A Dynamic Multi-Agent Based Scheduling for Flexible Flow Line Manufacturing System Accompanied by Dynamic Customer Demand

Danial Roudi

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Eastern Mediterranean University
December 2015
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Cem Tanova
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Mechanical Engineering.

Prof. Dr. Uğur Atikol
Chair, Department of Mechanical

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Mechanical.

Prof. Dr. Uğur Atikol
Supervisor

Examining Committee

1. Prof. Dr. Uğur Atikol
2. Assoc. Prof. Dr. Qasim Zeeshan
3. Asst. Prof. Dr. Dariush A.Firouzi

ABSTRACT

Nowadays, the manufacturing enterprises endeavor to avoid the risk of losing competition, and strive to respond precisely to the customer demands by improving their flexibility and agility, while maintaining productivity and quality. In this thesis a multi-agent based dynamic scheduling for flexible flow line manufacturing system is presented while considering dynamic customer demand. A case of unplasticized polyvinyl chloride (uPVC) door and window manufacturing company is considered to confirm the design of Multi-Agent System (MAS). A multi-agent dynamic scheduling system is developed based on Prometheus methodology^{TM}. The Prometheus methodology is employed for specifying, designing, implementing and testing agent oriented software systems in a detailed manner. Moreover, JACK agent platform is used as an implementation platform. Simulation test platform based on existing case study is developed for validation and verification of the proposed MAS. Two scenarios are defined to compare the conventional system with the proposed system. In the first scenario the acceptance rate for both conventional system and multi-agent based dynamic scheduling system is evaluated and compared. In the second scenario the effect of changing number of order is evaluated for both systems. The simulation results show that the proposed multi-agent dynamic scheduling system outperforms (acceptance rate, changing number of order, lead time) the conventional system. Furthermore, it was found that the proposed system performs better in terms of run time.

Keywords: Dynamic Scheduling, Multi-Agent System, Prometheus Methodology, JACK, Dynamic Customer Demand

ÖZ

Günümüzde, üretim işletmeleri rekabet kaybetme riskini önlemek, verimliliği ve kaliteyi korur, esneklik ve çeviklik geliştirerek müşteri taleplerine tam cevap için çalışıyoruz. Bu tez çalışmasında bir çok agent dinamik müşteri talebini göz önüne alarak sunulmuştur. Esnek akış hattı üretim sistemi için dinamik zamanlama tabanlı Plastikleştirilmemiş polivinil klorür (uPVC) kapı ve pencere üretim şirketi olgusu Multi-Agent Sistemi (MAS) tasarımı doğrulamak için kabul edilir. Bir multi-agent dinamik planlama sistemi Prometheus metodolojisine dayanarak, geliştirilmiştir. Prometheus metodolojisi belirterek tasarımı, uygulanması ve detaylı bir şekilde agent yönelimli yazılım sistemleri test etmek için kullanılır. Ayrıca, JACK agent platformu bir uygulama platformu olarak kullanılmaktadır. Mevcut durum çalışmaya dayanarak Simülasyon test platformu önerildi. MAS doğrulama ve onaylama için geliştirilmiştir. İki senaryo olarak önerilen sistem ile geleneksel bir sistemi karşılaştırma tanımlanır. Birinci senaryoda konvansiyonel sistem ve multi-agent tabanlı dinamik zamanlama sistemi hem de kabul oranı değerlendirilir ve karşılaştırılır. İkinci senaryoda düzenin sayısının değiştirilmesi etkisi, her iki sistem için değerlendirilir. Simülasyon sonuçları önerilen multi-agent dinamik planlama sistemi, geleneksel sistem (sipariş sayısını kurşun zamanı değiştirerek, kabul oranı) daha iyi performans göstermektedir. Bundan başka, önerilen sistem; çalışma süresi açısından daha iyi performans olduğu bulunmuştur.

Anahtar Kelimeler: Dynamic Scheduling, Multi-Agent System, Prometheus Methodology, JACK, Dynamic Customer Demand

TO MY FAMILY

ACKNOWLEDGMENT

Firstly, I would like to express my sincere gratitude to my friend Dr. Ali Vatankhah Barenji for the continuous support to my master study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of thesis. I could not have imagined having better advisor and mentor for my master study.

My sincere thanks also goes to, prof. Dr. Ugur Atikol and Prof. Dr. Majid Hashemipour, who provide me an opportunity to join as intern, without they precious support it would not be possible to conduct this research.

Besides my advisor, I would like to thank the rest of my thesis committee: Assoc. Prof. Dr. Qasim Zeeshan and Asst. Prof. Dr. Dariush A.Firouzi, for their insightful comment and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

Last but not the least, I would like to thank my family, my parents and to my sisters and brothers for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUML	Agent Unified Modeling Language
BDI	Beliefs, Desire, and Intentions
CPN	Color Petri Net
FLM	Flow Line manufacturing
FMCSP	Flow Line Manufacturing Cell Scheduling Problem
FMS	Flexible Manufacturing System
HHS	Hybrid Harmony Search
IMS	Intelligent Manufacturing System
JADE	Java Agent Development Environment
LCL	Lower Control Limit
MA	Manager-Agents
MAS	Multi-Agent System
MHS	Material Handling System
PDT	Prometheus Design Tool
RUP	Rational Unified Process
SD	Standard Deviation
SDFSTS	Sequence Dependent Family setup Times
YBG	Yaran Bahar Golestan
UML	Unified Modeling Language
uPVC	un-plasticized poly vinyl chloride
UCL	Upper Control Limit

Chapter 1

INTRODUCTION

1.1 Introduction

In the last two decade rapid advances in technology caused significant changes particularly in manufacturing environment; moving from a local economy towards a global economy in terms of increasing customization in manufacturing and relatively short-life cycle of manufactured goods, markets demanding for high quality at lower costs product, leading to "mass customization". Similarly, new revolutionary manufacturing concepts and emerging technologies often require updating and integration of current systems throughout new supervisory environments, to avoid their technological obsolescence [1].

Worldwide dependence of goods flow is rapidly growing and in this competition, the companies can no longer be seen acting standalone, and how they are organized and acting independently. In one side, the companies inclined to divide into small sub-companies, under supervision of mother company or not, each one concentration of few specialized ranges of products and having their specific business [2]. Moreover, the companies tend to share skills and knowledge, networking together to achieve global production. With these interpretations, the existence of manufacturing companies is more dependent on quick and swift response to market changes. In this respect, scheduling will appear as a success factor in current global competition to satisfy low cost products. Modern manufacturing is differentiated by their management, communication, performance, as well as by decision making and scheduling, which allow them to predict market needs [3].

The modern manufacturing companies in order to answer the products variety face two basic concepts: (1) building manufacturing plants and (2) enhancing the flexibility of the existing plant. In order to improve these two concepts flexibility of the system is the most favored alternative in manufacturing. To this extent, The Flexible Manufacturing System (FMS) which is a central integration of a group of flexible machining with an automated handling system that working under a centralized computer control is proposed. FMS present various layouts: Loop, Ladder, Open-Field, centered robot and Flow-line. The concept of flow line configuration can be defined as, materials flow between the workstations arranged in a line [4].

An automated uPVC door and window manufacturing Company (flexible flow line manufacturing) is the process of cutting, welding, and assembling parts. This fabrication is composed of several components with different shapes and sizes, although requires hundreds of operation with entrant flow, it is consists of some workstations that each contains one or even more machines. This kind of companies follow the flow line manufacturing system, therefore scheduling problem of this system is called flow line manufacturing cell-scheduling problem (FMCSP) [5]. Additionally, dynamic costumer demand (time based constraints) is added to the problems of this companies, so the scheduling problem can be considered as a real time system problems [6].

A manufacturing scheduling problem with limitation on schedule execution time can be considered as the scheduling problem of a real-time system [7]. The real-time systems are defined as those systems in which the correctness of the systems depends on both logical and time-based correctness [8]. The logical correctness refers to the satisfactions of resource capacity constraints and precedence limitation of operations [9]. The time-based correctness, namely timeliness, refers to the satisfactions of the time-based constraints such as interoperation time-based constraints and due dates. According to the Hyun Joong Yoon et al [5] real time system can be divided into two kinds of deadline, hard deadline and soft deadlines. The real time system with hard deadlines is the system in which time-based correctness is critical, whereas the one with soft deadlines is the system in which time-based correctness is important, but not critical. According to Ramamritham and Stankovic [10], the scheduling techniques of real-time systems are divided into static (offline) scheduling approaches and dynamic scheduling approaches. They perform offline feasibility or schedule ability analyses. For instance, hybrid harmony [11] search is used for solving the multi object FMCSP or A rate monotonic scheduling algorithm. This is known as a best static scheduling method, in which higher priorities are assigned to the jobs with a shorter period and also NP-hard method is used more to solving FMCSP in the static scheduling problem [12].

The traditional scheduling and optimization process (static scheduling), which always deals with a clear schedule and a fixed processing time, while for the actual processing problem, there are many uncertain factors, for example, changes in processing time, product demand, delivery, equipment failure, resources and production variations. By considering the dynamic interference of these factors concluding that the original dynamic scheduling cannot be implemented successfully. Therefore, the rescheduling model and its solution method are of highly important for the dynamic scheduling problem [13].

Dynamic scheduling techniques are advantageous in a system that uncertainty such as aperiodic jobs and machine failures can be taken into consideration [14]. Dynamic scheduling techniques are divided into dynamic planning-based approaches and dynamic best effort approaches [11]. In dynamic-planning-based approaches, schedule ability is checked at run time when a job arrives, and the job is accepted only if timeliness is guaranteed [15]. While, dynamic-best-effort approaches do not check schedule ability at all. Hence, dynamic planning-based approaches are adequate for the real-time systems with hard deadlines, whereas the dynamic best effort approaches are adequate for those with soft deadlines. There are quite few studies which consider dynamic scheduling of manufacturing system, especially dynamic scheduling of manufacturing system by considering dynamic customer demand [16]. For example: A multi-agent decision making system for flexible manufacturing system proposed [8, 14, 17, 18] for handling machine breakdown and optimizing machine utility focus on reconfiguration of control system [19]. However, dynamic scheduling and dynamic customer demand is not considered. In this way, constrains from both functions can also be fulfilled concurrently and an optimum integrated plan and schedule can then be produced.

1.2 Research Aims and Objectives

The aim of this research study is to present a multi-agent based dynamic scheduling decision-making system for flexible flow line (Automated door and Window manufacturing Company) with considering dynamic customer demand. Dynamic behaviors in this company such as diversification of production and reconfiguration are taken into consideration. The multi-agent dynamic scheduling system is developed based on Prometheus methodology ™. Prometheus methodology is a general-purpose design for the development of software agent systems in which it is not tied to any specific model of agency in software platform [20]. The multi-agent based dynamic scheduling system is completely modeled by the Prometheus Design Tool (PDT), which offers full support to Prometheus methodology. The proposed scheduling method is designed mainly for the work cell with time-based constraints although it is applicable for keeping the work cell free from time-based constraints.

In achieving this aim, the major objectives of the research can be stated as follow:

- Investigating the difficulties that exist in current scheduling and control system which can be potentially improved by proposed concept.
- Design software agent systems based on the multi-agent system.
- Implement a multi-agent based scheduling system in flexible flow line manufacturing system.
- Verifying the proposed multi-agent system with compering real case.

1.3 Research Methodology

The core of the research work is the Prometheus methodology for establishing agent based dynamic scheduling .We utilize and perform all four steps of the Prometheus methodology, and achieve to the goals, perception and actions of the system. By this representation multi-agent base dynamic scheduling will be designed for improving the shortcomings of the system. At the end, JACK as an agent platform is utilized to transition of detailed design into implementation. Simulation test platform for verification of design and implementation or in the other word, for evaluating the potential benefits of proposed multi-agent system is created.

1.4 Structure of Thesis

The rest of dissertation is organized as follows: In Chapter 2 (state of the arts) the flow line manufacturing system and its problems will be explained in detail. Moreover, the working mechanisms of multi-agent technology and how it can be helpful for scheduling and controlling of flow line manufacturing will be discussed. Since, the aim of this dissertation is presenting a multi-agent based dynamic scheduling decision-making system for flexible flow line manufacturing with considering dynamic customer demand; variety of approaches and tools for structural modeling of a business process will be introduce in this chapter. In chapter 3 the definition of Prometheus methodology is expressed; the design of proposed multi-agent system as well as a brief description of case study is expressed in chapter 4. In the chapter 5 the JACK agent platform for implementing proposed MAS is utilized and Simulation test platform for verification of design and implementation is developed. The reasons to develop simulation test platform also mentioned in this chapter. At the end, the result of simulation is compared with the real data of the case study. Chapter 6 consists of discussion and conclusion.

Chapter 2

LITRATURE REVIEW

2.1 Manufacturing System

A production firm is an organization with major communication and commercial infrastructures which main business is concentrated on the production facilities or fabrication of products. Transformation process can be as definition for production, which means the action of converting components/raw materials by using machinery and manual labors to finished or semi-finished products that have value in the market place and usually executed systematically [9].

In manufacturing world, Business, product, process are as key factors, which the integration of these elements are prerequisite by any production enterprise. The business element is involved in provision of goods, services, or both to clients, which is related to marketing and its distribution. The product element is bought as raw materials and sold as finished goods is related to the development of goods/products and design activities. The process element, which is the focus of this research work, is the steps through which raw materials are transformed into a final product or as a simple definition related to how to produce the products.

2.1.1 Classification of Manufacturing Systems

The production industries can be classified according to the type of production order [21]:

- Make-to-Stock, where products is manufactured and stored in warehouse by considering future demand such as textile and shoe industry;
- Assembly-to-Order, where finished products are just waiting for customer order to be assembled, such as the automobile industry;
- Make-to-Order, where the manufacturing line is starting to work just after receiving a customer order, such as in the case of production of machine tools;
- Engineer-to-Order, which is an expansion of make-to-order type, where products are designed and manufactured based on the customer specifications, such as uPVC door and window manufacturing company.

With this representation, the type of production for this dissertation's case study in terms of production order/customer demand is engineer-to-order production.

2.2 Flexible Manufacturing

Manufacturing industries to turn towards the problem of customized demand and products variable have two fundamental alternatives: build manufacturing plants, which follow the capacity and stock of finished products to make even the customer demands, or increase the flexibility of their manufacturing plants, which is related to volume and variety of products. Regardless of inherited complexity in implementation flexibility, still is the most preferred solution in manufacturing.

