UNIVERSITY OF
BIRMINGHAM

A UNIFIED SEMANTIC WEB-ENABLED FRAMEWORK FOR INTELLIGENT MANUFACTURING SUPPLY CHAIN

By

SALMAN SAEIDLOU

A thesis submitted to

The University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering

School of Engineering

University of Birmingham

December 2018

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

Abstract

The importance of well-constructed and efficient information synchronisation is critical to the successful tracking and monitoring of support processes in the context of manufacturing information flow. This project provides an innovative approach to exploring and addressing failure in traditional supply chain systems in the provision of dynamic and scalable information synchronisation. The latter has resulted due to the lack of protocols for the sharing of manufacturing data. This project aims to develop an accessible system whereby manufacturing supply chains can interact semantically in the exchange, interpretation and synchronisation of data. Contemporary semantic web effort in the supply chain field has been limited to the production and process knowledge areas; this project extends this by linking dynamic production data to create production execution knowledge. The novelty of the project is evidenced in the development of a synchronised data sharing system allowing better decision-making resources with intrinsic manufacturing intelligence. By identifying and addressing specific manufacturing challenges to mitigate disturbances within the supply chain, this project will have a significant impact on the proactivity of manufacturers and suppliers.

The overall aim of this research is to develop, test and demonstrate efficient creation and re-configuration of intelligent manufacturing supply-chain and manufacturing networks capable of rapid and dynamic response to changes and disturbances. The methodology used for this research mainly focuses on: (i) the development of an intelligent ontology-based data query system; (ii) the development of heuristics and optimisation of manufacturing systems through the use of agent-based technologies; (iii) the application of artificial intelligence techniques for simulating the manufacturing environment; and (iv) the use of intelligent software agents and linked data for predictive manufacturing.

A software platform is developed in order to simulate and form a highly adaptive supply chain based on a real industrial use case. The responsive characteristic of the system needs to be proactive, automated and self-regulated. This is realised through the development of a multi-agent system that would deploy autonomous agents, representing online physical systems, with capabilities of negotiation, cooperation, communication, learning and taking action. The results clearly illustrate the validity of the proposed data query, resource allocation and predictive models, when compared to the literature and a set of operation research benchmarks.

To my family for their support and patience

Acknowledgements

I would like to express my gratitude to all the people who have helped me and without their support it would not have been possible to conduct this research. I would like to include a special note of thanks to my PhD research supervisor Dr Mozafar Saadat who supported me throughout the completion of this study.

I would like to acknowledge with thanks GFM S.r.l. (Italy) for supporting this research by providing case study data and in particular Mr Luca Pasinetti for his kind support throughout this research. I thank my fellow researchers, Guiovanni Jules and Shahriar Parisouz for their help, motivation, support and for all the fun we have had while pursuing our research goals. I would like to thank Dr Marco Castellani for acting as the chairperson for my PhD viva interview and moreover, the valuable suggestions that enhanced this thesis from Dr Michael Packianather and Prof. Duc Truong Pham are greatly appreciated.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACATECH	German National Academy of Science and Engineering
ACL	Agent Communication Language
ADACOR	Adaptive Holonic Control Architecture
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AMCC	Avoid Maximum Current Completion time
API	Application Program Interface
ARIMA	Autoregressive Integrated Moving Average
ATeams	Asynchronous Teams
BBN	Bayesian Belief Network
BFS	Breadth-First Search
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CBM	Condition-Based Monitoring
CHC	Constrained Hill Climbing
CNP	Contract Net Protocol
CPS	Cyber Physical System
CRM	Customer Relationship Management
CSA	Chaotic Simulated Annealing
CTS	Convergent Tabu Search
DAML	DARPA Agent Markup Language

DARPA	Defence Advanced Research Projects Agency
DBHA	Decomposition-Based Holonic Approach
DES	Discrete Event Simulation
DF	Directory Facilitator
DR	Design Repository
ERP	Enterprise Resource Planning
FEA	Finite Element Analysis
FIPA	Foundation for Intelligent Physical Agents
FoF	Factories of Future
GA	Genetic Algorithm
GAC	Genetic Algorithm Control
GFM	Gruppo Fabricazione Meccanica
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HC	Hill Climbing
HMM	Hidden Markov Model
HMS	Holonic Manufacturing System
HTML	Hypertext Markup Language
IAS	Intelligent Application System
IDE	Integrated Development Environment
IIC	Industrial Internet Consortium
IIMMA	Improved Island Model Memetic Algorithm
IMS	Intelligence Maintenance System
IMS	Intelligent Manufacturing System
IoT	Internet of Things
IT	Information Technology
JADE	Java Agent Development Framework

JESS	Java Expert System Shell
JSSP	Job Shop Scheduling Problem
JVM	Java Virtual Machine
KBE	Knowledge Based Exchange
KNN	K-Nearest Neighbour
KPI	Key Performance Indicator
MAS	Multi Agent System
MASON	MAnufacturing's Semantics ONtology
MES	Manufacturing Execution System
MRO	Manufacturing Reference Ontology
MRP	Manufacturing Resource Planning
MSE	Manufacturing System Engineering
NIST	National Institute of Standards and Technology
NP	Non-deterministic Polynomial time
NT	Neighbouring Task search algorithm
OEM	Original Equipment Manufacturer
OIL	Ontology Inference Layer
OMDS	Ontology-based Manufacturing Design System
OPC	Open Platform Communications
OR	Operational Research
OSQR	Ontology-based Semantic Query Routing
OTD	On-Time Delivery
OWL	Ontology Web Language
PHM	Prognostic and Health Management
PKM	Product Knowledge Model
PLC	Product Life Cycle
PLM	Product Life cycle Management

PPP	Public Private Partnership
PROSA	Product Resource Order Staff Architecture
PSL	Process Specification Language
PSO	Particle Swarm Optimisation
QTY	Quantity
RA	Resource Agent
RCS	Resilience Control System
RDB	Relational Data Base
RDF	Resource Description Framework
RDFa	RDF attributes
RMA	Resource Management Agent
RUL	Remaining Useful Life
SA	Simulated Annealing
SCADA	Supervisory Control And Data Acquisition
SCM	Supply Chain Management
SMCP	Select Most Critical Pair
SME	Small and Medium Enterprise
SMSP	Select Maximum Sum Pair
SOA	Service Oriented Architecture
SOM	Self Organizing Map
SPT CNP	Shortest Process Time Contract Net Protocol
SQL	Microsoft Structured Query Language
SVM	Support Vector Machine
SWRL	Semantic Web Rule Language
TA	Task Agent
TMA	Task Management Agent
TS	Tabu Search

UI	User Interface
XML	eXtensible Markup Language

1 INTRODUCTION

1.1 Preamble

Due to the increasing complexity of the supply chain, there is a need for the synchronisation of the chain to extend product/process traceability and adaptability to disturbances upstream and downstream of the value chain. The platform emerging from the synchronisation of the value chain will significantly help to alleviate some of its major existing limitations. For instance, current manufacturing resource planning (MRP) lacks dynamism and scalability, and is constrained to a monolithic model of the value chain. It also follows a one-way communication paradigm. Another example is the reconfiguration of the manufacturing route and the rescheduling of the queuing orders with respect to the scheduled rectification of the resource back to full performance. The development of the corrective actions is very sensitive to the input data quality, relevance and completeness which originates from many sources: such as the supervisory control and data acquisition system (SCADA); the machine condition-based monitoring system (CBM); and production control and maintenance. In another scenario such as product development, a computer aided design (CAD) model encapsulates complex information about the components. However, the manufacturing processes only require a sub-set of the data involved in the CAD model and the identification of those specific data can improve the costs of product development and manufacture. For instance, the process of grinding does not require the entire CAD model. Likewise, computational stress analysis does not require the entire CAD model. Therefore, it is

important to efficiently make use of the right manufacturing data, at the right time and for the right application.

