens S7-300 type PLC and the entire system is equipped with an RS232 type RFID networks. Zone-A, configured with six RFID readers, contains two marking stations placed sequentially with capability to perform all operations by single station if one of them is failed. Zone-B, configured with two RFID readers, is installed to

satisfy unexpected plan change and buffer stock between Zone-A and Zone-D. It sets aside certain products and allows storing products as in a real system of production. Zone-C, configured with 11 RFID readers and four sorting stations (that sorts colored materials, required to be assembled, using linear camera). Zone-D, configured with six RFID readers, assembles the sorted and RFID tagged plates and tablets on the RFID tagged palet/support based on production order declared by a production initializer (or white-collar entities in the implementation architecture).

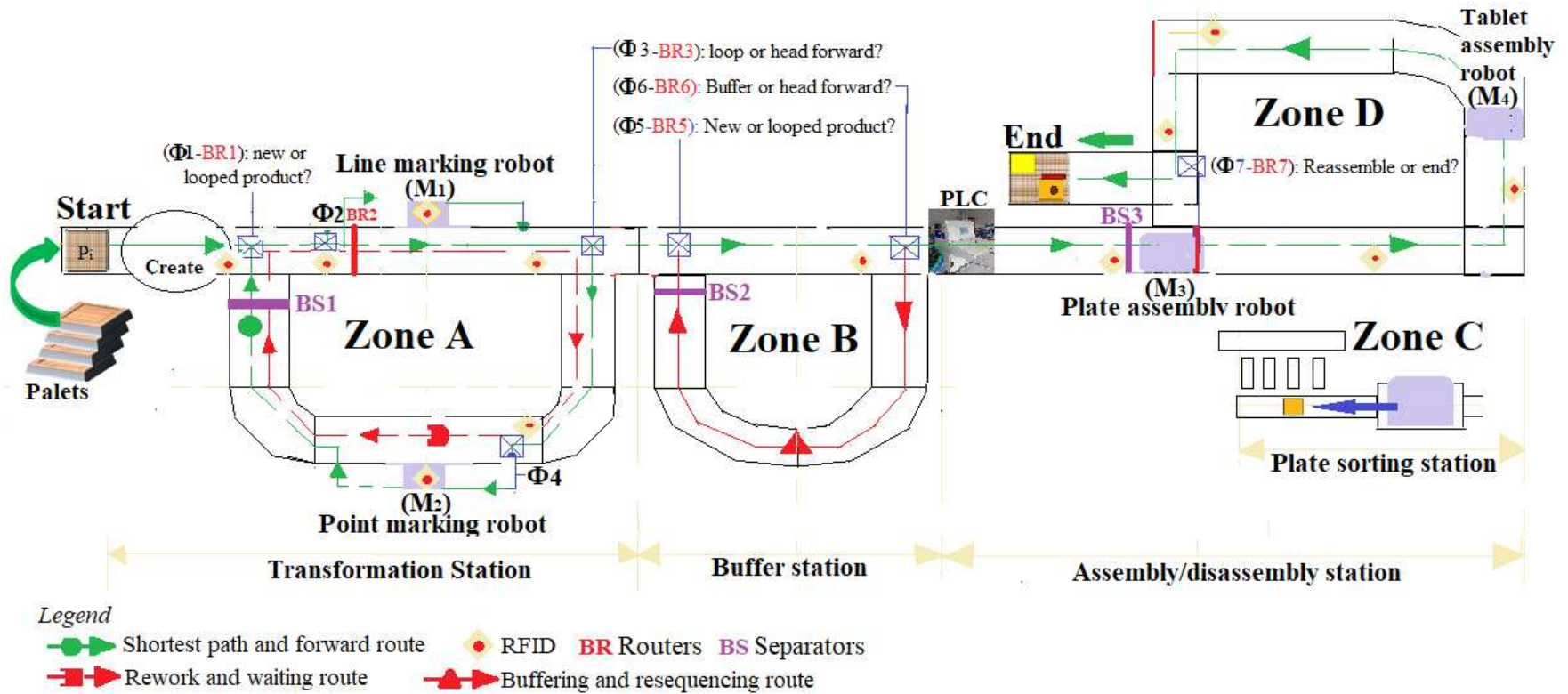


Fig. 4-2 Layout of the TRACILOGIS test-bed platform

4.2.1 Implementation architecture of the platform

This multi-agent system in the TRACIOLOGIS platform is developed through Java Agent Development framework (JADE), Fig. 4-3(a). JADE is a java software structure for multi-agent system development and applications compliant with Foundation for Intelligent Physical Agents (FIPA) standards (Bellifemine F. et al., 2010). It includes three basic components:

- An execution environment in which agents can *live* a library of classes
- A set of graphical tools for the management and
- Administration of active agents

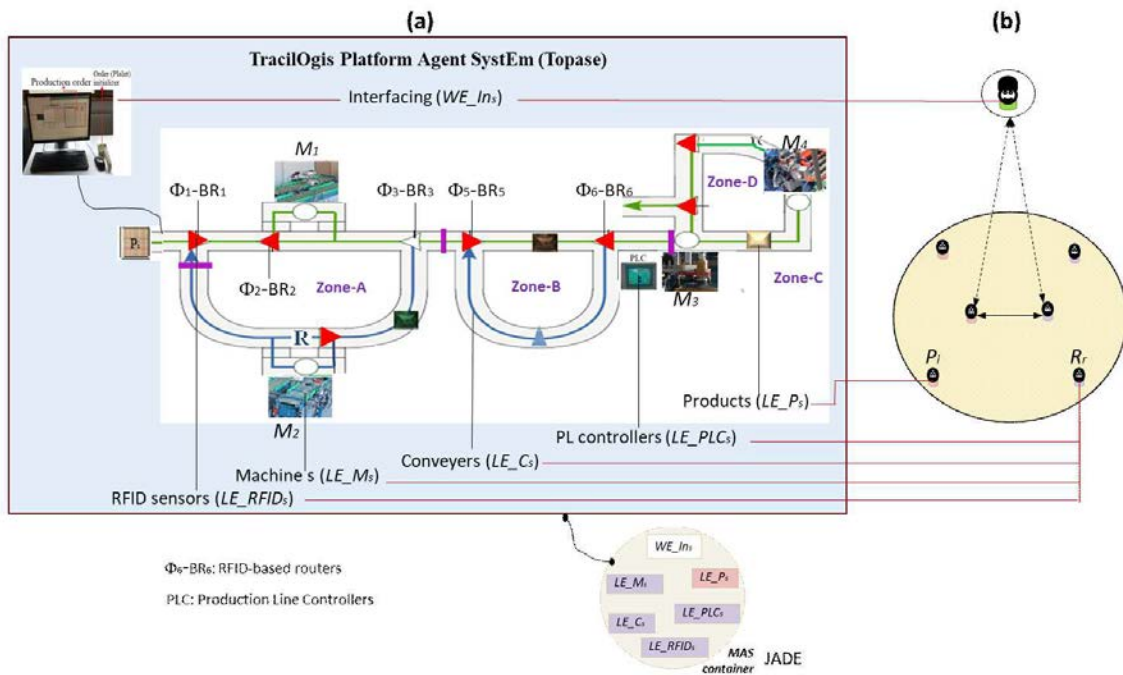


Fig. 4-3 The platform's implementation architectures

(a) The implementation architecture of the platform (b) The REDCA

Three agent groups namely product agents, resource agents, and interfacing agents constitute the JADE and make cooperative interactions until they reach on their common and final offer based on the pyramid of control architecture shown in Fig. 4-4. The product agents (LE_P_s in Fig. 4-3(a)), that have physical identifier (RFID tag) and logical identifier (product), are in charge of the correct execution of a given configuration, decides the way forward, transmits orders to resources, and validates or not the configuration changes sent by a configuration agent. The resource agents, which include reflex (automatic reactor to an event (or RFID sensors)), machines, PLCs, and Conveyers, are in charge of execution of commands sent by

interfacing agent and product agents to offer services such as physical transformation and spatial transformation. Four intelligent machines M_1 , M_2 , M_3 , and M_4 shown in Fig. 4-3(a) found the machine agents and are used for line marking, point marking, plate assembling, and tablet assembling respectively. In case of unpredictable failure of either machine, the other machines have capability to perform all products' operations left unprocessed. Meanwhile, a single machine can perform more than one operations of product i at the time of necessity. Finally, the interfacing agent interfaces via library developed inside the JADE. It is created to allow a user (or external services) to supervise (give orders and receive reports). These agents have pure utilitarian functions for allowing communication between basic agents (resource, product) with equipments (such as RFID reader, camera, sensor, database etc.).

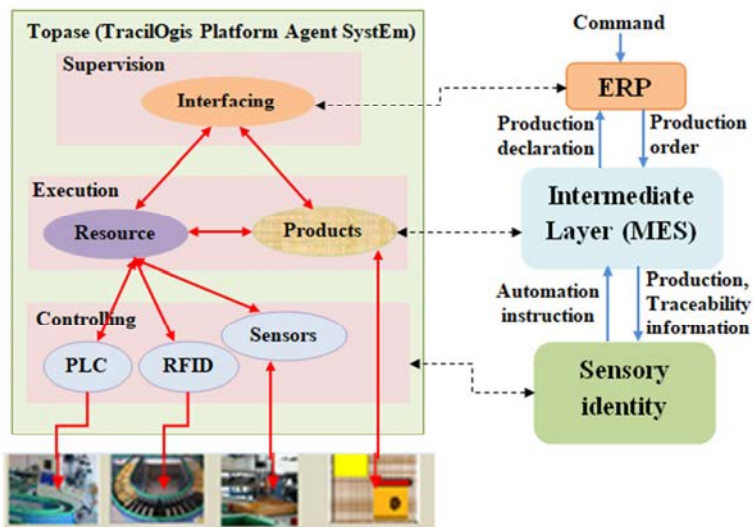


Fig. 4-4 The pyramid of control architecture of the platform (source: CRAN)

4.3 Simulation test results and discussion from the Negotiation-based control approach

MATLAB and Python coded simulation tests (with a minimum of 200 simulation replications) are executed to verify the performance of the negotiation-based control approach (NRC) over the pure reactive control approach (PRC).

4.3.1 Simulation configuration of the TRACILOGIS platform

The Cutting (resizing), drilling, sanding, and coating activities presented in Fig. 3-3 are transposed to the four machines M_1 , M_2 , M_3 , and M_4 shown in Fig. 4-2 respectively. Each machine entity has different processing times for each product entity created to be executed. The verification is mainly showed on a bottleneck machine M_1 .

Input data a central schedule for three manufacturing orders; MO_1 , MO_2 , and MO_3 , each with their own product types, Fig. 4-5, and due date is developed based on FIFO sequence $\{(MO_3) < (MO_2) < (MO_1)\}$. Meanwhile, for each manufacturing order, ten intelligent products are launched with $\{P_1, P_2, \dots, P_{10}\}$ to MO_1 , $\{P_{11}, P_{12}, \dots, P_{20}\}$ to MO_2 , and $\{P_{21}, P_{22}, \dots, P_{30}\}$ to MO_3 . The launched configuration considers the travel time of product entities from resource to resource and the processing time of each product on each machine entity; see Table 4-1. Meanwhile, note in the table that the processing time in M_2 for manufacturing order MO_2 is set to zero to indicate that product entities of this manufacturing order have no operation to be performed in machine M_2 and machine M_1 is represented two times to indicate it is required for another operation of same product type. The completion time and due date of each product type is calculated from the lead time of the launched platform. That is, the travel time (or conveyor speed of the platform) and the processing time of each operation of each product entity on each machine entity are considered.

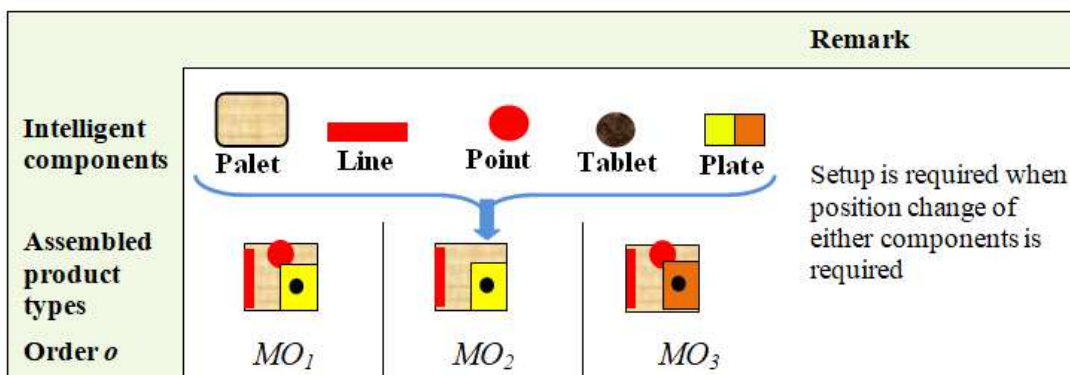


Fig. 4-5 Intelligent components and assembled product types of each manufacturing order

Table 4-1 Processing time of each product type P_i in each machine m

MO_o	Processing time of each product type (s)					C_i (s)	dd (s)
	M_1	M_2	M_1	M_3	M_4		
MO_1	3	1	4	9.5	7.7	162.2	400
MO_2	8	0	0	9.5	7.7	89.2	300
MO_3	11	3	0	9.5	7.7	174.2	600

From Fig. 4-2, the launched travel time from $\{start\ to\ M_1\}$, $\{M_1\ to\ M_2\}$, $\{M_2\ to\ M_1\}$, $\{M_1\ to\ M_3\}$, $\{M_2\ to\ M_3\}$, $\{M_3\ to\ M_4\}$, and $\{M_4\ to\ End\}$ are given as 5, 29.5, 47, 18.5, 60, 7.5 and 30 seconds respectively. Meanwhile, the travel time of a complete loop in zone-B is taken to be 43.5 seconds. Each product agent follows a standard routing sheet using the platforms' layout. For example, every product entity of MO_1 are expected to pass through M_1 for their first operation, move to M_2 for their next operation, loop back to M_1 for inspection (optional), travel to M_3 for their third operation, and finally route to M_4 for their last operation.

It is described in sub-section 3.3.1 that if the routing sheet continues based on the *FIFO* dispatch rule, it increases the completion time of products of MO_2 as they will be obligated to loop in Zone-B of Fig. 4-2 without added value until all products of MO_1 complete their route in Zone-A and proceed to machine M_3 for their next operation. This happens because, as per the initially launched execution, products of MO_1 need to serve in M_3 first. Hence, product entities of each manufacturing order create a system, shown at the beginning of Fig. 4-2, to be combined and dispatched SPT first rule $\{(MO_3) < (MO_1) < (MO_2)\}$. This system is created considering the bottleneck machine entity in order to fully utilize its capacity; step-4 in sub-section 3.3.1. While product and machine entities decide to make such change, critical ratio (step-5 in sub-section 3.3.1) is used with an objective to process product/s with shortest critical ratio first. The bottleneck machine as white-collar entity selects set of products from the set of candidates (products) sequenced. This continues until the completion of processing the set of visitors (all products of the manufacturing orders). This indicates that the intention of each product entity is to complete its routing sheet based on the sequence of earliest due date first. Consequently, from Eq. (3-5) and Table 4-1, the intention of each product type is computed as shown in Table 4-2 below.

Table 4-2 Computed intention (in seconds) of each product type

MO_o	Computed intentions				
	$v_{1(1)}$	$v_{2(2)}$	$v_{3(1)}$	$v_{4(3)}$	$v_{5(4)}$
MO_1	[5 3 8]	[37.5 1 38.5]	[85.5 4 89.5]	[108 9.5 117.5]	[125 7.7 144.2]
MO_2	[8 8 16]	[0 0 0]	[0 0 0]	[34.5 9.5 44]	[51.5 7.7 59.2]
MO_3	[16 11 27]	[56.5 3 59.5]	[0 0 0]	[119.5 9.5 129]	[136.5 7.7 144.2]

In the computed intention values, the exit time of product i from machine m (e_{km}) is the cumulative sum of travel time of the product to machine m (tt_{km}) and processing time of the product on the machine (p_{km}). After this computation, the product entity's intention is evaluated by a machine entity that the product is approaching and neighbor product entities j for execution.

4.3.2 Analysis and manipulation of the simulated results

Three performance indicators namely product lateness, machine utilization rate, and makespan are used to make the following analysis.

Production performance indicators the simulated result considering the product lateness (L_i) is illustrated in Fig. 4-6. Lateness measures how products are tardy because of constraints encountered in upper stream station or how products are early due to constraints occurred in downstream station. In the NRC, product entities have reduced lateness to a minimum of 24.59 seconds in one of their simulation replication. However, in the PRC, the minimum lateness is recorded to be 30.71 seconds. This shows that the product lateness in the launched routing sheet has been reduced by 19.92% as a result of the environment created for NRC. Moreover, the pattern in the NRC is more stable than that of the PRC, see Fig. 4-6, because the product entities in the PRC did not show interest to cooperate for minimizing the makespan instead they route for minimizing their individual completion time only. Following the reduction of product lateness, the machine utilization rate has also improved. For instance, as it is shown in Fig. 4-7, machine M_1 is utilized in NRC on average 18% better than in PRC.

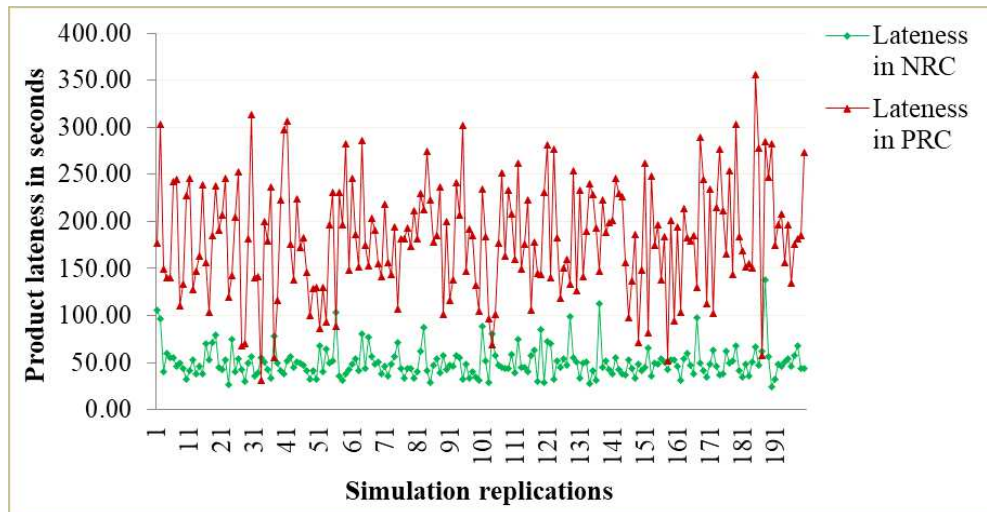


Fig. 4-6 Analysis of product lateness in both control approaches

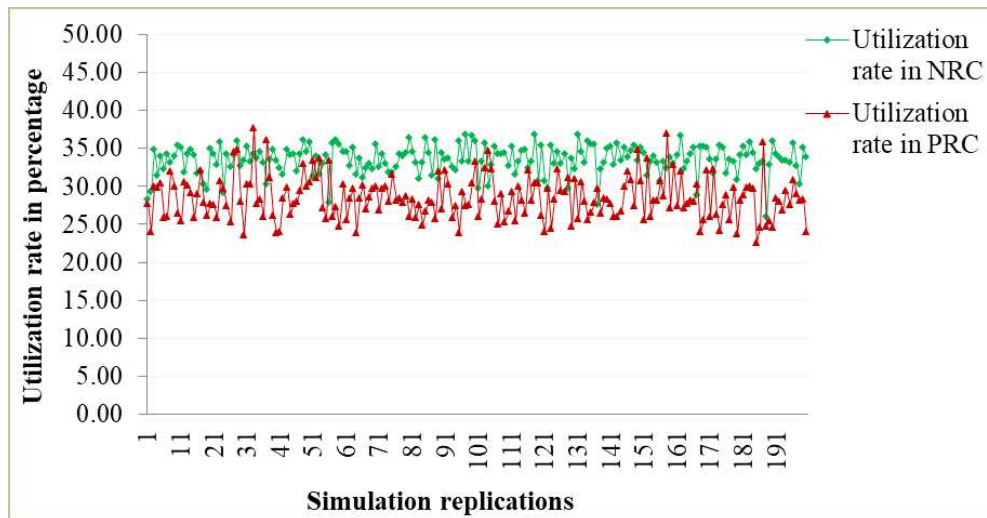


Fig. 4-7 Analysis of machine utilization rate in both control approaches

Makespan is also used to verify the performance of the NRC. As it can be observed from Fig. 4-8, once product agents in PRC set their sequential route, they showed interest to follow this route instead of setting another optimal route that help minimize makespan. But in the case of NRC, product and machine agents update their current route based on the unexpected event occurred so as to minimize its impact on the makespan. Accordingly, the simulation test has resulted, on average, 920.26 seconds of makespan in PRC and 777.88 seconds following the pursuance of NRC. This indicates that the makespan is reduced by 15.47% after employing the negotiation-based control approach.

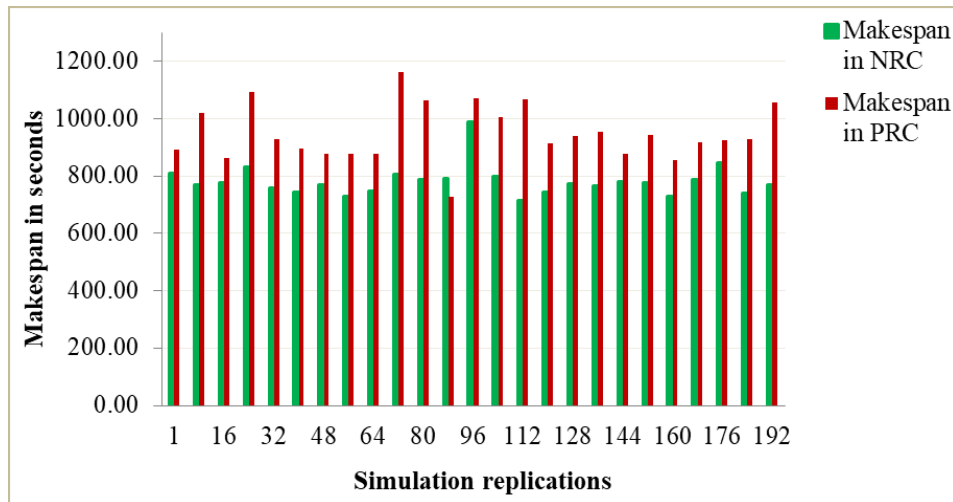


Fig. 4-8 Makespan analysis in both control approaches

On the other hand, a statistical probability characteristic was used to find patterns and predict certain outcomes from a set of population (number of simulation replications in our case). As part of this, a statistical parameter called performance density function (*pdf*)⁴² is used to show the effect of NRC in improving the machine utilization rate and hence minimizing the makespan. Accordingly, it is illustrated in Fig. 4-9(a) that the makespan in NRC has higher *pdf* value than the makespan in PRC to indicate that all replicated makespan values lies around their median. However, in the PRC, the *pdf* is lower to indicate unstable closeness of all replicated makespan values across their interval. The same pattern is observed in the probability density function of utilization rate of M_l , Fig. 4-9(b). Likewise, the *pdf* value increases with the decrease of makespan values in the NRC and vice-versa in the PRC. This happened because in the PRC, product entities always decide to choose their shortest path without the knowledge of the machine entity that they are approaching until they exit from the launched system. This may be successful for some product entities (perhaps for product entities that entered the launched system first) but not for the others. Because of the purely reactive decision taken by the product entities, the machine entity that they are approaching would not select any of these product entities and hence the product entities indefinitely loop in Zone-A or Zone-B in Fig. 4-2. This event increases the completion time of each product entity. However, in the NRC, as all the product entities together with the machine entities re-sequence themselves considering an agreed parameter, the simulated makespan values are very close to each other and hence reduced to indicate that

⁴²*Pdf* is a function of a continuous random variable, whose integral across an interval gives the probability that the value of the variable lies within the same interval. It is used to specify the probability of the random variable falling within a particular range of values, as opposed to taking on any one value.

a collectively negotiated and defined routing sheet is followed until their exit from the launched system.

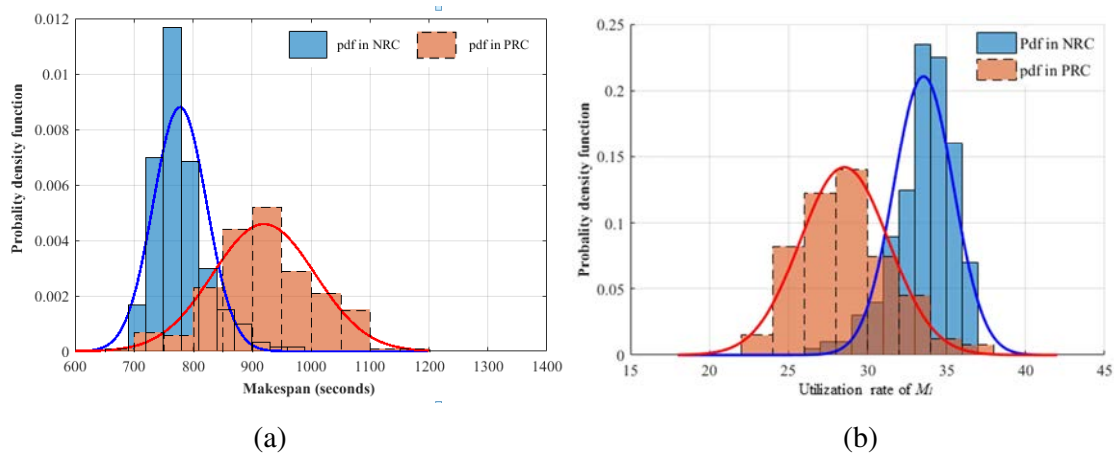


Fig. 4-9 Probability density function for both control approach

(a) pdf of makespan. (b) pdf of machine utilization rate

Correlation analysis the simulation experiment is also made to show the correlation of two performance indicators to evaluate the impact of the proposed control approach. For instance, as shown in Fig. 4-10, while the machine utilization rate of M_1 increases, the makespan decreases. However, the difference is shown at the percentage of variance in the makespan that the machine utilization rate explains collectively. In the dotted red regression line in Fig. 4-10, the cumulative effect of the machine utilization rate is shown at the left side of the regression line; less utilized and higher makespan. This happens because successor product entities arrive to the machine without understanding its current status and hence these products are obligated to wait close to the machine until they arrive to the machine for their next operation. The solid black regression line in Fig. 4-10 has reversed this tendency; the collective effect of the machine utilization rate is shown at the right side of the regression line. The makespan value is linearly shifting downward with the increase of the machine utilization rate. This happens because products are arriving to the machine keeping their sequence based on the initially computed critical ratio and a buffer size defined in constraint (3-3(b)). Meanwhile, even though the R-squared⁴³ value is similar in both control approaches, the place of variability closeness to the regression line is different to indicate a linear effect of the machine utilization rate to minimize makespan. Most of the replicated makespan values are closely positioned at the most optimal utilization rate of the resource after employing the NRC; see the most right side of the solid black line in Fig. 4-10.