2.2.1 Flexible Manufacturing System

The Flexible Manufacturing System (FMS) [22] notion combines the productivity of the manufacturing flow layout with flexibility of process plan/layout. In other word FMS is a central integration of a group of flexible machining with an automated handling system working under a centralized computer control in order to reach the maximum flexibility, reducing the transport time, the investment in equipment and setup times and more efficiency in production [22, 23]. Figure 1 an example of FMS, which include a set of work stations, mutually connected by conveyer and material handling system, under integrated computational system supervision.

Figure 1: Flexible Manufacturing System [24]

The FMS offer several advantageous which display a significant role in manufacturing industry [25]: enhance productivity (in some cases even 5 times more than current system), low cost production (50%), decrease the stocks and inventory (85%), enhance the quality of products, reduced response time, enhancing the products customization to clients demand, and etc. The inventory reduction can be important and vital enough in moderating the investment in significant software and hardware in order to create a flexible manufacturing system.

In spite of the main objective of the FMS being to achieve the flexibility, one of the main disadvantage is its inflexibility to the introduction of new products. Flexibility in FMS can be achieved while the particular range of products need to be manufactured, and it is becoming more complicating to insert or introduce a new product plans because of the complexity of automatically performance and the necessary adaptation.

2.3 Flexible Flow Line Manufacturing System

FMS present various layouts: Loop, Ladder, Open-Field, centered robot and Flow-line. The concept of flow line configuration, which is the focus of this research study, the materials flow between the workstations, arranged in a line, and it is proportional for systems in which the material progress is specified nicely. Flow line manufacturing is also called transfer lines and production lines. FLM systems comprise of material, workstations, and storage areas. Material flows from storage area to work stations; each activity on a material corresponding to special workstation and materials visit each workstation just once in a fixed sequence. As a simple definition for flow line manufacturing system (FLM), the output of previous machine is an input of next machine [26]. There is a first workstation where the material enter over and done with the system and a final work station which the material leaves the system. FLM can be also represented as a finite buffer and coming one after another as in the language of queening theory in a system [26]. In that situation, machines are named servers, storage areas are named buffers, and discrete parts are named jobs.

In automated manufacturing system, control plays a key role; FLM is mainly focusing on the manufacturing control, which is responsible of controlling the implementation of the manufacturing procedures.

The manufacturing control systems include the following main function [21]:

- Scheduling.
- Plan execution (dispatching, monitoring and reaction to disturbances).
- Process related functions.
- Pathological state handling (deadlock handling, etc).

The scheduling and its critical role in manufacturing as a main point of this research study is represented in follow.

2.4 Manufacturing Scheduling System

Problems of monitoring through the production process of the products is one of the concerning items in manufacturing control system which deal with, if it is being processed, assembled or even transported and look over carefully and critically in the factory.

As its mentioned, the manufacturing control system include some main functions, in which we focus on scheduling that is concerned with the transfer of operation to resources, within a shorter time and regarding to specific principles and standards. Short-term process planning and the capacity of the valid and attainable resource, which is provided by manufacturing plan, are used by scheduling algorithm in manufacturing execution system/factory plant.

2.4.1 Scheduling

Scheduling can be defined as the optimal access of jobs over the time to resources, where these tasks must to follow a set of constraints that express the temporal association between jobs and the corresponding capacity restriction of the resources [27]. In a simple definition, Scheduling is the procedure of controlling, organizing and making optimal work and workloads in a process of manufacturing. Scheduling is employed to allocate machinery resources and plant, scheme manufacturing processes, scheme human resources and procure materials [28, 29]. Scheduling According to Ramamritham and Stankovic scheduling approaches are divided into two static (offline) and dynamic (online). They announce that in operations research main focus on off-line scheduling while, computer science concentrate more on on-line scheduling [10].

2.4.2 Scheduling Problems and Strategies

Scheduling problems occur in all of the economic fields, from computer engineering to manufacturing technical skills. Dynamic rescheduling problem is an influential subject in current manufacturing system with the characteristic of combinatorial computation complexity. For example: problem of Job shop rescheduling was considered as reducing the total finalizing time under a restriction of the disruption due to the original scheduling [12].

The traditional scheduling and optimization process which always deals with a clear schedule and a fixed processing time, while for the actual processing problem, there are many uncertain factors, such as changes in processing time, product demand, delivery, equipment failure, resources and production variations. In the other word, the manufacturing scheduling due to its dynamic nature, highly combinatorial aspects and the ability to be implemented as a complex combinatorial problem [13].

The dynamic interference of these factors causes that the original off-line scheduling cannot be responsible in a proper way or hardy possible. Therefore, on-line scheduling and its solution method are more applicable and necessary to control shop-floor facilities and significant importance for the dynamic scheduling problems [30-34].

Recently a wide range of studies in the literature allocated to scheduling problems which is more specifically a non-polynomial (NP) problem [27].

Those systems where the properness of the systems depends on both temporal and logical correctness is considered as a real-time system in which the scheduling problem of these kind of systems as a small part of manufacturing scheduling problem are complex combinatorial optimization problems with temporal constraints [35].

2.5 Agents and Multi-Agent System

2.5.1 Agent

This research study is about designing and implementing intelligent agent systems. By this preface, one of the definition presented by (Wooldridge 2002) [36], which in turn is adapted from (Wooldridge and Jennings 1995) expressed in following: ‘An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives’ [37].

Being autonomous as one the software agent's basic properties, means that agents perform by their self and make their own decision, in the other words agents are independent. Situated as a second basic properties of software agent can be defined in terms of environment where agents tend to be implemented in challenging situation. This property makes agents be different in comparing with other software regarding to type of environment. More specifically, agent's environments normally are dynamic and uncertain. By means of dynamic environment is that, the agents cannot assume that the environment will remain static while they are trying to achieve a goal or in a different state, they change rapidly.

The manufacturing environments are uncertain and forecasting the future states of these environments is not possible at all; for an agent, in most cases, this is because of lack of wide-range information about their environment which is not possible generally and also the environment is not modified to agent's knowledge in a proper way. Finally, these environments are *not reliable* in that the agent's performances through agent's control may fail for reasons that are beyond an agent’s control.

Agents are normally situated in dynamic environments, which make agents responsible to large amount of changes in their environment. To state the matter differently, agents need to be reactive, reacting and respond to changes in their environments in a well-time mode.

Another key attribute of agents is that, they are proactive which means chase after goals over time. There are some properties of objective/goal/aim that makes agents more robust. Persistent as an example of these properties can be useful where agents will be enforced to achieve their goal even after failed attempts.

Although objects can be observed as having a goal and also be reactive, but actually, in the sense of multiple goals, and by considering these goals as being persistent and explicit, they are not proactive. By this representation, proactivity as another property of agents that distinguishes them form objects is proposed.

Since in this dynamic and challenging environment the concept of failure of plan/action is more sensible and not unexpected, so agents in response to such these failures need to recover them, in the other word they must be robust and flexibility as a natural approach to achieve robustness is introduced. By accessibility of a wide range of alternative to accomplish a given goal, the agent has options that can be performed should an action or plan fail. Robustness and Flexibility are also some of properties which distinguishing agents from objects.

As a final property, agents are social or as simple definition agents almost always required to interact with other agents.

An intelligent agent [38] is a piece of software that is:

- Situated – exists in an environment
- Autonomous – independent, not controlled externally
- Reactive – responds (in a timely manner!) to changes in its environment
- Proactive – persistently pursues goals
- Flexible – has multiple ways of achieving goals
- Robust – recovers from failure
- Social – interacts with other agents.

2.5.2 Multi-Agent

Nowadays the new methodology with suitable and proper architecture is needed highly to facilitate the manufacturing activities as a dynamic and adaptable one. For issuing this, the multi-agent approaches are providing favorable approaches through greater workshop to accomplish a large amount of workload and state of producing proposed earnings [39].

Multi-agent system, a branch in artificial intelligence which is consist of a collection of agents (two or even more) that situated to a particular use or purpose within a system, provides a new solution for resolving dynamic and distributed problems.

In general, different types of agent perform different tasks or functions, which may also represent different objects, with different ability and power to be used to do some special obligations. Additionally, based on a control structure, they can be organized dynamically. A definition for multi-agent system can be represented as a problem solver of linked network in a loose manner that act on each other to find a solution for problems that are further than the individual knowledge or capabilities of each problem solver [Durfee and Lesser 1989] [40]. The multi-agent system as one of the most promising control architecture based on the concept of scattered and allocated artificial intelligence have faith in for next generation manufacturing [Maturana and Norrie 1996, Lu and Yih 2001][41, 42].

The important distinguishing qualities of a multi-agent system are as following [43]:

- Each agent because of deficient information and incomplete capabilities for solving the problem has a restricted viewpoint;

- there is no system global control;
- data is distributed and not centralized;
- Computation is not simultaneous and synchronous

The modern manufacturing with new generation of systems is presented as intelligent manufacturing system (IMS) in which dynamic rescheduling problem with involving combinatorial computation complexity is an important subject [44].

The “agent-oriented manufacturing system” concepts as a foundation for making such system a reality have been demonstrated. For issuing this, the dynamic multi-agent based rescheduling system was proposed where to support the autonomous and independent decision for each individual agent or group of agents, the negotiation and communication mechanism between agents were addressed [12].

A Multi-agent based scheduling and controlling has attracted many researchers to find a separate schedule for operations and material handling systems or simultaneous schedule for both. Many researchers develop multi-agent based scheduling and controlling for manufacturing system [7, 23, 44] for supporting to Lack of distribution, Lack of reactivity, Lack of autonomous control systems in manufacturing system [16]. To this extent, a multi-agent based scheduling approach proposed for automated guided vehicles and machines within shop floor [45].

A multi-agent based system to simultaneous scheduling of flexible manufacturing system which is consists of machine groups and material handling system that working under a dynamic manufacturing environment is recommended [46]. To support modular reconfigurable production systems, the design and implementation of a multi-agent-based control architecture is suggested [47].

2.6 Multi-Agent Based Scheduling for Manufacturing System

Today's manufacturing facing immense compression to respond and react resolutely and rapidly to dynamic demand distribution. The flow line manufacturing system which is under the influence of industry development in a way that material handling as a part of modern features of machine can be more practical, for accomplish this purpose scheduling of flow line manufacturing system is the first requirement. In order to enhance the flexibility of manufacturing system and achieve data in a way that enable the best performance (solution) for quick react to dynamic demand and accumulate products, production scheduling and rescheduling approach should be taken into consideration simultaneously [32].

Scheduling and process planning tasks are wasting much time and very complicated; so to facilitate the dynamic and adaptable manufacturing activities, a new system with proper architecture is needed. The multi-agent based system is promising approach, specifically in scheduling part of flexible manufacturing system to achieve more profitability and productivity (Shen and Ghenniwa 2003) [39].

In order to figure out why agents are much more useful, we need to comprehend how the unique characteristic of agents transfer into properties of software systems that formulated and built utilizing agents. The usage and implementation of these properties depends on the application. In addition, it is prominent to understand how these software system properties relate to application types and application areas.

Agent technology has been over a wide area agreed and improved in implementation of the dynamic scheduling and distributed control system. Multi-agent based software platforms are becoming a key technology in modern manufacturing systems built in a distributed manner and usually granted with distributed intelligent functions, such as intelligent manufacturing systems (IMSs) [11, 45, 48].

Many researchers have attempted to support manufacturing activities by developing agent based architectures [45]. Since the late 1980s, a number of researchers have applied agent technology to perform production planning and control on the shop floor [49]. V Kaplanoğlu [50] proposed a real time dynamic scheduling system based on the agent-based system, which has the advantages of less sensitivity to fluctuations in demand or available vehicles than more traditional transportation planning heuristics (Local Control, Serial Scheduling) and provides a lot of flexibility by solving local problems . Kai-Ying Chen et al [51] Applying multi-agent technique in multi-section flexible manufacturing system. To set up the dynamic dispatching rules, a distributed agent based system is implemented, which assist the agents to choose suitable dispatching rule in pertaining dispatching region and also the whole-hog dispatching of manufacturing system resolved through the agents cooperation. Leitao [52] surveyed the literature in manufacturing control systems using distributed AI techniques. Mes et al. proposed a distributed agent-based solution to real-time, dynamic transport scheduling problems, which has the advantages of less sensitivity to fluctuations in demand or available vehicles than more traditional transportation planning heuristics (Local Control, Serial Scheduling) and provides a lot of flexibility by solving local problems [53]. Yazhi Li et al, consider the flow line manufacturing cell scheduling problem (FMCSP) with sequence-dependent family setup times (SDFSTs) for total tardiness and mean total flow-time minimization. Based on the mathematical model of this problem, a hybrid harmony search (HHS) is proposed [54]. Yazhi Li et al, If each family requires the same set of machines and each job is processed on each machine in the same technological order, then the problem is called flow-line (or flow-shop) manufacturing cell scheduling problem (FMCSP) [54].

All of the papers consider solving dynamic scheduling problem by using multi-agent systems. However, in their study there is more than one party (general contractor, subcontractors and their subcontractors) in the model. The system is used to control and resolve the difficulties for project participants in distributed projects, especially focusing on the dynamic solving of the scheduling problem. Fuzzy decision-making and the compensatory negotiation processes are used by the agent types. The methodology of Chen and Wang [55], vahid kaplaoglu can maintain a high degree of openness and flexibility due to their programming structure [50].

All studies use Java programming language and its standards as agent programming language. Up to now there is a few practical implementation of this approach in the real factory cases due to lack of standard way and there is a need for conducting more scientific research in this field.

By this representation, Agent based systems which provided an excellent opportunity for modeling and solving dynamic scheduling problems are proposed. Agent-based models consist of rule-based agents which are dynamically interacting. The agents within the systems, in which they interact with, can create real-world-like complexity. Based on that observation, scheduling (the process of arranging, controlling and optimizing work and workloads in a production process or manufacturing process) is proposed in which it is used to allocate plant and machinery resources, plan human resources, plan production processes and purchase materials to handle this type of problems. As it is mentioned in previous chapter, two kinds of scheduling, dynamic and static are defined. They are responsible in current challenging manufacturing area with a feature of uncertainty. The dynamic scheduling techniques to schedule the machine and material handling system (MHs) in a flow line manufacturing system through Multi-Agent System (MAS) approach can be advantageous and taken into consideration.

The main aim of this thesis is proposing MAS based dynamic scheduling for flow line manufacturing system. Our proposed approach not only works in the dynamic environment but also work in the static environment. This approach also could contribute to real manufacturing systems. The multi-agent dynamic scheduling system is developed based on Prometheus methodology ™ [20]. Based on that observation, the context of this study is to schedule the machine and material handling system (MHs) by means of emphasizing flexibility in a flow line manufacturing system through Multi-Agent System (MAS) approach.