An intelligent manufacturing system (IMS) would capture data from manufacturing processes and scheduling activities within any supply chain, align the information and fulfil the changing information needs of the stakeholders. An IMS would be a modular system consisting of small event-based planning applications of legacy systems and data driven modelling algorithms to holistically optimize the supply chain. Legacy systems involve discrete event simulation (DES), SCADA, computer aided systems for design (CAD), process planning (CAPP), manufacturing (CAM) etc. A pre-requisite for building an IMS is a sophisticated expert system which stores machine-interpretable data in a navigable structure. Ontology engineering can deliver on this requirement to produce the unified knowledge base. If the identification of the relevant data can be automated and passed on to the appropriate application, this would benefit the effectiveness of product development and production control. If the data is linked, then it can be retrieved by substantially more attributes and innovative query combinations. Such manufacturing intelligence, enabled by a swarm of intelligent agents, would empower manufacturers to query their manufacturing sites and perform activities like reporting, modelling, experimental design, system alerts, recommendations, extrapolation, prediction, simulation and optimization.

Industry 4.0 has provided a new lingua franca for manufacturing facilities, which integrates all the modern technologies for enhanced processing and system optimization of manufacturing systems. In a nutshell, Industry 4.0 can be conceptualized in three dimensions: end-to-end engineering throughout the product life cycle (PLC); horizontal integration in the

supply chain; and vertical integration through networked manufacturing systems (Kolberg and Zühlke, 2015; Stock and Seliger, 2016). However, in order to enable full interoperability among equipment and application in the manufacturing systems, there is a requirement for standard protocols for interactions and data flow management. Ontologies provide the crucial set of standards which can enable such integration and allow for the data to be transformed into knowledge models; these can then be used for further processing. The new sensor networks that cover the manufacturing facility have the capacity to produce large sums of data, referred to as "Big Data" in real-time streams; if processed appropriately, it can make the system highly efficient and intelligent. However, in order to make sense of the acquired data and use it for further processing, standard ontologies are required. In this way, ontologies enable the use of big data, which forms the backbone of an Industry 4.0 environment.

The data acquired if appropriately processed can be a source of vital information which can lead to machine learning and optimized decision making for smart manufacturing. In order to make the big data useful, a comprehensive systematic approach for handling and analyzing it is required, in order to construct knowledge models which can be understood across applications and value creation modules. A cyber physical system (CPS) can play an instrumental role in providing the platforms for acquisition and management of big data, along with the basic infrastructure required to implement the required changes (NIST, 2013).

End-to-end engineering refers to potentiating the manufacturing and design process with digitization and integration throughout the product life cycle from design to recycling. Horizontal integration refers to the interconnectivity of the systems conducted within the

supply chain network of a product life cycle, through their digitization and ontology-based analytics applications. Vertical integration refers to the interconnectivity and integration of the manufacturing processes within the company, through complete digitization of each domain, from manufacturing stations to marketing, sales and research and development (Hellinger and Seeger, 2011; Kagermann et al, 2013; Stock and Seliger, 2016; Mrugalska and Wyrwicka, 2017).

Cloud embedded communication technologies allow for the intelligent cross linking between two connected entities and an end-to-end smart solution with interactive user interfaces is enabled. The system hence has the capability to be completely digitized and monitored from remote locations, along with fully integrated enterprise environment.

There exists a plethora of literature on ontologies and their usage in manufacturing systems; however, there is a lack of studies that focus on the use of different ontologies to enable the big data acquired to be lucrative, and to apply the acquired knowledge to transform the value creation modules and their holistic collaboration.

Interoperability in the manufacturing domain of multiple software applications has evolved in the recent era as the biggest impediment to manufacturing processes in information technology (IT) enabled manufacturing facilities. Experts and software developers have worked together to tackle this seemingly insurmountable problem, by devising translation programs to enable the flow of data from a defined application to another, but they lacked the flexibility of two-way transfers. However, with ever increasing deployment of applications

and the complexity of data to be shared, the task of developing translation programs is no longer feasible, given that lack of standards requires multiple translation programs for interoperability of a single application. The solution to the problem is a set of standards encapsulated by ontology, defined for the global environment in the manufacturing domain; which can enable a standard mechanism for the exchange of product model data. Experts have resorted to developing ontology-based globally accepted applications that can use product model data to inter-operate with other applications and provide a standard protocol for communication within a manufacturing and design domain.

The advancement in supply chain management and manufacturing systems requires the firm to align their processes' plans and information systems to enable integration and collaborations between applications and individuals working throughout the supply chain. The bottleneck arises when the standards followed by the collaborating partners and their subsequent applications do not follow the same set of rules and standards. Hence, the development of comprehensive collaborative environments in firms has become overly complex with the advancement in IT enabled applications for each process in the product life cycle; which arises mainly due to lack of protocols for interoperability and sharing of data. As defined previously, an ontology-based semantic web has the ability to bridge this gap for modern production and manufacturing facilities, which can also be integrated with the post-production phases of a product's life cycle (Cai et al, 2014). The literature on the subject shows a trend towards the provision of developing globally accepted ontology-based semantics' agreements, in order to align the terminologies for interoperable applications; which is described as being the backbone for an integrated manufacturing system.

In the transformed industrial systems as marked by the Industry 4.0 era, cyber physical systems, integrated with intelligent analytics, can define a framework of big data integration to capitalize the knowledge from multiple sensor installations and historical data logs in order to make predictive intelligent decisions. For such systems researchers have defined communications protocols such as MTConnect and Open Platform Communications (OPC), which can enable data acquisition of multiple sensors simultaneously. Big data utilization requires the streams of data to be processed and analyzed through predictive analytics in real-time. Companies such as GE, Microsoft and IBM are working seamlessly to utilize the potential of big data in manufacturing. Big data transforming applications and agents consist of several components: application program interfaces (APIs), ontology layers, predictive analytics platforms, visualization tools etc. The processing of huge chunks of data through analysis of data streams coming from multiple machines in a facility simultaneously requires tremendous computing power; it is not feasible to monitor and control this in-house. Hence, companies providing big data solutions for manufacturing rely upon the concept of cloud computing; where the Internet plays a crucial role and the data is processed through external data servers, used on the basis of the requirements.

The research described in this thesis concentrates on analyzing the main concepts of ontology in the manufacturing domain and extrapolates the importance of ontologies in enabling insights from big data. The role of ontology and big data in manufacturing is exploited through the analysis of their practical applications in a manufacturing environment and their potential benefits are distilled. The interplay between disintegrated concepts from the literature is analyzed and fragments of each modern manufacturing system are used as building blocks to conceptualize a state-of-the-art manufacturing enterprise in line with the Industry 4.0 concept. Hence, this study demonstrates three important applications of a unified

semantic web-enabled framework: intelligent manufacturing data query, decentralised scheduling and predictive manufacturing.

1.2 Aim and objectives of the research

The aim of this research is to develop, evaluate and demonstrate a unified framework of semantic web technologies that enables rapid and dynamic response to enterprise-wide changes and disturbances within the concept of predictive manufacturing. The research outcome will emerge from the successful completion of the following objectives:

- A rigorous literature survey within the concept of intelligent manufacturing
- Extraction and identification of key semantics within the shop floor and the supply chain and performing data analytics to cluster meaningful data
- Development of a semantic web of manufacturing data through ontology development as the foundation of a knowledge-based information source
- Building an intelligent system to query the unified ontology for creating the initial data conditions for modelling, scheduling and production prediction
- Design of algorithms for an intelligent decision support system and the application of artificial intelligence (AI) and a multi agent system (MAS) for simulating a manufacturing environment

1.3 Research questions

The main questions derived from this thesis primarily address the importance of the concept of intelligent manufacturing systems, which are pursued through the aforementioned objectives. These questions are as follows:

- What factors need to be considered in developing manufacturing ontologies?
- How can semantic web technologies (ontologies) be used for efficient data queries in manufacturing networks?
- How can autonomy, self-organisation and adaptability in manufacturing scheduling using agent-based systems be modelled?
- Can the scalability of a multi agent decentralised scheduling system be enhanced using ontologies?
- How can the prompt delivery of orders in manufacturing networks be predicted?
- How can the conformity of manufacturing orders within a real industrial case study be predicted?
- How can the prediction process be automated through the use of ontology and knowledge-based reasoning?

1.4 Methodology

The research study utilizes content analysis by extracting the appropriate work from the literature and then constructing thresholds to assess the efficacy of these activities. In that regard, multiple ontological standards in the light of peer reviewed journals are assessed and analyzed to reach the findings for the constructed hypothesis. The approach allows the researcher to extract raw data and converge it to a defined theme of interest; hence making sense of the data and its parameters. The application of big data is analyzed through proposed industrial applications for the state-of-the-art manufacturing facilities in the literature. The quantitative and qualitative data regarding the benefits of application are also precipitated to demonstrate the efficacy of big data and ontology in the manufacturing domain.