⁴³R-squared is a statistical measure of how close data are to the fitted regression line

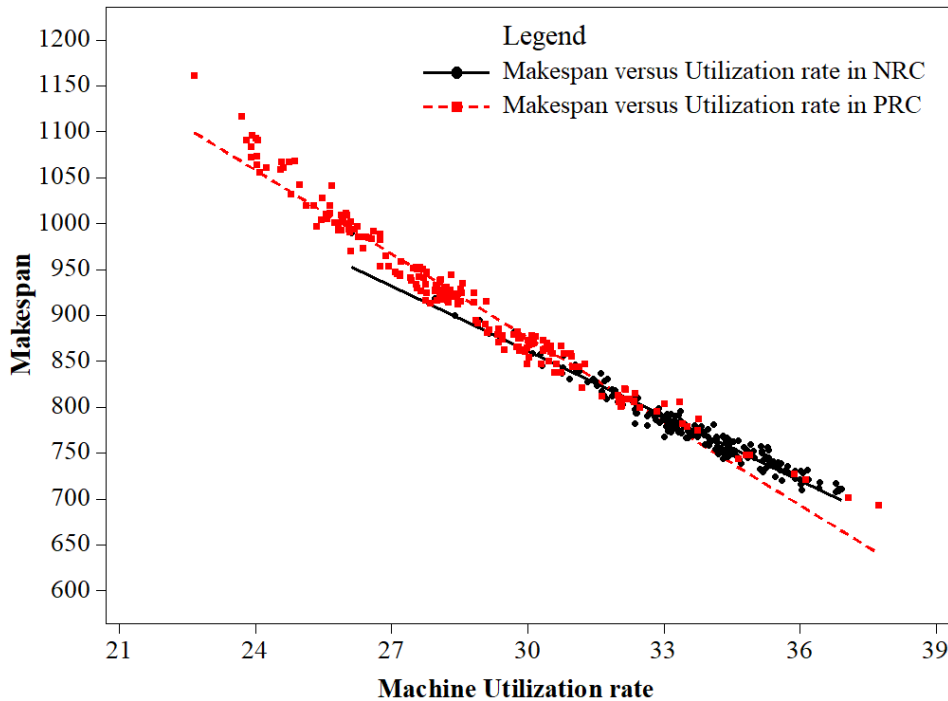


Fig. 4-10 Scatterplot of makespan versus utilization rate in both control approaches

To recap the discussion, three of the performance indicators have shown significant advantage of NRC in increasing adaptability and flexibility of a system to unexpected event within the launched routing sheet. Meanwhile, when decision change (e.g., resequencing) is queried by either of entity, all entities in the NRC are able to dynamically satisfy this change.

4.4 Simulation test results and discussion from the Consensus algorithm

The decision changing hypothesis has considered all cooperating product entities (through their leaders) of the affected manufacturing orders with simulation replications of 200.

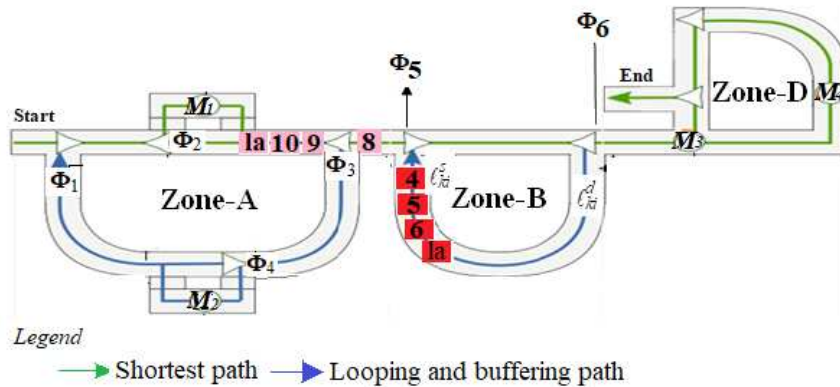
4.4.1 Simulation configuration of the platform

The shop floor presented in Fig. 3-8 is transposed to the TRACILOGIS platform for simulation experiment purpose. Accordingly, the simulation configuration of the platform is presented as shown in Fig. 4-11. After the occurrence of the unexpected event (called rush order), Zone-B is used to minimize cost penalties for trade-off between actual and planned completion time of product entities. It allows for buffering of semi-processed products as in a real production system. As it is shown in Fig. 4-11(a), Dynamic looping time (ℓ_{ki}^d), static looping time (ℓ_{ki}^s), and number of looping by product i (η_{ki}) in Zone-B have used to model this buffering such that,

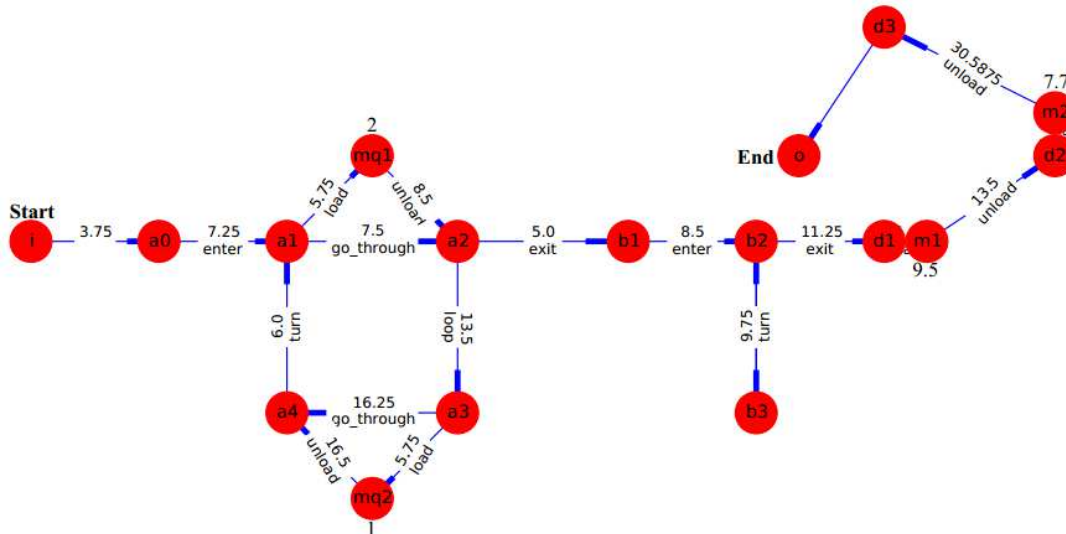
$$\eta_{ki} \begin{cases} \geq 2 & \text{if rush order is not fully satisfied} \\ =1 & \text{otherwise} \end{cases} \quad \forall i \in MO_{n-1}, \ell_{ki}^d = f(\eta_{ki})$$

$$\ell_{ki}^s \begin{cases} >0 & \text{if rush order is not fully satisfied} \\ =0 & \text{otherwise} \end{cases}$$

Product entities, router entities (BR6 in Fig. 4-2) and roller conveyor (controlled by PLC and RFID as controlling agents) are used to re-launch the platform. For instance, the static looping time at the end of Zone-B is controlled by RFID and BS2 separator in Fig. 4-2. The notations $M_1, M_2, M_3,$ and M_4 in Fig. 4-2 have the same meaning as $mq_1, mq_2, m_1,$ and m_2 in Fig. 4-11(b) respectively. Each machine has a processing time of 2.0, 1.0, 9.5, and 7.7 seconds for single operation of each product P_i respectively.



(a) Flow position of semi-finished products and the decision node they are approaching



(b) Operation times (in seconds) versus execution resources (routers and machines)

Fig. 4-11 Process graph of the TRACILOGIS platform

Input data a deterministic schedule, Fig. 4-12, for 12 manufacturing orders $MO_1, MO_2, \dots, MO_{12}$ is developed considering shortest path of the platform; Eq. (3-15) and Fig. 4-11(a). It presents the optimal schedule of each manufacturing order to complete the routing sheet. Each manufacturing order has its own product type (see Fig. 4-13) and the processing times of each operation of each product i on each machine m , see Table 4-3. In order to launch the routing sheet, the due date of each manufacturing order must be determined at glance. Accordingly, the due dates of each manufacturing order are set manually in the date they are received from their respective customers; the last column in Table 4-3.

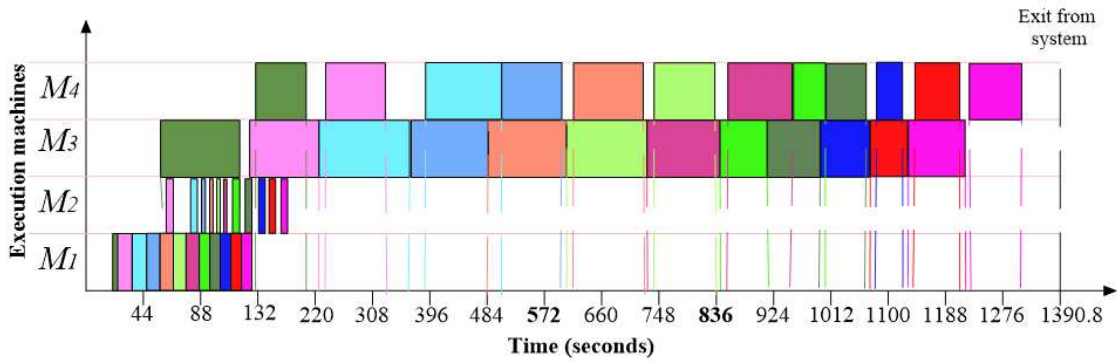


Fig. 4-12 Deterministic schedule (MPS) for 12 manufacturing orders

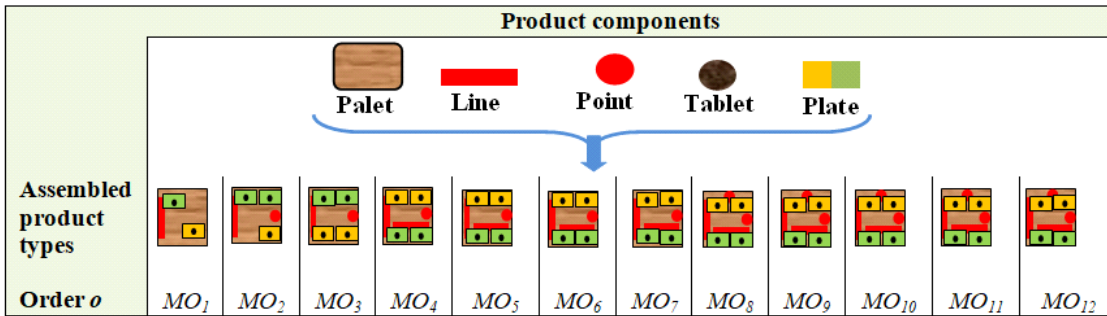


Fig. 4-13 Product types of each manufacturing order

As it is presented in Axiom 3-1, at the beginning of the execution system, these manufacturing orders were made to enter the platform based on the below dispatch list:

$$\{(MO_{12}) \prec (MO_{11}) \prec (MO_{10}) \prec (MO_9) \prec (MO_8) \prec (MO_7) \prec (MO_6) \prec (MO_5) \prec (MO_4) \prec (MO_3) \prec (MO_2) \prec (MO_1)\}$$

While they route according to this dispatch list, MO_3 is made to be rushed over MO_2 at time $t_e = 75$ seconds; which demands for the following dispatch list

$$\{(MO_{12}) \prec (MO_{11}) \prec (MO_{10}) \prec (MO_9) \prec (MO_8) \prec (MO_7) \prec (MO_6) \prec (MO_5) \prec (MO_4) \prec (MO_2) \prec (MO_3) \prec (MO_1)\}$$

Table 4-3 Processing time of each product i in each machine m

<i>Order o</i>	P_i	Processing time (s) of each product P_i on m				$C_i(s)$	dd (s)
		M_1	M_2	M_3	M_4		
MO_1	P_1	2	-	19	15.4	132.75	250
	P_2	2	-	19	15.4	153.75	
	P_3	2	-	19	15.4	174.75	
MO_2	P_4	2	1	28.5	23.1	289.25	600
	P_5	2	1	28.5	23.1	329.25	
	P_6	2	1	28.5	23.1	369.25	
	P_7	2	1	28.5	23.1	409.25	
MO_3	P_8	2	1	38	30.8	462.82	800
	P_9	2	1	38	30.8	493.32	
	P_{10}	2	1	38	30.8	523.82	
	P_{11}	2	1	38	30.8	554.32	
MO_4	P_{12}	4	1	38	30.8	599.32	950
	P_{13}	4	1	38	30.8	630.32	
	P_{14}	4	1	38	30.8	661.32	
MO_5	P_{15}	4	1	38	30.8	759.57	1200
	P_{16}	4	1	38	30.8	791.32	
	P_{17}	4	1	38	30.8	823.07	
MO_6	P_{18}	4	1	38	30.8	854.82	1350
	P_{19}	4	1	38	30.8	886.57	
	P_{20}	4	1	38	30.8	918.32	
MO_7	P_{21}	4	1	38	30.8	950.07	1400
	P_{22}	4	1	38	30.8	981.82	
	P_{23}	4	1	38	30.8	1013.57	
MO_8	P_{24}	4	2	38	30.8	1111.82	1600
	P_{25}	4	2	38	30.8	1142.82	
MO_9	P_{26}	4	2	38	30.8	1173.82	1700
	P_{27}	4	2	38	30.8	1204.82	
MO_{10}	P_{28}	4	2	38	30.8	1235.82	1800
	P_{29}	4	2	38	30.8	1266.82	
MO_{11}	P_{30}	4	2	38	30.8	1297.82	1900
	P_{31}	4	2	38	30.8	1328.82	
MO_{12}	P_{32}	4	2	38	30.8	1359.82	2050
	P_{33}	4	2	38	30.8	1390.82	

The Leader-Follower based communication approach, Fig. 4-14, is used by the product entities of the impacted manufacturing orders to fulfil this new dispatch list. The leader product agent⁴⁴ of MO_2 (P_7) is made to be precedence communication neighbor of the leader product agent of \widetilde{MO}_3 (P_{11}) and successor neighbor of that of MO_1 (P_3). The simulated communication between all neighbor product agents of these two manufacturing orders has emulated and validated the proposed algorithm. See Fig. 4-15 for the pseudo code, in Python, of these decision entities to make this communication.

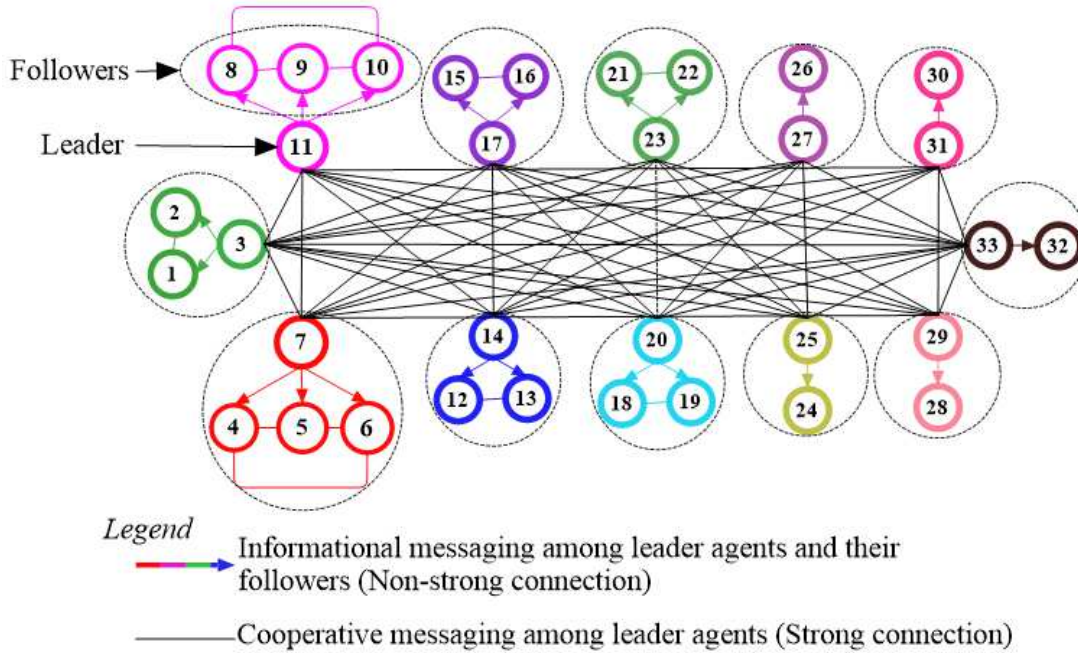


Fig. 4-14 The leader-Follower communication topology among the product agents (adapted from Notarstefano et al., 2011)

⁴⁴In this industrial context case, leader product agent is the last product of each manufacturing order

Defining three groups of decision entities

Control.get_products ():

List product types (type, quantity, due date) to enter to the system,
 Rush_time (#time when the rush order declaration happens), $t_e = 75$ seconds
 Rush_id (#id of the order to change, MO_3) = 3
 Rushed_time (# due date required to be rushed by MO_3) = 290 seconds

Return

Control.get_machines ():

Create machine agents
 Machines['mq1'] = MachineAgent ('mq1')
 Machines['mq2'] = MachineAgent ('mq2')
 Machines['m1'] = MachineAgent ('m1')
 Machines['m2'] = MachineAgent ('m2')

Return machines

Control.get_conveyors ():

Create a conveyor agent representing this physical component
 Conveyors['a1'] = conveyorAgent ('a1', 7.25)
 Conveyors['d1'] = conveyorAgent ('d1', 11.25)

Return conveyors

End

Fig. 4-15 Pseudo code for the three groups of decision entities

From the six decision nodes Φ_s shown in Fig. 4-11(a), decision nodes Φ_5 and Φ_6 are chosen to test the required consensus. During the emulation, all the product agents of successor manufacturing orders to MO_3 are assumed to behave same as that of the currently interacting product agents of MO_2 and MO_3 by adapting to the rush order considering their current state. Accordingly, the communication on the other decision nodes behaves (for instance, Φ_2 and Φ_3) same evolution as in the state-based decisions made in Φ_5 and Φ_6 .

All the leader agents negotiate with each other and share the agreed offer to their followers for implementation; Fig. 4-14. As each product agent P_i has its own programmed intention, Eq. (3-5), the decision process starts by sensing the cumulative sum of informational state of neighboring agents. Each product agent of the rush order re-computes its intention $v_{k(m)}$ and broadcasts to defined service providing and neighbor agents $j \in N_i$. The neighbor (or responding) agents acknowledge the broadcasted intention and respond by updating their current state; the Python pseudo code of this *update state* is shown in Fig. J-1 of Appendix J. Hence, the leader agent and its followers of the second manufacturing order

(P_7, P_6, P_5, P_4 ; the responding agents) are obligated to update their current state and loop in Zone-B as the router in Φ_6 (controlled by PLC agent) blocks these product entities from heading to machine M_3 . This continues until the leader product agent and its followers of the rushed order (P_{11}, P_{10}, P_9, P_8) pass Φ_5 in order to be processed in next machine M_3 before that of products of MO_2 ; see the controlling Python pseudo code of this *loop request* in Fig. J~2 of Appendix J.

On the other hand, Fig. 4-16, while products of MO_2 , MO_3 and their successors are cooperating to satisfy the rush order (in a distributed reactive control mode), product entities of MO_1 (which are not affected by the rush order) are in execution based on the centrally developed routing sheet. This confirms the role of the recursive hybrid control architecture presented in section 3.2 to satisfy unexpected perturbation(s) occurred. This type of hybrid decision is also presented in the work of Pach et al. (2014b).

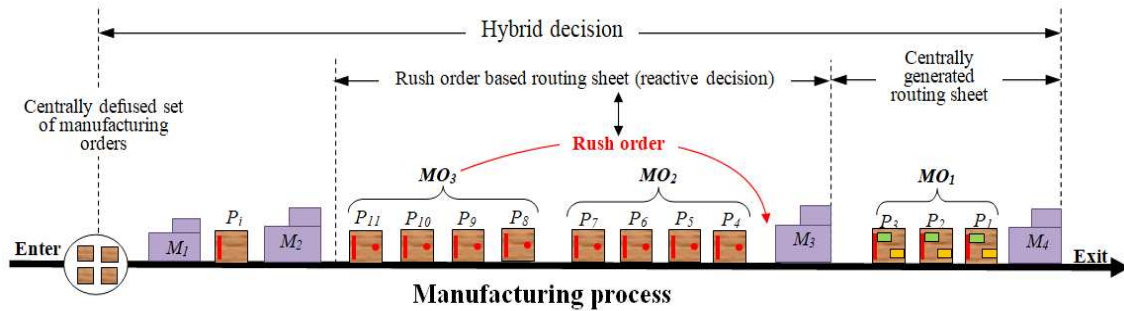


Fig. 4-16 Role of hybrid decision to satisfy the rush order

4.3.2 Analysis and manipulation of the simulated results

Three performance indicators namely informational state (x_i), lateness (L_i), and makespan (C_{max}) have considered prompting the performance and are discussed through convergence analysis, production performance indicators, and sensitivity analysis in the following paragraphs.

Convergence analysis it is stated earlier that the required consensus is mainly showed in zone-B of Fig. 4-11(a). Product agents of the rush order and that of the preceding manufacturing order have agreed to support a decision for the best interest of all members. They follow the *update intention for the best of the whole* sequencing and routing. Accordingly, it is modeled in Eq. (3-13) that consensus would be achieved if the *variability* of interest among all product agents approaches towards zero; where x_i and x_j in Eq. (3-13) represent $\{x_8, x_9, x_{10}, x_{11}\}$ and $\{x_4, x_5, x_6, x_7\}$ respectively in this case.

Such that, from Algorithm 3-2, $\begin{cases} \text{if } |x_i - x_j| \rightarrow 0, \text{ agreed} \\ \text{if not, otherwise} \end{cases}$

While supporting this decision, the location of product agents, Fig. 4-11(a), is determined by the priority request. If the sum of the difference of state value of neighbor agents is greater than a threshold $\beta_2=1$, product agents of the proceeding manufacturing order will loop in zone-B otherwise not. As shown in Fig. 4-17, the simulation results have shown rapid convergence (closeness to the invariant centroid state $x_i=1$ or the black dotted line in the figure) in consensus-based decision rather than its non-convergence in myopic decision-making algorithm, Fig. 4-18. The intention of product agents of the rush order was to lower their completion time at least by 290 seconds, Fig. 4-17. As a response to this intention, the preceding product agents have shared the state to show their agreement by increasing their completion time at least by 290 seconds. This continues until the rushed product agents completely pass Φ_5 in Fig. 4-11(a) and this is confirmed when the difference of state value of conflicting product agents start to be below the threshold $\beta_2=1$, Algorithm 3-2. To examine this agreement level, it is taken for instance, the state of the two leader agents x_{11} and x_7 with their last time-evolved simulation replications of 0.998035 (or the minimum value) and 0.938344 (or the maximum value) respectively. Eq. (3-13) gives $|0.998035-0.938344| = 0.05969$ which implies that the variability of interest γ among these product agents is close to zero; thus, fast convergence to the invariant centroid state $x_i = 1$ is obtained.

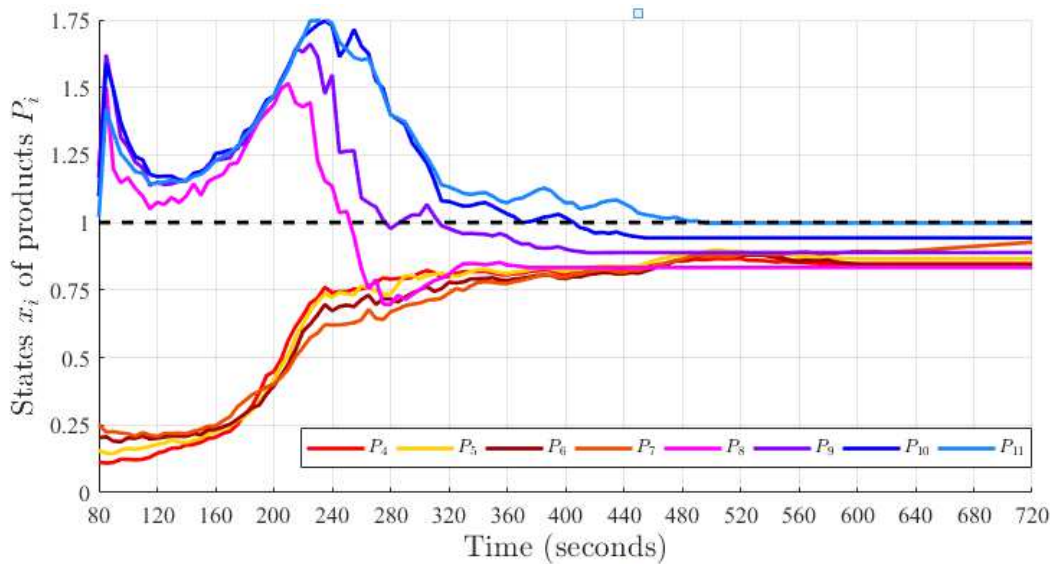


Fig. 4-17 Time based evolution of state in the consensus-based decision-making algorithm

On the contrary, it is presented in Fig. 4-18 that each product agent decides to choose a destination with the shortest completion time scheduled centrally. Product agents of the preceding manufacturing order become myopic as they use *change the production intention for only own goal* routing. This indicates that these product agents were not interested to consider the designed state so as to satisfy the rush order and hence a conflict results. As a result, these agents have higher probability to loop in zone-B as many as η_{ki} without value addition. This has happened because the PLC agent in the MAS container has received a command message from the leader product agent of the rushed order to block products who are proceeding to him not to go to M_3 through router agent BR6 in Fig. 4-2 To examine this myopic decision, consider again the simulated state value of the leader agent of the rush order and that of the proceeding manufacturing order which are recorded as 0.301589 and 1.547314 respectively. Eq. (3-13) once more gives a value of $|0.301589-1.547314| = 1.24573$ which implies that the variability of interest among these two agents is large enough or non-convergent (perhaps much delayed convergence). This has happened because the former agent has tried to shift-down its completion time but that of the later agent to keep its original completion time and hence initial state.