Chapter 3

METHODOLOGY

3.1 Introduction

In the previous chapters, the concepts that are used by intelligent software agents were explained. To this extent, in current chapter we develop these concepts and utilize an appropriate software system based on the intelligent agent by providing a detailed answer to entrant disturbances.

In addition to high-level steps such as ‘specify the system’ or even ‘identify the system’s goals’, a usable methodology needs to provide detailed guidelines explaining how these steps are carried out. Also how these steps are performed throughout the process.

As the process is followed, in a same time the design artifacts are extracted, which are used to get more information about the system and its design. For instance, in Unified Modeling Language (UML) a class diagram catches the classes in a system and their association.

In spite of the fact that, design artifacts could all be clearly described using a natural language like English and this not worth having for some reasons: natural languages are innately having many possible interpretations, not structured, and also do not capture definite and particular types of information. Therefore, a methodology should prepare in advance a process with detailed notations and guidelines, which are used to depict the design artifacts.

3.2 Why Prometheus Methodology?

Although there are many existing methodologies for designing and implementing software, particularly object oriented design and analysis have been widely studied and developed, but in fact, object oriented techniques are not perfectly possible to build agent oriented system. For more explanation; agents and objects do have resemblances, but the differences are consequential. It is possible to take advantage of object oriented techniques to design agent systems. However the adaptation is not natural and there is a distinction between end resulting design and good exploit of agents.

For instance, one of important features of agents is being proactive, that they follow their own schedule over time which is realized in terms of goals. A methodology that upholds proactive agents need to leave nothing to the imagination and modeling of the goals, which is not a part of object oriented methodologies at all. In contrast, Prometheus methodologies goals is different in comparison with other methodologies, it supports the design of proactive agents.

Some other aspects of Prometheus which make it distinct from other object oriented methodologies:

- The provision with supplies of a process for determining the types of agents in the system
- Treating messages as components in their own right, not just as labels on arcs.

- This enable a message (or an event) to be treated by numerous access point/multiple plans, which is critical in accomplishing robustness and flexibility.
- Characterizing apperception and react from messages, and looking with great detail at percept processing.

Agents are placed in a particular situation in an environment, and it is crucial to clarify the interface between agents and their environment. Frequently percept processing is significant those are located in the real world and capture their percepts from noisy device such as video cameras.

- Recognize and differentiate the active components like agents and plans form passive components like beliefs and data; (in agent oriented modeling all objects modeled as passive one).
- One outlook of agents [56] attributes to mental attitudes, such as desire and beliefs to agents. If we endorse to this view, then we are interested in design of methodology to address these aspects. Non-agent methodologies do not ascribe mental attitude at all but it is worthy of mentioning that some agent oriented like MaSE [57], do not subscribe to this view also do not address mental attitude. In contrast of other methodologies, Prometheus methodology do capture mental attitude during the analysis and design processes.

Note that although there are big differences between object oriented and Prometheus methodologies, they are also sharing their common features and characteristics. Despite that agent oriented methodologies are not good at all for agent oriented system's engineering, besides they are relevant and many aspects of Prometheus methodology have been based on object oriented techniques. For example [58], use case scenarios are accommodated from standard practice (Jacobson *et al*. 1992) [59].

Interaction diagrams are Unified Modeling Language (UML) sequence diagrams; Agent Unified Modeling Language (AUML) (http://www.auml.org/) (itself an extension of UML) is used directly, and the Rational Unified Process (RUP) (Kruchten 1998) [60] and Prometheus share a similar procedure in implementing an iterative process over clearly outlined phases.

We have given reasons that prevailing methodologies do not provide adequate support for proposing good agent oriented system like UML and RUP. In the following we give concise report of agent oriented, Prometheus methodology which employed in this research study.

3.3 Prometheus Methodology: A Brief Overview

The Prometheus methodology, which is employed for specifying, designing, implementing and testing agent, oriented software systems in a detailed manner. Besides detailed processes, it also defines a range of artifacts that are produced along the way.

Prometheus artifacts relate back to the agent concepts. For instance, percepts and actions are seized in the specification phase of system; the detailed design phase outcomes in events, plants and beliefs; and the entities utilize in the assorted diagrams that directly parallel to the concepts.

Notice that all the artifacts are structured that is significant in preparing tool support in advance of the methodology.

The *Prometheus* methodology comprise of three main phases, described in Figure 1.

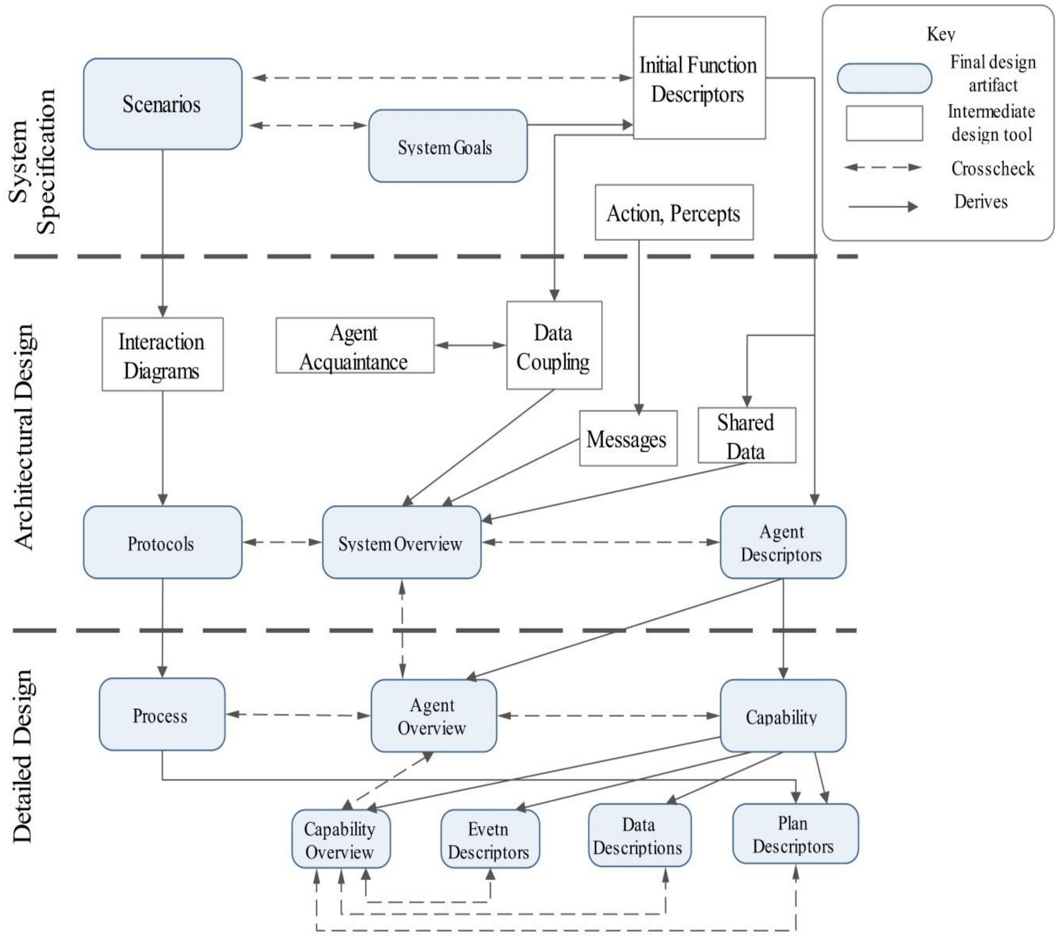

Figure 2: The Phases of the Prometheus Methodology [59]

1. The system specification phase adjusts and identifies the aims and basic usefulness/functionalities of the system, together with inputs (percepts) and outputs (actions).
2. The architectural design phase utilizes the information sent out from a previous phase to decide and settle how agents will interact and which types of agent will contain in the system.
3. The detailed design phase looks at the individual inner life of each agent and how it will perform its tasks in the frame of the overall system.

An implementation as a last phase of Prometheus methodology is omitted from figure because of the dependence on chosen implementation platform.

In following the detailed description of overall structure of Prometheus methodology's phases for more clarification is proposed.

3.3.1 System Specification

The system specification phase focuses on the following:

- Identifying the system goals.
- Improving used case scenarios depicting with the system’s operation.
- Identifying the basic usefulness and functionality quality of the system.
- Specifying the interface between the system and its environment in terms of percepts and actions.

These four steps do not carried out in sequence; on the other word, one will shifts and replaced between them. For instance, since we adding a goal, the use case scenario demonstrating how this goal can be achieve might be added. The scenario might embrace new goals with new functionalities.

System goals are functional and applicable in which its captures at a high level and this is what the system needs to be able to do. Because goals are well enough at high level description, they inclined to less likely to alteration over time as compared to functionalities. Identifying the goals consequence in a group of goals that have been collected, where each with a name and descriptive statement.

Use case scenarios are defined as an example of the system's operation. They will be more practical in the case of being specific than goals and consequently tend to easier visualize and accessible.

However do not cover the entire issue. Regardless of details, a use case scenario is composed of a progression which is coming as steps that take place during the system's operation, along with a description of the context in which this sequence of steps could occur. The source of use case scenarios is object oriented design; but the details are different.

Functionalities formed by grouping related goals and data, percepts and actions. For instance, in electronic bookstore definition we may embrace functionalities such as stock purchasing functionality or delivery handling functionality in which these functionalities should be specific enough to describe.

The environment that agents will be situated is also specified which is defined in terms of percepts (the information which is coming from the environment) and actions (agents affect's progress to the environment).

3.3.2 Architectural Design

The architectural design phase is focused on the following:

- Deciding which types of agent will execute and develop the agent descriptors.
- By utilizing the system overview diagram capture the overall structure of system.
- By utilizing the interaction protocols and diagrams describe the dynamic behavior of the system.

The architectural design major decision is which types of agent should exist. This is executing by grouping functionalities into agent types. Each types of agent consist of one or even more functionalities in which, there are a large number of possible groupings. Decision making on a logical grouping is instructed by careful attention of cohering and coupling. An agent acquaintance diagram is used to help estimating whether a reasonable decision was made and a data coupling diagram is used to help instructing this decision. Once a grouping is chosen, each agent type is described using an agent descriptor form.

Once we identify upon the agents in the system that which agents carry out the specific action on the external environment as well as which agents react to which percepts. We also specify the messages between agents and settle the majority shared data repositories. These items characterized from the overall design of the system in the system overview diagram. System overview diagram most probably is the significant result in the design process where it links the data, agents, external input and output and displays the agents' communication.

The communication paths are displayed by system overview diagram, specifically, which agents talk to which other agents, or which messages are followed by which other messages. The timing of communication is not captured in this part; it is initially captured in scenarios. By means of communication is the activity between agents which is followed by each other. This communication is displayed in agent interaction diagrams. Interaction diagrams, same as scenarios display one possible sequence of messages between agents. For describe all possible interactions, the interaction protocol, depicted using Agent UML is developed [61].

3.3.3 Detailed Design

Detailed design concentrates on improving each agent's internal structure and how it will accomplish its obligation in the frame of the system. Essentially, we gradually purify each agent or as a simple definition we improve its style by explaining capabilities, plans, inner occurrence and structure of itemized data structures which is starts with clarifying agents' internal regarding to its capabilities. Then the internal structure of each capability is depicted. These are refined in turn by turns till whole capabilities events, plans and data. In a same time, we keep working on improving the dynamics of whole system by purifying interaction protocols into process specification.

The detailed design process focuses on the following:

- The refinement of agents in terms of capabilities, giving the agent overview diagram and capability descriptors
- The development of process specification.

The capability descriptors and agent overview are similar and comparable to the agent descriptors and overview diagram, however in present moment concentrating within a single agent. The process specifications as defined by the protocol specification provide detailed scenery of an individual part of agent in a certain process which is at a global level.

The detailed design process, also focusing on:

- Design of the plans in terms of capabilities and the events which is generated by these plans, as captured in capability overview diagrams.
- Specification of the plan's algorithm, besides associated data and detailed specification of events. All these captured in data, plan and event descriptors.

The capability overview diagrams and agent overview diagrams are similar in style where plans must point out that which incoming event is the trigger event. In developing the plans as a part of dynamic specification of the system, the process specification as well as structural information is taken into consideration for example what triggers them. The descriptors provide the necessary detail to move into implementation which is depends on aspects of the implementation platform.

We committing just in this final part of detailed design to a specific implementation style. More specifically, we assuming that plans are trigged by events which make it possible to have multiple plans that can handle a given event type, where the choice of plan to be used is determined at run time. This assumption is parallel to entirely class of implementation platforms including BDI systems such as JACK.

According to the literature review, we realize that exist gaps in the field of flow line manufacturing system, specifically in scheduling and controlling. By this consideration, we are going to design MAS based dynamic scheduling and controlling system based on the Prometheus methodology. For better understanding, a real case is utilized in order to validate of proposed system.

Chapter 4

A CASE STUDY OF DESIGN DYNAMIC MULTI-AGENT SCHEDULING SYSTEM

4.1 A Short Description of YBG

YBG is a small enterprise that produces engineer-to-order uPVC door and window. YBG is located in the north of Iran and provides doors and windows to representatives all over IRAN. The company is made up of two main departments: an administrative office, located in a downtown area with major communication and commercial infrastructures that expedite relations with partners and clients and the promotion of new products, and a production facility located a few kilometers away in the industrial area.

The production process involves the production of the frames of the windows/doors and several assemblies phase; in addition, test and quality control phases are performed.

The window components, such as fittings, profiles, and glasses, are ordered to partner companies that manufacture them according to the windows/doors designs. The windows frames are produced in the YBG plant from first substances (uPVC Profiles). Nearly fifteen models of doors and windows are manufactured: Tilt and turn windows, slide hung, top light, sliding folding, center hinge/pivot. Three profile qualities are available: high quality with two different colors, medium with five different colors and economic one with two different colors. Figure 2 shows the plant layout and summarizes the process flows of uPVC manufacturing company.