The simulation platform development, modelling and evaluation were conducted using the following concepts and tools:

- The case study data was gathered from GFM S.r.l. (Italy) group and also from operational research libraries.
- The knowledge captured in the designed ontology was created using Protégé ontology editor. The ontology is designed based on the holonic paradigm and extrapolated to model the manufacturing environment limited to the scope of data query, job allocation and scheduling. Being based on the object-oriented paradigm, ontology has classes and instances of classes called objects. The ontology allows us to graphically investigate the relationship forming between objects of classes (that are defined in our model) during simulation.

- The holonic paradigm is used in this research to provide a list of definitions and generic data structures that guide the modelling of manufacturing systems and facilitate the analysis phase of software development.
- A multi-agent system (MAS) paradigm was used to model and simulate job allocation and scheduling in manufacturing networks. The Java Agent Development Framework (JADE TiLab) was used to model a group of manufacturing service providers and a central company (as supply chain integrator). The software was implemented using a Java integrated development environment (IDE). The MAS requires development of software modules, ontology, interfaces with databases and adaptation of optimisation techniques.
- Three machine learning and data mining techniques were used and a prediction model was developed in R language for manufacturing purposes based on an industrial case study.

1.5 Thesis layout

Chapter 1 is an introduction to the research with an overview of the current manufacturing systems and the practices discussed to address the need for an intelligent data interoperability system within the manufacturing supply chain. The two main enablers of the factory of the future, namely, Industry 4.0 and the semantic web are also addressed. The research aim and objectives, research questions and methodology are also highlighted alongside the dissertation layout to give an insight into the research presented in this thesis.

Chapter 2 provides a comprehensive literature review on an intelligent manufacturing system and its enablers. It elaborates on the application of ontologies and the semantic web in the manufacturing domain and reviews the concept of big data and Industry 4.0 as the trending research within the manufacturing research community. It also provides an insight into the state-of-the-art of holonic manufacturing systems (HMS) and agent-based technologies, which are used as the main concepts of modelling and simulation in this research. An overview of the predictive manufacturing paradigm is also provided in the literature survey. The outcome from the research on these models and techniques shed light on the existing research gap and hence, delivers a framework for the understanding of the research attempted in this thesis.

Chapter 3 investigates the development of an intelligent data query framework through the use of semantic web technologies for manufacturing purposes. The primary objectives of the ontology-based data query were to develop an efficient and scalable data interoperability and retrieval system; in order to find the most relevant query results with minimum message cost, most hits per query and least response time. This chapter explains the idea of ontology and the application of the same in the manufacturing domain. A computer simulation software is developed and presented based on a real case study of distributed networks of manufacturing workshops. In this chapter, a semantic query algorithm is developed where query results are returned by investigating the semantic richness of each workshop. Results were compared with a semantic-free search mechanism based on key performance indicators.

Chapter 4 presents a decentralised scheduling system that enables intelligent agents to interact with each other in order to find an optimum solution. The structure and interaction

protocol of the agents are adapted from the heuristic techniques. A number of job shop scheduling case studies are presented to define simulation constraints. These problem instances are the basis for research evaluation and validation within the operation research community. The fundamental need for self-sequencing operation plans in autonomous agents has led this research to the development of algorithms. These allow the resolution of the combinatorial optimisation of a global schedule; which consists of the fixed process planned jobs and also requires operations offered by manufacturers. The research has achieved a novel hybrid optimisation algorithm to allow agents to schedule operations whilst cutting down on the duration of the computational analysis, as well as improving the performance extensibility amongst others.

Chapter 5 uses the findings in the previous sections and makes use of historical data from the industrial case study to deliver a prediction model, where the future instances of the system can be anticipated. Accordingly, the motivating scenarios or examples of the role of prediction in manufacturing are addressed through user-specific workshops with the project stakeholders. To be more precise, the questions that stakeholders like to ask of the prediction system are presented. Based on the results, raw data that are extracted from GFM are pre-processed and presented in this chapter followed by a predictive modelling algorithm, which is based on regression and data mining techniques.

Chapter 6 is dedicated to the summary of the thesis and the concluding remarks. It provides the main contributions and the impacts of this research with some recommendations for future research directions.

2 LITERATURE REVIEW

2.1 Introduction

Intelligent systems have been a prominent field of research for the manufacturing domain in recent years. The intelligent manufacturing system (IMS) is a system of manufacturing entities that are autonomous, distributed and intelligent. Its architecture enables the entities to cooperate and the manufacturing system to self-regulate when disturbed. The aim of this section is to elaborate on the advancement in this area and to review the emerging technologies and underlying intelligence in such systems. This chapter presents the literature that has helped with defining the intelligent data query problem, as well as the modelling of manufacturing scheduling problems and prediction framework.

2.2 Manufacturing ontology

Ontology is defined as a data model which provides information about a domain, with the aid of relationships established among various concepts under that particular domain (Gruber, 1993; Guarino et al, 2009). Organizational knowledge is encapsulated to build a set of knowledge bases that can combine details such as those of rules, policies, processes, definitions and relationships. Ontology has been applied in many different areas and domains, including manufacturing. Simple to complicated processes are easily explained by ontology in the manufacturing domain; this makes it simpler to create patterns, algorithms and frameworks for the automation of any process.

2.2.1 Ontology engineering

Ontology represents a specified, formal way to comprehensively share a complex concept, encapsulating a broad set of terminologies, for cross-domain integration and collaboration. Formal ontology incorporates complex semantics of relative concepts, their controlling rules, constraints, values, attributes and properties (Schalkoff, 2011). An ontology representation language is a set of categorically defined rules and in-depth semantics; one such example is the web ontology language (OWL).

A five-stage methodological approach is widely accepted as the design process of ontology, originally proposed by Uschold et al (1998):

1. Purpose and scope identification
2. Conceptualization and knowledge acquisition
3. Reconstruction and integration of multiple relating ontologies
4. Specifications
5. Documentation and evaluation

Ontology presents a formal tool for bridging the gap between interdisciplinary and interdisciplinary collaboration and knowledge sharing; which plays a significant role in multiple fields of application. According to Kulon et al (2006), a lack of ontology creates multiple bottlenecks for every repetition of an engineering task, emanating from impediments in cross-disciplinary data sharing; hence, considerably slowing down the repetitive processes. In such circumstances, the delays are multiplied with each design loop, badly affecting the

cyclic processes. Kim et al (2006) demonstrated that a lack of ontology results in designers using individualistic approaches and terminologies to document a product design, which makes it difficult for other designers to reconstruct and implement it. Ontology, as a categorical design of a conceptualization, plays an instrumental role in collaborative infrastructures through integrating applications, sharing cross-field information, enabling interoperability, and reconstructing knowledge (Cho et al, 2006).

2.2.2 Ontology in manufacturing domain

The process specification language (PSL) was proposed by Schlenoff et al (2000) from the National Institute of Standards and Technology (NIST), to define a taxonomy to capture and exchange discrete manufacturing process information. Process specification language is a neutral standard language for process specifications; such as production scheduling, manufacturing process planning, workflow, business process reengineering, simulation, process modelling and project management. The neutrality of PSL allows process specifications to be translated to and from software-proprietary file formats.

For the domain problem of keeping track of the cost in the manufacturing process, Lemaignan et al (2006) proposed an upper ontology in the manufacturing domain to be a common ancestor for more domain-specific ontologies. MASON (manufacturing's semantics ontology) addresses a similar problem domain to that from the NIST. The difference is that PSL is written in resource description language (RDF); while MASON uses the more up-to-date semantic language called ontology web language (OWL). Moreover, MASON shows a concrete application in expert systems for automatic cost estimation and in semantic-aware

multi-agent systems. The work in this thesis, on the other hand, looked at an intelligent and efficient data query system in a manufacturing network which is missing in the literature.