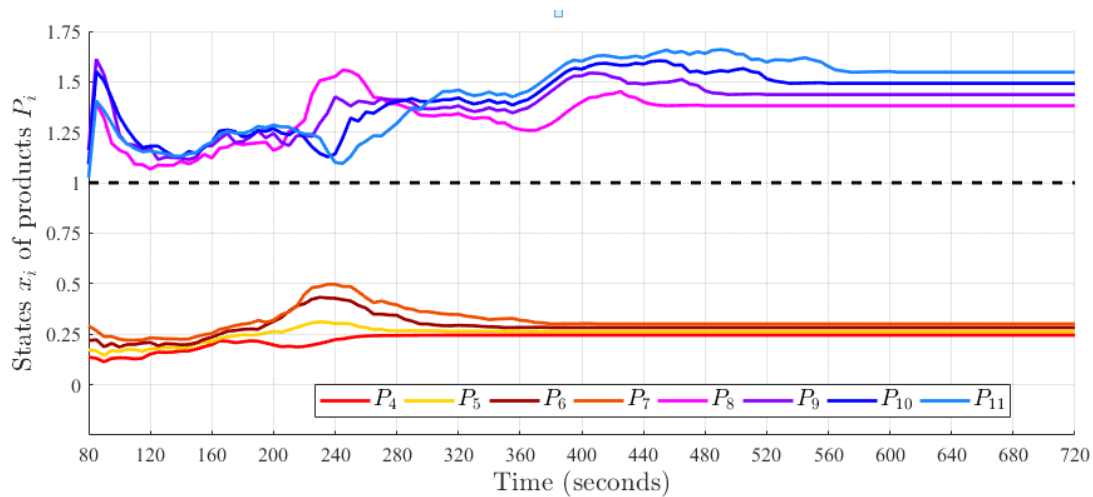


Fig. 4-18 Time based evolution of state in the myopic decision-making algorithm

From these two calculated state values, it could be understood that the disagreement level among product agents for the rush order offer in myopic decision is 95.21% higher than that of the consensus-based decision, Fig. 4-19. Even though there was strong rush order priority by product agents of MO_3 , product agents of MO_2 did not show interest to satisfy this offer in the myopic decision.

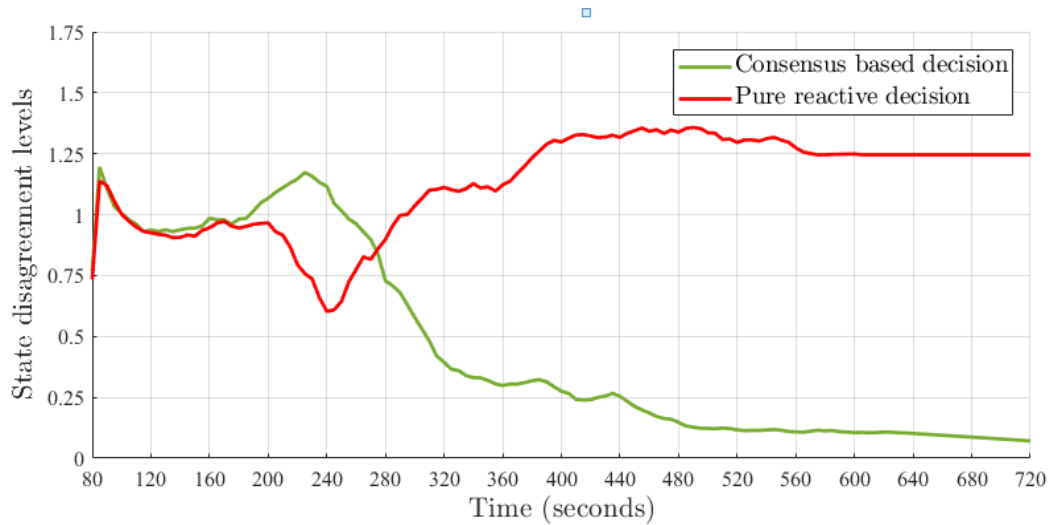


Fig. 4-19 State disagreement ($|x_i, x_j|$) level for the two decision-making algorithms

Production performance indicators lateness and makespan are taken as key production performance indicators. The simulation results considering the product lateness (or specifically the product tardiness) is presented in Fig. 4-20 where the negative values indicate product earliness and positive values for tardiness. In myopic decision-making algorithm, there is no clear behavioral pattern to show either the continuous tardiness or earliness of products in the execution process; the instability shown by the dotted red line in Fig. 4-20. As soon as the rush priority is sent by the leader agent of MO_3 , all the product agents of the proceeding manufacturing order have shown strong interest to be processed as early as possible ignoring the state of the rush order. For example, P_7 of MO_2 has showed, on average, 290.65 seconds earliness in this decision not to satisfy the rush priority by P_{11} of MO_3 . P_{11} in turn becomes, on average, 99.89 seconds ($\cong 5\eta_{ki}$) tardy from its priority intention.

On the contrary, in consensus-based decision, the pattern shows gradual and stable increment to satisfy the rush order; the dotted blue line in Fig. 4-20. Taking same example as the myopic decision, the leader agent of MO_2 has reduced its earliness to 69.42 seconds and the tardiness of that of MO_3 to 18.44 seconds ($\cong 1\eta_{ki}$). This indicates that, as a result of implementing the consensus-based decision, tardiness of all manufacturing orders has lowered by 81.53% than using the myopic decision.

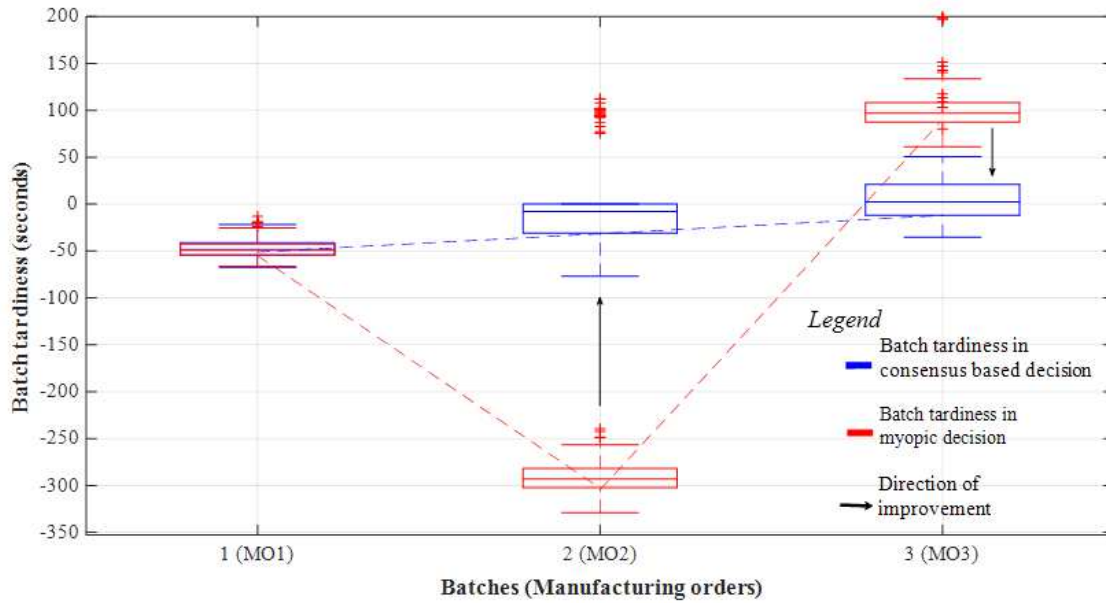


Fig. 4-20 Batch lateness for both control approaches

The simulation result has also showed the impact of product's disagreement level on makespan. Even though the makespan in both decision-making algorithms has scored a value higher than the centrally planned schedule (a routing sheet without rush order), the consensus-based decision has comparatively improved the deliverability. As shown in Fig. 4-21, the mean completion time in consensus-based decision is recorded as 1437.09 seconds or 12.55% lower than that of the myopic decision. This deviation has come from the myopic behavior of preceding product agents to response and set near optimal resequencing. The simulated completion times of each product of each manufacturing order is shown in Fig. J-3 in Appendix J. Meanwhile, product agents in the myopic decision-making algorithm might temporarily think that this behavior would lead them to quickly arrive to the next machine. However, as the rush order is proposed by the leader agent of the rushed order as white-collar entity, these product agents will be obligated to queue or buffer in front of the machine; higher buffer size (or constraint (3-3(b)) is not satisfied). It is all this waiting time that led the deliverability of manufacturing orders more delayed in the myopic decision-making algorithm than in the consensus-based decision-making algorithm.

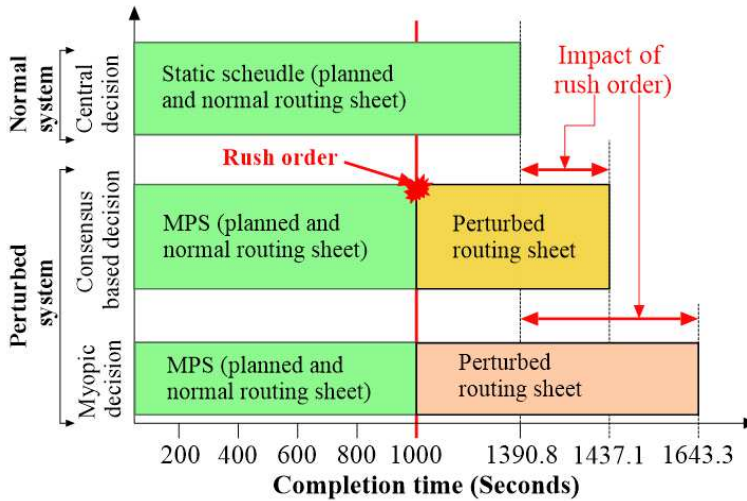


Fig. 4-21 Mean completion time of all manufacturing orders

Sensitivity analysis sensitivity analysis was also used as a performance indicator so as to evaluate the designed state. From statistical concept, sensitivity analysis is a technique used to determine how independent variable values can impact a particular dependent variable under a given set of assumptions. Fig. 4-22 shows this analysis considering makespan & state disagreement level as dependent variables and due date as explanatory variable. The values of makespan and state disagreement level are revealed in the z-axis. Three main colors namely dark blue (lower part of z-axis), light blue, and yellow (upper part of z-axis) are used to indicate the values. In the consensus-based decision, the deliverability of the rushed manufacturing order has reduced (shown by the white downward arrow in Fig. 4-22(a)) by C_3 .

However, the deliverability of the preceding manufacturing order has increased (shown by the white horizontal arrow in Fig. 4-22(a)) by C_2 . Even though the completion time of the leader agent of MO_2 has increased, the makespan of the system has significantly reduced, Fig. 4-21. Such fulfilment to margin changes at minimized cost has come from the interest of all product agents to reduce their variability until their common goal is achieved. Consequently, the completion time of MO_3 is scored to be lower than 650 seconds; the lower part of z-axis in Fig. 4-22(a) or option 2 in the figure. If the product agents of both manufacturing orders were deciding myopic, the completion time of MO_3 is expected to be higher than 650 seconds. No change from position 1 to position 2, in Fig. 4-22(a), as products of MO_2 did not show interest to increase their completion time from the initial plan. However, products of MO_3 are interested to complete their completion time as per the new plan which in turn leads to conflict among the products.

The state disagreement level at steady state is also shown in the z-axis of Fig. 4-22(b). When the margin of MO_2 and MO_3 are large enough, see the right upper part of Fig. 4-22(b) or the darkest blue one, there is no variability of interest between product agents and hence Eq. (3-13) is satisfied. However, when the margin of either MO_2 or MO_3 becomes lower, see the lowest left part of Fig. 4-22(b), product agents strive only to choose a destination with shortest completion time scheduled initially. At this time, they have to continuously compute their state and compare with their neighbor agents to minimize the disagreement level and hence to satisfy the rush order priority.

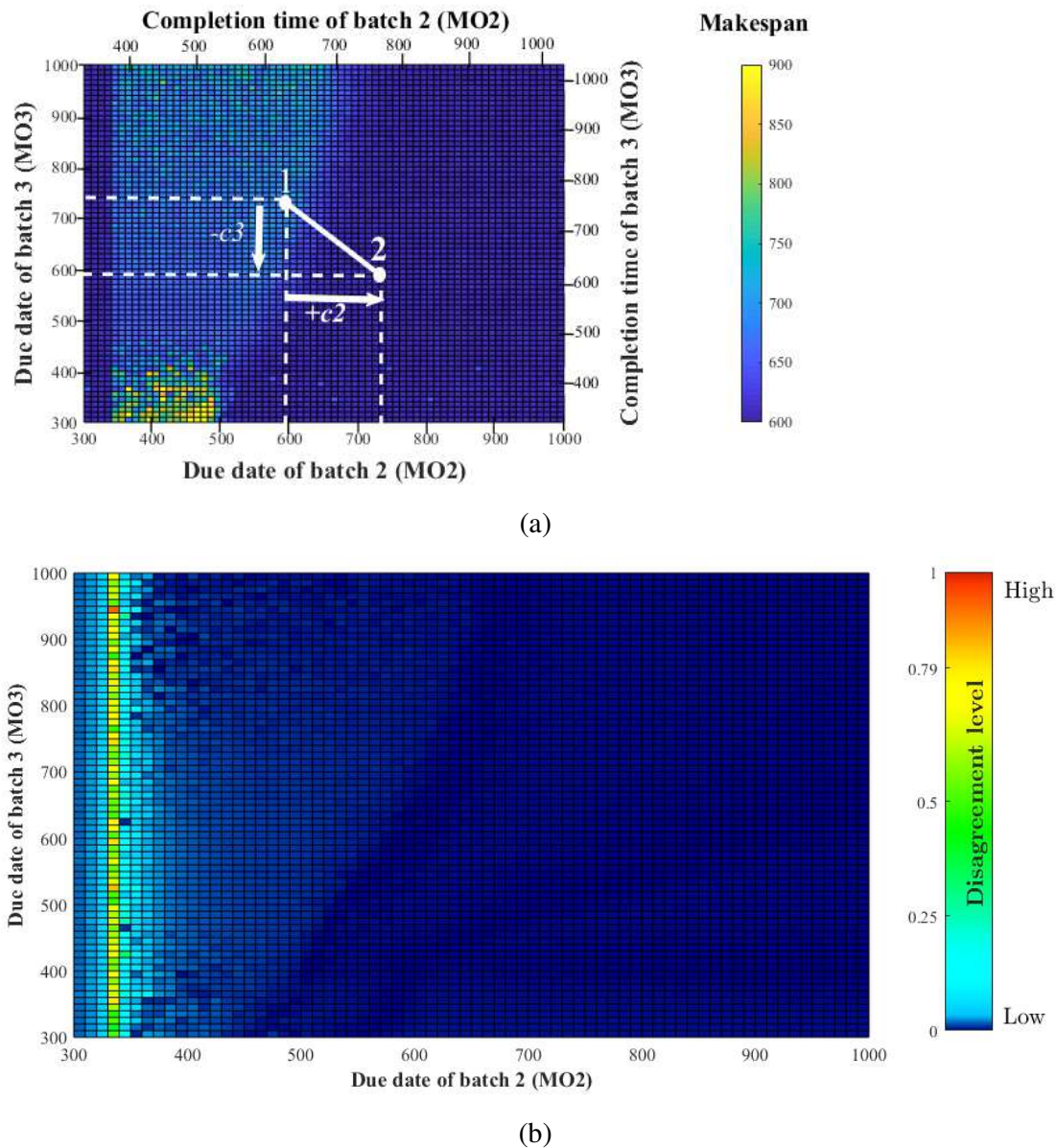


Fig. 4-22 Sensitivity analysis between performance indicators

(a) Due date versus makespan. (b) Due date versus disagreement level

4.5 Real applicability implementation of the proposed control approaches

Full tests of proposed control architectures and decision-making algorithms on real manufacturing systems or platforms are rarely addressed in the literature (Jimenez et al., 2017). Most of the works ends either at their simulation experimental stage or its applicability implementations are tested on test-bed platforms. As part of fulfilling this limitation, the industrial applicability of the proposed control approaches in this study are implemented using a java software structure called TracilOgis Platform Agent SystEm (topase), see for example the the topase configuration and communication method among agents in Fig. J~4 of Appendix J. During the applicability implementation tests, different number of palets⁴⁵ are used; 3 palets, 5 palets, and 9 palets. These different numbers of palets are considered to show the scalability and flexibility of the proposed decision-making algorithms. Subsequently, results obtained using nine products are presented in this study because the implementation platform has a maximum number of nine palets.

The probabilistic environment within the platform has been traced and the required information is shared among decision entities using RFID technology. Meanwhile, the required messaging is evaluated using FIPA-ACL communication language. A contract-net and broadcasting type of communication methods are coded using JADE to initialize data, localize the RFID tags, present current state of neighbor agents, allocate tasks of product agents, etc. Three agent types namely product agents (NegoProductAgent and ConsensusProductAgent), resource agents (machine agents, routers, conveyors, PLC), and interfacing agents (RFID and ERP) are used within the probabilistic environment; see Fig. J~5 in Appendix J.

In order to show the applicability implementation of the negotiation based control approach, production is made to begin at Φ_1 in Fig. 4-2 where the first RFID reader detects the first palet and production ends immediate to product's operation completion in M_3 , see Fig. 4-2. There is no need for products to go to M_1 for line marking and M_2 for point marking. The applicability test is made to show how a defined queue size can be implemented on the work space of M_3 in Fig. 4-2. According to constraint 3-3(b), the defined queue size is two. The products *update their position* through the NegoProductAgent; Fig. 4-23(a),. Once the position is updated, the machine M_3 in turn *informs its ready to work*, Fig. 4-23(b), based on

⁴⁵Palet is a product that includes a hardware part and control part localized with RFID tag. It directly represents the initial raw material of a manufacturing order to be assembled with all necessary parts such as plates in Fig. 4-2. This frequency of palets also indicates the number of implementation tests conducted.

the defined queue size and the completion of plate assembly operation for the product type on board. If two semi-processed products are queued on the work space of M_3 , the successor products loops in Zone-B of Fig. 4-2 keeping their critical ratio until one of the queued product starts to head to M_3 for its next operation. This continues until the completion of all operations of the products launched, see Fig. 4-24.

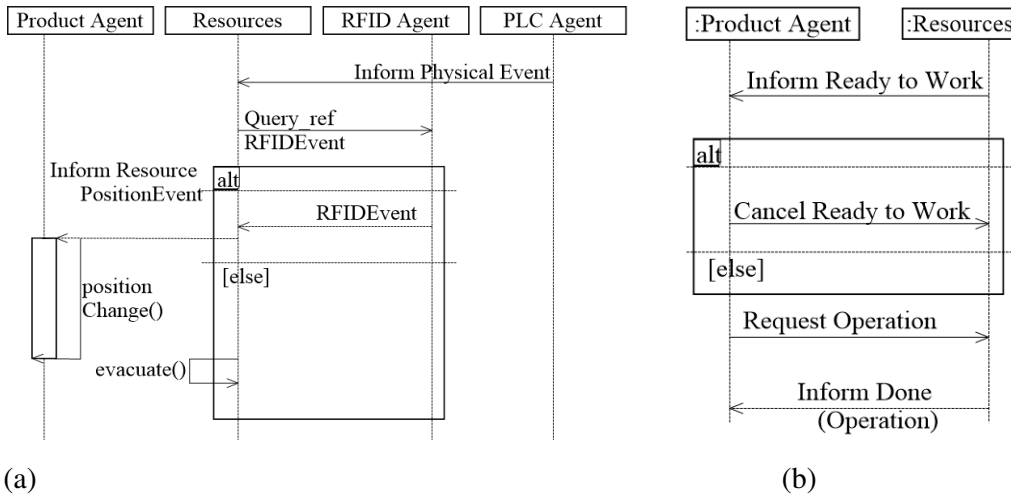


Fig. 4-23 UML topase sequence diagram

(a) for product's *state update position*. (b) for resource's *inform ready to work*.

Results of the applicability implementation test is shown in Fig. 4-24 and Table 4-4. The real completion time of each product is 99.99% same as their initially estimated completion time. Know that the estimated completion time in Table 4-4 is calculated after each product left machine M_3 . Two products are only late by only two seconds from the total amount of the estimated completion time. Compared to the real completion time in Fig. 4-24, the results in Table 4-4 include the product's travel time from M_3 to *end* and the processing time of the last operation of products in M_4 of Fig. 4-2.

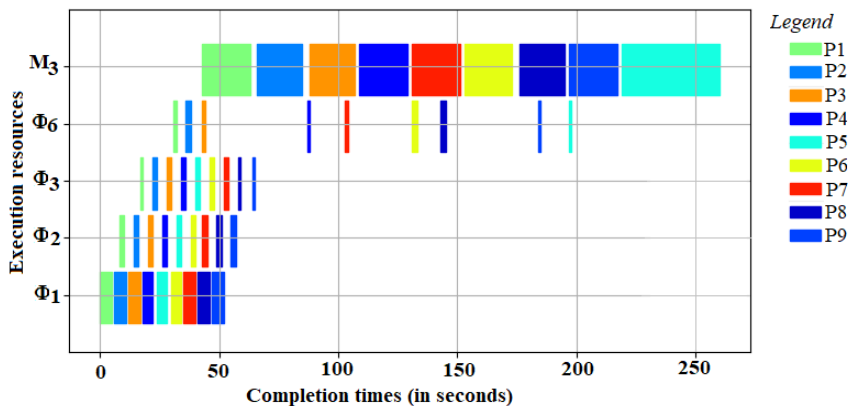


Fig. 4-24 Gantt chart for real completion time and sequence of each product

Table 4-4 Estimated completion time versus real completion time for each product

Order	Product_id	Production start(s)	Estimated C_i (s)	Real C_i (s)	$Tard_i$ (s)
MO_1	P_1	0	115	116	1
	P_2	6	137	137	0
	P_3	12	159	159	0
MO_2	P_4	17	183	183	0
	P_5	23	311	311	0
	P_6	29	225	225	0
MO_3	P_7	35	203	203	0
	P_8	41	245	245	0
	P_9	47	268	269	1

In order to show the impact of the sociability-based decision, the implementation test was also conducted to satisfy convergence and rushing priority using the consensus algorithm. This is revealed on resources Φ_1 and Φ_5 of Fig. 4-2 respectively. Meanwhile, during this test, the product_ID in Fig. 4-25 represents the first three product types in Fig. 4-13 respectively and products are launched on the system based on the sequence of $P_3 < P_2 < P_1$. The convergence and the rushing decisions are satisfied based on the availability of maximum amount of processing times of operations of each product type and a bottleneck machine M_3 . The product with highest processing times gets highest rushing priority in all converging points; see the entrance and exit sequence of the products in Fig. 4-26. To show the convergence, the LEs used the communication approached presented in Fig. 4-23(a) and product P_3 is released 40 seconds before the arrival of P_2 to Φ_1 and hence it had highest chance to pass Φ_1 before P_2 . However, it kept its current position in front of Φ_1 in order to satisfy the convergence decision; it waited until P_2 passes Φ_1 ; this is shown at the beginning of Fig. 4-26.

Product_ID	Due date	Tag_ID	Start date	Estimated end	Real end	Planned operations	Performed operations
1	250	E007000002199DCA	Tue Jul 02 15:51:56	Tue Jul 02 15:57:20	Tue Jul 02 15:56:54		
2	300	E004010000314399	Tue Jul 02 15:52:03	Tue Jul 02 15:56:32	Tue Jul 02 15:56:06		
3	700	E007000002199DB5	Tue Jul 02 15:52:49	Tue Jul 02 15:56:46	Tue Jul 02 15:56:20		

Fig. 4-25 Precedence relationship of real production start date and production end time of each product type

While the products proceed based on this sequence, machine M_3 as bottleneck is considered to satisfy the rushing decision. Even though P_1 had scheduled plan to arrive M_3 before that of P_2 , Product P_2 has arrived M_3 before P_1 , as it has highest processing time within the routing sheet, by agreeing with P_1 to loop in Zone-B of Fig. 4-2. This rescheduling is shown in Φ_5 of Fig. 4-26. During the communication, product P_3 has also got chance to be processed in M_3 before P_1 . This has happened because the amount of processing time of its operation in M_3 has less impact on the entire makespan. This is shown by the *position change()* function in Fig. 4-23(a).

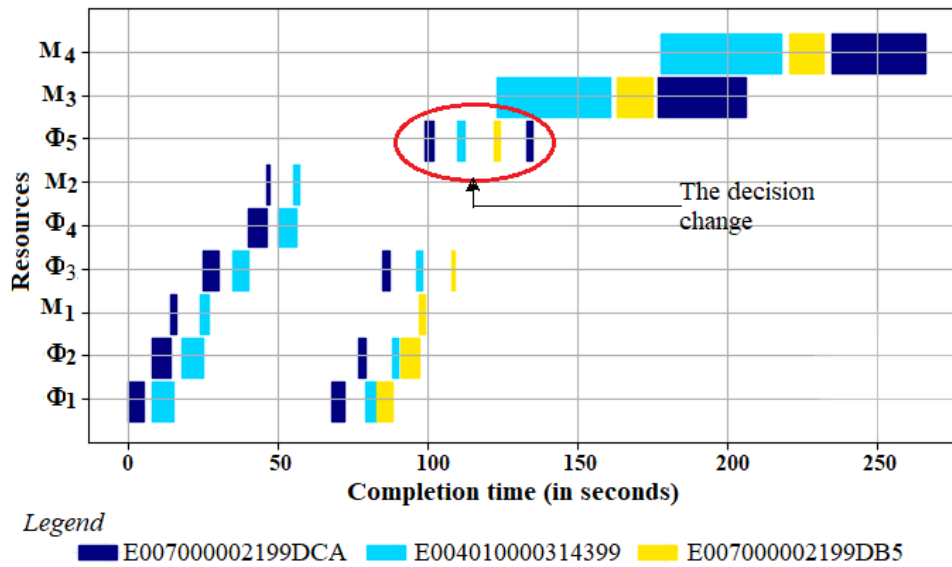


Fig. 4-26 Gantt chart for the real completion time of each product

4.6 Summary and discussion

This chapter mainly presented the implementability of the proposed control approaches using the experimental test-bed platform. Simulation and real applicability tests were launched to verify and validate the proposed algorithms by considering two industrial scenarios each with their own control problem. Local entities mainly the product and resource entities perform a self-organization decision; this self-organization may be either between products or products and resources, while they face a perturbing event. A supervising decision is also performed by white-collar entities mainly product or resource entities while the local entities face conflicting events among themselves. The resource entity, for example, changes itself from local decision maker to white-collar entity if the product entities started to decide by themselves. This is clearly presented in defining queue size in the first industrial context.