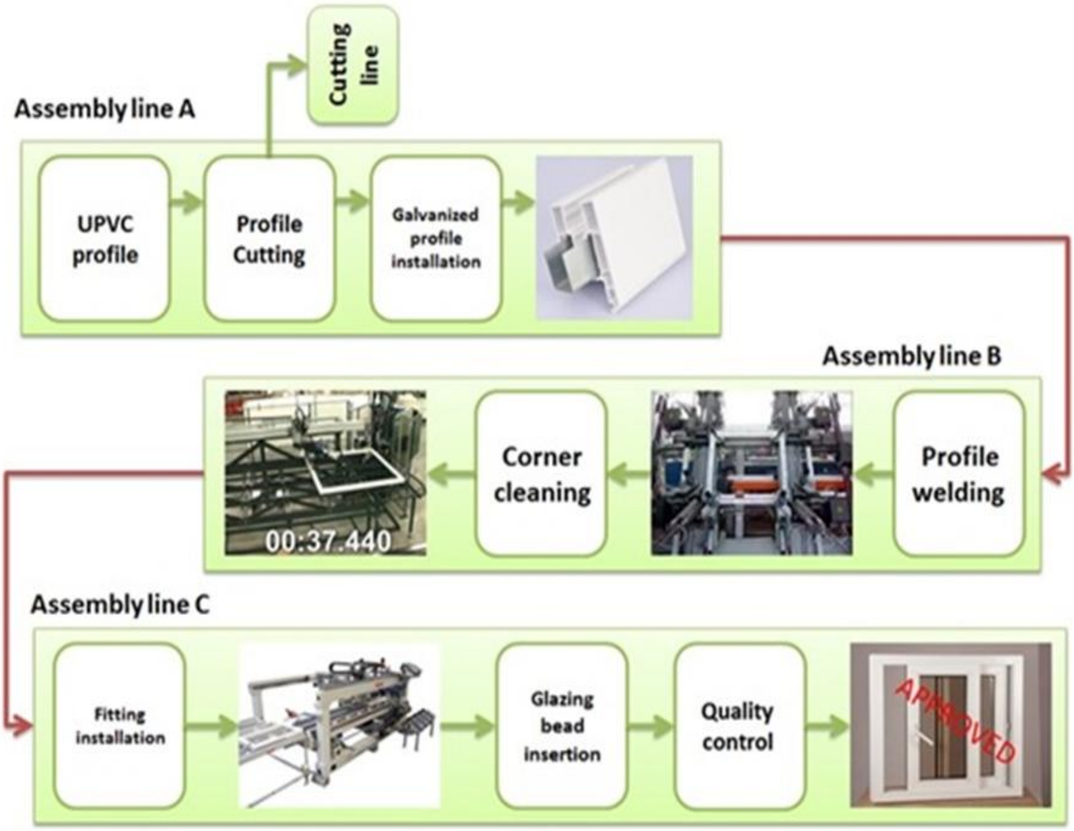

Figure 3: Layout of the uPVC manufacturing company [62]

First substance (uPVC profiles) and finished windows components are stored in four different warehouses. uPVC profiles are standard lengths long enough to manufacture the largest frame size and undergo quality control according to a defined protocol. Finished uPVC components arrive with quality control already certified.

After uPVC profiles arrive at the buffer areas where the cutting center machine is programmed to cut the profiles and labeled them, according to specified orders. Once cut, and stored in a buffer area, then galvanized profiles are installed inside the profiles. uPVC profiles which are installed galvanized profiles are then sent to welding, a process consisting of two phases: welding and corner cleaning, performed by two different machines: a welder and corner cleaner, respectively. The frames are thus ready to install fittings. Two different lines are specialized to install the fittings to frames. Glazing bead insertion is a last process which is performed. Here we can name them window/door. Finished windows/doors are stored in the specific warehouse and quality will be controlled. Windows/doors are finally stored in a finished product warehouse, waiting for their delivery to the customers.

The firm is satisfied with this information system; however, management feels that some modifications are necessary for reacting in timely manner to occurred disturbances in production system. This need is likely to involve multi-agent based scheduling.

The problems that exist in current scheduling and control architecture which can be potentially improved by multi-agent based dynamic decision making are as follows:

- The manufacturing system is scheduled by static scheduling
- The stations (machines) have no autonomous scheduling unit for their operations
- The system lacks the real time scheduling and is not flexible in the case of dynamic customer demand.
- The scheduling of this system in the dynamic environment is NP-hard problem.

Development of a multi-agent based dynamic decision system to address these problems is justified as follows:

1. When the dynamic customer demands accrue, the dynamic decision making system can schedule the system in the dynamic manner.

2. Development of multi-agent based dynamic decision system can be fined optimal scheduling in the machine fail disturbance.
3. The proposed system makes autonomous station level
4. proposed MAS create real time communication in the system

The suggested architecture is clarified by means of a sample of flow line manufacturing system that is comprised of a three stations cell. This company (YBG) is located in the north of IRAN. The purpose is applying multi-agent founded scheduling for dynamic customer demand.

4.2 Design of the Proposed Multi-Agent System

The Prometheus methodology defines [63] a detailed process for specifying, designing, implementing and testing/debugging agent-oriented software systems. In addition to detailed processes (and many practical tips), it defines a range of artifacts that are produced along the way. The Prometheus methodology consist of four steps, the first three steps of this methodology is in design of any agent oriented software are same but the last step namely implementation step is different, In this study Jack will be selected as platform for implementing proposed MAS in the future work.

4.2.1 System Specification Design

System specification phase is first part of Prometheus methodology, System specification design phase consist of four sub phases: Analysis Overview, Scenario Overview, Goal Overview, System Role Overview. Specification of system goals is designed in the Goal Overview diagram, resulting in a list of goals and sub goals, with associated descriptors. This phase is responsible for identification of system goals, development of set of scenarios that have adequate coverage of the goals, functionalities that are linked to one or more goals, negotiations among the types of agents and scenarios of the system. Figure 4 shows Goal Overview diagram of the system.

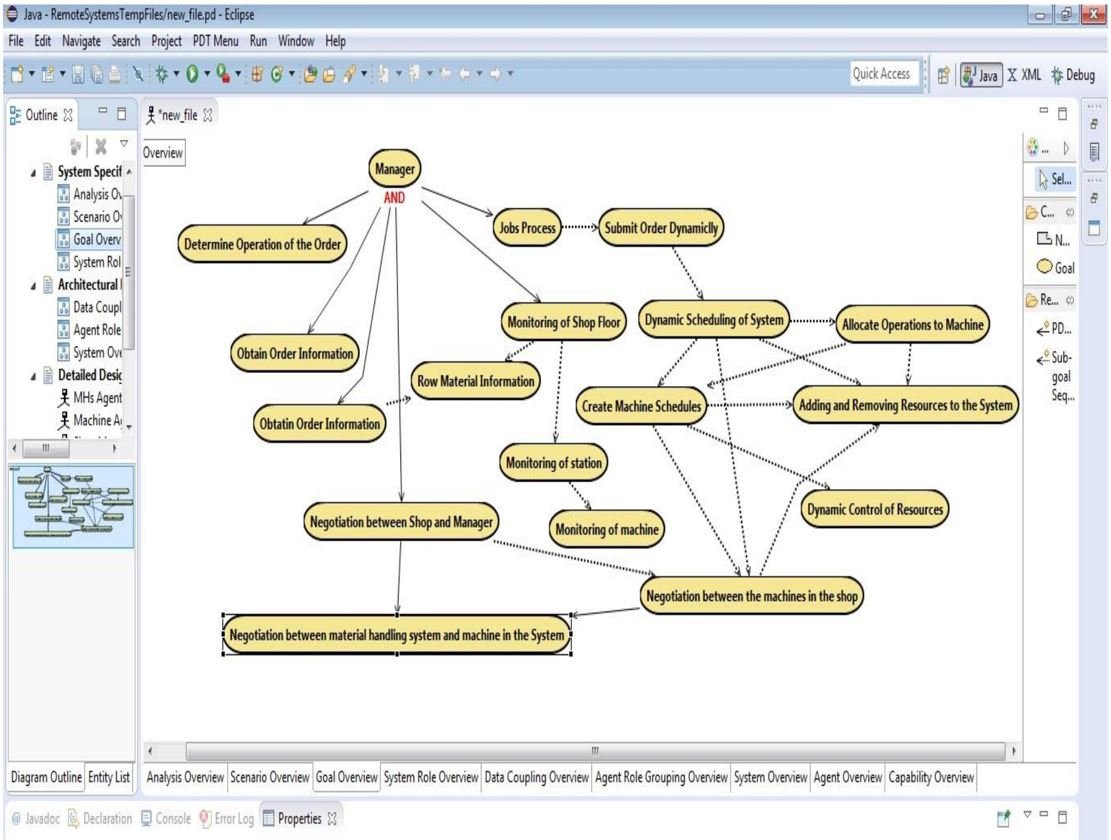

Figure 4: Goal Overview Diagram of the System

Scenario Overview was developed by set of scenarios that have adequate coverage of the goals, and which provide a process oriented view of the system that have to be developed. System Role Overview defined set of functionalities that are linked to one or more goals, and capture a limited piece of system behavior. Figure 5 shows System Role Overview of the System in which, there are four main roles in the system: *Manger role*, *Shop Management role*, *cell role*, *Negotiation Management role*.

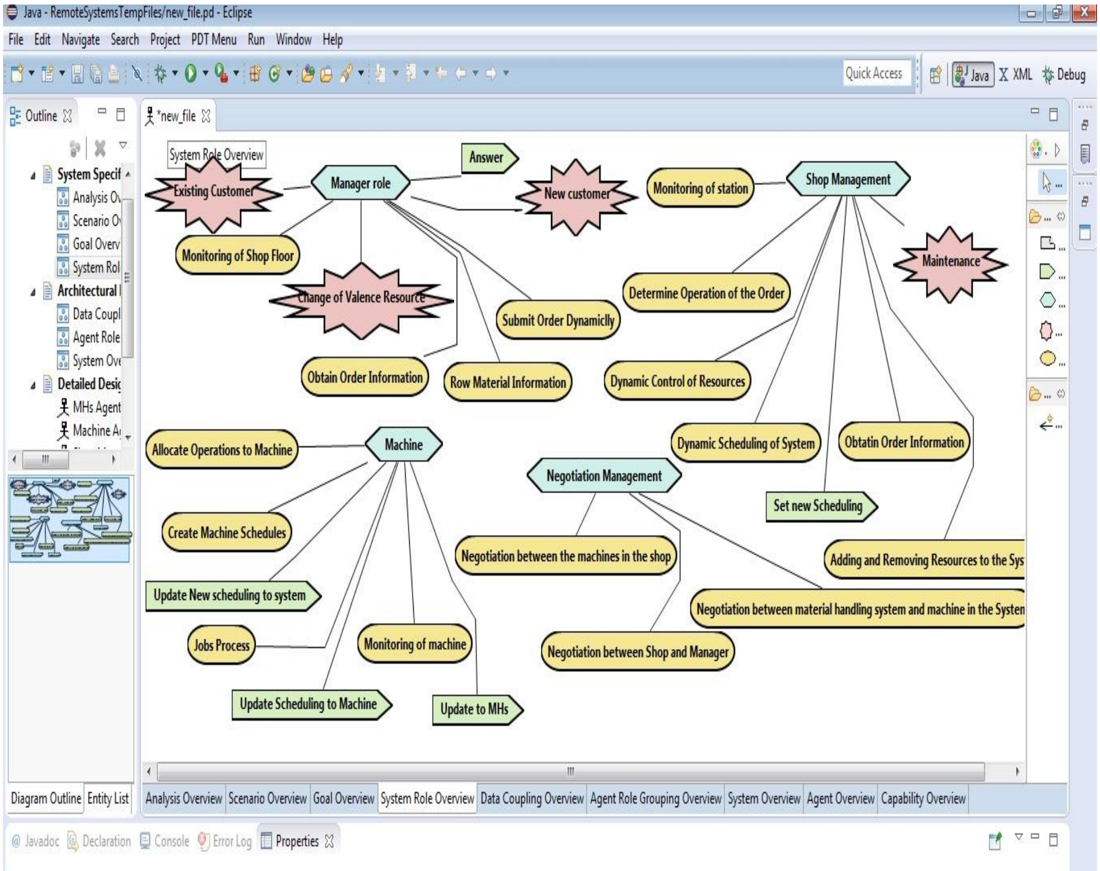

Figure 5: System Role Overview of the System

The sub goal is also designed in the system specification stage. For example four-sub goals of the Machine Scheduling after unpredictable orders arrived goal is defined; “Machine is busy and has task”, “Machine is free and has a task”, “Machine is free and has no task”, “Machine is loaded and has no task”.

4.2.2 Architecture Design

This stage includes the identification of agent types according to Prometheus methodology in which the roles of the agents in the system are determined. This phase consist of three parts namely; “Data Coupling Overview”, “Agent Role Grouping Overview” and “System Overview”. The negotiation protocols between agent types are designed in this phase. A system overview diagram is given in Figure 6. All agents are defined at this stage namely; “Manger Agent”, “Shop Manager-

Agent", "Cell Agent", "MHs agent", "Scheduler Machine-Agent", "MHs Resource Agent", "Machine Resource Agent". The last two agents are interface agents other five agents are software agents that used for dynamic scheduling decision-making system. The proposed system follow top-down approach but by considering real time negotiation between all types of agent. All negotiation protocols between agents are defined at Figure 4 diagram by arrows. Protocols consist of "Order Protocol", "Shop Protocol", "Material Handling System Negotiation Protocol", "Machine Negotiation Protocol", "Resource Protocol" and "Machine Resource Protocol".