In an extensive and rigorous review of the literature, Sanfilippo and Borgo (2016) have focused on the semantics of feature notions. They have highlighted that the current feature-based CAx systems such as computer-aided design (CAD), computer-aided engineering (CAE), computer-aided manufacturing (CAM), and computer-aided process planning (CAPP) are missing a common understanding of what counts as a feature and the way features are conceptualised. They have emphasised the product knowledge conceptualisation problems within current product life cycle management (PLM) systems and the shortcomings of existing unifying approaches. The proposed ontology-based product knowledge representation would allow reliable data integration while preserving the intended semantics. This will give an insight into the importance of semantic web technologies for the current manufacturing systems. In this thesis, an attempt was made to demonstrate the advantages of using ontologies for data queries in a quantifiable manner.

In another study, Wei et al (2009) proposed an ontology-based approach to address manufacturing design problems. Their system is capable of integrating the whole design process within the product knowledge management to support locating, reusing, integrating and transferring the design knowledge. Their work can be improved and extended by developing more diverse domain-specific engineering ontologies to achieve a more extensive multidisciplinary integration. In contrast, the presented data query system in this thesis is not limited to one aspect of the manufacturing supply chain such as product design. The proposed

system is designed in a scalable manner to accommodate data integration to the ontology within different tiers of the manufacturing domain.

Lin and Harding (2007) have proposed a manufacturing system engineering (MSE) ontology which acts as a mediated meta-model for inter-enterprise collaboration. Their ontology is structured into classes, properties and instances. However, this method still involves time and cost overheads, as the individual partners must map their vocabularies to the MSE ontology at the outset; this can be a slow process with the current manual methods. Therefore, this research is limited due to this ineffective process, which may be a major obstacle for its large-scale use in information integration in a global supply chain. To ease this problem, formal mapping could become semi-automated via algorithms and heuristics, identifying the shared characteristics between the two ontologies.

Usman et al (2013) presented a practical manufacturing reference ontology (MRO) for product design and production domains. Their work is motivated by the lack of a core set of manufacturing concepts for unification of terminologies across various strands of manufacturing domains. The emphasis was on the formalisation of these core concepts for a semantic-aware system, where the MRO will support the development of domain-specific manufacturing ontologies. Their proposed ontology was tested by a qualitative method of inserting two types of facts in the system, namely: facts which are violating the formal classifications of the concepts; and those which conform to the formal classification. The experimentation part was mainly focused on the feature concepts, as with the work carried out by Sanfilippo and Borgo (2016). In this thesis, we looked at a semantic-aware query system in a manufacturing network, rather than a comprehensive ontology development,

which we believe is missing within the manufacturing research community. The contrast between the work of Usman et al (2013) and Lin and Harding (2007) and the work presented in this thesis is shown where this thesis made use of the ontology editor Protégé to create an ontology as a tool for data query purposes. While the scope of their work is mainly focused on the development of an efficient and practical core manufacturing ontology.

In a separate research study, Cai et al (2011) presented an ontology-based system called ManuHub which acts as a mediator for manufacturing service providers, by efficient and automatic retrieval of the required manufacturing services. Their work is based on service oriented architecture (SOA); and there are two limitations in their work. Firstly, there is the question of how to merge their ontology into existing upper-level and domain-specific ontologies in the real world; and secondly, how to test their idea on a dynamically changing manufacturing data set.

In order to develop a simulation platform for the formation process of networks of manufacturers, Jules et al (2013) and Jules et al (2015) have developed an ontology based on the product-resource-order-staff architecture (PROSA), to model the inter-enterprise communication during the formation process. The ontology also models the process of resource auctioning which is part of the formation process.

Saeidlou et al (2016) have proposed a hybrid search algorithm where a learning process is combined with data extraction from the ontology. This learning technique leads to a more intelligent data query system. The framework used set the foundation for the following

research, where an intelligent ontology-based data query mechanism is compared with a semantic-free mechanism as well as with the concept proposed by Himali et al (2012).

In order to enable the required interoperability in the manufacturing and design domain, Ahmed et al (2007) devised an ontology for knowledge sharing based on a set of protocols for retrieving design knowledge, indexing information and searching process and product specifications. In a study conducted by Cho et al (2006), semantic heterogeneity was realized to be the biggest obstacle to smooth integration of digital libraries of applications in an advanced manufacturing facility. In order to mechanize reliable data sharing protocols, Cho et al (2006) developed a meta-ontology for collective visualization of data in multiple-part libraries. The researchers described a domain-specific ontology with their atomic structures resonating with meta-ontology. Zhou and Dieng-Kuntz (2004) analyzed and discussed models based on knowledge-based exchange (KBE), aimed at adding and integrating data to the base geometric data to enable the modular exchange of product design and manufacturing. They demonstrated the flexibility of the KBE environment which can enable extraction and addition simultaneously. Application exchange for features and data was analyzed by Nawijn et al (2006); the researcher proposed multiple ontological approaches for this purpose as well. They modelled both common intermediate ontologies and participating ontologies. However, the rules for the mapping of information and exchange were manually defined; which posed certain limitations for large scale applications (Nawijn et al, 2006). Patil et al (2005) developed a product modelling language, which characterized every element of modelling in a standard format from the scratch of product design till its inception.

However, the main issue found in the literature on the subject of ontology in manufacturing is that a comprehensive ontological approach from the product design phase through the

manufacturing and life cycle has not been proposed; although, a few researchers attempted to create interoperable environments by manual exchange parameters for multiple applications (Gao et al, 2007; Wei et al, 2009). Some researchers however, successfully managed to integrate multiple phases of manufacturing ontology, but the major gap seems to be filled by the standard formulating organizations and regulatory authorities. Such selected comprehensive frameworks for ontology in manufacturing are discussed later in this thesis.

The ontology gives the appropriate meaning to stored data through constructing databases attached to standard terminologies; which can then be utilized to automate the process and create intelligent self-regulating systems. Such a semantics' web is used for the connectivity of machines working in a facility using the standard protocols. Ontology encompasses the classes: all the class hierarchy, properties and relationships between them. Moreover, an ontology has instances or individuals and also facilitates rule-based reasoning. On the other hand, in a knowledge graph or a semantic network, data/triples are only stored in the form of nodes and edges. This implies that ontology contains the structure and schemas, whereas a semantic network only captures the data. Therefore, it can be argued that an ontology would be metadata for a knowledge graph or semantic network.

The literature shows that the ontologies for modelling manufacturing activities can be divided among four groups and levels: top level ontologies, domain level ontologies, task level ontologies, and application level ontologies (Khilwani et al, 2009).

Top level ontologies represent the atomic building blocks for the applicable domain. Domain specific vocabulary is defined in the domain and task level ontologies. Domain specific concepts are defined in application level ontologies.

2.2.3 Ontology-based manufacturing design systems (OMDS)

As described above, ontologies provide a coded structure for semantics to be used as a standard to enable interoperability and knowledge sharing between the stakeholders and applications in a product design and manufacturing process. However, intensive knowledge sharing is required to optimize design specifications and rectify the manufacturing processes for better efficiency and enhanced quality. Such knowledge includes the design rules, specifications, rationale and constraint; and with the advancements in technology knowledge regarding previous experiments from multiple facilities throughout history, design and manufacturing processes can be benefactors in this process.

To enable high levels of information sharing and interoperability, researchers have proposed the ontology-based manufacturing design system (OMDS) (Wei et al, 2009), which utilizes a product knowledge model (PKM) and intelligent application systems (IAS) to enable cross platform and cross application interoperability, with the ability for history based assessments of the design and manufacturing process to be conducted.

According to the proposed framework of OMDS, a central design repository (DR) lies at the centre of the system which houses multiple domain and task specific repositories of semantics. The DR encapsulates design rules, industrial standards and domain ontology as encoded by the central authorities in collaboration with knowledge engineers. The DRs are

further structured and stored according to their domains in the PKM. The systems hence store all the information provided to it by the design engineers regarding the geometry of the components as well as design specifications. The intelligent system then allows the design engineers to conduct any changes by simply changing any specification input or defining a new geometrical dimension; the system updates all the components and application ontologies simultaneously and comprehensively for better visualization of the product design. The system then allows the quality engineers to export the design of the product to a finite element analysis (FEA) platform in order to conduct simulations to test the performance of the product as intended. The tested design can again be modified according to the results. In this way, the extensive costs of the trial-and-error approach can be mitigated and the design and implementation cycle can be repeated many times efficiently. Hence, the product model allows fast and effective design testing and simultaneous modifications such that any individual engineer can conduct the process.