Coming to the results, the simulated results have verified that the sociability-based reactive decision made by contributing entities has brought significant improvements in tackling perturbation(s) occurred and hence in increasing resource utilizability, adaptability, and flexibility of a system compared to purely reactive decision; see Fig. 4-21. Despite the pure reactive decision has advantage to be the most generic in a rapidly changing context and regardless of communication protocols used and most disturbances addressed, the performance remains poor. This in turn has made the launched system to minimize its makespan by considering all necessary constraints. Meanwhile, when decision change (e.g., re-sequencing) is queried by either of decision entity, all entities in the sociability-based reactive decision-making approach have abled to dynamically satisfy this change through cooperative negotiation.

The applicability implementation test results have also validated the simulated results with some deviations caused by assumptions made during the simulation tests. It has clearly showed how product entities can keep their current position to satisfy some instances requested either by neighbor product entities or resource entities (this resource entity has been served as supervisory entity or local entity depending on the scenario in execution). In a real manufacturing system, production satisfaction is perceived if customers receive their finished products within the estimated completion time. Accordingly, in the applicability implementation experiment, the proposed decision-making algorithms have abled to almost equalize the estimated completion time of products with their real completion time; see Table 4-4 and Fig. 4-25.

GENERAL CONCLUSION AND FURTHER WORKS

In spite of the promising advancements on control systems and implementation paradigms, several issues remain yet unsolved. Some of these issues include:

- Autonomy of decisional entities and legal aspects to represent sociability challenges
- Development of interaction mechanisms and protocols; which are highly comprehended with challenges such as tendency towards centralization, specificity, difficulty to guarantee minimal level of operational performance etc.
- Design of decision models and control architectures to induce conceptual challenges
- Type and frequency of perturbations addressed

In order to structurally and behaviorally model and implement these challenges, plenty of researchers have been used Holonic reference architectures and Multi-agent system paradigms. Both of them provide an overall production performance and ensure reactivity face to unpredicted events. The holonic control architectures have been studied by the manufacturing control community and clearly showed that they are capable to minimize impact of perturbations on a holonic system. However, the setup for holonic structure requires high effort and hence development of holonic control applications is higher. Besides, the sociability-based multi-agent system has shown its effectiveness for controlling communication problems in pure computer science discipline. Nevertheless, its applicability for the manufacturing control problems was highly comprehended with challenges such as difficulty to guarantee minimal level of information and operational performance, difficulty of cooperation, lack of commercial platforms etc. Therefore, exploring on different agent-based control algorithms that help manufacturing components cooperate for their common goal is still a demanding research area.

On the other hand, in order to implement the holonic and agent-based control architectures, different structural and behavioral switching and interacting approaches have been introduced and used in the previous decades. Bio-inspired and physical-inspired interaction approaches have been importantly exercised and still they are importantly serving. However, decision entities in these approaches interact based on specific metrics and with limited reasoning capabilities. For instance, evaporation of pheromones with time in the biological interaction approach and reduction of attraction of intensity with distance in the potential field type interaction are important limitations. Another type of interaction approach that creates and manages metrics depending on probabilistic environment of the decision

entities is the sociability-based interaction approach. The probabilistic environment is traced and the required information is shared among the decision entities using the current state of each entity. Indeed, in this study, this human-inspired interaction mechanism is chosen to explore its use as a support for decision-making in reacting to unexpected events occurred. This is chosen over the others as it has not been studied as much as using the intelligence of other animals to solve the problems of manufacturing systems. Meanwhile, the human intelligence fits to smart and intelligent technological advances such as Internet of Things (IoT), Cyber Physical Systems (CPS), Cloud technologies, etc. The focus, of Industry 4.0, on the end-to-end digitization of all physical assets and integration into digital eco-systems with all value chain partners can also be achieved if decision entities within manufacturing systems cooperate as intelligent as human being.

The aim of this doctoral study was to propose human-inspired interaction approach (by means of negotiation and consensus-based decision-making algorithms) with agent-based hybrid control architecture. These decision-making algorithms for product driven control systems were implemented recursive hybrid control architecture. The performance of these algorithms was articulated using a real test-bed platform and compared with another decision-making approach called pure reactive control approach. Pure reactive is chosen as comparative control approach because agents in the platform were initially instantiated for this approach such that product agents calculate their production range at every move the shortest path to their final destination and machine agents are passive which only wait for instructions from product agents. Besides, the decision-making algorithms are verified on the basis of simulation experiment and validated with applicability implementation tests on the TRACILOGIS test-bed platform. During the applicability implementation tests, different number of palets has been used. These different numbers of palets were considered to show the scalability and flexibility of the proposed control approaches.

The first work was focused on negotiation between product and resource agents where the notion of intention of the agents produced was used to solve the problem of work-in-progress buildup caused by machine constraint. The idea was to get the best compromise after the occurrence of the disturbance in order to finish the production of late products as quickly as possible. This work is clearly presented in Mezgebe et al. (2018a), Mezgebe et al. (2018b), and Mezgebe et al. (2019b). The product entities and machine entities have worked cooperatively such that product entities compute their intention and broadcast into a system created that incorporates a machine entity that the product/s is approaching and neighbour

product entities j , $\forall j \in N_i$. After each product entity broadcast its intention, machine entities are expected to being fully utilized in order to return the initial cost of the machines within short period of time. Taking this precondition, product prioritization has to be validated (or supervised) by machine entities through calculating their utilization rate for every broadcasted intention of the product. The machine entity structurally shifts to supervision level from its local behavior in order to select products from set of candidates and hence to reduce and/or prevent occurrence of high buffer size. This is clearly presented by defining queue size using critical ratio as decision-making parameters in the applicability implementation experiment accompanied.

Product lateness, machine utilization rate, and makespan were chosen as key production performance indicators to validate this control approach. Three of these performance indicators have shown significant advantage of the proposed approach in adapting unexpected event occurred and minimizing its impact on manufacturing process. This has ensured that the impact of sociability-based decision-making to deliver produced products to their customer near to the planned date and it also confirmed the suitability of the proposed control approach for industrial applicability.

However, it has left to fully enrich this approach as decision entities (or agents) may decide alone by rejecting the negotiated offer while they are on their route. This rejection may occur because the decision entities do not negotiate using a common state x_i . At this instant, another decision-making algorithm is proposed through Mezgebe et al. (2019a). It was proposed using consensus theory (control/command issue) with an objective to make agents asymptotically reach a final agreement using a common state x_i for the best interest of the whole. This algorithm is implemented to solve rush order as perturbation, which has a big impact on the performance of manufacturing systems. RFID tagged product entities carrying their information and data and machine entities with their execution capacity were made to execute the system. A machine entity with higher product operation's processing time was chosen as constraint during the decision. Continuous negotiation and convergence to a common offer is required before these entities mainly the product entities start decision. A product entity (which represents the entire manufacturing order) with urgent priority structurally shifts to white-collar entity to declare the rush order. Then after, it turned back to its local behavior in order to negotiate with the proceeding product entities by presenting its current state.

Convergence state, product lateness, and makespan are used as key performance indicators to compare results of this algorithm with that of PRC (or myopic decision-making for convenience). Accordingly, the consensus-based decision making has shown clear behavioral pattern (or stability) to satisfy the rush order as a perturbing event. This was achieved by making product agents to minimize non-value adding activities created after the rush order. The results obtained from convergence state, for example, presented that product entity's disagreement level for the rush order offer in the PRC becomes extensively higher than that of the consensus-based decision. Consequently, this disagreement level has made product entities of all manufacturing orders to exit from the routing sheet tardy.

Finally, by considering few decision parameters from both decision-making algorithms, some concluding remarks and discussions can be drawn as follow.

Harmonized estimated completion time and real completion time in order to create an end-to-end alignment between customer's deliverability requirement and manufacturing process of a shop floor, harmonizing the planned completion time of products with the real completion time is important. This requires efficient assignment of tasks to resources within manufacturing system and this makes real customers to receive their orders as early as possible.

Achieving due date schedule changes should be observed according to product due date values. For example, in industrial context-I, critical ratio is used by machine entity to verify the negotiation-based reactive control approach. When critical ratio of a product is too low or when the given due date is very short, product will not be supervised by the machine and switch to pure reactive mode in order to complete its route within the given short due date, see Eq. (3-6). Thus, in theory, it is possible to observe the emergence of misbehavior where all product entities switch to non-collective decision (or the pure reactive) in an extremely strained situation of all products.

Resource field view this is also viewed while decision entities negotiate to reach at their final goal. For instance, for the first industrial context, machine entities select a defined size of products from set of candidates and visitors which will be processed in their work space. The size of this space has an important impact on the proposed control approaches. If the size is low, lack of proactivity is observed because the machine only selects the first or the closest products. If the size is huge, a meaningless waiting time will occur because the selection of distant product will let other closer products to wait during the routing sheet. This is also

verified and validated using the simulation test and applicability implementation experiments presented in section 4.3, section 4.4, and section 4.5.

Size of waiting queue it corresponds to the number of products simultaneously selected by the machine entity that they are approaching. If the value of this size is low, the machine shortage risk is high. If the value is high, it might lead to pure reactive mode by most products since each product arriving to the machine is directly selected. Specifically, the minimal queuing time and maximal arrival time of a newly selected product must be compared in order to prevent the machine shortage.

Further work

It is illustrated in chapters three and four that the contribution of this study relies on designing a control system and decision-making algorithms to deal with perturbed manufacturing systems. We are aware that the proposed control approaches have been concentrated in solving problems sourced from internal and external environments using the academic experimental test-bed platform. Accordingly, promising results (both from the simulation experiment and applicability implementation test) have been demonstrated from these control approaches. According to the results, many works are already engaged in our laboratory to define an optimal number of communicating agents, define communication methods among agents, to define other decision states etc. in regard of the expected partner company's objectives. However, the application implementation tests using the TRACILOGIS platform was not fully addressed. Thus, additional experiments and tests must be conducted considering:

- a) Saturated routing sheet or complex communication constraints in order to see the scalability and flexibility of the control approaches
- b) Large scale of product and resource entities (by considering more than nine products) in order to further investigate the adaptability of these entities in more complex environment

On the other hand, the implementation architecture is closely positioned to distributed reactive control system. This indicates that even though the white-collar entities are involved in the decision-making process, the required decisions are made at local level. This clearly presents the limitation of the proposed control system and decision-making algorithms. In order to bring significant impact of the decision entities, they must equally oscillate towards the distributed reactive control mode and centrally predictive control approach depending on the type of perturbation(s) a system faces. The slack interaction of the white-collar entities

must be changed to direct interaction and a full study of decision parameters needs to be showed. This will be the significant work expected either as a continuation of this study or by some other external researchers. Game theory, particle swarm optimization etc. can also be used to design the required algorithms as alternative interaction approaches for the sociability-based interaction mechanism introduced herein. This cross check-up must also be taken as the future work of this study. Besides, the concept of human-inspired in this study is used to show the interaction and/or communication mechanism used by decision entities. However, the study did not address a man-machine interaction to optimize such a disturbed manufacturing system. Integrating human in to the manufacturing execution system that help to optimally manage a disturbance must be considered the future works. Trentesaux and Millot (2016) have worked on this topic.

Last but not least, identification and discussion of perturbations rather than the perturbations addressed in this study must be studied relative to control architectures and decision parameters used to reduce and model them respectively. This should be done by expecting that it will give a broad insight to researchers and scholars which perturbations do have higher impact on manufacturing system and which do not. It also helps to design a new control architecture which is not yet addressed. Finally, The TRACILOGIS platform based applicability implementation tests conducted in this study must be mutated into real manufacturing system in order to see its effectiveness and to help detect some other problems. Moreover, the study has to be extended to other industrial sectors such as logistics management, transportation system etc.

REFERENCES

- 1) Adhau, S., Mittal, M. L., & Mittal, A. (2012). A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. *Engineering Applications of Artificial Intelligence*, 25(8), 1738-1751
- 2) Amgoud, L., Belabbès, S., & Prade, H. (2005, July). Towards a formal framework for the search of a consensus between autonomous agents. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems* (pp. 537-543). ACM
- 3) Anderson, C., & Bartholdi, J. J. (2000). Centralized versus decentralized control in manufacturing: lessons from social insects. *Complexity and complex systems in industry*, 92-105
- 4) Aqlan, F., Ahmed, A., Ashour, O., Shamsan, A., & Hamasha, M. M. (2017). An approach for rush order acceptance decisions using simulation and multi-attribute utility theory. *European Journal of Industrial Engineering*, 11(5), 613-630.
- 5) Aydogan, R., Baarslag, T., Hindriks, K.V., Jonker, C.M., Yolum, P. (2013). Heuristic-Based Approaches for CP-Nets in Negotiation. In *Complex automated negotiations: Theories, models and software competitions* (pp. 113-123). Berlin, Heidelberg: Springer
- 6) Bae, S. (2019). Recursion. In *JavaScript Data Structures and Algorithms* (pp. 99-115). Apress, Berkeley, CA
- 7) Baïna, S. (2006). INTEROPERABILITE DIRIGEE PAR LES MODELES: Une Approche Orientée Produit pour l'interopérabilité des systèmes d'entreprise (Doctoral dissertation, Université Henri Poincaré-Nancy I)
- 8) Barbosa, J., Leitão, P., Adam, E., & Trentesaux, D. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Computers in Industry*, 66, 99-111
- 9) Bel-Enguix, G., Grando, M. A., & Jiménez-López, M. D. (2007, September). An interaction protocol for agent communication. In *International Central and Eastern European Conference on Multi-Agent Systems* (pp. 62-72). Springer, Berlin, Heidelberg.
- 10) Bellifemine, F., Caire, G., Trucco, T., & Rimassa, G. (2002). Jade Programmer's Guide-Jade version 2.6

- 11) Bellifemine, F., Caire, G., Trucco, T., & Rimassa, G. (2010). Java Agent Development Framework (JADE) Programmer's guide. TILAB, formerly CSELT and University of Parma
- 12) Benjaafar, S., Heragu, S. S., & Irani, S. A. (2002). Next generation factory layouts: research challenges and recent progress. *Interfaces*, 32(6), 58-76.
- 13) Blum, C. and Sampels, M. (2004). An Ant colony optimization algorithm for shop scheduling problems. *Journal of Mathematical Modelling and Algorithms*, 3(3), 285-308.
- 14) Borangiu, T., Răileanu, S., Berger, T., and Trentesaux, D. (2015). Switching Mode Control Strategy in Manufacturing Execution Systems. *International Journal of Production Research* 53(7): 1950–1963
- 15) Borisoff, D., & Victor, D. A. (1989). *Conflict management: A communication skills approach*. Englewood Cliffs, NJ: Prentice Hall.
- 16) Botti, V., & Giret, A. (2008). *ANEMONA: A multi-agent methodology for Holonic Manufacturing Systems*. Springer Science & Business Media.
- 17) Botti, V., Barber, F., Crespo, A., Onaindia, E., García-Fornes, A., Ripoll, I., & Hernández, L. (1995). A temporal blackboard for a multi-agent environment. *Data & knowledge engineering*, 15(3), 189-211.
- 18) Brennan, R. W. (2000). Performance comparison and analysis of reactive and planning-based control architectures for manufacturing. *Robotics and Computer-Integrated Manufacturing*, 16(2), 191-200.
- 19) Browne, J., Dubois, D., Rathmill, K., Sethi, S. P., & Stecke, K. E. (1984). Classification of flexible manufacturing systems. *The FMS magazine*, 2(2), 114-117.
- 20) Cao, Y., Yu, W., Ren, W., & Chen, G. (2012). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1), 427-438
- 21) Cardin, O., & Castagna, P. (2009). Using online simulation in Holonic manufacturing systems. *Engineering Applications of Artificial Intelligence*, 22(7), 1025-1033
- 22) Cardin, O., Derigent, W., & Trentesaux, D. (2018). Evolution of holonic control architectures towards Industry 4.0: A short overview. *IFAC-PapersOnLine*, 51(11), 1243-1248
- 23) Cardin, O., Trentesaux, D., Thomas, A., Castagna, P., Berger, T., & El-Haouzi, H. B. (2017). Coupling predictive scheduling and reactive control in manufacturing hybrid

- control architectures: state of the art and future challenges. *Journal of Intelligent Manufacturing*, 28(7), 1503-1517.
- 24) Caridi, M., & Cavalieri, S. (2004). Multi-agent systems in production planning and control: an overview. *Production Planning & Control*, 15(2), 106-118
 - 25) Chirn, J. L., & McFarlane, D. C. (2000, September). A holonic component-based approach to reconfigurable manufacturing control architecture. In *Proceedings 11th International Workshop on Database and Expert Systems Applications, 2000* (pp. 219-223). IEEE
 - 26) CHRISTENSEN, J. (1994). Holonic manufacturing systems-initial architecture and standards directions. In *1st Euro. Conf. on HMS*, Hannover, Germany, 1994
 - 27) Devezas, T., & Sarygulov, A. (2017). *Industry 4.0: Entrepreneurship and Structural Change in the New Digital Landscape*. Gewerbestrasse 11, 6330 Cham, Springer, Switzerland
 - 28) Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269-271.
 - 29) Dilts, D. M., Boyd, N. P., & Whorms, H. H. (1991). The evolution of control architectures for automated manufacturing systems. *Journal of manufacturing systems*, 10(1), 79-93
 - 30) Dimarogonas, D.V., Kyriakopoulos, K.J. (2007). On the rendezvous problem for multiple nonholonomic agents. *IEEE Trans. Autom. Control*, 52(5), 916-922
 - 31) Dimopoulos, Y., & Moraitis, P. (2006, December). Multi-agent coordination and cooperation through classical planning. In *Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology* (pp. 398-402). IEEE Computer Society.
 - 32) Dolgui Alexandre & Proth Jean-Marie (2010). *Supply Chain Engineering, useful Methods and Techniques*. Springer Science & Business Media, ISBN: 1-84996-017-8, London Dordrecht Heidelberg, New York, USA
 - 33) Dorri, A., Kanhere, S. S., & Jurdak, R. (2018). Multi-agent systems: A survey. *IEEE Access*, 6, 28573-28593
 - 34) Duffie, N. A. (1996). Heterarchical control of highly distributed manufacturing systems. *International Journal of Computer Integrated Manufacturing*, 9(4), 270-281
 - 35) Duffie, N. A. (2008). Challenges in design of heterarchical controls for dynamic logistic systems. *Dynamics in Logistics*, 3-24

- 36) Ehteshami, B., Petrakian, R. G., & Shabe, P. M. (1992). Trade-offs in cycle time management: hot lots. *IEEE Transactions on Semiconductor Manufacturing*, 5(2), 101-106.
- 37) El Haouzi H, Pétrin JF, Thomas A (2009) Design and validation of a product-driven control system based on a six sigma methodology and discrete event simulation. *Prod. Plan. Control*, 20(6), 510–524
- 38) El Haouzi, H. (2008). Approche méthodologique pour l'intégration des systèmes contrôlés par le produit dans un environnement de juste-à-temps: Application à l'entreprise Trane (Doctoral dissertation, Université Henri Poincaré-Nancy I)
- 39) El Haouzi, H. B. (2017). Contribution à la conception et à l'évaluation des architectures de pilotage des systèmes de production adaptables: vers une approche anthropocentrée pour la simulation et le pilotage (Habilitation à diriger des recherches, Université de lorraine)
- 40) El Haouzi, H. B., Thomas, A., & Charpentier, P. (2013). Toward adaptive modelling & simulation for IMS: The Adaptive Capability Maturity Model and future challenges. *IFAC Proceedings Volumes*, 46(7), 174-179
- 41) ElMaraghy, H. A. (2005). Flexible and reconfigurable manufacturing systems paradigms. *International journal of flexible manufacturing systems*, 17(4), 261-276.
- 42) Espejo, R., & Reyes, A. (2011). Organizational systems: Managing complexity with the viable system model. *Springer Science and Business Media*, URL: <https://books.google.com/books>
- 43) Farid, A. (2004). *A review of holonic manufacturing systems literature*. Univ. Camb. Inst. Manuf. Camb. UK Tech Rep
- 44) Feld, W. M. (2000). *Lean Manufacturing: tools, techniques, and how to use them*. CRC press.
- 45) Fisher, K. (1999). Agent-based design of holonic manufacturing systems. *Robotics and autonomous Systems*, 27(1-2), 3-13
- 46) Fister Jr, I., Yang, X.S., Fister, I., Brest, J. and Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*
- 47) García, A. P., Oliver, J., & Gosch, D. (2010, October). An intelligent agent-based distributed architecture for Smart-Grid integrated network management. In *IEEE Local Computer Network Conference* (pp. 1013-1018). IEEE
- 48) Geissbauer, R., Vedso, J., & Schrauf, S. (2016). *Industry 4.0: Building the digital enterprise, 2016 global industry 4.0 survey*. PwC, Munich.

- 49) Gilmore, J. H., & Pine, B. J. (1997). The four faces of mass customization. *Harvard business review*, 75(1), 91-102.
- 50) Giovannini, A. (2015). A knowledge representation framework for the design and the evaluation of a product variety (Doctoral dissertation, Université de Lorraine)
- 51) Giret, A., & Botti, V. (2006). From system requirements to holonic manufacturing system analysis. *International journal of production research*, 44(18-19), 3917-3928.
- 52) Governatori, G., Dumas, M., Ter Hofstede, A. H., & Oaks, P. (2001, May). A formal approach to protocols and strategies for (legal) negotiation. In *Proceedings of the 8th international conference on Artificial intelligence and law* (pp. 168-177). ACM.
- 53) Gunasekaran, A. (1999). Agile manufacturing: a framework for research and development. *International journal of production economics*, 62(1-2), 87-105
- 54) Gunes, S. Peter, T. Givargis, and F. Vahid (2014). A survey on concepts, applications, and challenges in cyber-physical systems. *THIS*, 8(12), 4242–4268
- 55) Gupta, M. C., & Boyd, L. H. (2008). Theory of constraints: a theory for operations management. *International Journal of Operations & Production Management*, 28(10), 991-1012
- 56) Hankel, M., & Rexroth, B. (2015). The reference architectural model industrie 4.0 (rami 4.0). *ZVEI, April*, 410.
- 57) Herrera, C., Belmokhtar-Berraf, S., Thomas, A., & Parada, V. (2016). A reactive decision-making approach to reduce instability in a master production schedule. *International Journal of Production Research*, 54(8), 2394-2404.
- 58) Herrera, C., Thomas, A., & Parada, V. (2014). A product-driven system approach for multilevel decisions in manufacturing planning and control. *Production & Manufacturing Research*, 2(1), 756-766
- 59) Herrera, C., Thomas, A., & Vera, V. (2013). Simulation of a Hybrid Product-Driven System for Manufacturing Planning and Control. *IFAC Proceedings Volumes*, 46(7), 69-74
- 60) Herrera, C., Thomas, A., Belmokhtar, S., & Pannequin, R. (2011). A viable system model for product-driven systems. In *International Conference on Industrial Engineering and Systems Management, IESM 2011* (CDROM)
- 61) Holling, C. S. (1973). Resilience and stability of ecological systems. *Annual review of ecology and systematics*, 4(1), 1-23

- 62) Holvoet, T., Weyns, D., and Valckenaers, P. (2009). Patterns of Delegate Mas. *Third IEEE International Conference on Self-adaptive and Self-organizing Systems, 2009. SASO'09. IEEE. September 1-9. doi:10.1109/SASO.2009.31*
- 63) Indriago, C., Cardin, O., Rakoto, N., Castagna, P., & Chacòn, E. (2016). H²CM: A holonic architecture for flexible hybrid control systems. *Computers in industry*, 77, 15-28
- 64) Isern, D., Sánchez, D., & Moreno, A. (2011). Organizational structures supported by agent-oriented methodologies. *Journal of Systems and Software*, 84(2), 169-184
- 65) Jarvis, J., Jarvis, D., & Rönquist, R. (2008). *Holonic execution: A BDI approach (Vol. 106)*. Springer Science & Business Media
- 66) Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. J., & Sierra, C. (2001). Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2), 199-215.
- 67) Jimenez, J. F., Bekrar, A., Zambrano-Rey, G., Trentesaux, D., & Leitão, P. (2017). Pollux: a dynamic hybrid control architecture for flexible job shop systems. *International Journal of Production Research*, 55(15), 4229-4247
- 68) John, H. & Blackstone, Jr. (2010). *American Production and Inventory Control Society Dictionart*. Thirteenth edition, APICS The Association for Operations Management, USA
- 69) Kao, H. A., Jin, W., Siegel, D., & Lee, J. (2015). A cyber physical interface for automation systems-methodology and examples. *Machines*, 3(2), 93-106.
- 70) Karanasos, K., Rao, S., Curino, C., Douglas, C., Chaliparambil, K., Fumarola, G. M., & Sakalanaga, S. (2015, July). Mercury: Hybrid Centralized and Distributed Scheduling in Large Shared Clusters. In *USENIX Annual Technical Conference* (pp. 485-497).
- 71) Kaur, M., Sandhu, M., Mohan, N., & Sandhu, P. S. (2011). RFID technology principles, advantages, limitations & its applications. *International Journal of Computer and Electrical Engineering*, 3(1), 151.
- 72) Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing* (pp. 187-219). Springer, Boston, MA.
- 73) Klein, T. (2008). Le kanban actif pour assurer l'interopérabilité décisionnelle centralisé/distribué Application à un industriel de l'ameublement (Doctoral dissertation, Université Henri Poincaré-Nancy I)
- 74) Klimeš, J. (2014). Using formal concept analysis for control in cyber-physical systems. *Procedia Engineering*, 69, 1518-1522

- 75) Knabe, T., Schillo, M., & Fischer, K. (2002). Improvements to the FIPA contract net protocol for performance increase and cascading applications. In *In International Workshop for Multi-Agent Interoperability at the German Conference on AI (KI-2002)*.
- 76) Koch, V., Kuge, S., Geissbauer, R., & Schrauf, S. (2014). *Industry 4.0: Opportunities and challenges of the industrial internet*. Strategy & PwC.
- 77) Koestler, A. (1989). *The Ghost in the Machine*, Arkana Books, London
- 78) Komma, V. R., Jain, P. K., & Mehta, N. K. (2011). An approach for agent modeling in manufacturing on JADE™ reactive architecture. *The International Journal of Advanced Manufacturing Technology*, 52(9-12), 1079-1090
- 79) Koren, Y. (2010). *The global manufacturing revolution: product-process-business integration and reconfigurable systems*. John Wiley & Sons.
- 80) Kraus, S. (1997). Negotiation and cooperation in multi-agent environments. *Artificial intelligence*, 94(1-2), 79-97.
- 81) Kraus, S. (2001). *Strategic negotiation in multiagent environments*. MIT press
- 82) Labib, A. W., & Yuniarto, M. N. (2005). Intelligent real time control of disturbances in manufacturing systems. *Journal of Manufacturing Technology Management*, 16(8), 864-889
- 83) Laws, A. G., Taleb-Bendiab, A., & Wade, S. J. (2005, September). Genetically modified software: Realizing viable autonomic agency. In *Workshop on Radical Agent Concepts* (pp. 184-196). Springer, Berlin, Heidelberg
- 84) Lee, D. and Spong, M.W. (2007). Stable flocking of multiple inertial agents on balanced graphs. *IEEE Trans. Autom. Control*, 52(8), 1469-1475
- 85) Lee, E. A. (2006, October). Cyber-physical systems-are computing foundations adequate. In *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap* (Vol. 2, pp. 1-9)
- 86) Lee, J., Bagheri, B., & Kao, H. A. (2015). A Cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18-23.
- 87) Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7), 979-991
- 88) Leitão, P., & Restivo, F. (2006). ADACOR: A Holonic architecture for agile and adaptive manufacturing control. *Computers in industry*, 57(2), 121-130
- 89) Leitão, P., Barbosa, J., & Trentesaux, D. (2012). Bio-inspired multi-agent systems for reconfigurable manufacturing systems. *Engineering Applications of Artificial Intelligence*, 25(5), 934-944