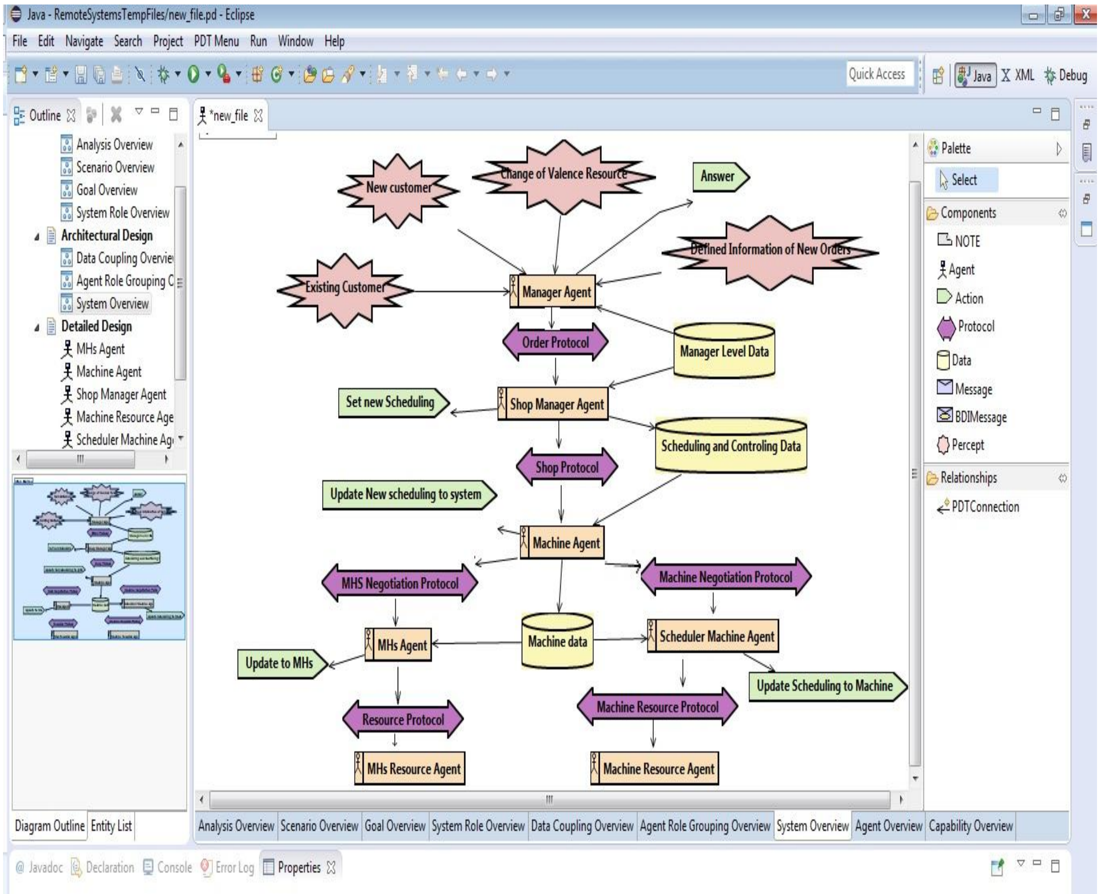

Figure 6: System Overview Diagram in the Architectural Design Stage

In order to describe all interactions protocol we develop interaction protocols by depicted using the Agent UML (AUML). Figure 7 shows an example negotiation protocol between Cell Agent and Scheduler Machine-Agent and MHs Agent. This negotiation shows that how Cell Agent negotiation between Scheduler Machine-Agent for updating new scheduling to the Machine and concurrently communication with Material Handling system for transferring material to the machine and this communication between Machine-Agent and MHs Agent created by MHs Negotiation Protocol. This negotiation protocols are coded in Prometheus ™ software.

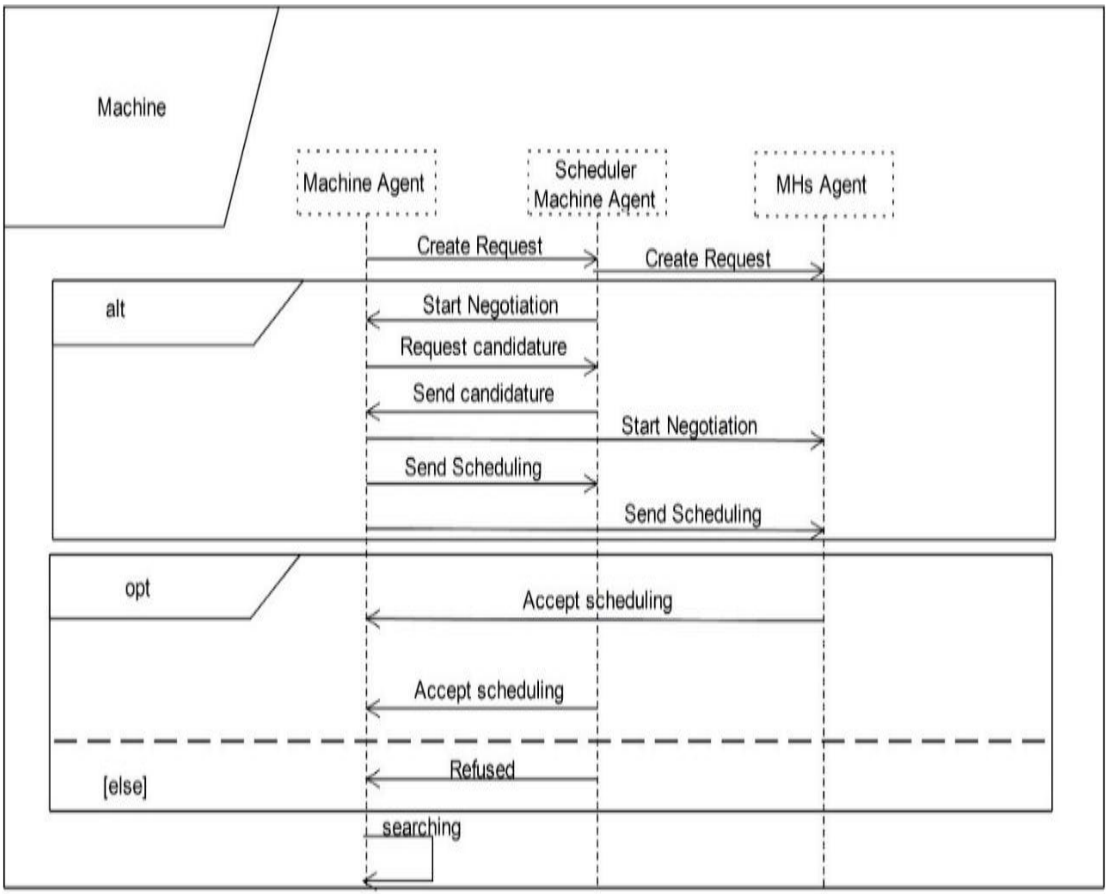

Figure 7: Cell Agent Negotiation between Scheduler Machine-Agent for Updating New Scheduling

4.2.3 Detailed Design

Detailed design is done for each of the agent type one by one in this step. Types of agents at this stage take the message from the event of their environment or other agents, which operate on their plans, and thus they act according to the record in their data base.

For example Manager-Agent (MA) is responsible to managing customer and updating new order to the system. Manager-Agent uses its belief sets, plans and message events so that it will accomplish this task. Manager-Agent architecture is shown in Figure 8 in the form of Prometheus TM design view.

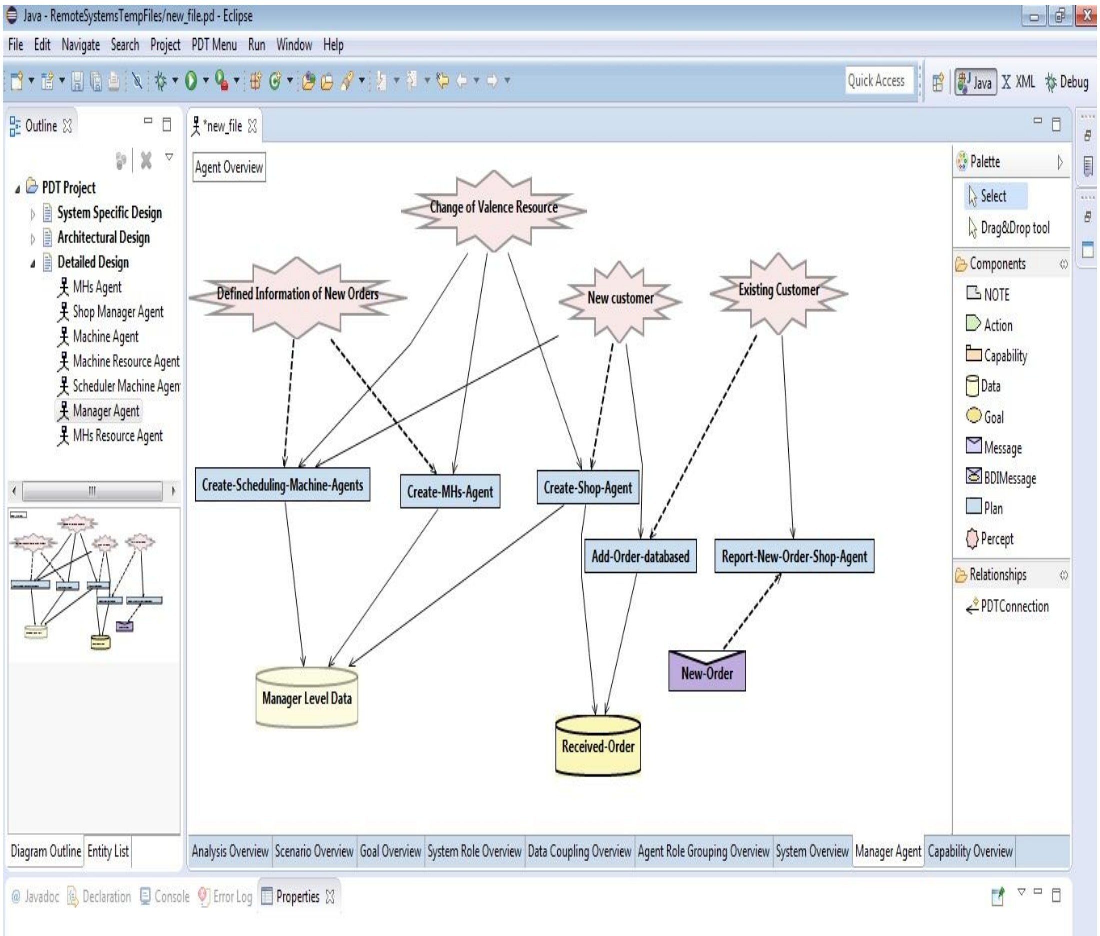

Figure 8: Manager-Agent Architecture

Machine-agent is responsible for managing and controlling the cell level of factory and this agent consist of two sub agents namely MHs Agent and Scheduler Machine-Agent. Detail design of this agent is in the Figure 9 is shown.

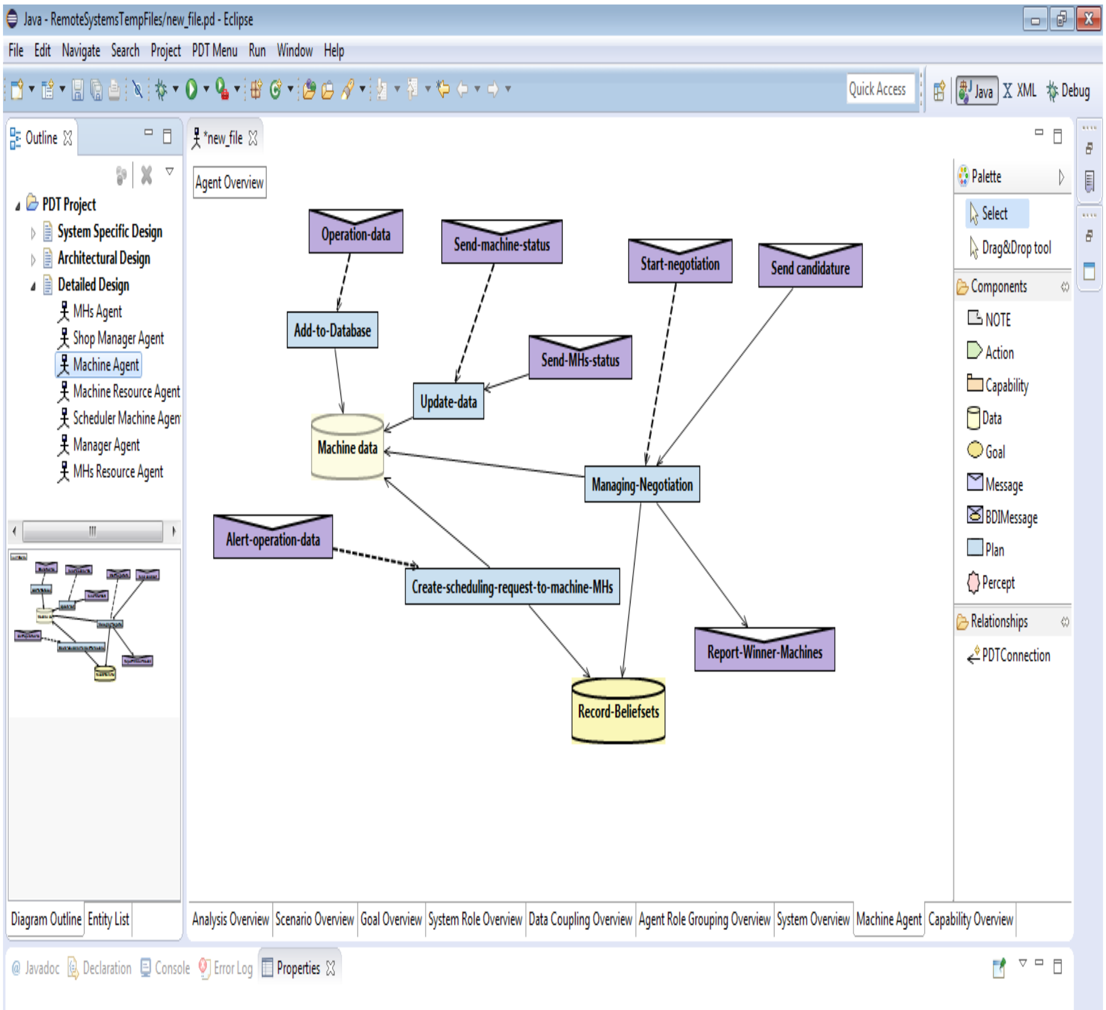

Figure 9: Detail Design of Machine-Agent

The other important agent is has an important role in rescheduling and dynamic scheduling of cell level is Scheduler Machine-Agent. This agent consists of two data based Machine–status and Machine-Negotiation-Results. The detailed design of this agent illustrated in Figure 10.