There are five major components of an OMDS:

2.2.3.1 User interface

The OMDS provides the facility of data access from remote locations as well as within the facility through a web-based user interface (UI). The design engineers can access databases and files based on standard semantics through the UI. The interface is also a tool for visualization, where engineers can grasp the scope of the design through interactive and dynamic tools.

2.2.3.2 Design repository

The design repository, as explained previously, houses multiple data sets and databases, such as the ontology base, the rule base, the engineering analysis base. These bases are indexed and organized in accordance with the knowledge model. The case base is a combination of product, component and part cases. The rule base houses the configuration, rules, variant rules and restrictions on the design. The defined ontologies for the PKM are stored in the ontology base. Manufacturing resources are stored in the manufacturing information base.

2.2.3.3 Knowledge management

Knowledge management includes three main components, i.e. acquisition, presentation and examination. Knowledge management in a manufacturing ontology domain deals with the rules and ontology defined for interoperability. After acquisition of knowledge regarding a product design and manufacture through multiple sources of human induction and literature, the defined ontology is used to present the acquired knowledge in a standard manner. The OWL coding standards are used by the experts to deploy knowledge bases in semantic web rule language (SWRL). The creation and maintenance of ontologies is carried out through additional tools such as Protégé. Researchers have used Protégé to further create APIs based on Java (Zhao and Liu, 2008).

2.2.3.4 Intelligent application systems

For the retrieval, transfer and reintegration of product design knowledge, the OMDS uses intelligent application systems. Multiple domain databases, knowledge engines and application software are used as backhand applications for enabling IAS processes. The system utilizes smart reasoning of the design constraint rules as defined in the databases, to

administer any design changes conducted by the engineers and recommend alterations based on previously installed knowledge. The system interactively and dynamically supports the design process through recommending design modifications and allowing the engineers to consider industrial standards in the real time of application usage.

2.2.3.5 Workflow management

The system deploys information access tools to give a standard visualization to the end users and provide a base tool for exploiting the specifications. Sun et al (2007) proposed an effective way to enable integration of the design product and process knowledge, hence enabling effective workflow management.

2.3 Big data and ontology in manufacturing domain

With the intervention of smart machines and robotics into the manufacturing supply chain, large chunks of data are collected from multiple sensors, which enable the optimization of design and manufacturing processes. Recently, such huge chunks of data are summarized by the term “Big Data”, which represents seamless information regarding the system being received at every instant of the process. Given the extent of the information, researchers have delved deep on the capability of algorithms to spot trends in the data and make sense of the big data in order to further optimize the manufacturing process. Such data mining techniques can be used in various processes across the supply chain; from where extensive data is being collected at every instant from deployed sensors and machines, such as dirty data from machines for reliability centred maintenance, predictive manufacturing for maximal

optimization and reduction in production times. Ontology is required to make sense of big data in a collaborative environment such that the databases can be structured and accessed through a comprehensive platform and decisions can be made.

Error! Reference source not found. below demonstrates some of the major components of a big data enabled intelligent system along with their purpose. One such comprehensive and reliable solution is that of Predix offered by GE. The platform combines remote telemetry data being acquired in real time from the factory and tags it into relevant information through a set of ontologies aimed at making sense of data for the control engineers. The system provides the facility of utilization as well as automation, as it allows control engineers to set flags for different thresholds. The predictive analytics and big data handling model allows the system to become self-aware and learn from the historical events in order to enhance decision making. Such systems give way to self-aware and self-maintenance machines powered with intelligent decision making and machine learning.

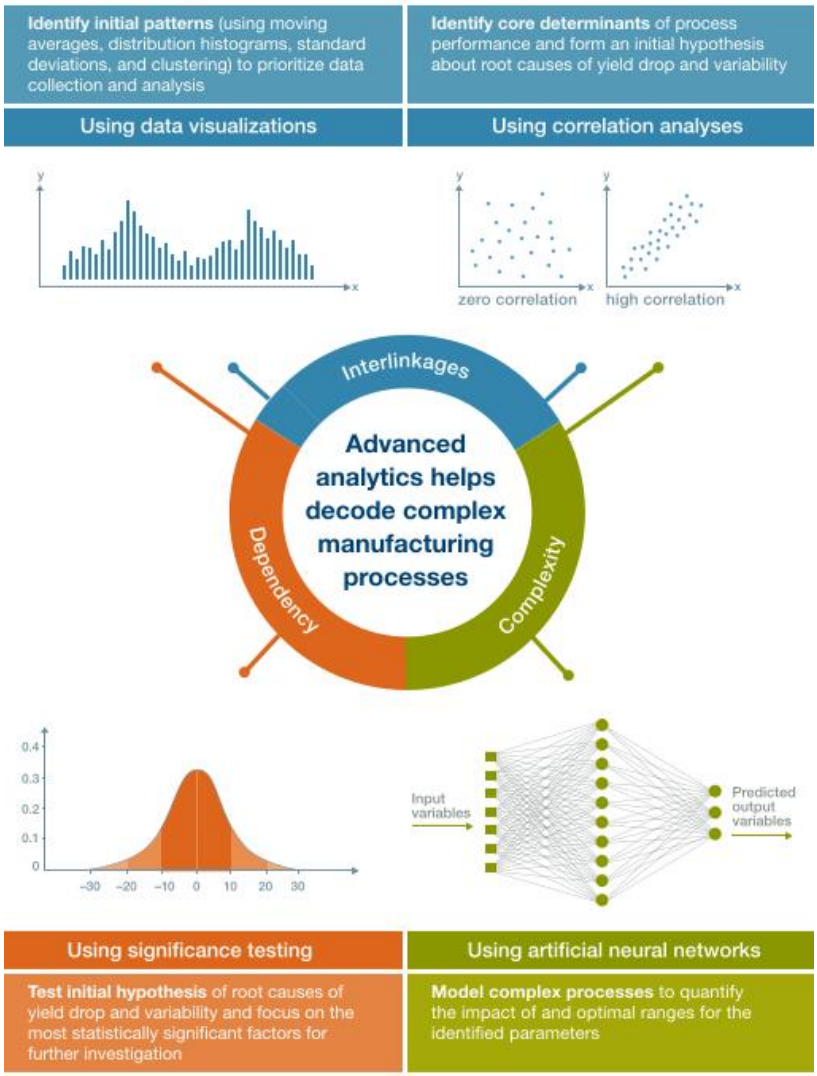

Figure 2.1 Process of using big data for intelligent decision making in manufacturing systems (Auschitzky et al, 2014)

Big data can also be a product of human intervention from distributed sources such as the conceptualization of the design of a similar product from multiple design engineers throughout the system's history; such data can be mined to produce and predict the best possible design keeping in view the data acquired from the testing phase of each design. Hence, with each new design being realized in the system, the distributed repositories get updated and the rectification process enhances, enabling smart design decisions.

Big data is also associated with the internet of things and in this case the internet of production machines, which can use the data transferred through applications to be self-aware and interact with each other to make collaborative intelligent decisions without human intervention. In this way, big data forms the backbone for artificially intelligent manufacturing systems. Such cyber-physical systems form the basis for the Industry 4.0 revolution, which utilizes the big data management tools to enhance manufacturing processes and automate it for optimal design and performance. This can also be transcribed into the domain of predictive manufacturing (Shen and Norrie, 1999).

The use and integration of big data into the design and manufacturing process can only be achieved through the help of ontologies, as multiple knowledge models are to be used for data acquisition; and standards are required for enabling interoperability between the machines to empower them for collaborative peak performance.

2.3.1 Industry 4.0: the role of ontology and big data

The concept of Industry 4.0 sent shock waves through the manufacturing industry with its recent inception in 2015 through a research article published by Kagermann (2015); which led to its official embrace by the German National Academy of Science and Engineering (acatech) through publishing its manifesto. However, European regulatory authorities and organizations identify the Industry 4.0 concepts in the public-private partnership (PPP) for the factories-of-future (FoF) domain (European Factories of the Future Research Association, 2013). On the other hand, the Industrial Internet Consortium (IIC) is the official promulgator of the Industry 4.0 concepts in the US (IIC, 2015).