- 90) Leitao, P., Colombo, A. W., & Restivo, F. J. (2005). ADACOR: A collaborative production automation and control architecture. *IEEE Intelligent Systems*, 20(1), 58-66.
- 91) Liang, Y.C. and Smith, A.E. (2004). An Ant colony optimization algorithm for the redundancy allocation problem (RAP). *IEEE Transactions on reliability*, 53(3), 417-423
- 92) Lin, G. Y. J., & Solberg, J. J. (1992). Integrated shop floor control using autonomous agents. *IIE Trans*, 24(1992), 57-71.
- 93) Mabin, V. (2010). *Goldratt's "Theory of Constraints" Thinking Processes: A Systems Methodology linking Soft with Hard (1999)*
- 94) Madureira, A., Pereira, I., Pereira, P., & Abraham, A. (2014). Negotiation mechanism for self-organized scheduling system with collective intelligence. *Neurocomputing*, 132, 97-110.
- 95) Maturana, F., Shen, W., & Norrie, D. H. (1999). MetaMorph: an adaptive agent-based architecture for intelligent manufacturing. *International Journal of Production Research*, 37(10), 2159-2173
- 96) McArthur, S. D., Davidson, E. M., Catterson, V. M., Dimeas, A. L., Hatziaargyriou, N. D., Ponci, F., & Funabashi, T. (2007). Multi-agent systems for power engineering applications—Part I: Concepts, approaches, and technical challenges. *IEEE Transactions on Power systems*, 22(4), 1743-1752
- 97) McCullen, P., Saw, R., Christopher, M., & Towill, D. (2006, January). The F1 supply chain: adapting the car to the circuit-the supply chain to the market. In *Supply chain forum: an international journal* (Vol. 7, No. 1, pp. 14-23). Taylor & Francis
- 98) McFarlane, D., Sarma, S., Chirn, J. L., Wong, C., & Ashton, K. (2003). Auto ID systems and intelligent manufacturing control. *Engineering Applications of Artificial Intelligence*, 16(4), 365-376
- 99) Mezgebe, T. T., Demesure, G., El Haouzi, H. B., Pannequin, R., Thomas, A. (2019a). CoMM: A Consensus algorithm for Multi-agent based Manufacturing system to deal with perturbation. *International Journal of Advanced Manufacturing Technology, Springer*, 105(9), 3911-3926
- 100) Mezgebe, T. T., El Haouzi, H. B., Demesure, G., & Thomas, A. (2018b). A Negotiation-based control approach for disturbed industrial context. *IFAC-PapersOnLine*, 51(11), 1255-1260

- 101) Mezgebe, T. T., El Haouzi, H. B., Demesure, G., Pannequin, R., & Thomas, A. (2019b). Multi-agent systems negotiation to deal with dynamic scheduling in disturbed industrial context. *Journal of Intelligent Manufacturing*, 1-16. DOI: 10.1007/s10845-019-01515-7
- 102) Mezgebe, T. T., El Haouzi, H. B., Demesure, G., Pannequin, R., & Thomas, A. (2018a). A Negotiation Scenario Using an Agent-Based Modeling Approach to Deal with Dynamic Scheduling. In *Service Orientation in Holonic and Multi-Agent Manufacturing* (pp. 381-391). Springer, Cham
- 103) Misra, J. (1986). Distributed discrete-event simulation. *ACM Computing Surveys (CSUR)*, 18(1), 39-65
- 104) Mohammadi, A., & Akl, S. G. (2005). *Scheduling algorithms for real-time systems*. School of Computing Queens University, Tech. Rep.
- 105) Monteiro, T., Anciaux, D., Espinasse, B., Ferrarini, A., Labarthe, O., Montreuil, B., & Roy, D. (2008). The Interest of Agents for Supply Chain Simulation
- 106) Monteserin, A., & Amandi, A. (2011). Argumentation-based negotiation planning for autonomous agents. *Decision Support Systems*, 51(3), 532-548
- 107) Moore, K. L., & Lucarelli, D. (2007). Decentralized adaptive scheduling using consensus variables. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 17(10-11), 921-940.
- 108) Moore, R., & Scheinkopf, L. (1998). *Theory of Constraint and Lean manufacturing: Friends or Foes*. Chesapeake Consulting Inc
- 109) Mukhopadhyay, S. K., & Sahu, S. K. (1996). Priority-based tool allocation in a flexible manufacturing system. *International journal of production research*, 34(7), 1995-2018
- 110) Nagarajan, M., & Sošić, G. (2008). Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *European Journal of Operational Research*, 187(3), 719-745
- 111) Nicholas, J. (2018). *Lean production for competitive advantage: A comprehensive guide to lean methodologies and management practices*. Productivity Press
- 112) Notarstefano, G., Egerstedt, M., & Haque, M. (2011). Containment in leader-follower networks with switching communication topologies. *Automatica*, 47(5), 1035-1040.
- 113) Novas, J. M., Van Belle, J., Saint Germain, B., & Valckenaers, P. (2013). A collaborative framework between a scheduling system and a holonic manufacturing execution system. In *Service orientation in holonic and multi agent manufacturing and robotics* (pp. 3-17), Springer, Berlin, Heidelberg

- 114) Noyel, M., & Pisaneschi, T. (2011). Mise en place d'un système RFID pour une entreprise de panneaux laqués haute finition (Doctoral dissertation, UHP-Université Henri Poincaré).
- 115) Noyel, M., Thomas, P., Charpentier, P., Thomas, A., & Beaupretre, B. (2013). Improving production process performance thanks to neuronal analysis. *IFAC Proceedings Volumes*, 46(7), 432-437
- 116) Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control*, 51(3), 401-420
- 117) Olfati-Saber, R., Fax, J.A. and Murray, R.M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215-233
- 118) Olfati-Saber, R., Murray, R.M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 49(9), 1520-1533
- 119) Ottaway, T. A., & Burns, J. R. (2000). An adaptive production control system utilizing agent technology. *International Journal of Production Research*, 38(4), 721-737
- 120) Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of scheduling*, 12(4), 417-431
- 121) Ou-Yang, C., & Lin, J. S. (1998). The development of a hybrid hierarchical/heterarchical shop floor control system applying bidding method in job dispatching. *Robotics and Computer-Integrated Manufacturing*, 14(3), 199-217
- 122) Pach C, Berger T, Bonte T, Trentesaux D (2014b). ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. *Comput. Ind*, 65(4), 706-720
- 123) Pach, C., Bekrar, A., Zbib, N., Sallez, Y., & Trentesaux, D. (2012). An effective potential field approach to FMS holonic heterarchical control. *Control Engineering Practice*, 20(12), 1293-1309
- 124) Pach, C., Berger, T., Sallez, Y., Bonte, T., Adam, E., & Trentesaux, D. (2014a). Reactive and energy-aware scheduling of flexible manufacturing systems using potential fields. *Computers in Industry*, 65(3), 434-448.
- 125) Panizzolo, R., & Garengo, P. (2013). Using theory of constraints to control manufacturing systems: a conceptual model. *Industrial Engineering and Management*, 3(3), 1-9.
- 126) Pannequin, R. (2007). Proposition d'un environnement de modélisation et de test d'architectures de pilotage par le produit de systèmes de production (Doctoral dissertation, Université Henri Poincaré-Nancy I)

- 127) Pannequin, R., & Thomas, A. (2012). Another interpretation of Stigmergy for product-driven systems architecture. *Journal of Intelligent Manufacturing*, 23(6), 2587-2599
- 128) Pannequin, R., Morel, Thomas, A. (2009). The performance of product-driven manufacturing control: An emulation-based benchmarking study. *Computers in Industry*, 60(3), 195-203
- 129) Park, H. S., & Tran, N. H. (2011). An autonomous manufacturing system for adapting to disturbances. *The International Journal of Advanced Manufacturing Technology*, 56(9-12), 1159-1165
- 130) Pfeiffer A., Kádár B., Monostori L., Karnok D. (2008). Simulation as one of the core technologies for digital enterprises: assessment of hybrid rescheduling methods. *Int. J. of Comput. Integr. Manuf.*, 21(2), 206–214
- 131) Pine, B. J., & Davis, S. (1993). *Mass customization: the new frontier in business competition*. Harvard Business School Press, Boston
- 132) Pujo, P., Broissin, N., & Ounnar, F. (2009). PROSIS: An isoarchic structure for HMS control. *Engineering Applications of Artificial Intelligence*, 22(7), 1034-1045.
- 133) Quintanilla, F. G., Cardin, O., L'anton, A., & Castagna, P. (2016). A modeling framework for manufacturing services in service-oriented holonic manufacturing systems. *Engineering Applications of Artificial Intelligence*, 55, 26-36
- 134) Rahwan, I., Ramchurn, S.D., Jennings, N.R., Mcburney, P., Parsons, S. and Sonenberg, L. (2003). Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4), 343-375.
- 135) Ren, W., & Beard, R. W. (2008). Overview of consensus algorithms in cooperative control. *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*, 3-22.
- 136) Rey, G. Z., Bonte, T., Prabhu, V., & Trentesaux, D. (2014). Reducing myopic behavior in FMS control: A semi-heterarchical simulation–optimization approach. *Simulation Modelling Practice and Theory*, 46, 53-75
- 137) Rey, G. Z., Pach, C., Aissani, N., Bekrar, A., Berger, T., & Trentesaux, D. (2013). The control of myopic behavior in semi-heterarchical production systems: A holonic framework. *Engineering Applications of Artificial Intelligence*, 26(2), 800-817.
- 138) Rojko, A. (2017). Industry 4.0 concept: background and overview. *International Journal of Interactive Mobile Technologies (iJIM)*, 11(5), 77-90
- 139) Rosenschein, J. S., & Zlotkin, G. (1994). *Rules of encounter: designing conventions for automated negotiation among computers*. MIT press.

- 140) Ryu, K., & Jung, M. (2003). Agent-based fractal architecture and modelling for developing distributed manufacturing systems. *International Journal of Production Research*, 41(17), 4233-4255
- 141) Saad, S. M., & Gindy, N. N. (1998). Handling internal and external disturbances in responsive manufacturing environments. *Production Planning & Control*, 9(8), 760-770
- 142) Saadat, M., Tan, M. C. L., & Owliya, M. (2008, September). Changes and disturbances in manufacturing systems: A comparison of emerging concepts. In *2008 World Automation Congress* (pp. 1-6). IEEE
- 143) Saharidis, G. K., Dallery, Y., & Karaesmen, F. (2006). Centralized versus decentralized production planning. *RAIRO-Operations Research*, 40(2), 113-128.
- 144) Schwaninger, M. (2006). Design for viable organizations: The diagnostic power of the viable system model. *Kybernetes: The International Journal of Systems & Cybernetics*, 35(7-8), 955-966.
- 145) Seyboth, G. S., Dimarogonas, D. V., & Johansson, K. H. (2013). Event-based broadcasting for multi-agent average consensus. *Automatica*, 49(1), 245-252.
- 146) Sha, D. Y., & Lin, H. H. (2010). A multi-objective PSO for job-shop scheduling problems. *Expert Systems with Applications*, 37(2), 1065-1070.
- 147) Shah, R., & Ward, P. T. (2007). Defining and developing measures of lean production. *Journal of operations management*, 25(4), 785-805.
- 148) Shannon, R. E. (1998, December). Introduction to the art and science of simulation. In *1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)* (Vol. 1, pp. 7-14). IEEE.
- 149) Sharma, P. (2015). Discrete-event simulation. *Int. J. Sci. Technol. Res*, 4(04), 136-140
- 150) Shi, Y. (2001, May). Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)* (Vol. 1, pp. 81-86). IEEE.
- 151) Shingo S. & Dillon A. P. (1989). *A study of the Toyota production system: From an Industrial Engineering Viewpoint*. CRC Press
- 152) Sierra, C., Jennings, N. R., Noriega, P., & Parsons, S. (1997, July). A framework for argumentation-based negotiation. In *International Workshop on Agent Theories, Architectures, and Languages* (pp. 177-192). Springer, Berlin, Heidelberg.
- 153) Sinha, A., Ghose, D. (2006). Generalization of linear cyclic pursuit with application to rendezvous of multiple autonomous agents. *IEEE Trans. Autom. Control*, 51(11), 1819–1824

- 154) Skorin-Kapov, J., & Vakharia, A. J. (1993). Scheduling a flow-line manufacturing cell: a tabu search approach. *The International Journal of Production Research*, 31(7), 1721-1734
- 155) Smith, R. G. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, (12), 1104-1113.
- 156) Sousa, P., Ramos, C., & Neves, J. (2007). Scheduling in holonic manufacturing systems. In *Process Planning and Scheduling for Distributed Manufacturing* (pp. 167-190). Springer, London.
- 157) Stich, V., Schmidt, C., Meyer, J. C., & Wienholdt, H. (2009, May). Viable production system for adaptable and flexible production planning and control processes. In *Proceedings of POMS 20th Annual Conference. POMS, Orlando, Florida, USA*
- 158) Su, H., Wang, X., Lin, Z. (2009). Synchronization of coupled harmonic oscillators in a dynamic proximity network. *Automatica*, 45(10), 2286–2291
- 159) Suda, H. (1989). Future factory system formulated in Japan. *Japanese Journal of Advanced Automation Technology*, 1(1), 15-25.
- 160) Tang, D., Gu, W., Wang, L., & Zheng, K. (2011). A neuroendocrine-inspired approach for adaptive manufacturing system control. *International journal of production research*, 49(5), 1255-1268
- 161) Tanweer, M. R., & Sundaram, S. (2014, April). Human cognition inspired particle swarm optimization algorithm. In *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)* (pp. 1-6). IEEE
- 162) Thierry, C., Thomas, A., & Bel, G. (2008). *Simulation for supply Chain management: An Overview*. ISTE Ltd and John Wiley and Sons Inc., ninth edition.
- 163) Thomas, A., Borangiu, T., & Trentesaux, D. (2017). Holonic and multi-agent technologies for service and computing oriented manufacturing. *Journal of Intelligent Manufacturing*, 28(7), 1501-1502
- 164) Tonino, H., Bos, A., de Weerd, M. and Witteveen, C. (2002). Plan coordination by revision in collective agent based systems. *Artificial Intelligence*, 142(2), 121-145.
- 165) Tony Arnold, J. R., Chapman, S. N., & Clive, L. M. (2008). *Introduction to Materials Management*. Sixth edition, Pearson Education International
- 166) Trentesaux, D. (2009). Distributed control of production systems. *Engineering Applications of Artificial Intelligence*, 22(7), 971-978

- 167) Trentesaux, D., & Millot, P. (2016). A Human-Centered Design to Break the Myth of the “Magic Human” in Intelligent Manufacturing Systems. In *Service orientation in holonic and multi-agent manufacturing* (pp. 103-113). Springer, Cham
- 168) Trentesaux, D., Tahon, C., & Ladet, P. (1998). Hybrid production control approach for JIT scheduling. *Artificial Intelligence in Engineering*, 12(1-2), 49-67
- 169) Trzyna, D., Kuyumcu, A., & Lödding, H. (2012). Throughput time characteristics of rush orders and their impact on standard orders. *Procedia CIRP*, 3, 311-316.
- 170) Tursi, A., Panetto, H., Morel, G., & Dassisti, M. (2007). Ontology-based products information interoperability in networked manufacturing enterprises. *IFAC Proceedings Volumes*, 40(19), 85-93.
- 171) Ueda, K. (1992). A concept for bionic manufacturing systems based on DNA-type information. In *Human Aspects in Computer Integrated Manufacturing* (pp. 853-863). Elsevier.
- 172) Umpleby, S.A. (2007). Viable system model. In *International encyclopedia of organization studies*. 4, 1616-1617. Sag
- 173) Valckenaers, P. (2018, June). ARTI reference architecture–PROSA revisited. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing* (pp. 1-19). Springer, Cham.
- 174) Valckenaers, P., & Van Brussel, H. (2016). *Design for the unexpected: From holonic manufacturing systems towards a humane mechatronics society*. ISBN: 0-12-803696-6 , Butterworth-Heinemann
- 175) Valckenaers, P., Van Brussel, H., Verstraete, P., & Saint Germain, B. (2007). Schedule execution in autonomic manufacturing execution systems. *Journal of manufacturing systems*, 26(2), 75-84
- 176) Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Computers in industry*, 37(3), 255-274
- 177) Vander Veen, D. J., & Jordan, W. C. (1989). Analyzing trade-offs between machine investment and utilization. *Management Science*, 35(10), 1215-1226.
- 178) Verstraete, P., Saint Germain, B., Valckenaers, P., Van Brussel, H., Belle, J., & Hadeli, H. (2008). Engineering manufacturing control systems using PROSA and delegate MAS. *International Journal of Agent-Oriented Software Engineering*, 2(1), 62-89
- 179) Vlassis, N. (2003). A concise introduction to multiagent systems and distributed AI. *Intelligent Autonomous Systems, Informatics Institute, University of Amsterdam*

- 180) Vlk, M., & Bartak, R. (2015). Replanning in Predictive-reactive Scheduling. *Association for the Advancement of Artificial Intelligence*.
- 181) Vollmann, T. E., Berry, W. L., Whybark, D. C. (1997). *Manufacturing Planning and Control Systems*. Third edition, USA
- 182) Wang, W. P., & Chen, Z. (2008). A neuro-fuzzy based forecasting approach for rush order control applications. *Expert Systems with Applications*, 35(1-2), 223-234
- 183) Wang, X., & Shao, J. (2015). Consensus for Discrete-Time Multiagent Systems. *Discrete Dynamics in Nature and Society*, 2015
- 184) Weiss, G. (Ed.). (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press
- 185) Wilson, L. (2009). *How to implement lean manufacturing*. McGraw Hill Professional
- 186) Wong, T. N., Leung, C. W., Mak, K. L., & Fung, R. Y. (2006). Dynamic shop floor scheduling in multi-agent manufacturing systems. *Expert Systems with Applications*, 31(3), 486-494
- 187) Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.
- 188) Wroblewski, M. T. (Updated February 12, 2019). What Is a Blue-Collar Worker and a White-Collar Worker? Available at <https://smallbusiness.chron.com/bluecollar-worker-whitecollar-worker-11074.html>. Date accessed [September 22, 2019]
- 189) Wunsch, D., & Bratukhin, A. (2007, September). Multilevel order decomposition in distributed production. In *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)* (pp. 872-879). IEEE
- 190) Xiong, W., & Fu, D. (2018). A new immune multi-agent system for the flexible job shop scheduling problem. *Journal of Intelligent Manufacturing*, 29(4), 857-873.
- 191) Yang, T., Ma, J., Hou, Z. G., Peng, G., & Tan, M. (2007, November). A multi-agent architecture based cooperation and intelligent decision making method for multirobot systems. In *International Conference on Neural Information Processing* (pp. 376-385). Springer, Berlin, Heidelberg
- 192) Zambrano, G., Pach, C., Aissani, N., Berger, T., & Trentesaux, D. (2011, June). An approach for temporal myopia reduction in heterarchical control architectures. In *2011 IEEE International Symposium on Industrial Electronics* (pp. 1767-1772). IEEE.
- 193) Zattar, I. C., Ferreira, J. C. E., Rodrigues, J. G. G., & De Sousa, C. H. B. (2010). A multi-agent system for the integration of process planning and scheduling using

- operation-based time-extended negotiation protocols. *International Journal of Computer Integrated Manufacturing*, 23(5), 441-452
- 194) Zhang, L. M., Dahlmann, C., & Zhang, Y. (2009, November). Human-inspired algorithms for continuous function optimization. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems* (Vol. 1, pp. 318-321). IEEE.
- 195) Zimmermann, E., El Haouzi, H. B., Thomas, P., Pannequin, R., Noyel, M., & Thomas, A. (2018). A Case Study of Intelligent Manufacturing Control Based on Multi-agent System to Deal with Batching and Sequencing on Rework Context. In *Service Orientation in Holonic and Multi-Agent Manufacturing* (pp. 63-75). Springer, Cham.
- 196) Zimmermann, E., El Haouzi, H. B., Thomas, P., Thomas, A., & Noyel, M. (2017, July). A hybrid manufacturing control based on smart lots in a disrupted industrial context. In *20th IFAC World Congress, IFAC 2017*

APPENDICES

Appendix A Industry 4.0: Evolution and Requirements

Evolution and definition of Industry 4.0

Industrial requirements towards the current level have been caused by their need to shift from traditional control system to the current dynamic and agile control system (mainly characterized with smart and intelligent decision entities). This helps them to be reactive and adaptable to dynamic environment and hence shorter product lifecycle & customized products (Trentesaux, 2009). To reach at this stage, it has passed through different evolutionary development levels, Fig. A~1. Industry 4.0 is a strategic initiative started in 2001 by German government and distributed over the world as *strategy 2020 high tech Action Plan*. It fits to smart and intelligent technological advances such as Internet of Things (IoT), Cyber Physical Systems (CPS), Cloud technologies, etc. It refers to a further developmental stage in the organization and management of an entire value chain process involved in manufacturing industry.

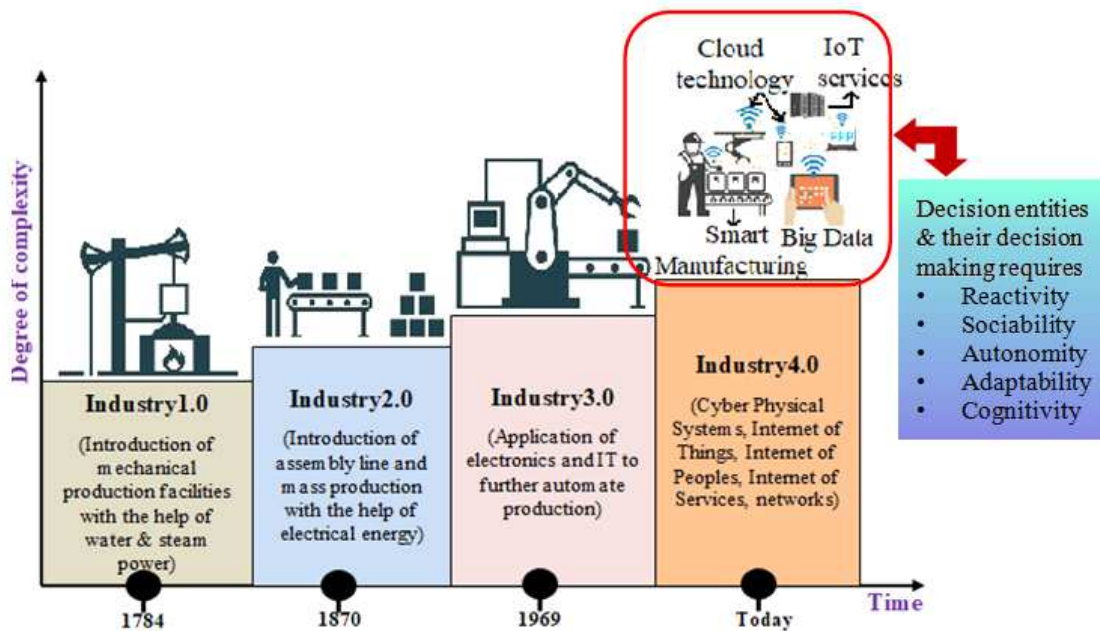


Fig. A~1 Evolutionary approach of Industry 4.0 (adapted from <https://techtzupah.com/evolution-of-industrial-revolution-4-0>)

Unlike the Industry 3.0, which was focused on the automation of single machines and processes, Industry 4.0 focuses on the end-to-end digitization of all physical assets and integration into digital Eco-systems with all value chain partners (Geissbauer et al., 2016). Meanwhile, it is a new level of organization and control over an entire value chain of the life

cycle of product and it is always geared towards increasingly individualised customer drivers and requirements, Fig. A~2. Thus, current manufacturing industries have to grow in intelligent environment such that a gap between real world and digital world would be diminished or reduced. Three main components represent the digital eco-system namely intelligent products, resources, and human. The intelligent products control their routes and decision changes encountered after unexpected disturbance while the resources exchange information, trigger actions, and control themselves (self-organize) independently (El Haouzi, 2017). The human expert integrates the digital system as the other two do not have capability design an initial production plan and/schedule without him.

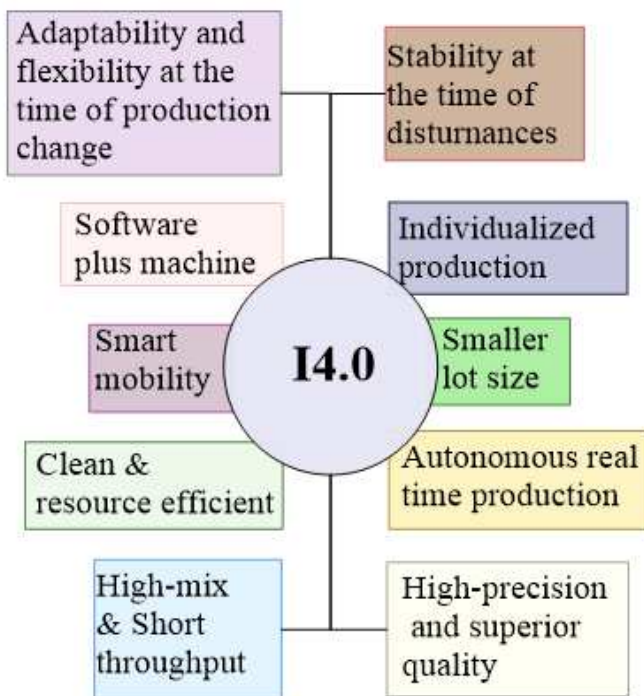


Fig. A~2 Drivers, Requirements, and Characterization of Industry 4.0 (Source: Koch et al., 2014; Benjaafar et al., 2002)

In addition to the drivers and requirements presented in Fig. A~2, Industry 4.0 has also become re-known for the following four main characteristics (Devezas and Sarygulov, 2017),

- a) **Vertical networking of smart production systems** uses Cyber Physical Production systems (CPPSs) to enable plants to react rapidly to changes in demand or stock levels and to faults. Plants organize themselves and enable production that is customer specific and individualized. This requires data to be extensively integrated.
- b) **Horizontal integration via a new generation of global value chain networks** these local and global networks are real time optimized networks that enable integrated

transparency, offer a high level of flexibility to respond more rapidly to problems and faults, and facilitate better global optimization.