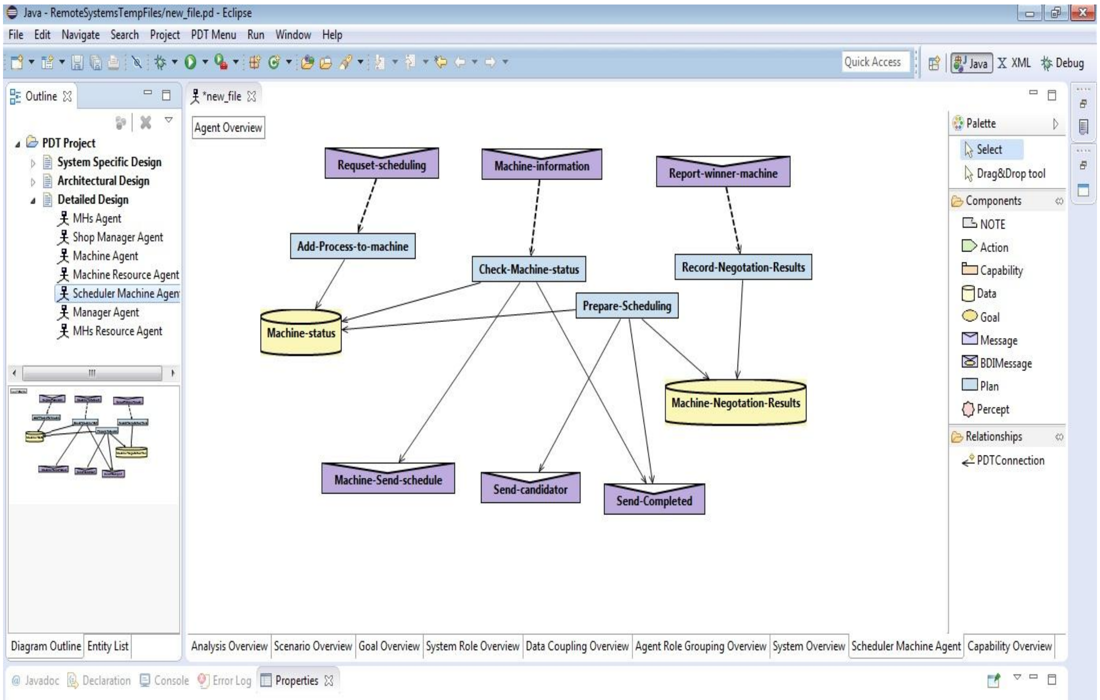

Figure 10: Detail Design of Scheduler Machine-Agent

4.3 Decision Making Mechanism

Algorithm for rescheduling of the system for dynamic customer demand is proposed at this section. Figure 11 is sequence diagram of decision-making mechanism in the proposed multi-agent system. The Manager-Agent informs the new or unpredictable order to the Shop Manager-Agent. Shop Manager-Agent send related question to the Cell Agent and this agent ask from Scheduler Machine-Agent and MHs agent. Scheduler Machine-Agent by having real time communication with Machine Resource Agent send related information to the Cell Agent and this agent by considering the information of Scheduler Machine-Agent answers the question asked by Shop Manager-Agent. Shop Manager-Agent by considering information from cell level make a decision and send to Manager-agent, if Manager-Agent conform this decision, it will send related information to Shop Manger-Agent, and Shop Manager-Agent create new scheduling and new sub agent and send to the Cell Agent and MHs

Agent. Cell Agent send new data to the scheduler Machine-Agent and this agent create update new scheduler to the machine.

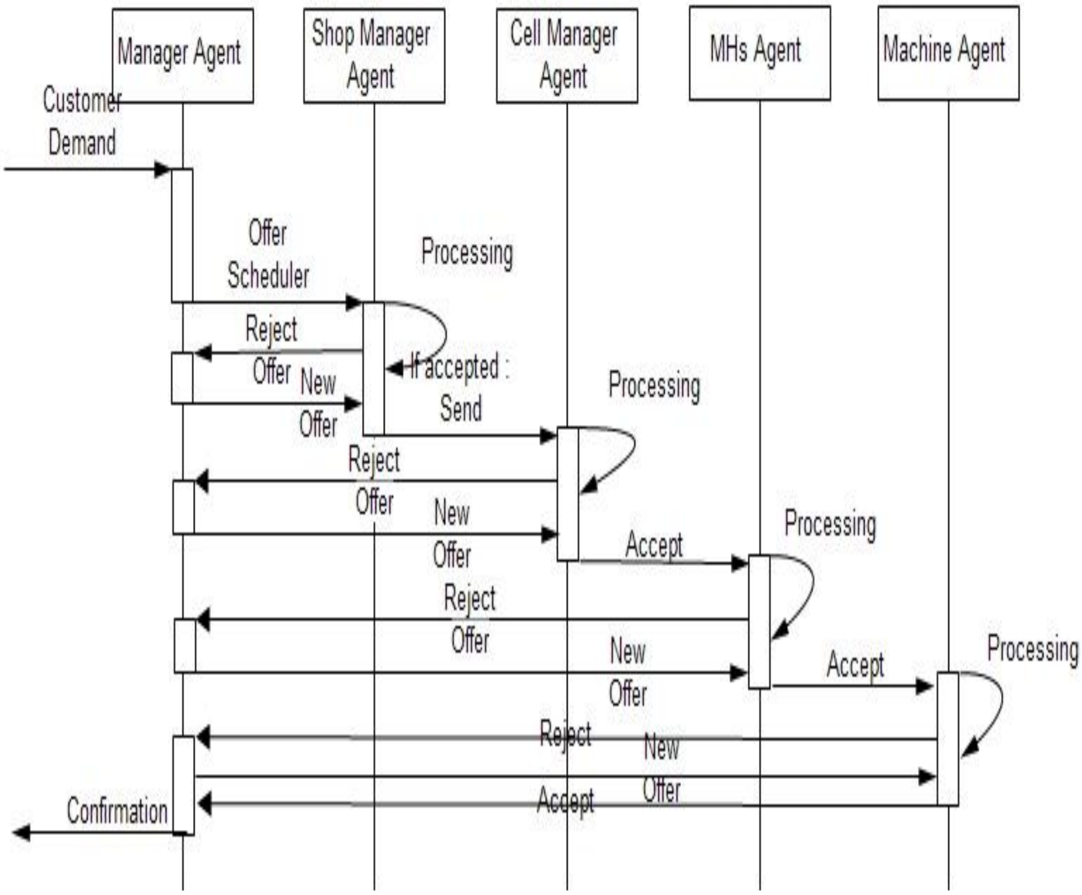

Figure 11: Sequence Diagram of Decision-Making Mechanism

Chapter 5

IMPLEMENTATION AND VALIDATION

5.1 Introduction

The Prometheus design process leads to diagrams and descriptors which depict a range of design entities like agents, goals, capabilities, functionalities, plan, percepts, data, actions, protocols, messages and scenarios. However, all these design entities are not inflected through to implementation. For instance, also functionalities are utilized to determine the agent types but they do not match/correspond to any run-time entity. Confidently, those entities are produced in the detailed design phase to wit, agents, plans, capabilities, beliefs and messages, and also goals, actions and percepts. In this chapter, we briefly observe how we endure the detailed design transition to implementation, and how particular agent platform provide a direct outlining of these concepts.

5.2 Agent Platform

The Prometheus methodology consists of four steps, the first three steps of this methodology in designing of any agent oriented software is common but the last step namely implementation step is different. The high heterogeneity and the vast amount of the available agent platforms for implementing proposed MAS is a fact. Hence, choosing the right or most suitable platform for a given problem is still a challenge for the developers.

The growing complexity and extent of this knowledge creates the need for more powerful tools. To this end, some available data to discover which platforms support technologies, particularly semantic web technologies, that better fit this goal is proposed as a table by kravari [64].

In table 1 [64], the values of criteria, some basic and usability characteristic with operating characteristics and finally Pragmatics of each agent platform is presented.

Table 1: Platform Properties

Platform	Open Source	Simplicity	learnability	Scalability	performance	robustness
JACK	No	User friendly/ Rich in features GUI	Good	High	High	High
JADE	Yes	User friendly, useful GUI, many familiar features	Easy (many examples)	High	High	High
JADEX (BDI)	Yes	User-friendly GUI/Many features	Average	High	Good	High
Agent Factory	Yes	Simple/Poor in features GUI	Average	Good	Good	Average
Agent Builder	No	Simple/Poor in features GUI	Easy	Good	Good	High
Agent Scape	Yes	Simple/Poor in features GUI	Average	Good	Good	Good
AGLOBE	Yes	Simple/Poor in features GUI	Easy	High	High	High

Platform	stability	Programming language	User support	Supporting Prometheus methodology	Cost
JACK	High	Java, JACK Agent Language (JAL) (plus use of XML)	High (extensive documentation and supporting tools)	Yes	Unknown, Free trial version
JADE	High (very quick, fast agent communication)	Java	High (FAQ, mailing list, defect list, API, docs)	Yes	Free
Jadex (BDI)	High	Java (plus use of XML)	Average (docs, mail contact)	Yes	Free
Agent Factory	Good	Java, AFAPL, Agent Speak,	Good (docs, mailing list, forum)	**?**	Free
Agent Builder	Good	KQML, Java, C,C++	Good (Consulting training, example, FAQ, docs, defect reporting, mailing list)	**?**	Agent Builder Lite $99 Agent Builder Pro $925-$4,525
Agent Scape	Good	Java (plus use of XML)	Good (forum, email, docs)	**?**	Free
AGLOBE	High	Java	Average (docs, mail contact)	**?**	Free

Here, in consideration of utilizing which agent platform, it is important to be cognizant that not all agent platforms are provided for plan-based agents. To this end, Padgham [58] divide agent platform into three groups:

1. Those that concentrated upon internal agent reasoning and support goals, plans, and so on. Examples: PRS (Georgeff and Lansky 1986; Ingrand et al. 1992) [65], UMPRS (Lee et al. 1994), JAM (Huber 1999) [66], JACK (Busetta et al. 1999) [67], DECAF, Zeus, AgentBuilder and JADEX (which is an extension of JADE).
2. Those that focus on inter-agent communications. More recent platforms in this class tend to conform to the FIPA standards. Examples: JADE, Zeus, OAA (Cheyer and Martin 2001) [68].
3. Those that focus on mobile agents. Examples: Grasshopper, D'Agents, Aglets

The first class is most applicable in the condition that supports implementing designs. JACK is utilized as an agent platform in this research study which is listed as being a class 1 platform and also it provides support for agent messaging and for locating agents by name.

In following, we focus on clarifying how the detailed design can be implemented using a plan-based agent system. Particularly, the close matches between the concept that supported by JACK and the detailed design results is demonstrated. We begin by briefly describing the JACK platform.

5.3 JACK

JACK is an agent platform based on the beliefs, desire, and intentions model. Agents in this platform presented as having a plan which precipitated by events and also messages that are considered as a sub type of event. Entities declarations in JACK are consist of: agents, event, capabilities, plans and belief sets. Each declaration connected to other entities; for instance, an agent declaration itemize what belief sets it has, what plan and capabilities have within, what events its internally posts, what messages it sends and receives.

5.4 Automatic Code Generation

In this part, the advantage outcome of a Detailed Design phase which is also named as a last phase of Prometheus methodology is taken which is map the skeleton code in JACK naturally. The 'JACK Development Environment' (JDE) as a tool in JACK Intelligent Agents environment supports this automatic skeleton code from a graphical design.

Implementing software and generation code was manually begun from design part, which is possible from the implementation and design to diverge. This deposed to make gap between implementation and design part [17]. To fill the gap, a methodology need to also propose purified design models to be implemented directly in a valid programming language.

Prometheus methodology pursues this attitude, and this is one of the prominent advantages, which differentiate it from other methodologies. The detailed design phase proposes models that quite near to the concepts used in JACK [66]. Approximately most concepts of Prometheus are directly outlined to JACK. For instance, a Prometheus agent exactly turned to JACK agent. JACK also supports capabilities (Busetta et al. 1999) [66] and so capabilities in Prometheus directly outlined to JACK capabilities. Note that some entities (Actor, Goal, Protocol, Role, and Scenario) are not transformed into JACK concepts, for example: JACK does not have concepts corresponding to percepts and actions, so percepts represented as events and actions are simply performed in the frame of the plan body using Java code. Also, JACK does not implement goals directly. Instead, the acquisition of a new goal modeled as an event. Table 2 shows simply which Prometheus entities are translated into their equivalent JACK concepts.

Table 2: Mapping Prometheus Modeling Concepts into JACK Concepts

Prometheus entity	JACK concept
Agent	**Agent**
Capability	Capability
Percept	Event
Plan	Plan
Data	Belief Set
Action	---

The process of code generation is begun by PDT tool and converting this code to JACK. He/she select the directory of code generation and push generate button to automatically generate a JACK folder that contains several subfolders (agents, data, capabilities, plans, events). The same happen for the other entities (capabilities, message, data and plan) in the model, except folder and file extension. Then, JACK Developer Environment (JDE) to import the generated code by PDT is used.

For this, five steps should be done;

1. The Compiler Utility submenu available in Tools menu is chosen in JDE;
2. Selected Convert Non JDE JACK for converting existing JACK code;
3. The folder that contains the code generated by PDT is introduced into content list
4. After defining address and folder name,

5. Finally Generate button is pressed; the new JDE project will be obtained.

Here the internal structure for the files and their supplements are disparate to be readable by JDE. At the end, after making some Java classes and finishing the whole generated 'gcode', the JACK program by utilize the facilities that provided by JDE can be easily executed and transferred into Java.

5.5 Simulation Platform

SMEs include number of machines that connected and working with different robots and also storage system and material handling. Analyzing this type of system in a real company environment is not practicable; because of the resource development for each individual machine is time consuming. Also to recreate the conditions of various experimental examinations when trying to compare assorted scheduling systems [69]. Simulation test platform is used to overcome this problem which is acts as a real system. The advantages of this simulation platform are considering both hardware and software together and its results will be to real implementation.

In this part simulation test platform for evaluating the potential benefits of proposed multi-agent system is created based on this work. The developed simulation test platform is illustrated in Figure 12. This platform contains two main modules [16]:

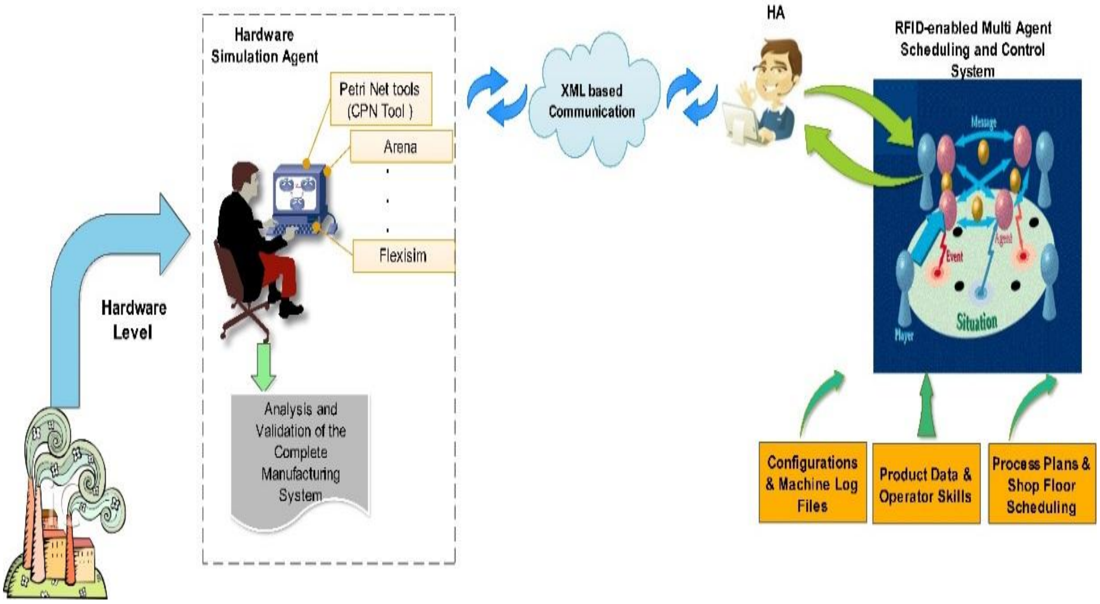

Figure 12: The Developed Simulation Test Platform [16]

a) Hardware simulation agent module: This belongs to the Color Petri Net (CPN) model of the system and is used to analyze the company behavior.