2.4 Cyber physical systems for manufacturing

Cyber physical systems (CPS) are used in production, logistics and services in manufacturing facilities, which enable self-adaptive and self-aware machines; they can utilize real time comprehensive information through virtual interaction to adjust production patterns intelligently. A cyber physical system is a web of tightly knitted natural and human made systems, that interact with each other in real time through virtual networks and use computational power to make sense of the information exchanged, which is acted upon automatically using control systems. The interacting systems represent the physical space while the cyber space is represented by the computation, control and computational systems. The data acquired through a network of sensors and other data acquisition devices, along with intelligent control systems, produces constant streams of data in real time- this is referred to as big data (Lee et al, 2013). The data acquired, if appropriately processed, can be a source of vital information, which can lead to machine learning and optimized decision making for smart manufacturing. In order to make the big data useful a comprehensive systematic approach for handling and analyzing the data is required, in order to construct knowledge models which can be understood across applications and value creation modules. Cyber physical systems can play an instrumental role in providing the platforms for acquisition and management of big data, along with the basic infrastructure required to implement the required changes (Wang et al, 2015). Such intelligent features enabled by CPS can radically transform the industrial operations and truly enable the Industry 4.0 environment. Hence, a CPS enables the complete integration of big data through the use of ontologies with the manufacturing environment, enabling smart communication and automated control.

Architecture of CPS systems for manufacturing:

The implementation of CPS systems, starting from data acquisition to value creation through smart control CPS, requires infrastructure defined by 5-level architecture, known as 5C, which consists of categorical guidelines and methodologies for deploying a fully enabled CPS in a manufacturing domain. The figure below demonstrates the functions and attributes of 5C architecture for CPS in the manufacturing domain.

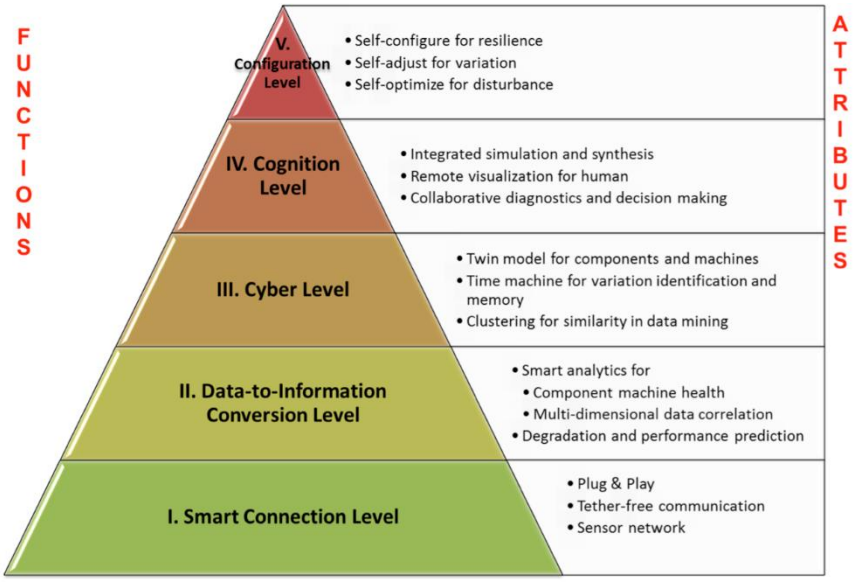

Figure 2.2: CPS architecture in manufacturing: 5C (Bagheri et al, 2015)

The 5-level architecture layers include:

2.4.1 Smart connection

To develop a cyber-physical application the most important step is to deploy data acquisition tools to accurately and reliably collect data, which can be used for decision making. Data can be collected through multiple platforms either directly or indirectly. A perception layer might be implanted through sensors and controllers to acquire raw data from the machines and environment, or enterprise manufacturing systems can be used for indirect data collection which includes manufacturing execution system (MES), supply chain management (SCM)

and enterprise resource planning (ERP). Hence, to acquire data from multiple machines and equipment, a reliable and robust data collection approach shall be employed, such that data can be acquired in constant streams without any glitches. For transferring the acquired data to controlling hubs, fully established protocols shall be used such as the MTConnect (Vijayaraghavan et al, 2008). An equally important aspect of the first level of architecture for 5-C is the use of reliable data collection instruments and sensors. The issue is that if the acquired data is not reliable or is harmed due to some glitches, the whole system can come to a halt and machines can take unfavourable decisions automatically based on empirical evidence.

2.4.2 Data-to-information conversion

In order to extract meaningful information and insights, techniques for making sense of the acquired raw data are required. For the information conversion level, several tools are available for data handling and processing. In this phase, the ontologies established for manufacturing enterprises as discussed in previous sections are used, terminologies are employed to convert raw data into useful information which can be used by machines to become self-adaptive and self-aware.

2.4.3 Cyber set-up

The central information hub is the cyber level in the architecture, which provides the web for intricate connections and interactions between the involved entities. A machine network is formed through the flow of information into cyber level from each connected machine or sensor. After acquiring massive machine data, the individual status of each machine is identified within the bounds of the system; such specific information is constructed through

the use of big data analytics' algorithms. Hence, such identification of individual machines among a fleet allows the machine to compare itself with others and judge the current state of its performance. This mechanism allows for the implementation of predictive analytics using big data where the historical data regarding similar machines is integrated in prediction models to anticipate the future performance of the system.

2.4.4 Cognition

At this level a thorough analysis of the complete system as a whole is established. This level allows the controlling authorities to make accurate decisions based on data driven insights. This level allows for all sorts of smart decisions from prioritizing operational tasks to enabling reliability centred maintenance. This level also utilizes data analytics to implement exploratory techniques for better visualization of the data for informed decision making by the users.

2.4.5 Configuration

This level constitutes the feedback mechanisms from the virtual to the physical. It enables machines to self-adapt and self-configure through executing supervisory control of the overall system. It also maintains the efficiency of the system by taking preventive decisions and taking corrective measures, hence, the name resilient control system (RCS).

2.5 Predictive manufacturing enabled by big data and ontology

The objective estimation of the manufacturing capacity and readiness of a manufacturing facility is instrumental in enhancing production efficiency. Such objectivity is possible

through complete transparency and monitoring of the value creation modules involved in the product lifecycle; through quantification of uncertainty using acquired knowledge from the field of operations. In a real factory environment, systems are vulnerable to multiple glitches and uncertainties which undermine the assumed optimal performance and timely equipment availability (Lee et al, 2013). Predictive manufacturing allows the level of transparency required in an organization to accurately predict the possible causes of failure in the future and inform intelligent systems to adjust in real time to deal with the anticipated uncertainties in real time. To enable precise predictions and estimations of the capabilities in real time, advanced prediction tools are required, along with reliable systems that can acquire big data and process it into information which can then be used to apply predictive analytics' algorithms to derive insights; leading to informed and smart decisions both automatically and through user interaction (Lapira, 2012). Smart machines enabled through networks of smart sensors connected in a virtual environment have been adopted in the new concept of the Internet of Things (IOT) based automation systems; this allows for data acquisition that can be used in predictive manufacturing. Predictive manufacturing systems, just like CPS allows the self-aware capabilities to the machines in a manufacturing environment and uses the big data to extract intelligent insights about the future of the system, such that artificial intelligence is enabled and the system's behaviour can be maximally optimized to meet current demand. The most essential part of a predictive system is the powerful computational systems deployed either in-house or through cloud computing infrastructure to deploy complex algorithms and process tremendous amounts of data for predictive modelling algorithms.

One important application in predictive manufacturing systems is prognostic and health management (PHM) (Lee et al, 2013). The data acquired through the working machines is processed through predictive modelling tools to predict the possible failure of any equipment involved and also to analyze the future performance of machines; which can then be enhanced through maintenance only when it is required. Such an environment allows for proactive solutions for possible uncertainties in the future which can greatly help in managing downtimes and prevent losses in performance of a manufacturing system (Bughin et al, 2010).