- c) **Through-engineering across the entire value chain** this engineering occurs seamlessly during the design, development and manufacture of new products and services. New products need new and/or modified production systems. The development and manufacture of new products and production systems is integrated and coordinated with product life cycles, enabling new synergies to be created between product development and production systems.
- d) **Acceleration through exponential technologies** it is the impact of exponential technologies as an accelerant or catalyst that allows individualized solutions, flexibility, and cost savings in industrial processes.

Reference Architecture for Industry 4.0

In order to execute, considering the concept of Industry 4.0, a new manufacturing order received from their customers, manufacturing industries use their available resources and technologies. The big issue at this time is how to customize the available resources to the new and personalized customer order. To address this problem, a Reference Architecture Model for Industry 4.0 (RAMI4.0), Fig. A~3, was devised in Germany (Hankel and Rexroth, 2015 in Rojko, 2017). It is a meta-model to play an important standardization role in the Industry 4.0 production system (Rojko, 2017).

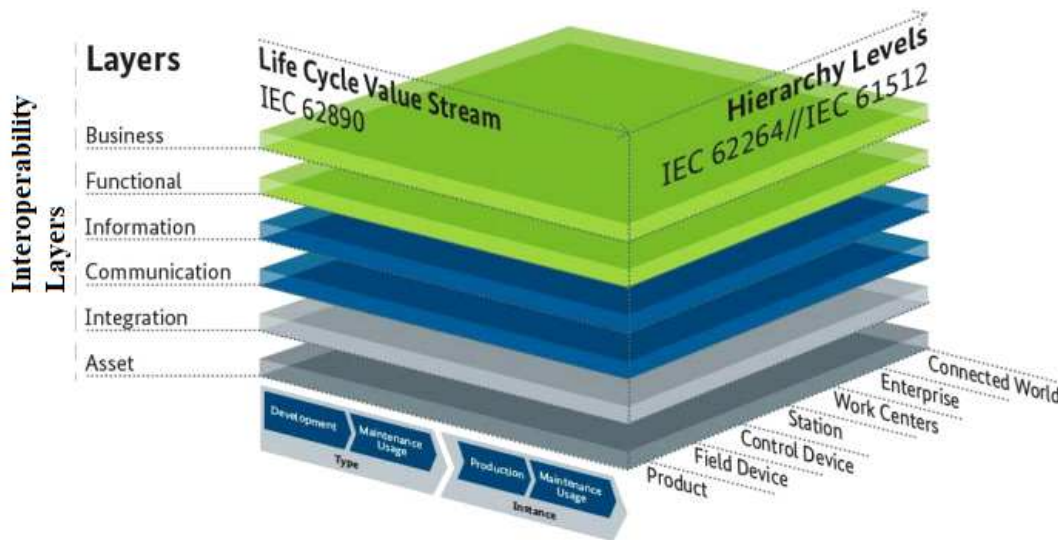


Fig. A~3 Reference Architecture Model for Industry 4.0 (Source: Hankel and Rexroth, 2015)

Appendix B Cyber Physical Systems (CPS)

The fact that current computers are smaller in size but with greater computing power has led to a propagation of physical objects with virtual systems called Cyber Physical Systems (CPS). The first definition of CPS backs to 2006 by US National Science Foundation, Lee (2006), but through time, the definitions converge to Gunes et al. (2014) and Lee et al. (2015) which this study is also interest on. According to Gunes et al. (2014) and Lee et al. (2015), CPS is defined as smart system that have cyber technologies, both hardware and software, that are deeply embedded in & between physical components of smart manufacturing, that can be seen as interacting networks of physical and computational components, Fig. B~1.

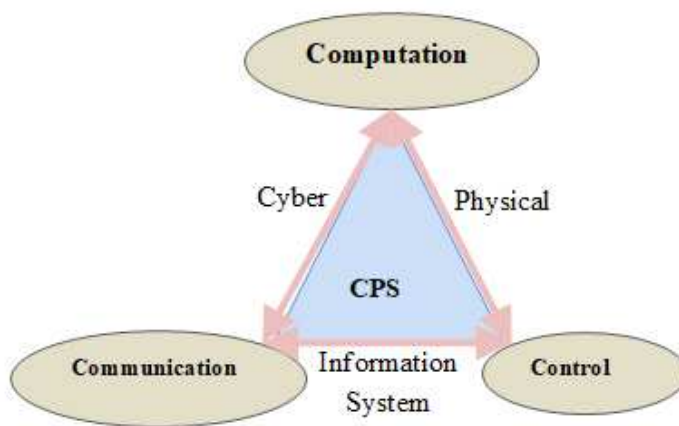


Fig. B~1 Conceptual view of CPS (Source: Klimeš, 2014)

CPS manages interconnected systems between its physical assets and computational capabilities to reach a goal of intelligent, resilient, and self-adjustable (adaptable) resources. Meanwhile, it links the IT with mechanical (machines) and electrical components that then communicate with each other via embedded technological possibilities such as wireless network sensor, intelligent actuators, embedded infotonics, Radio Frequency Identification (RFID) etc. This merging is expected to bring responsiveness and sustainability to the system through the cooperation, self-organization, and making autonomous decisions. Lee et al. (2015) have also presented the following five levels of CPS in order to realize the required merging and cooperation among the physical components and virtual systems.

- Level-I: ability to connect to a network
- Level-II: capability to deal with information and its transcription
- Level-III: Knowability about their environment
- Level-IV: Cognitive ability
- Level-V: adaptability in case of unexpected disturbance

The virtual system must be close to the physical world which generates a large amount of data to support the cooperation.

Recent works applying the CPSs concepts could be found in hybrid subject domains such as holonic manufacturing system, multi-agent system, autonomous distributed systems, product-driven system, big data etc. In the framework of these applicability, Kao et al. (2015) has illustrated a high-level view of a Cyber-Physical Analytics platform with self-learning capabilities, Fig. B~2. It takes into account technical approach linking a physical world and a Cyber Physical System on maintenance of an automation factory. As shown in the figure, in terms of the physical world, one first must select which fleet of assets to build a cyber-physical model for and what attributes of the asset are important and have value to the end user. They considered a fleet of machine tools; based on the needs of the particular plant or end user, the fleet of systems could also be at the component or subsystem level. Once deciding on the appropriate physical space, it is necessary to decide on the physical assets to model and what type of information is of value to the end-user. In this case, the cyber-physical interaction between the physical asset world and the cyber representation shall be considered. The link between the physical world and the cyber-physical interaction is the machine data (sensory data, data from controllers, repair history etc.) and factory data (e.g. overall equipment effectiveness). This set of heterogeneous information from a fleet of assets should be time-stamped, that capture the fleet histories over time. The time machines would capture signatures (feature values) from the sensory data of the machine as well as the utilization history, maintenance logs etc. to sum up, the above example and other studies disclose that the expectations towards CPS are versatile and enormous; robustness, autonomy, self-organization, predictability, efficiency, interoperability, to list few.

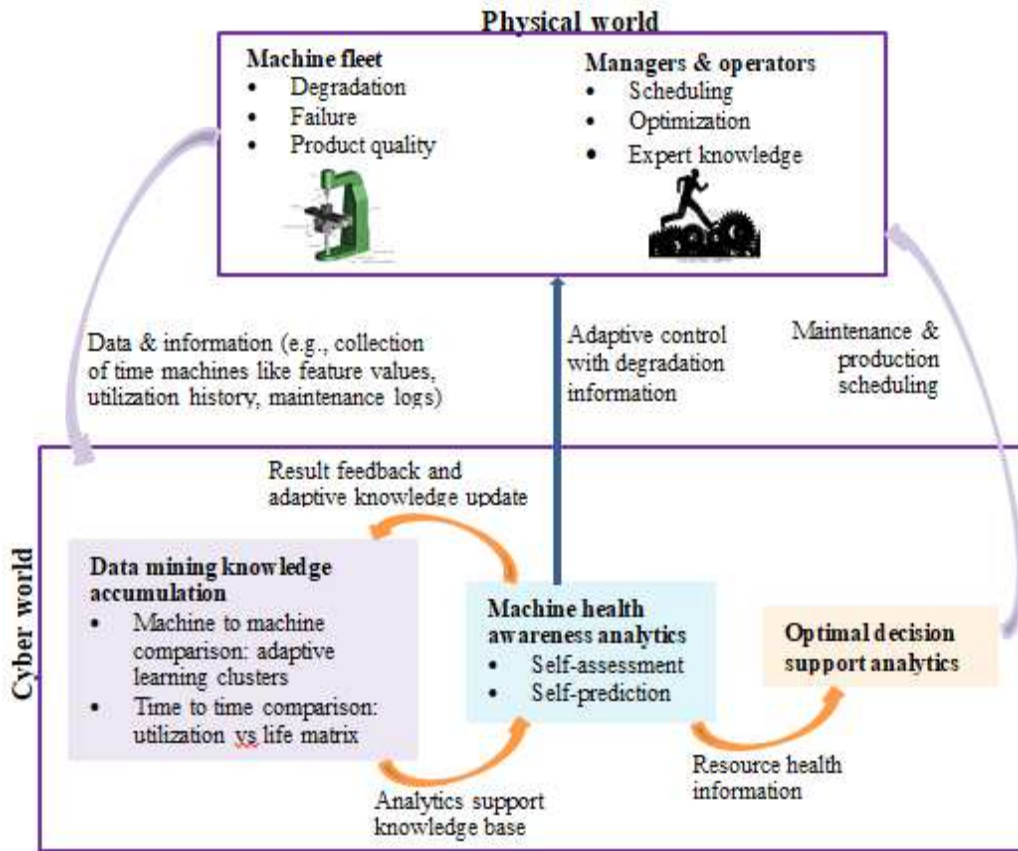


Fig. B-2 Cyber Physical Analytics Platform with Self Learning Capabilities (Source: Kao et al., 2015)

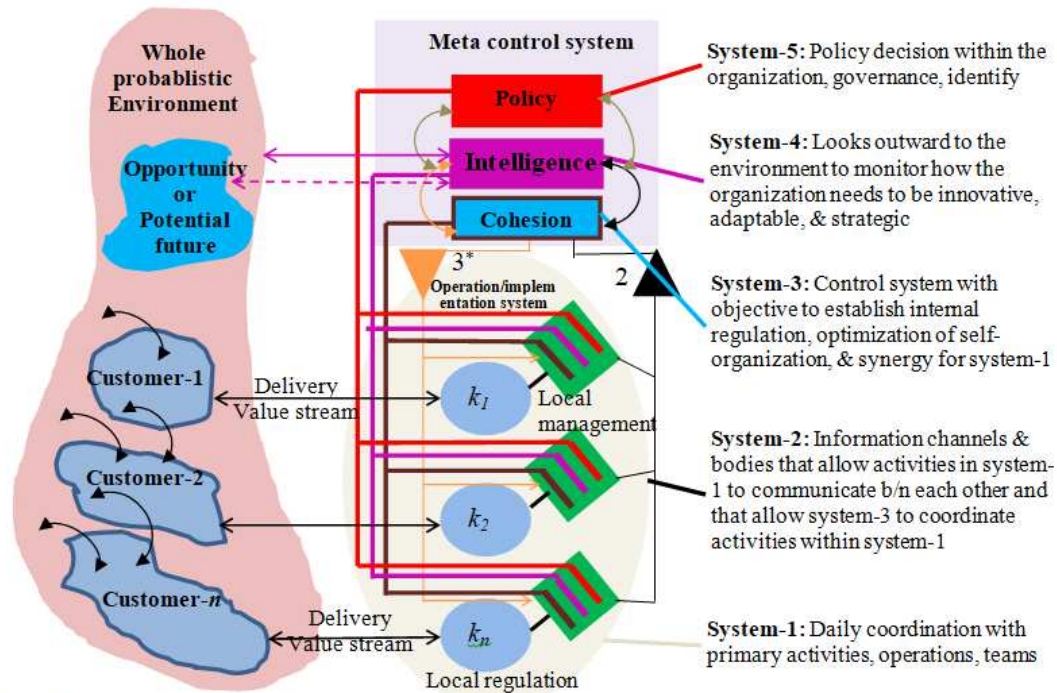
Appendix C Viable System Model (VSM)

Viable System Model, Fig. C~1, was described by British Cybernetician Stafford Beer in his books as *Brain of Firm* (1972), *The Heart of Enterprise* (1979), and *Diagnosing the System for Organizations* (1985) (Umpleby, 2007). It allows proposing an organizational model based on the structure of human nervous system focusing on lower level activities taking into account system components (like products, machines, conveyors, operators etc.) that perform assigned tasks. It is derived from striving mechanism of biological organisms to face a changing environment around them by diminishing a command based control system characterized with top-down decisional capability and encouraging a flat recursion. It explains how some systems succeed in being viable. As it is presented in Fig. C~1, this recursive and cybernetic model incorporates three collaborative components and/or units:

- Meta-control unit (steps 3, 4, and step 5)
- Operational (or implementation) system unit (step 1 and step 2) and
- Probabilistic external environment unit which is integrated in order to realize the overall harmonization and coordination of the system

VSM is also a powerful tool to steer interactions among these three units in directions that produce effective structural mechanisms and as a result it manages complexity of organizational system by integrating local and global processes (Espejo and Reyes 2011; Schwaninger 2006). Inside the operating unit, same configuration exists with local regulation and local management and all entities decide at their level which confirms a recursive property to the system. Furthermore, VSM helps to understand how decisional entity's interaction produce shared communication spaces, whether real or virtual, with particular structures. For instance, shared communication spaces populated by people with uneven power are likely to produce hierarchical structures, which themselves become the media to constitute interactions with uneven distribution of power.

VSM has been chosen to model research findings by previous different studies such as Herrera et al. (2011), Stich (2009), Laws (2005) etc. Herrera et al. (2011) for instance has presented that production management, autonomous distributed systems, modeling communities of autonomous agents etc. uses the application domains of VSM.



Legend

- Globally both 2 and 3* represents the cooperation and information exchange channels
- Specifically 2 represent overall harmonization and coordination and 3* represents coordination and monitoring, sporadic audits, anomaly detection, and learning potential.
- k_n represents n number of operations

Fig. C-1 Viable System Model for system adaptability (Source: Stafford Beer cell)

Appendix D TRACILOGIS test-bed platform

TRACILOGIS is an abbreviation for **TRACEABILITY**, **Identification** and control by Product for **LOGISTICS** Chain. It demonstrates several educational domains related to automation unit, production control unit, or fabrication unit with principal objectives to:

- Assess the impact of new and smart RFID technology (the RS232 RFID technology in, Fig. D~1)
- Provide a test environment for the implementation of scenarios for production of different products, see Fig. D~2, in a centralized, distributed or hybrid context
- Implement, test and compare different traceability techniques within a supply chain in general and timber in particular
- Confirm running modes for production control and demonstrate new production decision

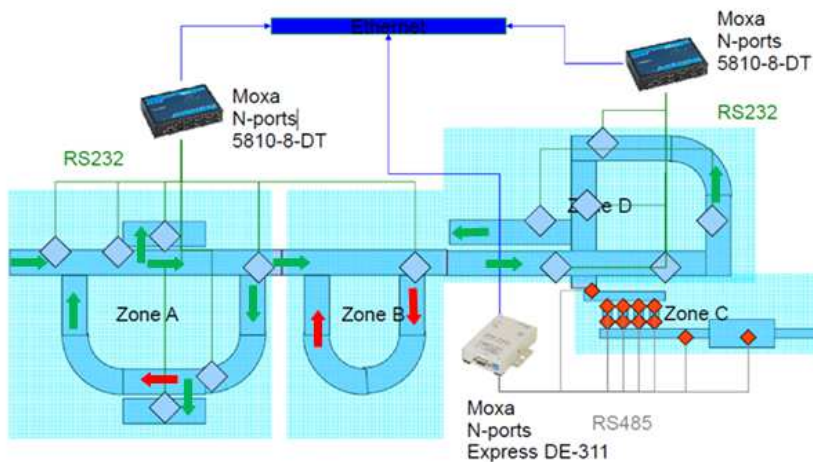


Fig. D~1 Flow chart of RFID networks (Source: CRAN)

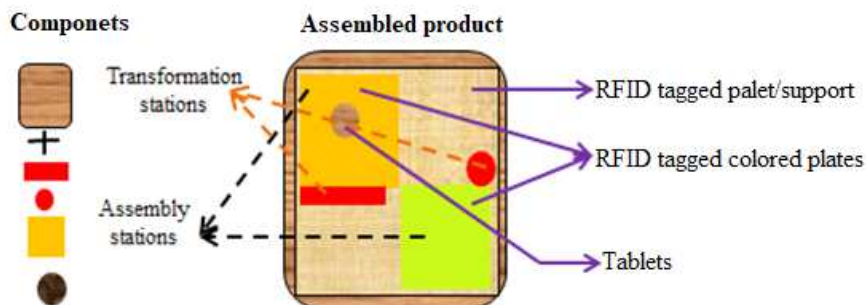


Fig. D~2 Intelligent components and assembled product type

The three agent groups in the platform that make cooperative interactions until they reach on their common and final offer using the control architecture shown in Fig. 4-4 are presented as follow.

(a) **The product agent (or Actor)** it includes physical identifier (RFID tag) and logical identifier (product) and it is in charge of the correct execution of a given configuration, decides the way forward, transmits orders to resources, and validates or not the configuration changes sent by a configuration agent. In a base scenario, it is the smartest element. The product, meanwhile, has a perception of the wider environment allowing him to build his life cycle. To build its career, the product agent has several behaviors, including the main ones are described by activity diagrams. A first behavior allows the product agent to discover its environment by building a resource graph (their position and their services). From its historical data and position at an instant time t , it decides the way forward using Dijkstra's algorithm to calculate its shortest path. See its UML diagrams in Fig. D~3 and Fig. D~4

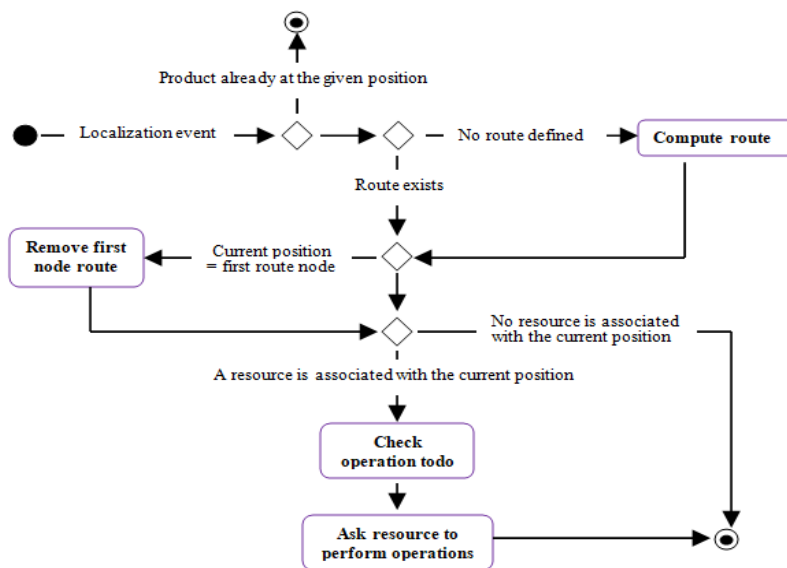


Fig. D~3 UML of product's main behavior (Source: CRAN)

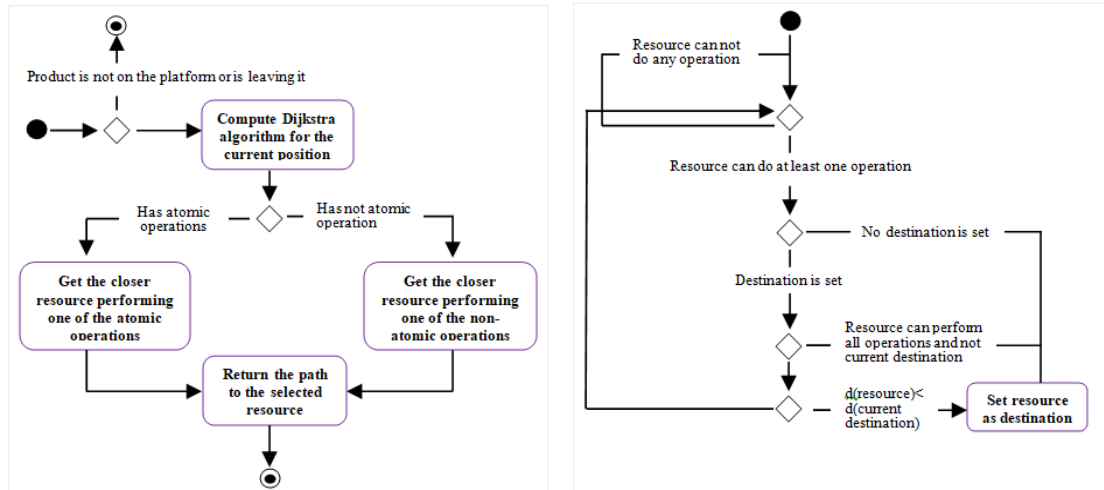


Fig. D-4 The UML of creating product's route (Source: CRAN)

(b) **The executing agents** it includes reflex (automatic reaction to an event, sensor presence) and resource agents (execution of commands sent by a supervisor agent & product agents to offer services such as physical transformation and spatial transformation).

(c) **The interfacing/supervision agents** it includes the PLC (automation control), the RFID (communication agents/readers), and the configuration (creating an interface and configuration management of product agents). The RFID reader (which includes RFID agent linked to one or more readers (Ethernet / RS232), RFID agent linked to all readers (Ethernet/RS485), and Interface via library developed internally (Java)), interfaces via library developed internally (Java). This interfacing agent has been created to allow a user (or external services) to supervise (give orders and receive reports). These agents have pure utilitarian functions for allowing communication between cognitive agents (resource, product) with equipment (RFID reader, camera, sensor, database etc.).

The three communication and control levels shown in Fig. 4-4 are connected via an Ethernet network and they communicate using a Java software structure shown in Fig. D-5 and ACL shown in Fig. D-6. This multi-agent system is developed through Java Agent Development framework (JADE). It is a software platform for multi-agent system development and applications compliant with Foundation for Intelligent Physical Agents (FIPA) standards. It includes three basic components:

1. An execution environment in which agents can *live* a library of classes
2. A set of graphical tools for the management and
3. Administration of active agents

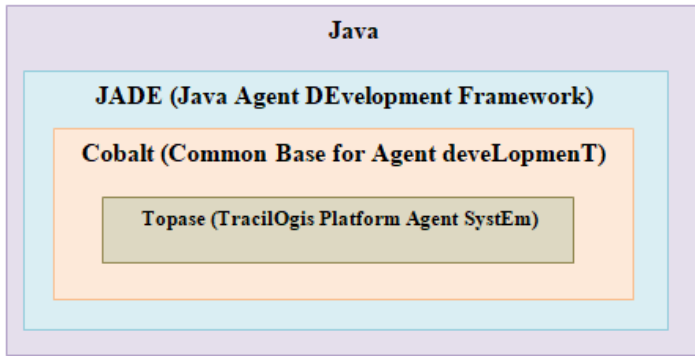


Fig. D~5 The Java software structure (Source: CRAN)

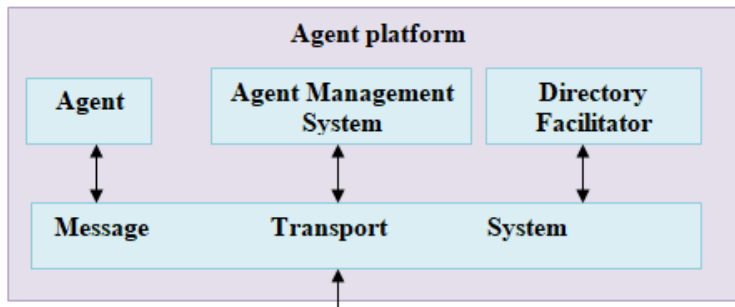


Fig. D~6 FIPA standard Agent Communication Language, ACL (Source: CRAN)

Meanwhile, this platform uses two libraries, Fig. D~5; a generic library to facilitate the development of Cobalt software agents (COMmon Base for Agent deveLopmenT) and a specific library for operating system control Topase (TracilOgis Platform SystEm agent). The developed system is made up of operating agents like Directory Facilitator (DF) that allows managing services offered by product and resource agents, and Agent Management System (AMS) who exerts supervisory control over access to and use of the agent platform shown in Fig. D~6.

Finally, while the three agent groups make cooperative interactions, the multi-agent system, developed with FIPA compliant JADE, manages the sending and receiving of all the actions to realize the event-driven automation, Fig. D~7(a), Fig. D~7(b), Fig. D~7(c).