(b) The proposed MASs with added Hybrid-agent module, which is related to the scheduling and control architecture of the system.

The physical actions that take place in the manufacturing environment are characterized by hardware simulation. The established communication model in the CPN is facilitating real time communication among the players. The illustrated figure 12 is hardware simulation-based on CPN model.

In order to testify the efficiency of the rescheduling model and the scheduling system, the petri net model of company is designed. The obtained results by proposed MAS based scheduling and control system were compared with real result of company in the July month.

5.6 Results and Discussion

This section discusses some preliminary evaluations of the proposed MAS in terms of the dynamic customer demand, number of excess products and the production lead-time when decrease in volume is requested during ongoing process. No setup time was considered, and it was assumed that negotiation failure would never occur. The transporting operations were performed by robot and conveyor, and orders were queued in order of arrival. Each transport action required 1 min. For each part, average processing time equal 20 min. The adopted time unit was 1 second, as in the standard time data provided. The warm-up period of 200 min used for fill the machine queues and to obtain steady state results.

All basic information such as dynamic customer demand and performance of system is based on the case study. For comparison of the conventional system with the proposed one, we acquired the data of the company in July of 2015. This month was the most important period in terms of the dynamic customer demand from the viewpoint of the company.

In our first scenario from the viewpoint of company's manufacturing process, customers divided in two groups. First customers who are ordering high volume of company’s products and other customers with low volume order. The mentioned company by utilizing modern machinery in its field has a daily average production capability of 150 doors and windows. The daily average production has a direct association with series of size. It means that to what extent the sizes are constant and equal, a series of similar orders will result in dramatic increase of daily average production.

This factory set daily 100 limits to the first types of orders and the rest of production capability is devoted to second types of customers with low volume order. This procedure brings to works in order to manage the fluctuation which is caused by differentiates in size and reaches the predetermined daily average production which is produce 150 doors and window.

In this scenario, the main focus is on the second type of customers which make disturbances to the system. We seek an improvement in acceptance of second types of customers as well as first type while devising a comprehensive scheduling in terms of MAS.

It should be noted that, among every working day normally 5 to 10 people for contract with different deadline and volume attend to administration office or company's representatives. The company in terms of these dead times and volume try to accept or reject client's requirements.

Table 3 illustrated customer demand of July month for company.

Table 3: Dynamic Customer Demand in July Month

Day	**Door**	**Window**	**Total**	**Day**	**Door**	**Window**	**Total**
1	2	13	15	**17**	0	0	**0**
2	3	17	20	**18**	47	20	67
3	0	0	0	**19**	0	73	73
4	26	22	48	**20**	7	32	39
5	0	0	0	**21**	14	44	58
6	4	22	26	**22**	15	35	50
7	15	40	55	**23**	17	40	57
8	15	23	38	**24**	0	0	0
9	13	22	35	**25**	0	34	34
10	7	43	50	**26**	0	34	34
11	0	0	0	**27**	14	16	30
12	0	56	56	**28**	0	41	41
13	5	48	53	**29**	14	44	60
14	16	44	60	**30**	7	43	50
15	4	46	50	**31**	0	0	0
16	28	29	57	**Total Demanded Products**			**1156**

After running simulation platform for proposed MAS, the dynamic demand was sent to system. Table 4 shows total accepted parts from each day in July than compare the result of proposed and conventional system.

Table 4: Total Accepted Parts from each Day in July in terms of Conventional and MAS

Day	Accepted by MASs	Accepted by conventional system	Day	Accepted by MASs	Accepted by conventional system
1	15	15	**16**	49	42
2	20	20	**17**	0	0
3	0	0	**18**	50	37
4	42	40	**19**	50	50
5	0	0	**20**	39	39
6	26	26	**21**	50	46
7	50	48	**22**	50	48
8	38	36	**23**	50	46
9	35	33	**24**	0	0
10	50	48	**25**	34	34
11	0	0	**26**	34	34
12	50	50	**27**	30	30
13	50	48	**28**	41	40
14	50	46	**29**	50	46
15	50	47	**30**	50	47
			Total Acceptance	1053	996

We defined acceptance rate for each system, as Ra;

$$Ra = \frac{\sum_{1}^{n} \text{acceptance of system}}{\sum_{1}^{n} \text{demeand of customer}} \qquad 1<day<n$$

We observed an improvement of 6 % in the rate of acceptance for multi-agent based dynamic scheduling system (0.91) to the conventional system (0.861) in July. This rate shows that number of product in the MASs is more than conventional system because the system can accommodate rescheduling using the dynamic model.

Second scenario is a changing number of orders; on the other hand reduce the number of excess products by accepting order changes during the time that possible. The production time for special contract which is related to special client (first type of customer) has a direct relationship with product and production plan for the day in question, and the delay time from the original order to the acceptance of the changed order which are evaluated as part of our comparison.

Firstly, by considering the production schedule we discuss how many excess products would be produced when a disturbance get in throughout the system by changing in number of product, specifically reducing the number of requested products during the production process. The amount of excess products was acquired by simulation. In the simulation, the number of products decreased from 100 to 50 products, requested by client. The order change time was varied from 1 to 24 working hours. Figure 13 demonstrates the simulation results. If the change in order occurs at initial 3 work hours of the production, the conventional system is able to keep 0 excess products. This system achieves 50 excess products if the order is changed after 6 working hours. Meanwhile, MASs will kept 0 excess products during the initial 9 working hours. Also, the number of excess products increases gradually, if the order is changed between 9-21 hours, and the system produce 50 excess products if the order is changed after 21 hours. This is because MASs can cancel manufacturing the products before the components are processed by system and concurrently can reschedule system in a dynamic manner.

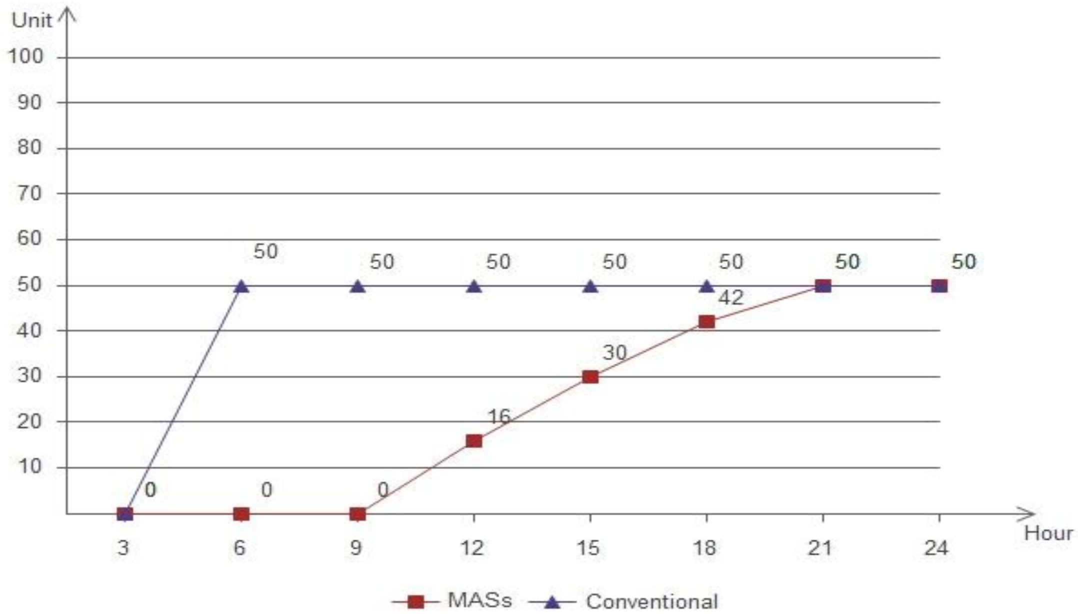

Figure 13: Comparison of MAS and Conventional one in terms of Order Change

Our main target in second scenario is production of 50 products during the first 24 hours after receiving special customers' order. As it is obvious, this aim for both systems are satisfied with differentiate that, the decision making time for MASs is equal to 12 hours. In contrast, this for conventional system is just 3 hours. Consequently, the slope of graph representing conventional system is 16.6; on the other hand the graph's slope calculated for the MASs is 4.1. This demonstrates higher agility of MASs in comparison with former system in terms of ad hoc events.

In the simulation tests, the two scenarios were also evaluated by analyzing the manufacturing lead time. Manufacturing lead time is the total time required to manufacture an item, including the order-preparation, queue, setup, process, move, inspection, and put-away times.

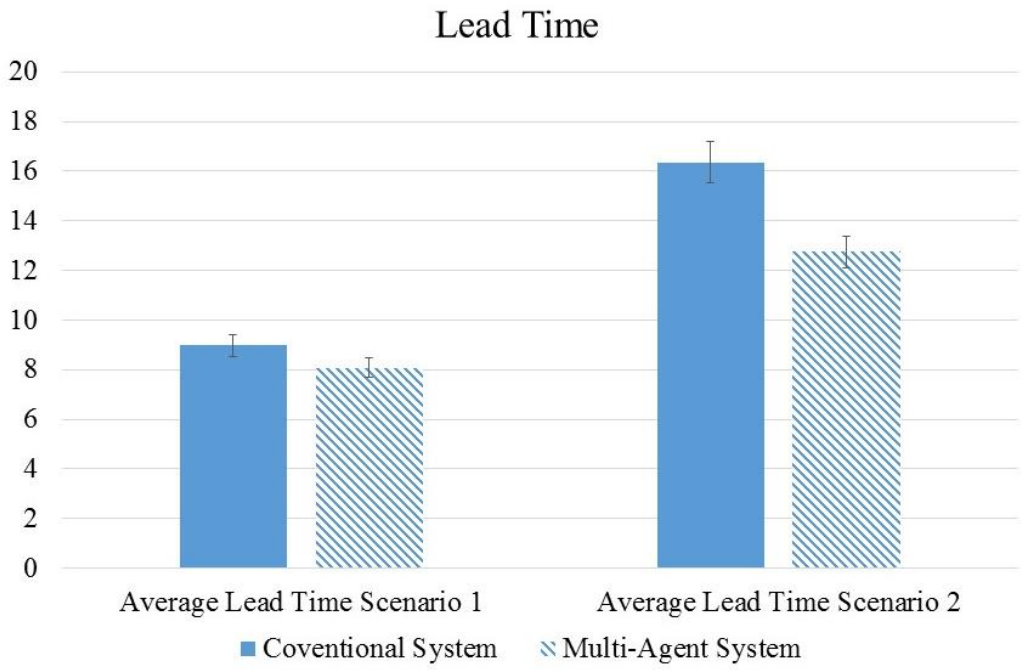

Figure 14: Manufacturing Lead Time

Table 5: Lead Time Results

Lead Time					
Scenario	Type	SD	Mean	LCL 95%	UCL95%
1	A	1.43	8.98	8.529	9.431
	B	1.297	8.08	7.671	8.489
2	A	2.267	16.363	15.649	17.078
	B	1.788	12.756	12.192	13.32

Chapter 6

CONCLUSION

6.1 Conclusion

Dynamic rescheduling method is widely used in the modern production planning. One of crucial ways for solving the problems exist in modern manufacturing planning is a Multi-agent based scheduling system. Moreover, a novel multi-agent scheduling based on the Prometheus methodology by considering dynamic customer demand is proposed for flow line manufacturing system. The proposed system is aimed to overcome the short comings of the current flow line manufacturing system to give flexibility and re-configurability and rescheduling. This system follows top-down approach by considering real time negotiation between all types of agent which are labeled as; “Manger Agent”, “Shop Manager-Agent”, “Cell Agent”, “MHs agent”, “Scheduler Machine-Agent”, “MHs Resource Agent”, “Machine Resource Agent”. JACK agent platform for implementing proposed multi-agent based scheduling and control system is used. At the implementation phase of the proposed system for flow line manufacturing (uPVC door and window manufacturing company) we have taken advantage of the results of a Prometheus detailed design which is naturally map the (skeleton) code in JACK.

Consequently, simulation test platform for evaluating the potential benefits of proposed multi-agent system is created for an uPVC door and window manufacturing company. This platform is created based on hybrid-agent. The advantages of this simulation platform are considering both hardware and software together and its results will be near to real implementation. Some preliminary evaluations of the proposed multi-agent based dynamic scheduling are discussed. These evaluations are in terms of the dynamic customer demand and the number of excess products when a decrease in volume is requested during ongoing process. Also the results of simulation test platform were compared with real factory data in July of 2015.

For these comparison two scenarios was used in simulation platform. We observed an improvement of 6 % in the rate of acceptance for multi-agent based dynamic scheduling system (0.91) to the conventional system (0.861). This shows that, the new system exhibits superior performance in term of dynamic scheduling and improved the flexibility of production planning by accepting extra new orders. These are because the real time negotiation between agents can take immediate action to reschedule tasks (improve rescheduling). Second scenario (changing number of order) by considering illustrated graph, the time for changing number of order for customers throughout the conventional system is between 0 up to 3 working hours. Whereas, this change order time for proposed multi-agent dynamic scheduling system is comprehensively increased; in a way that the customer can change the order between 0 up to 9 working hours.

As a result of these scenarios, it shows that the proposed multi-agent dynamic rescheduling system enhances the efficiency and productivity of manufacturing system significantly. Also, the agility and reconfiguration of new system in terms of ad-hoc events are properly increased.

REFERENCES

[1] Barenji, R. V., Hashemipour, M., Barenji, A. V., & Guerra-Zubiaga, D. (2012). *Toward a framework for intra-enterprise competency modeling.* Paper presented at the Advances in Computational Tools for Engineering Applications (ACTEA), 2012 2nd International Conference on.

[2] Barenji, R. V., Hashemipour, M., & Guerra-Zubiaga, D. A. (2013). Toward a modeling framework for organizational competency *Technological Innovation for the Internet of Things* (pp. 142-151): Springer.