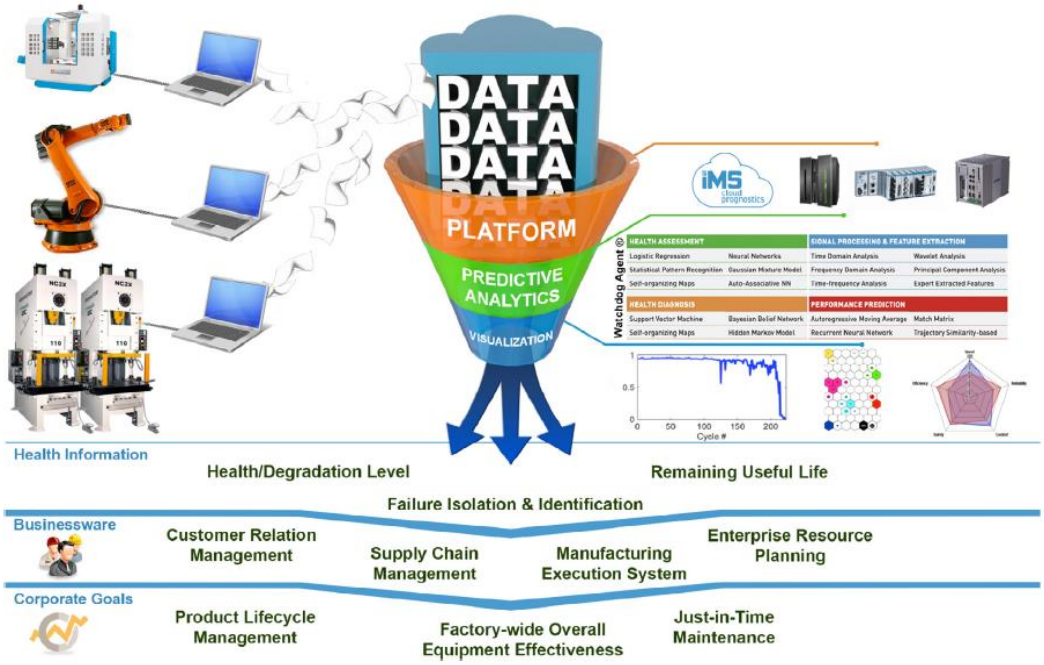

Figure 2.3 End-to-end conceptualized system framework for enabling predictive analytics for manufacturing; Watchdog Agent (Lee et al, 2013)

Figure 2.3 demonstrates a conceptual framework of a predictive manufacturing system. The system uses a smart web of sensors deployed at the facility to monitor the assets through streams of data being acquired and processed in real time. The raw data constitutes sensor signals based on quantities such as temperature, vibration, pressure, which is then tagged through a computational system where each asset is identified as a separate entity and data is

distributed into chunks of tagged information. Data mining algorithms exploit a combination of real time and historical data to make precise calculations about the current situation relatively. After the data is acquired, the connectivity is enabled through virtual nodes based on reliable communication protocols such as MTconnect and OPC, which enable a constant flow of data back and forth into the computational servers (Wang and Wang, 2016).

The aggregated data combined into a single hub is collected; which is referred to as "Big Data". The collected big data is then processed through multiple components of a transforming agent which includes an integrated platform for transforming the data through the use of ontologies into knowledge models. Then predictive analytics and visualization tools are applied to the knowledge models and useful insights are enabled for the users (Lee et al, 2013).

There are many different qualities of such deployment platforms for predictive analytics, given that there is no limit to optimization and prediction in the future. Hence, a greater amount of data mining and processing requires greater computational powers; which mainly depends upon the available costs of investments, desired ease of deployment and other individual references. Watchdog Agent is one such predictive analytics platform available on the market which is developed by the Centre for Intelligence Maintenance Systems (IMS). Another powerful system with cloud computing capabilities is deployed by GE which is known as "Predix", especially tailored multiple manufacturing factories. Such platforms do not only provide tools for predictive analytics but also provide platforms for information sharing interactive visualization and automation extension capabilities.

2.5.1 Watchdog Agent for predictive manufacturing

The Watchdog Agent provides an end-to-end solution for a predictive manufacturing capability which is powered by the Centre for Intelligent Maintenance Systems (IMS). The algorithm can be categorized into four major categories:

- Feature extraction and signal processing
- Health assessment
- Performance prediction
- Fault diagnosis

Error! Reference source not found. demonstrates the model used in the cyber-physical system integration with the algorithm deployed through Watchdog Agent.

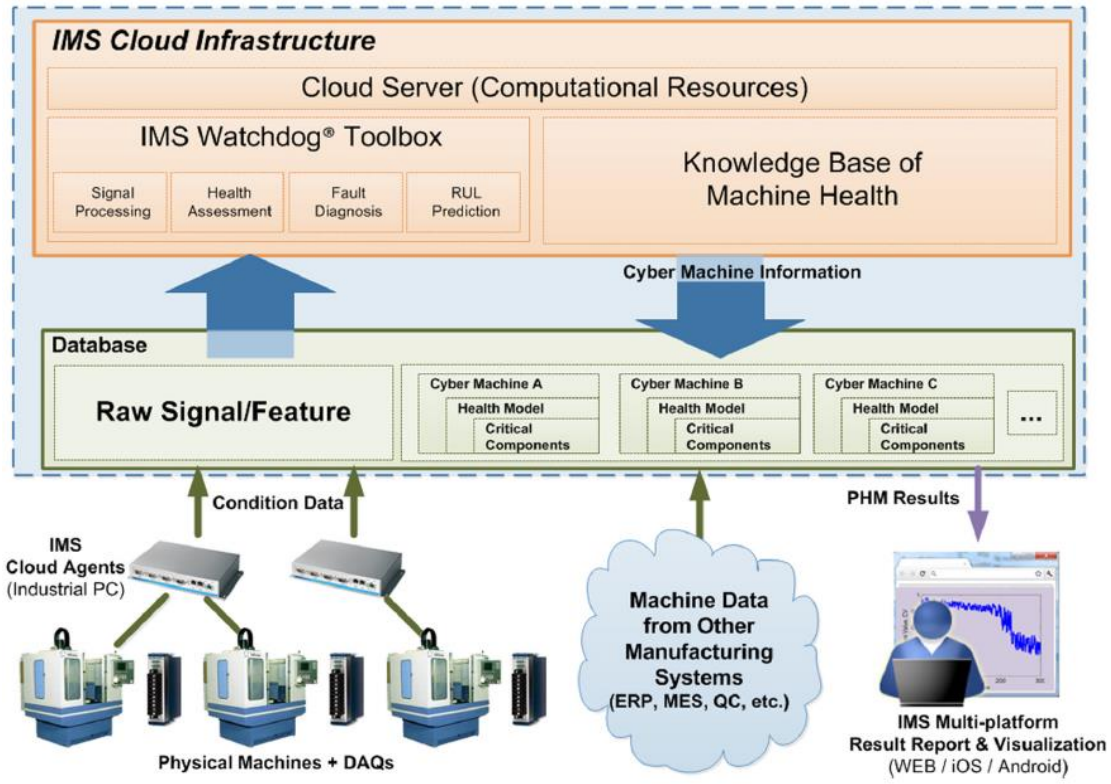

Figure 2.4 Predictive analytics and control enabled through a cyber-physical model. (Lee et al, 2013)

The intricate sets of data as acquired in raw format can be transformed into knowledge models which can be further transformed using visualization tools for easy to understand demonstrations of health information, such as current performance, requirements for maintenance, the expected life of an asset and failure predictions; which can be conveyed using risk charts, fault maps or degradation curves (Lapira, 2012).

Furthermore, the deployed algorithms can be used for optimal control and management of enterprise systems by allowing multiple systems to access insights tailored according to their unique needs for decision making, the systems that can be served include customer relationship management (CRM), ERP, SCM and MES. The enabled transparency and future prediction allows the management to make informed and accurate assessments for decision making and the facility wide assets can be managed and monitored through centralized platforms. Just-in-time maintenance is an essential feature for cost effectiveness in a manufacturing facility which is effectively enabled by Watchdog Agent. The system enables smart control through closed-loop lifecycle design, where historical information can be used in parallel with real time data streams by the algorithm to derive optimal solutions.

2.5.2 Adaptive clustering for self-aware machine analytics (prognostics using big data)

The data analysis capability at the cyber level of a cyber-physical system determines the extent to which the complexity of the operations can be executed. A powerful computational system, or a cloud connection with greater processing power and virtual machines, allows for complex data mining processes which can lead to enhanced system performance and enable fully self-aware machines. The data acquired from the perception layer in the architecture is grouped into different clusters of machines which is autonomously transformed into knowledge models; while in the meantime, machine learning algorithms allow the system to make intelligent insights and learn during the process. The knowledge model for the system

includes various kinds of information regarding the performance of the machines; such as machine health condition, work regimes, degradation patterns and failure modes. The acquired and segmented information is further utilized for optimal control enabling failure avoidance and optimization through cognitive abilities of the cyber layer. The traditional data processing algorithms are limited in their capability to use multiple factors in parallel and integrate them with historical information to handle complex situations; hence autonomous machine learning and deep data mining is made possible through a two-step knowledge extraction and machine learning methodology for performing cluster-based prognostics and health assessment.