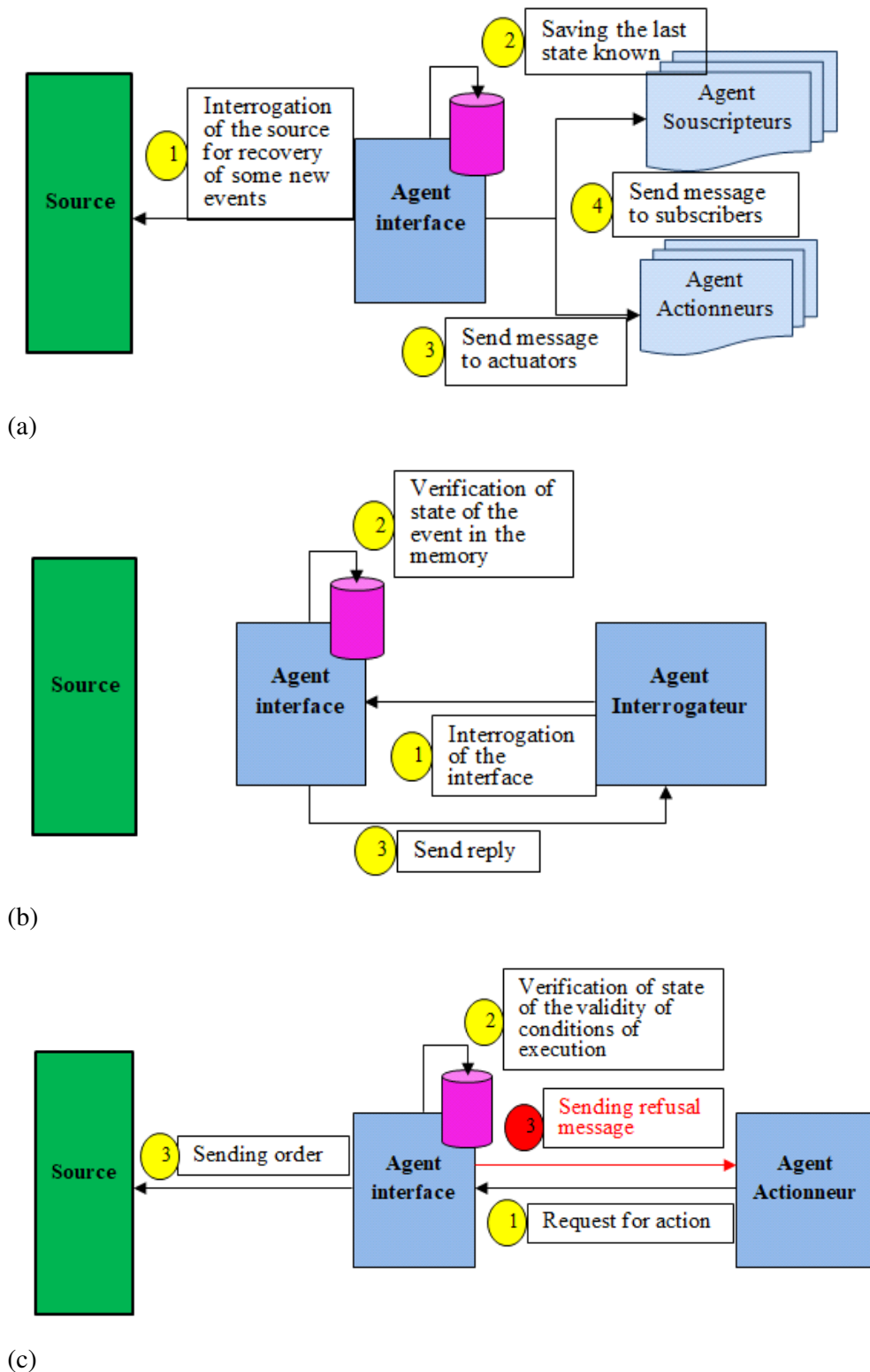


Fig. D~7 The cooperative interactions among agents (a) Subscription: message sent as soon as the event takes place (b) Interrogation: response based on last event occurred (c) Request for action by PLC only. (Source: CRAN)

Appendix E Radio Frequency Identification (RFID)

While product driven manufacturing plants are at their intelligent executing process, it is obvious to face unexpected events from raw material warehouse through work-in-process queue up to finished products & services stock waiting for distribution. In such circumstance, it is necessary to track and trace these events through the entire supply chain (see the work of McCullen et al. (2006) and McFarlane et al. (2003) for the role identification in agile and adaptable supply chain). To ensure this and hence shorten the product lifecycle, all the manufacturing entities generally and intelligent products & resources specifically should continuously communicate among each other and interactively react to their stochastic environment so that their decision-making capabilities will be enhanced. As it is presented in Trentesaux et al. (2009), even though their deployment cost is not easy, there are many embedded and non-embedded technological possibilities, such as barcode, mobile robots (AGV), wireless network sensor, embedded infotronics, Pick by light (foolproofinf), Radio Frequency Identification among many others.

Though it is limited for collision of information transaction, security and privacy issue for example, RFID HF network technology has been widely accepted as it interlinks physical system with information system to remotely retrieve and receive data between decisional entities. It is applicable in inventory management of manufacturing plant, work-in-process, entire supply chain, and in truck industries like automotive industry, waste management, logistics, retail industry, pharmaceutical industry etc. Additionally, it has capability to reduce and/or eliminate industrial problems like stock loss or shrinkage which might encountered due to theft, defects, time-dated items, inaccurate location of items, or inaccessible items (Dolgui Alexandre and Jean-Marie, 2010; Kaur et al., 2011). RFID are small objects, such as self-adhesive labels, that can be glued or incorporated into objects and even implanted in living organisms, Fig. E~1, with an objective to remotely retrieve and receive data using labels called *radio labels*.



(a) The RFID reader (b) The RFID tagged pallet/support (c) RFID tagged plate

Fig. E~1 The RFID HP network technology

As part of this, the TRACILOGIS platform presented in Appendix D is configured with total of 25 RFID⁴⁶ readers in order to trace and identify production events and to facilitate communications among the decisional entities. All the executor agents and the roller based simulator/conveyor are tagged with an RFID reader and the products (both the support/platter and pallet) are embedded with RFID tag so as to store information helpful for communicating its route. This has proved to increase speed of material flow, drastically reduce workload and in-turn labor cost, reduce WIP queue, minimize parts loss etc.

⁴⁶See the work of Noyel & Pisaneschi (2011) for the role of this RFID in the platform

Appendix F Lean Manufacturing

Lean manufacturing is a Japanese production philosophy that focuses on abolishing or reducing wastes, Fig. F~1, and on maximizing or fully utilizing activities that add value from the customer's perspective (Nicholas, 2018). It was introduced in Japan after world war-II to help them manage their materials and human resources which itself evolved from Taiichi Ohno's experiments and initiatives over three decades at Toyota Motor Company (Shah and Ward, 2007). In addition, it is a present-day instance of the recurring theme in human history toward increasing efficiency, decreasing waste, material management (Tony Arnold et al., 2008) and using empirical methods to decide what matters, rather than uncritically accepting pre-existing ideas. It combines the advantages of craft and mass production, while avoiding the high cost of the former and the rigidity of the later.



Fig. F~1 The seven lean wastes (adapted from <http://quotesgram.com/quotes-on-waste-lean>)

This business model distills the essence of lean approach into five key principles and shows how the concepts can be extended beyond automotive production to any company or organization, in any sector; (Feld, 2000).

1. *Specify* what does and does not create *value* from the customer's perspective and not from the perspective of individual firms, functions and departments
2. *Identify* all the steps necessary to design, order and produce the product across the *whole value stream* to highlight non value adding waste
3. Make those actions that create *value flow* without interruption, backflows, waiting or scrap

4. Only make what is *pulled* by customer
5. Strive for *perfection* by continually removing successive layers of waste as they are uncovered

These principles are fundamental to the elimination of wastes. To get company's focus on customers need, they must define the value streams inside their company (all the activities which are needed to provide a particular product or service) and, later, the value streams in their wider supply chain as well. To satisfy customers, industries will need to eliminate or at least reduce the wasteful activities in their value streams that their customers would not wish to pay for. Next they have to find a way of setting the direction, fixing targets and seeing whether or not change is actually occurring. They need an internal (and later external) framework to deliver *value* for their customers to make the change.

The first stage in lean production is the understanding of wastes; just specifying the types of wastes, and types of activities to detect these wastes within and around the organization. Improved productivity leads to leaner operations, which in turn help not to expose further waste and quality problems in the system. The systematic attack on waste is also a systematic attack on the factors underlying poor quality and fundamental management problems. In Toyota Production System, seven types of wastes were identified by Shingo and Dillon (1989) and Fig. F~1 and Table F~1 shows these seven types of wastes with their descriptions.

Table F~1 Type and description of wastes (Source: Shingo and Dillon, 1989)

S/N	Waste	Description
1	Overproduction	Producing too much or too soon, resulting in poor flow of information or goods and excess inventory
2	Defects	Frequent errors, product quality problems, or poor delivery
3	Unnecessary inventory	Excessive storage and delay of information or products, resulting in excessive, often when a simpler approach may be more effective
4	Inappropriate processing	Going about work process using the wrong set of tools, procedures or systems, often when a simpler approach may be more effective
5	Excessive transportation	Excessive movement of people, information or goods, resulting in wasted time, effort and cost
6	Waiting	Long periods of inactivity for people, information or goods
7	Unnecessary motion	Poor workplace organization, resulting in poor ergonomics, for example excessive bending or stretching and frequently lost items

While thinking about these lean wastes, there are three types of activities that always happen within supply chain of an enterprise (Wilson, 2009):

1. *Value adding activity* those activities that, in the eyes of the final customer, make a product or service more valuable. Examples would include drilling, complicated cutting, facing, turning, grinding etc. within carpentry factories. A value adding activity is simple to define; just industries can ask themselves if they as a customer would be happy to pay for it!
2. *Necessary non value adding activity* those activities that, in the eyes of the final customer, do not make a product or service more valuable but are necessary unless the existing supply process is radically changed. Such waste is more difficult to remove in the short term and should be a target for longer term or radical change. An example would be: inspecting every product at the end of a process because the process uses an old machine which is known to be unreliable.
3. *Non-value adding activity* those activities that, in the eyes of the final customer, do not make a product or service more valuable and are not necessary even under present circumstances. These activities are clearly *wastes* and should therefore be the target of immediate or short term removal. An example of non-value adding activity would be transferring a product from one storage area to other storage area so that it will decrease excess inventory.

Meanwhile, lean production is a holistic view that emphasizes the interconnectivity and dependence among a set of five key/primary elements (Feld, 2000):

- Manufacturing flow,
- Organization,
- Process control,
- Metrics, and
- Logistics

Appendix G Multi-agent Systems (MAS)

In addition to monitoring interconnected system between physical assets and its computational capabilities using CPS, an agent-based production system treats unexpected changes and disturbances as business as usual. It considers manufacturing process as a non-stop *going concern*. For example, on a shop floor, schedules might acquire a reputation for rapidly becoming invalid because of frequent changes and disturbances. Therefore, a multi-agent production control treats these changes and disturbances as business as usual; it considers manufacturing process as a non-stop *going concern*. This makes agents to interconnect usually, ongoing, and as non-terminating one. See the work of Monteiro et al., 2008 for the interest of agents in supply chain. Examples of agents include humans, robots, or software agents:

- *Human agents*: having eyes as sensors, hands as actuators
- *Robotic agents*: having cameras as sensors, wheels as actuators or
- *Software agents*: having a graphical user interface as sensor and actuator

A term autonomous is often used to refer to an agent whose decision-making relies to a larger extent on its own perception than to prior knowledge given to it at design time. In most domains of reasonable complexity, an agent will not have complete control over its environment. It will have at best partial control, in that it can influence it, Fig. G-1(a). From the point of view of the agent, this means that the same action performed twice in apparently identical circumstances might appear to have entirely different effects, and in particular, it may fail to have the desired effect. Hence, the flexibility of agent incorporates different properties such as:

- (a) *Reactivity* perceives its environment and responds in a timely fashion to changes in order to satisfy its design objectives
- (b) *Proactiveness* aims to exhibit goal-directed behavior by taking the initiative in order to satisfy its design objectives
- (c) *Social ability* capable of interacting with other agents (and possibly humans) such as cooperating, coordinating, communicating etc.
- (d) *Embodied*: situated in the environment

This indicates that agents in all but the most trivial of environments must be prepared for the possibility of failure. This leads to say that environments are in general assumed to be non-deterministic and inaccessible. Such a system that consists of a group of agents that can potentially interact with each other is called Multi-agent systems (Wooldridge, 2009; Botti

and Giret, 2008), Fig. G~1(b). The multi-agent design incorporates the approach from object-oriented design which states that the problem has to be a part of the solution in order to make robust and flexible software.

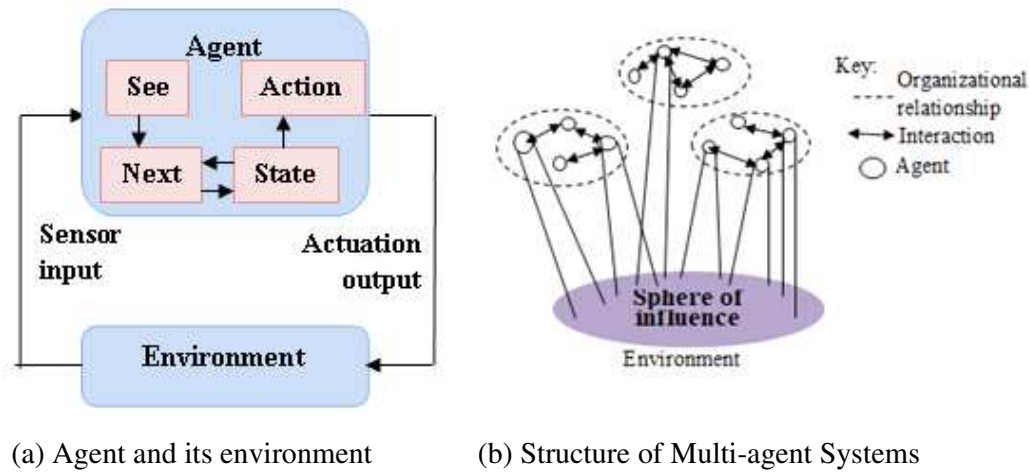


Fig. G~1 The Multi-agent Systems (Source: Wooldridge, 2009)

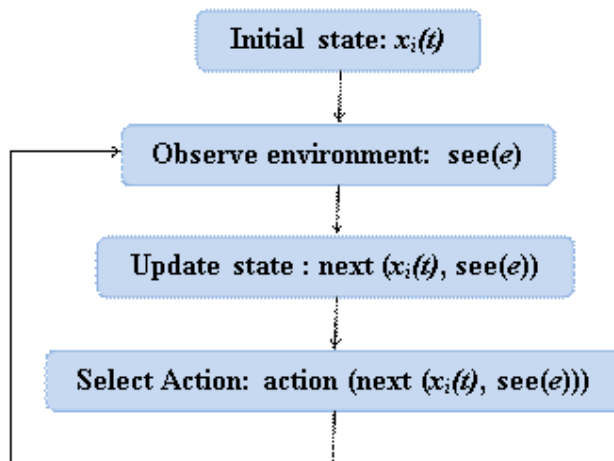


Fig. G~2 Agent control loop (Source: QUINTANILLA in winter school on IMSS⁴⁷)

Multi-agent systems differ from single agent system with the environment they execute; single agent system has been developed for static environments because these are easier to handle and allow for a more rigorous mathematical treatment (Vlassis, 2003). In MAS, the mere presence of multiple agents make the environment appear dynamic from the point of view of each agent, with the control loop they follow, Fig. G~2. Contrary to single-agent system, the control in MAS is typically distributed (decentralized). This means that there is no central process that collects information from each agent and then decides what action

⁴⁷Winter school on IMS² was a school prepared in IUT Nantes (University of Nantes) for masters and doctoral students by Olivier Cardin (MdC in University of Nantes) from Jan. 15-Jan. 18, 2019

each agent should take. The decision-making of each agent lies to a large extent within the agent itself, and with the level of its knowledge. In single-agent system, the agent knows its own actions but not necessarily how its surrounding is affected by its actions. However, in MAS, the levels of knowledge of each agent about the current state can differ substantially.

Characteristics of multi-agent environment multi-agent environments provide an infrastructure specifying communication and interaction protocols, Fig. G~1(b). It contains agents that are autonomous and distributed, and may be self-interested or cooperative. In addition to this, multi-agent environment is characterized by the following characteristics (Weiss, 1999):

- a) *Know-ability* to what extent is the environment known to the agent
- b) *Predictability* to what extent can it be predicted by the agent
- c) *Controllability* to what extent can the agent modify the environment
- d) *Historical* do future states depend on the entire history, or only the current state
- e) *Teleological* are parts of it purposeful, i.e., are there other agents
- f) *Real-time* can the environment change while the agent is deliberating

Cooperation and consensus of agents one of the key issues in cooperative control of manufacturing agents used in agent-based shop floor simulator, see Table G~1 for different agent types, is the ability of distributed agents to coordinate their actions and avoid overlapping. In a decentralized environment, cooperation between agents may not come natural since every agent tries to optimize its own behavior. In typical complex scenarios, it may not be clear to an individual agent how its own behavior is related to the global performance of the multi-agent system.

Table G~1 Agents in agent-based shop floor simulator (Source: Komma et al., 2011)

Agent	Role
AGV agent	Represents a typical AGV
Arrival-queue-agent	Represents a system arrival-queue
Departure-agent	Represents the queue at the system departure place of parts
Machine-agent	Represents a typical machine with its input and output buffers on the shop floor
Part-agent	Represents a typical job or part on the shop floor
Part-generator-agent	Represents an agent that facilitates the arrival of part-agents at the shop floor

Thus, in order to act rationally and interact within a shop floor or any other environment, an agent should take both the past and the future into account when choosing an action and when it cooperates with other nearby agents (Vlassis, 2003). The past refers to what the agent has perceived and what actions it has taken until time t and the future refers to what the agent expects to perceive and do after time t . More description and role of Multi-agent systems can be grasped on references such as (Wooldridge, 2009; Dimopoulos and Moraitis, 2006; Amgoud et al., 2005) among many others.

Applications and limitations of MAS multi-agent systems are applicable mainly in complex software system and in factory process control which demand complex and digital production systems (Wooldridge, 2009; Weiss, 1999). Some of them are:

- *Speedup and efficiency* due to the asynchronous and parallel computation of agents, and this can result in an increased overall speed
- *Robustness and reliability* the failure of one or several agents does not necessarily make the overall system useless, because other agents already available in the system may take over their part
- *Scalability and flexibility* the system can be adopted to an increased problem size by adding new agents
- *In need of adaptability* for disturbances (e.g., machine breakdowns, missing tools, missing materials etc.) and changes (e.g., new technology, new markets, new orders, new organizations etc.)
- *Distributed resource scheduling and planning* which emphasizes how agents should coordinate their schedules to avoid and resolve conflicts over resources, and to maximize system output

Appendix H Interaction protocols

In order to make agent-based production system to treat unexpected changes and disturbances, creating an interaction environment called interaction protocol for all decisional entities within a system is important. According to Caridi and Cavalieri (2004), *protocol* is a sematic structure and content of messages exchanged among autonomous agents. Hence, an interaction protocol, Fig. H~1, is defined as a messaging media or communication protocol that specifies the information exchange between an agent and all of its neighbors on a given agent's network without affecting its autonomy (Jimenez et al., 2017; Reza Olfati-Saber, 2007; Bel-Enguix et al., 2007).

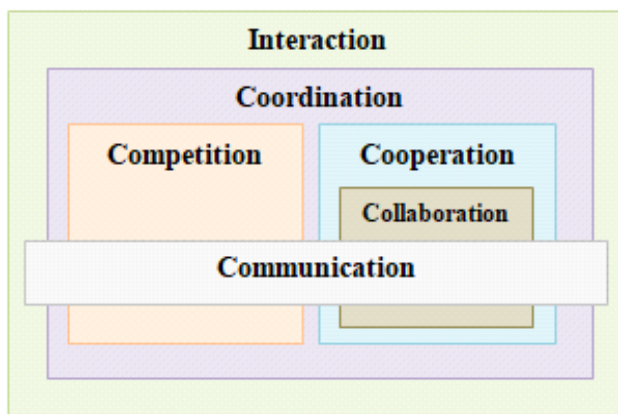


Fig. H~1 MAS interaction topology (Source: QUINTANILLA in winter school on IMS²)

In Fig. H~1, coordination is mechanism for ensuring agent's desired relationships (such as sequence, complementarity etc.). Likewise, communication in Fig. H~1 is presented to illustrate information exchange, collaboration, and negotiation among agents to reach agreement in presence of conflict. The communicative act has its own preconditions that need to be true before an agent starts to execute the action. Similarly, having sent the message, the sender's message might be ignored by receivers as the receivers have an autonomous decision capability. That is why, an interaction protocol should be created with triggering confirmation by sender agent whether the sent message has an effect or not. Globally, different interaction protocols such as Contract-net (Smith, 1980), Stigmergy (Pannequin and Thomas, 2012), Blackboard (Botti et al., 1995), etc. have been introduced. More specifically, Foundation for Intelligent Physical Agents (FIPA), (Bellifemine et al., 2002), has listed a number of interaction protocols to build agent communications. For instance, *FIPA-recruiting* (Contract Net protocol), *FIPA-request* (peer to peer protocol), *FIPA-query* (broadcasting protocol) are some of the commonly practiced interaction protocols. They have

an initiator role (an agent that triggers a communication) and responder role (an agent engaged in communication after being contacted by some other agents).

Contract Net Protocol (CNP) CNP, Fig. H-2, allows an initiator agent to send a call for proposal based on its requirements to a set of N respondents (Knabe et al., 2002). Accordingly, all the N respondents send their proposal within the given proposal submission deadline. After this, the initiator evaluates their proposals, and finally accepts one of the best proposals (or even he/she can reject all the proposals if they do not satisfy its requirements). Here, responders do not have any decision altering power during the interaction. CNP is applicable to online dispatching, meeting scheduling, and in flexible manufacturing system.

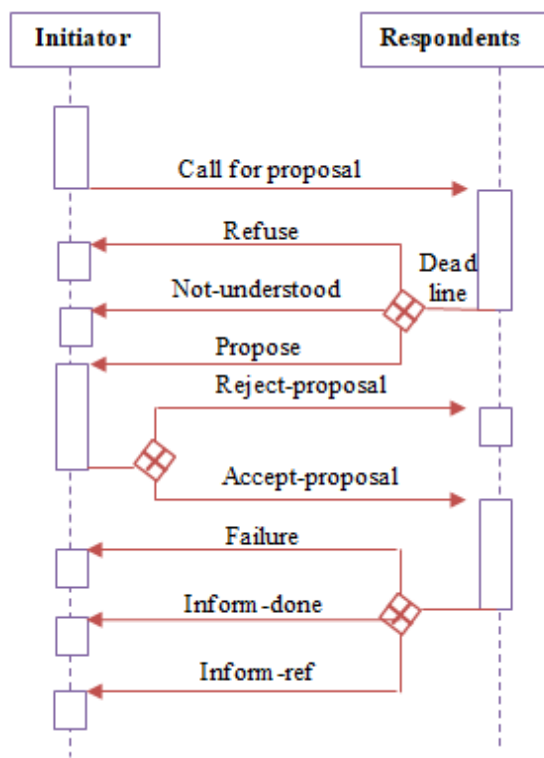


Fig. H-2 The FIPA Contract Net Protocol (Source: Knabe et al., 2002)

Peer-to-Peer Protocol it is the variant of the Contract Net Protocol operating in head-to-head mode; the initiator agent simply unicast request to a single responder among N responding agents, Fig. H-3. As a response to this request, a single responder from the N respondents replies that *yes I am available or I am busy*. If it is busy, the initiator again requests another candidate (still from the N respondents) for similar service and continues to repeat the process until the initiator satisfies its requirements.

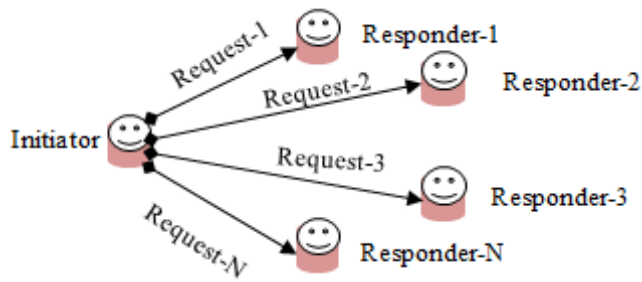


Fig. H~3 Peer-to-Peer Protocol (adapted from <https://en.wikipedia.org/wiki/Peer-to-peer>)

Broadcasting Protocol is a fundamental operation in all kinds of networking systems. It is mainly applicable in dynamic environments with events such as releasing orders, searching optimal routes for these orders, notifying important execution processes through sensors and actuators etc. Unlike the peer-to-peer protocol, broadcasting protocol is considered as a simple variant of Contract Net Protocol operating in general broadcasting mode. The initiator (or the client) contacts all the responders (service providers) through broadcasting its requirements to execute him/her intention, Fig. H~4, and the respondents in turn acknowledge the client to provide him/her the required services considering their current state. Here it should be noted that as the respondents did not reply to each query of the initiator/client, the client should ahead know the current state of all respondents whether they are capable to entertain its broadcasted messages/intentions or not. See the work of Seyboth et al. (2013) for the role of broadcasting to multi-agent average consensus.

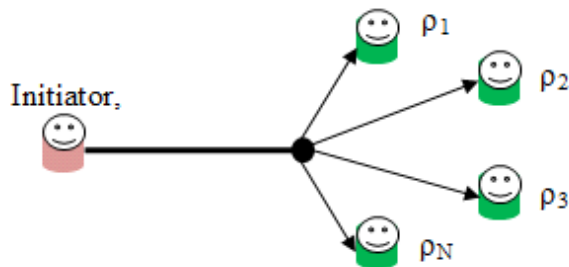


Fig. H~4 Broadcasting protocol (adapted from [https://en.wikipedia.org/wiki/Broadcasting_\(networking\)](https://en.wikipedia.org/wiki/Broadcasting_(networking)))

Appendix I Holonic Manufacturing System (HMS)

The holonic concept was developed by the philosopher Arthur Koestler in order to model the evolution of biological and social systems Koestler (1989). According to Koestler(1989), Holon is a combination of a Greek *holos* = whole, with suffix *-on* which, as in *proton* or *neutron*, suggests a particle or part. Two things urge Koestler to concentrate on Holon concepts:

- On the one hand, these systems develop stable intermediate forms during evolution that are self-reliant
- On the other hand, in living and organizational systems, it is difficult to distinguish between *wholes* and *parts*; almost everything is both a whole and a part, at the same time

He observed that, in living organisms and in social organizations, which are entirely self-supporting, non-interacting entities did not exist. From this observation, it is concluded that, the strength of holonic organization, or holarchy, is that it enables the construction of very complex systems that are nonetheless efficient in the use of resources, highly resilient to disturbance (both internal and external), and adaptable to changes in the environment in which they exist.

The Holonic concept in manufacturing context was developed by Suda (1989) and Christensen (1994) as a response to a growing perception that Japanese manufacturing firms lacked competitiveness in a global manufacturing environment. Suda (1989) hypothesized that the cause of this inability to compete was rigid manufacturing practices that did not have the necessary agility and responsiveness in increasingly volatile markets. Thus, a holarchy system (Farid, 2004; Botti and Giret, 2008; Cardin & Castagna, 2009), Fig. I-1, which includes the following three elements, is proved to gain an international competitiveness through its agile, robust, responsive, flexible, and adaptable characteristics.

Holon is an autonomous and cooperative building block of a manufacturing system for transforming, transporting, storing, and/or validating information and physical objects. Holons may create and execute their own plans and follow their own strategy.

Holarchy Holon is cooperative since it works with other holons on mutually developed and agreed plan. This cooperation creates holarchy, self-organization, which defines basic rules for cooperation of the holons by limiting the autonomy of holons.