[3] Groover, M. P. (2007). *Automation, production systems, and computer integrated manufacturing*: Prentice Hall Press.

[4] Dilts, D. M., Boyd, N. P., & Whorms, H. (1991). The evolution of control architectures for automated manufacturing systems. *Journal of manufacturing systems, 10*(1), 79-93.

[5] Yoon, H. J., & Shen, W. (2008). A multiagent-based decision-making system for semiconductor wafer fabrication with hard temporal constraints. *Semiconductor Manufacturing, IEEE Transactions on, 21*(1), 83-91.

[6] Brennan, R. W., & Norrie, D. H. (2001). Agents, holons and function blocks: distributed intelligent control in manufacturing. *Journal of Applied Systems Studies, 2*(1), 1-19.

[7] Lu, S. H., & Kumar, P. (1991). Distributed scheduling based on due dates and buffer priorities. *Automatic Control, IEEE Transactions on*, *36*(12), 1406-1416.

[8] Vatankhah Barenji, A., Vatankhah Barenji, R., & Hashemipour, M. (2013). *Structural modeling of a RFID-enabled reconfigurable architecture for a flexible manufacturing system.* Paper presented at the Smart Objects, Systems and Technologies (SmartSysTech), Proceedings of 2013 European Conference on.

[9] Vatankhah Barenji, R., Hashemipour, M., & Guerra-Zubiaga, D. A. (2015). A framework for modelling enterprise competencies: from theory to practice in enterprise architecture. *International Journal of Computer Integrated Manufacturing, 28*(8), 791-810.

[10] Ramamritham, K., & Stankovic, J. (1994). Scheduling algorithms and operating systems support for real-time systems. *Proceedings of the IEEE, 82*(1), 55-67.

[11] Li, Y., Li, X., & Gupta, J. N. (2015). Solving the multi-objective flowline manufacturing cell scheduling problem by hybrid harmony search. *Expert Systems with Applications, 42*(3), 1409-1417.

[12] Zhao, F., Wang, J., Wang, J., & Jonrinaldi, J. (2012). A dynamic rescheduling model with multi-agent system and its solution method. *Strojniški vestnik Journal of Mechanical Engineering, 58*(2), 81-92.

[13] Rossi, A., & Boschi, E. (2009). A hybrid heuristic to solve the parallel machines

job-shop scheduling problem. *Advances in Engineering Software, 40*(2), 118 127.

[14] Rossi, A., & Boschi, E. (2009). A hybrid heuristic to solve the parallel machines job-shop scheduling problem. *Advances in Engineering Software, 40*(2), 118 127.

[15] Zhong, R. Y., Huang, G. Q., Lan, S., Dai, Q., Zhang, T., & Xu, C. (2015). A two-level advanced production planning and scheduling model for RFID enabled ubiquitous manufacturing. *Advanced Engineering Informatics*.

[16] Barenji, A. V., Barenji, R. V., & Hashemipour, M. (2016). Flexible testing platform for employment of RFID-enabled multi-agent system on flexible assembly line. *Advances in Engineering Software, 91*, 1-11.

[17] Barenji, A. V. (2013). *An RFID-based distributed control system for flexible manufacturing system.* Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ).

[18] Barenji, R. V., Barenji, A. V., & Hashemipour, M. (2014). A multi-agent RFID-enabled distributed control system for a flexible manufacturing shop. *The International Journal of Advanced Manufacturing Technology, 71*(9-12), 1773-1791.

[19] Barcnji, A. V., Barenji, R. V., & Sefidgari, B. L. (2013). *An RFID-enabled*

distributed control and monitoring system for a manufacturing system. Paper presented at the Innovative Computing Technology (INTECH), 2013 Third International Conference on.

[20] Thangarajah, J., Padgham, L., & Winikoff, M. (2005). *Prometheus design tool.* Paper presented at the Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems.

[21] Leitao, P. J. P. (2004). *An agile and adaptive holonic architecture for manufacturing control.* University of Porto.

[22] Johnstone, R., & Kurtzhaltz, J. E. (1984). Flexible manufacturing system: Google Patents.

[23] Haq, A. N., Karthikeyan, T., & Dinesh, M. (2003). Scheduling decisions in FMS using a heuristic approach. *The International Journal of Advanced Manufacturing Technology, 22*(5-6), 374-379.

[24] http://www.feng.unimas.my/departments/mechanical/37-facilities/1074-flexible-manufacturing-system-laboratory.html

[25] Rembold, U., Nnaji, B. O., & Storr, A. (1993). *Computer integrated manufacturing and engineering*: Addison-Wesley Longman Publishing Co., Inc.

[26] Dallery, Y., & Gershwin, S. B. (1992). Manufacturing flow line systems: a

review of models and analytical results. *Queueing systems, 12*(1-2), 3-94.

[27] Shen, W., & Norrie, D. H. (1999). Agent-based systems for intelligent manufacturing: a state-of-the-art survey. *Knowledge and information systems, 1*(2), 129-156.

[28] Jia, Z., & Ierapetritou, M. (2004). Efficient short-term scheduling of refinery operations based on a continuous time formulation. *Computers & chemical engineering, 28*(6), 1001-1019.

[29] Magalhães, M. V., & Shah, N. (2003). *Crude oil scheduling.* Paper presented at the Proceedings of the 4th Conference on Foundations of Computer-Aided Process Operations.

[30] Błażewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. (2007). *Handbook on scheduling: from theory to applications*: Springer Science & Business Media.

[31] El Khayat, G., Langevin, A., & Riopel, D. (2006). Integrated production and material handling scheduling using mathematical programming and constraint programming. *European Journal of Operational Research*, *175*(3), 1818-1832.

[32] Qiu, L., Hsu, W.-J., Huang, S.-Y., & Wang, H. (2002). Scheduling and routing algorithms for AGVs: a survey. *International Journal of Production Research, 40*(3), 745-760.

[33] Sabuncuoglu, I., & Hommertzheim, D. L. (1992). Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system. *The International Journal of Production Research, 30*(5), 1059-1079.

[34] Ulusoy, G., Sivrikaya-Şerifoğlu, F., & Bilge, Ü. (1997). A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. *Computers & Operations Research, 24*(4), 335-351.

[35] Isaksen, U., Bowen, J. P., & Nissanke, N. (1996). System and software safety in critical systems. *The University of Reading, Whiteknights, United Kingdom*.

[36] Wooldridge, M. (2009). *An introduction to multiagent systems*: John Wiley & Sons.

[37] Woolridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review, 10*(2), 115-152.

[38] Padgham, L., & Winikoff, M. (2005a). *Developing intelligent agent systems: A practical guide* (Vol. 13): John Wiley & Sons.

[39] Shen, W., & Ghenniwa, H. (2003). Editorial of the special issues on agent-based manufacturing process planning and scheduling. *Journal of Intelligent Manufacturing, 14*(5), 427-428.

[40] Durfee, E. H., & Lesser, V. R. (1990). *Negotiating task decomposition and allocation using partial global planning.* Paper presented at the Distributed artificial intelligence: vol. 2.

[41] Lu, T.-P., & Yih, Y. (2001). An agent-based production control framework for multiple-line collaborative manufacturing. *International Journal of Production Research*, *39*(10), 2155-2176.

[42] Maturana, F. P., & Norrie, D. H. (1996). Multi-agent mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing, 7*(4), 257-270.

[43] Sycara, K. P. (1998). Multiagent systems. *AI magazine, 19*(2), 79.

[44] Colombo, A. W., Schoop, R., & Neubert, R. (2006). An agent-based intelligent control platform for industrial holonic manufacturing systems. *Industrial Electronics, IEEE Transactions on*, *53*(1), 322-337.

[45] Erol, R., Sahin, C., Baykasoglu, A., & Kaplanoglu, V. (2012). A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. *Applied Soft Computing, 12*(6), 1720-1732.

[46] Sahin, C., Demirtas, M., Erol, R., Baykasoğlu, A., & Kaplanoğlu, V. (2015). A multi-agent based approach to dynamic scheduling with flexible processing capabilities. *Journal of Intelligent Manufacturing*, 1-19.

[47] Barata, J., Camarinha-Matos, L., & Cândido, G. (2008). A multiagent-based control system applied to an educational shop floor. *Robotics and Computer-Integrated Manufacturing, 24*(5), 597-605.

[48] Kumari, S., Singh, A., Mishra, N., & Garza-Reyes, J. A. (2015). A multi-agent architecture for outsourcing SMEs manufacturing supply chain. *Robotics and Computer-Integrated Manufacturing*.

[49] Wong, T., Leung, C., & Tang, H. (2008). *A multi-agent system framework for manufacturing planning and control.* Paper presented at the Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on.

[50] Kaplanoğlu, V. (2014). Multi-agent based approach for single machine scheduling with sequence-dependent setup times and machine maintenance. *Applied Soft Computing, 23*, 165-179.

[51] Chen, K.-Y., & Chen, C.-J. (2010). Applying multi-agent technique in multi-section flexible manufacturing system. *Expert Systems with Applications, 37*(11), 7310-7318.

[52] Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence, 22*(7), 979-991.

[53] Mes, M., Van Der Heijden, M., & Van Harten, A. (2007). Comparison of agent-

based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research, 181*(1), 59-75.

[54] Barenji, A. V., & Değirmenci, C. (2015). Robot Control System based on Web Application and RFID Technology. In *MATEC Web of Conferences* (Vol. 28, p. 04001). EDP Sciences.

[55] Chen, Y.-M., & Wang, S.-C. (2007). Framework of agent-based intelligence system with two-stage decision-making process for distributed dynamic scheduling. *Applied Soft Computing, 7*(1), 229-245.

[56] Gergely, G., Nádasdy, Z., Csibra, G., & Biro, S. (1995). Taking the intentional stance at 12 months of age. *Cognition, 56*(2), 165-193.

[57] DeLoach, S. A. (2001). Analysis and Design using MaSE and agentTool: DTIC Document.

[58] Padgham, L., & Winikoff, M. (2003). Prometheus: A methodology for developing intelligent agents *Agent-oriented software engineering III* (pp. 174-185): Springer.

[59] Jacobson, I. (1992). Object oriented software engineering: a use case driven approach.

[60] Kruchten, P. (2004). The Rational Unified Process. Object Technology Series:

Addison-Wesley, Reading.

[61] Huget, M.-P., Odell, J., & Haugen, Ø. Mariam “Misty” Nodine, Stephen Cranefield, Renato Levy, and Lin Padgham. Fipa modeling: Interaction diagrams. On www. auml. org under “Working Documents”, 2003. *FIPA Working Draft (version 2003-07-02).*

[62] http://www.proform-ybg.com

[63] Padgham, L., & Winikoff, M. (2005b). Prometheus: A practical agent-oriented methodology. *Agent-oriented methodologies*, 107-135.

[64] Kravari, K., & Bassiliades, N. (2015). A Survey of Agent Platforms. *Journal of Artificial Societies and Social Simulation, 18*(1), 11.

[65] Georgeff, M. P., & Lansky, A. L. (1986). Procedural knowledge. *Proceedings of the IEEE, 74*(10), 1383-1398.

[66] Huber, M. J. (1999). *JAM: A BDI-theoretic mobile agent architecture.* Paper presented at the Proceedings of the third annual conference on Autonomous Agents.

[67] Busetta, P., Rönnquist, R., Hodgson, A., & Lucas, A. (1999). Jack intelligent agents-components for intelligent agents in java. *AgentLink News Letter*, *2*(1), 2.

[68] Cheyer, A., & Martin, D. (2001). The open agent architecture. *Autonomous*

Agents and Multi-Agent Systems, 4(1), 143-148.

[69] Zhong, R. Y., Huang, G. Q., Lan, S., Dai, Q., Chen, X., & Zhang, T. (2015). A big data approach for logistics trajectory discovery from RFID-enabled production data. *International Journal of Production Economics, 165*, 260-272.

APPENDIX

Appendix A: Glossary

Agent: Agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives

Agility: Agility is a term applied to an organization that has created the processes, tools, and training to enable it to respond quickly to customer needs and market changes while still controlling costs and quality.

Buffer: In manufacturing, the concept of buffering is defined as maintaining enough supplies to keep operations running smoothly.

Dynamic Scheduling: Dynamic scheduling is a method in which the hardware determines which instructions to execute.

Entrant Flow: We say that when a job returns to a facility it reenters the flow at that facility, and consequently we call the shop a re-entrant flow shop.

Flexible Flow Line Manufacturing: FMS present various layouts: Loop, Ladder, Open-Field, centered robot and Flow-line. The concept of flow line configuration, which is the focus of this research study, the materials flow between the workstations, arranged in a line.

Flexible Manufacturing System: The Flexible Manufacturing System (FMS) is a central integration of a group of flexible machining with an automated handling system working under a centralized computer control.

Hybrid-Agent: Hybrid-agent is developed base on the mobile agent. It is flying between the existing agents and gathers information from the agents and transfer to the hardware simulation platform, the duty of hybrid-agent is like the ant-colony system.

JACK: JACK is an agent platform based on the beliefs, desire, and intentions model.

Manufacturing system: A production firm is an organization with major communication and commercial infrastructures which main business is concentrated on the production facilities or fabrication of products.

Multi-Agent: Multi-agent system, a branch in artificial intelligence which is consists of a collection of agents (two or even more) that situated to a particular use or purpose within a system provides a new solution for resolving dynamic and distributed problems.

NP-hard Problem: NP-hardness (non-deterministic polynomial-time hard), in computational complexity theory, is a class of problems that are, informally, "at least as hard as the hardest problems in NP".

Prometheus Design Tool: The Prometheus Design Tool (PDT) supports the structured design of intelligent agent systems. It supports the Prometheus methodology, but can also be used more generally.

Prometheus Methodology: Prometheus methodology is a general-purpose design for the development of software agent systems in which it is not tied to any specific model of agency in software platform.

Re-Configurability: Denotes the Reconfigurable Computing capability of a system, so that its behavior can be changed by reconfiguration.

Scheduling: Scheduling is the process of arranging, controlling and optimizing work and workloads in a production process or manufacturing process

Series of size: The sizes of products that are going to be equal with each other

Short-life cycle product: Product life cycle is the cycle through which every product goes through from introduction to withdrawal or eventual demise.

Static Scheduling: statically scheduled machine, in which the compiler determines the order of execution.

Top-down and bottom-up approach: Top-down and bottom-up are both strategies of information processing and knowledge ordering, used in a variety of fields including software, humanistic and scientific theories , and management and organization. In practice, they can be seen as a style of thinking and teaching.