Holonic manufacturing System (HMS) a holarchy that integrates the entire range of manufacturing activities from order booking through design, production and marketing to

realize the agile manufacturing enterprise. See the works of Quintanilla et al. (2016) and Indriago et al. (2016) for the role of holonic manufacturing system in manufacturing services. It modernizes the perceptions of the natural hierarchy and joins its advantages with a heterarchical control approach. It is modular and cooperates by message passing and integration.

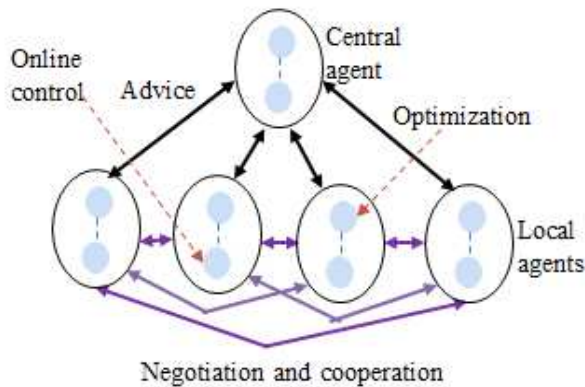


Fig. I-1 A Hierarchy system (Source: Farid, 2004)

Fig. I-2 also demonstrates how the holonic vision directly links salient business drivers to the necessary manufacturing system attributes. It elaborates that the smartness of attributes is mandatory based on real time data (A data where its correctness depends not only on the logical result of computation but also on the time at which the results were produced).

- *Smart*: Holons should be specific, measurable, achievable, reliable, and timely
- *Based on real time data*: integrating holons at ideal scenario will be valueless; thus, realistic data and information results accurate products

This magnifies that a manufacturing control system for production processes is composed of informational processing module as well as physical processing module such as resources, products, client work orders, etc. These software modules and the physical entity, bonded by means of an appropriate communication network, represent a Holon. The physical processing part is optional; some examples of holons without a physical processing part are work-order holons, planning holons, scheduler holons, etc. The physical processing part is again divided into: the physical processing itself, which is the hardware that executes the manufacturing operation; and the physical control, controllers like NC, CNC, and DNC that controls the hardware operation.

			Decentralized architecture	Product/resource based architecture	Abstract/generalized interactions	Flexible acquaintances/interactions	Reactive capabilities	Proactive capabilities	Self-organizations
	*	Robustness	*	*		*	*		
	*	Scalability	*	*		*		*	*
	*	Re-configurability	*	*	*	*		*	*
	*	Flexibility			*	*	*	*	
*		Well defined/transparent behavior			*	*	*	*	
*		Intuitive/transparent structure	*	*					*
*		Minimal system structure		*					*
*		Standardization			*				
Complexity									
Responsiveness									

Fig. I~2 Linkage between business drivers and control system (Source: Farid, 2004)

On the other hand, the information processing part is made up of three modules:

- The Holon’s kernel or decision-making which is in charge of the Holon’s reasoning capabilities and decision-making,
- The inter-holon interface for the communication and interaction between holons
- The human interface for input (operation commands) and output (state monitoring) data

Moreover, Botti and Giret (2008) have proposed an agent-based architecture, Fig. I-3, for the information processing part illustrated in the holon’s general view by incorporating three components:

- Holons with autonomous control over the machine behavior they are associated with,
- Two or more holons with ability to cooperate when and wherever it is necessary and,
- Holons with ability to act in multiple organizations called holarchies

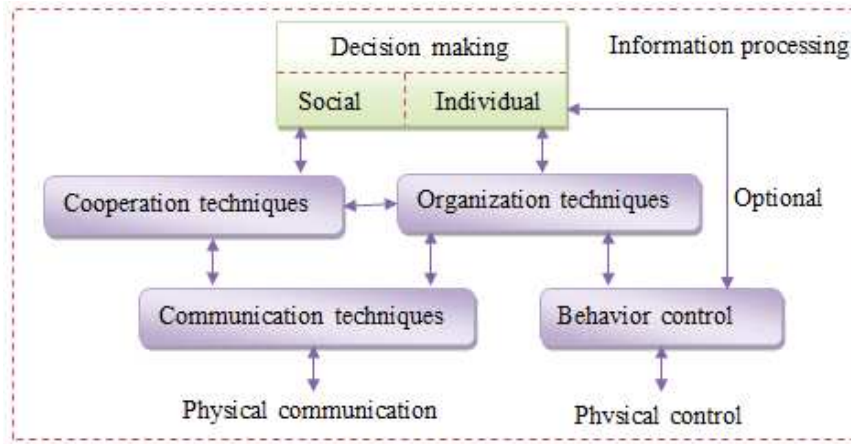


Fig. I-3 Agent-based architecture of holons (Source: Botti and Giret, 2008)

Holon's plans and strategies are communicated from the decision-making module to the behavior control in order to translate them into hardware operations. Besides, the cooperation interactions are initiated by the decision-making module and executed by the specific cooperation techniques using the communication techniques (domain ontologies and languages). In order to reorganize the manufacturing controlling processes, the Holon needs techniques. These techniques are used to monitor other component actions and to analyze the controlling process. In this way, the holons can figure out opportunities for improvement and can start a negotiation process for reorganization (organization techniques) that is executed by means of cooperation standards.

On the other hand, in terms of functionality, a holon may be considered as a composition of an intelligent controlling system (head) and a processing system (base), Fig. I-4. The control system includes different elements such as:

- PMC (process/machine control) to execute controlling plans for the running processes
- PMI (process/machine interface) to provide the logic and physical interface for the processing system through a communication net
- HI (human interface) to provide the human-readable interface
- IHI (interholon interface) which is in charge of inter-holonic communication

And the processing system (base) incorporates all the processing modules needed to carry out the production activities. Likewise, agents are used to manage high-level planning strategies (head), while function blocks manage real-time process/machine low-level control (base). Moreover, there is a holonic kernel running over the function blocks in order to provide the necessary interface between the agents and the base.

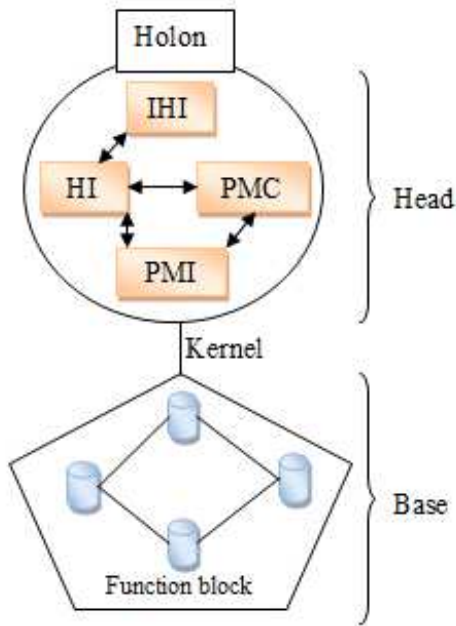


Fig. I~4 Head and base of holons (Source: Farid, 2004)

Holonic modeling: Fig. I~5 illustrates the transition of conventional hybrid systems into the holonic paradigm of model system. The conventional hybrid system was composed of three parts namely controller, interface, and plant. But in the holonic hybrid system, the controller is represented by product and order holons, the interface is by the logical part of the resource holons, and finally the production system by the physical part of the resource holons.

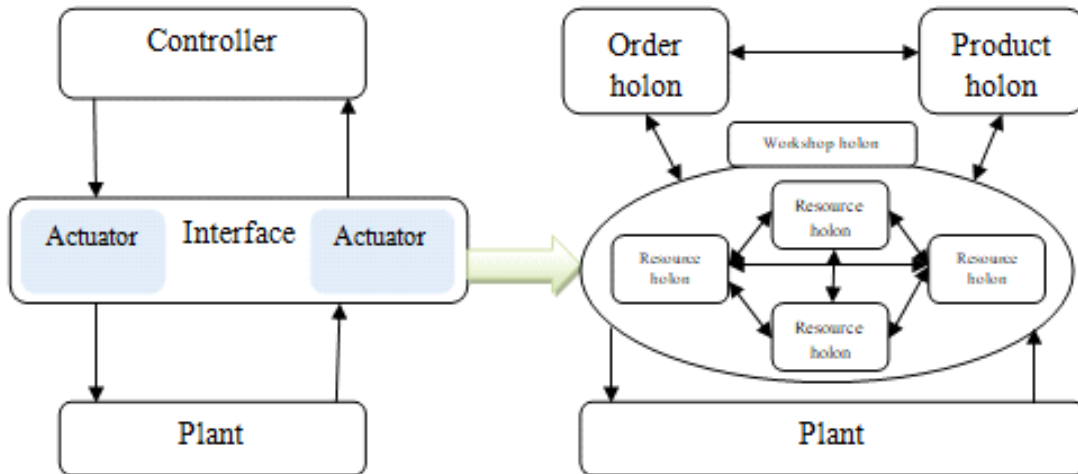


Fig. I~5 Holonic architecture of hybrid system (Source: Borangiu et al., 2015)

Appendix J MatLab and/or Python pseudo codes, Java codes, UML diagrams, and results for the proposed algorithms

Defining the *update state*

```

def update_state(self, now):
    p_leader = self leader()
    C* = self C*
    C = self estimate_completion_time(now)
    if self due_date < C*:
        print("WARNING: due date ({} ) is less than C* ({} )".format(self due_date, C*))
    if C < C*:
        print("WARNING: completion ({} ) is less than C* ({} ), updating C* (new value is {} )"
              .format(C, C*, C*0.98))
        self C* = C*0.98
        C* = self C*
        old_dd = self due_date
        self due_date = max(self original_due_date, min(avrage_C, self original_due_date + self slack_time))
    if absolute_value(old_dd - self due_date) > 1e-6:
        log("[Product Agent] t = {}, due date shift. Original due date: {}, estimated completion:
            {}, allowed slack time {}, new due date {} ({} )"
            .format(now, self orig_due_date, avrage_C, self slack_time, self due_date, -old_dd
                    + self due_date))
        self state = (Eq. (3-16))

    if not self pid in state_log:
        state_log[self pid] = []
        state_log[self pid].append((now, self state))

```

Fig. J-1 A Python pseudo code for *update state* of the product agents within the consensus algorithm

Define the *looping request*

```

def loop (self, now, router_id)
    if not Consensus:
        return False
    Log ("[Product Agent] [Consensus] t = {} product {} is being asked loop or exit by {}".format (now, self.pid, router_id))
    neighbors_leaders = set()
        if router_id == 'br2':
            neighbors = [p for p in products.values() if p.will_go_to('b2', now +40)]
        else
            neighbors = []

    neighbors_leaders.add(self leader())
    for n in neighbors:
        n.update_state(now)
        neighbors_leaders.add(n leader())
    C* = {}
    for p in neighbors_leaders:
        L = [ A[p pid - 1][n pid - 1]*(n state - p state) for n in neighbors_leaders]
        C*[p] = sum(L)
    for n in neighbors_leaders:
        if not n == self leader() and C*[self leader()] - C*[n] > LOOP_THRESHOLD:
            log("[Product Agent][Consensus] t={{} product {} choose loop (neighbors: {})".format(now, self pid, dictionary([(p pid,v) for (p,v) in C* items()]]));
            return True

    Log ("[Product Agent][Consensus] t={{} product {} choose exit (neighbors: {})".format(now, self pid, dictionary([(p pid,v) for (p,v) in C* items()]]));
    return False

```

Fig. J-2 A controlling Python pseudo code of *loop request* for the consensus algorithm

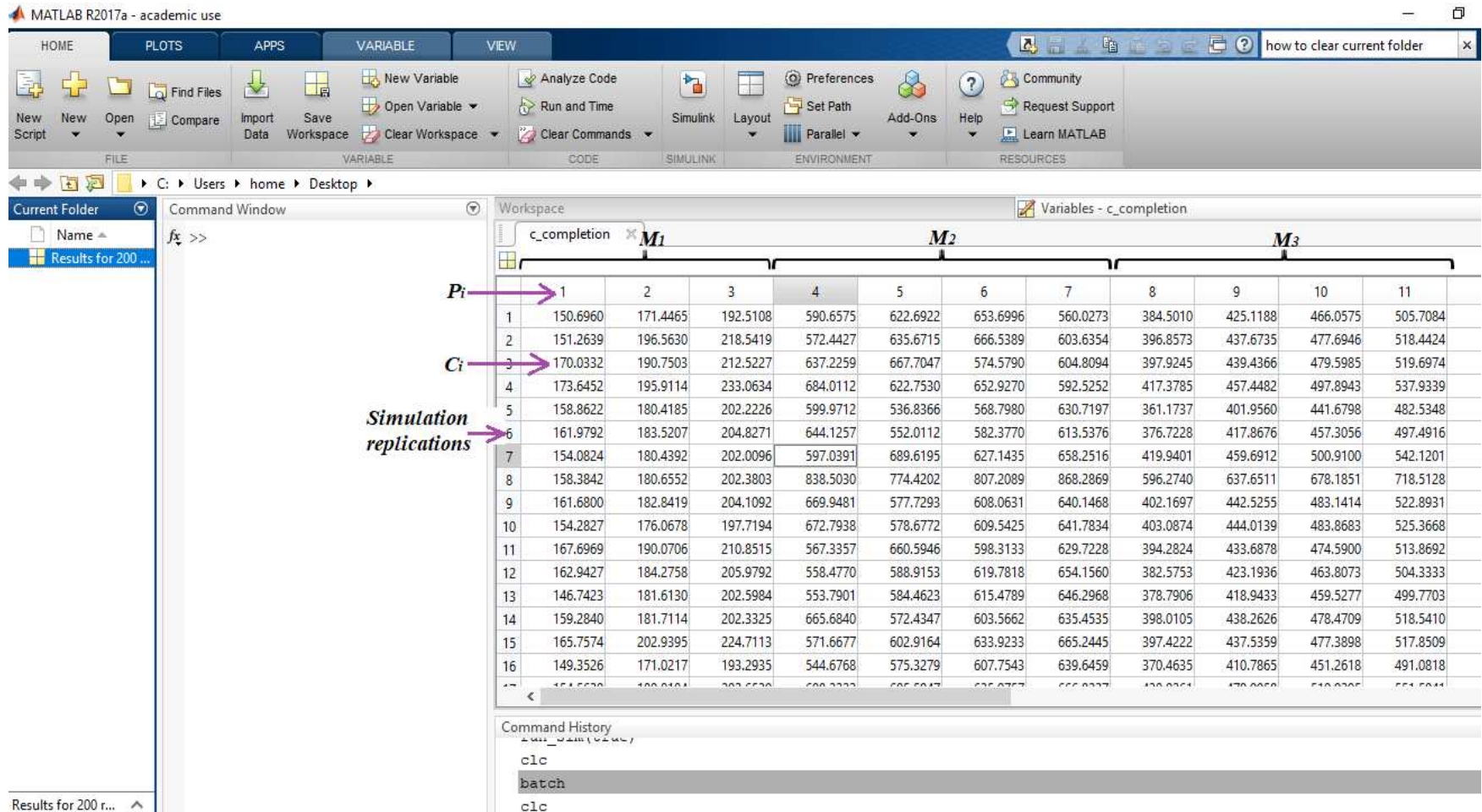


Fig. J-3 Completion times of each product of each manufacturing order for the consensus algorithm

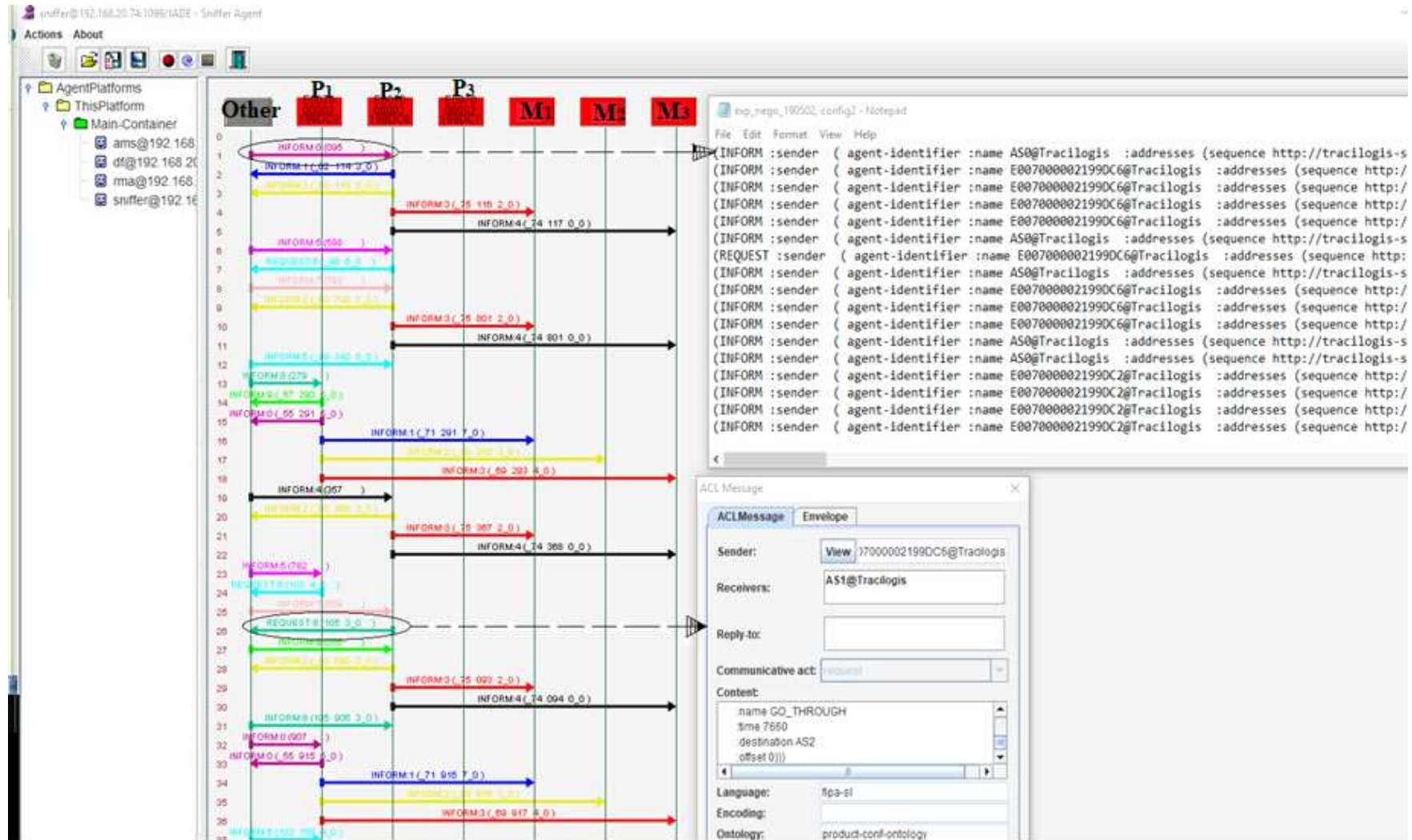
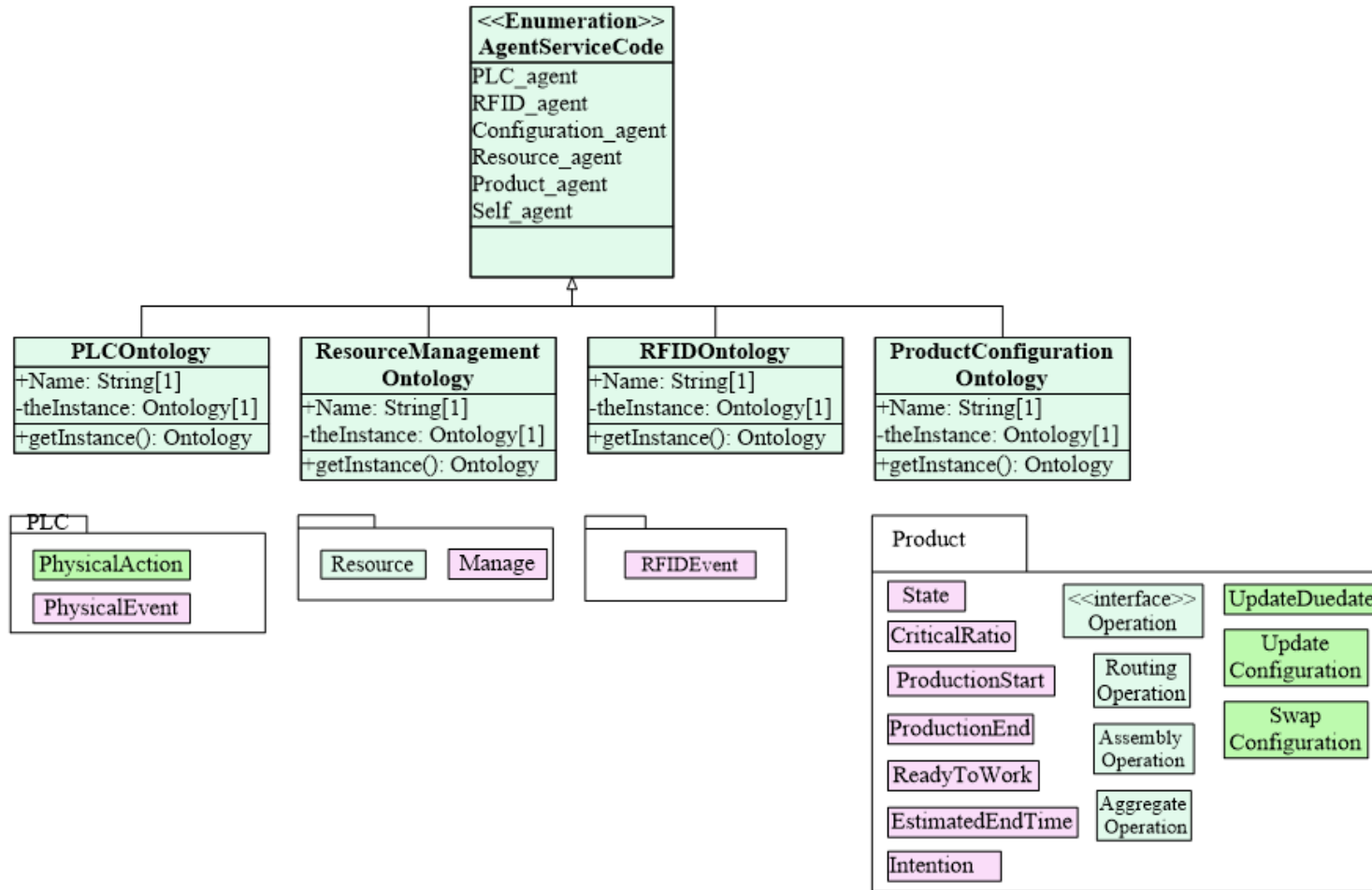
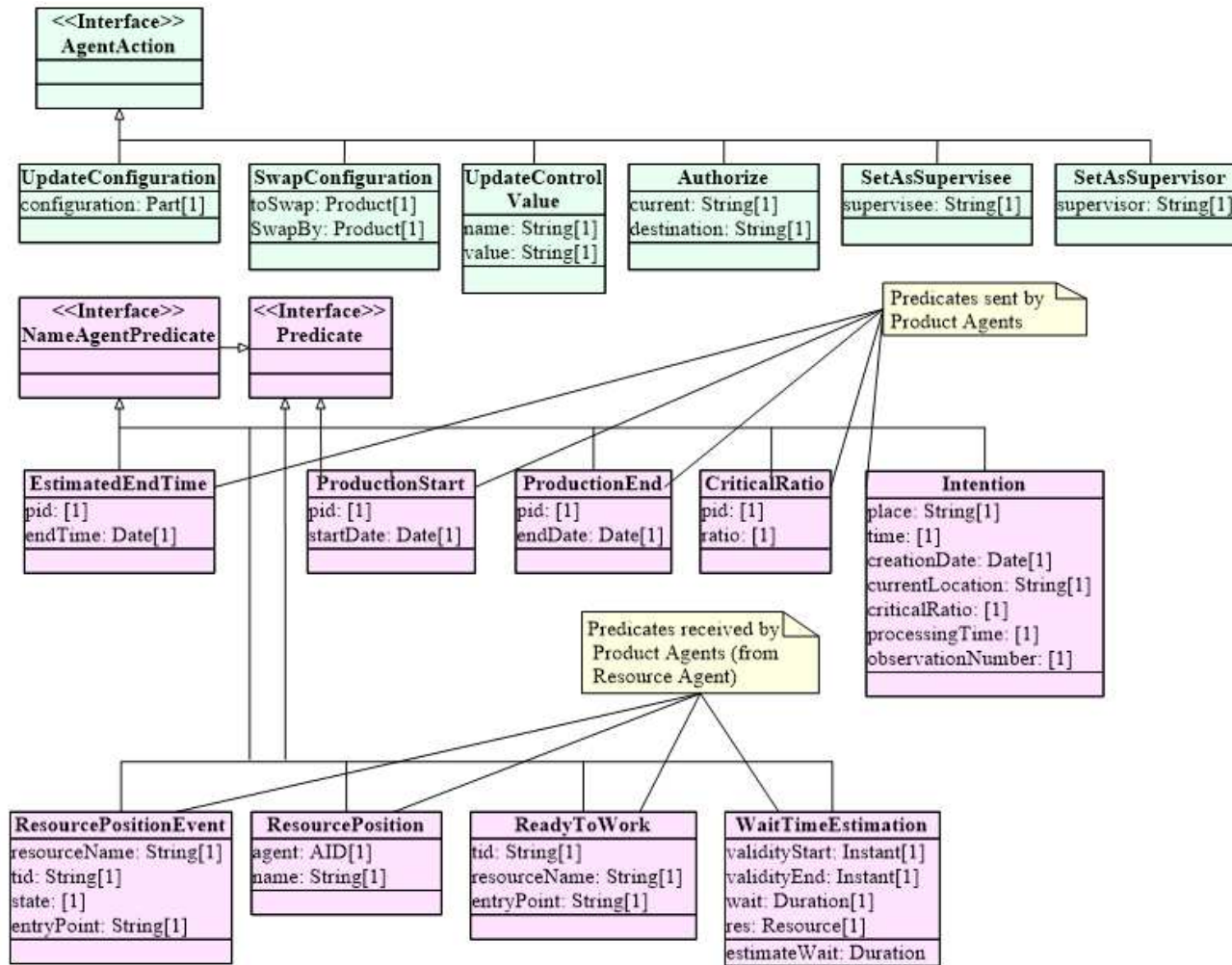


Fig. J~4 Screenshot of topose configuration and communication among agents in one of the implementation experiment



(a)



(b)

Fig. J-5 UML class diagram (a) for topase-ontology of all agents (b) for topase product agent messaging ontology