Similarity among machines working to handle a certain situation always exists; machines operating under similar conditions or performing a similar task may have similar health conditions and performance curves. Hence, to reduce the complexity of the processing to be performed on big data it is feasible to form machine clusters at the cyber level which differentiate the data into chunks of similar information and then perform prognostics on the clusters.

Autonomous clusters are made using predefined and reliable algorithm models such as the Gaussian mixture model (GMM) and self organizing map (SOM) which enable unsupervised learning; such clusters can be based upon any sets of factors to be differentiated, such as machine conditions or a working environment. A state-of-the-art method deployed for clustering uses online updates on clustering algorithms; such algorithms utilize multidimensional distance measurement to compare existing clusters to the latest input by using one cluster as a sample for comparison.

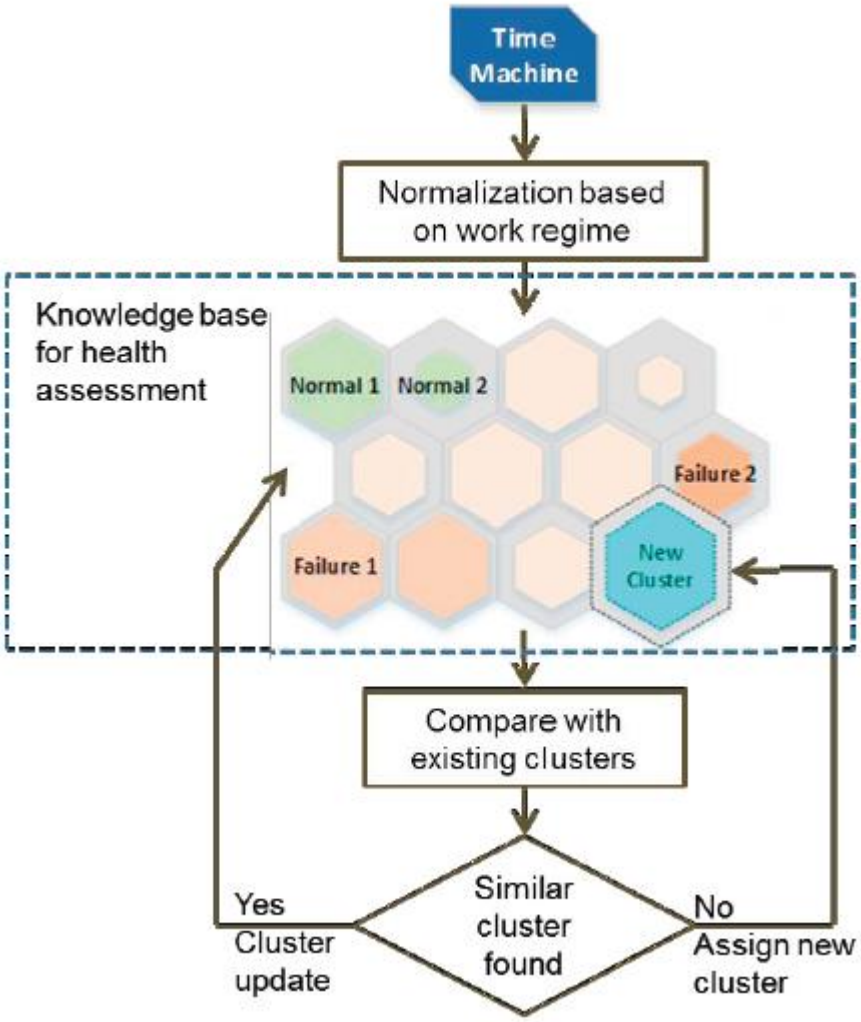

Figure 2.5 Multidimensional distance measurement algorithm for testing similarity in clusters using online model (Bagheri et al, 2015)

The sample-based search using multidimensional distance measurement can be executed to obtain two results (the mechanism is demonstrated in Figure 2.5):

2.5.2.1 Cluster found is similar:

In the case of identified similarity, the machine which is associated with the sample cluster will be identified as having a certain prognostic condition or the performance of a certain level, as predicted by the matching sample which is already tested to have a specific

condition with certainty. In parallel, the machine learning algorithm will allow the cluster identification algorithm to spot the difference between the latest sample and the established sample, such that the learned information can be used in the future models; hence, enabling complete cognition of the prognostics system.

2.5.2.2 Cluster found is not similar:

The algorithm in such a case will keep on comparing samples in the data set until it reaches its maximum count for out-of-cluster samples, then it can be established that no similar sample is available in the online model. The model hence comprehends that the tested sample predicts new behaviour in the machine which has not been identified in the past; consequently, the model will update the clusters to represent the new identified behaviour for future predictions. This allows the clustering algorithm to quickly adapt to new conditions and learn while performing the tests.

Given that all the samples are available online for the adoption system, the system's ability to predict and adapt new patterns becomes gigantic and it can identify behaviour which is not possible to be detected by any predefined tests in the theory. As the data to be monitored, and clustering data to be compared expands, the system has to adapt to big data and deep data learning techniques to become even more efficient. This is where big data plays an important role and allows the prognostic system to identify very minute changes in the behaviour, hence allowing better performance and predictive techniques.

Similarly, this testing technique for multiple clusters allows for prognostic health analysis; which further extends the procedure to measure the remaining useful life (RUL). This is enabled through the mechanization of stress and degradation history using previous and

current cluster data for the machines. A general purpose prediction algorithm is used which enables better performance and life prediction based on the relationship between stress and life. Complex learning algorithms with their enhanced reliability and deep analytics' power are being used for prognostics such as the hidden Markov model (HMM) and Bayesian belief network (BBN). Such an algorithm allows the usage of very complex big-data-based models to relate different degradation rates among the clusters based on multiple factors such as stress history, or in truly cognitive systems, through precision fluctuation techniques in the acquired data streams.

2.6 Horizontal integration in collaborative networks (distributed/ holonic manufacturing)

Collaborative manufacturing environments and collaborative manufacturing has huge potential for small and medium scale manufacturing facilities, where value is created through adaptability and control and producing customized products is a central premise, with a limited number of resources (Deen, 2013; Jules et al, 2015). Market opportunities can be fully materialized through collaborative horizontal integration which allows for balancing risks and optimizing processes for maximum resource utilization. The available capacities can be multiplied through self-adaptive systems and further investments are minimized in case of design alterations and rule modification. Hence, high agility can be enabled through collaborative manufacturing networks by adapting to volatile markets and individualized customer needs. The need for integration and coordination drastically increases when production sites and clusters of machines are to be dealt with individually. The system also requires vertical as well as horizontal data integration virtual networks, which can integrate knowledge models from all domains of an organization. Hence, there is a constant need for

the communication of employees across departments and their ability to make sense of the information being acquired to communicate effectively (Deen, 2013).

Another important facet of a collaborative and efficient manufacturing environment is the capability of outsourcing precise processes to other collaborators and conducting cross-organization communication effectively using standard ontologies with other networks. Such radical expansion can change the business model for manufacturing facilities as cross-organizational integration becomes a necessity for optimal operations. Such organizations with business models dependent upon outsourcing are referred to as virtual corporations (Jules et al, 2013; Jules et al, 2015). However, such corporations face many obstacles and one of the biggest is the absence of trust, as companies have to share confidential information across networks for complete collaboration and high levels of reliability are required.

Hence, along with data sharing networks, such horizontal and cross-organization collaboration requires data to be secure and protected from external intrusions into the system. The role of ontologies becomes crucial here as more enhanced knowledge models are required which can deliver only the precise information needed for collaboration and the rest of the knowledge models can be secured. A high level of agility is required for supply chain management and commodity flows have to be tracked along with data concerning customer satisfaction and logistics.

The virtual networks and the sensors deployed across the value chain produces massive amounts of data flow in real-time which have to be processed and segregated instantly for optimal operations, and sent after processing and characterization to the appropriate destinations. This data, also referred to as big data, requires intricate processing systems to be

converted into knowledge models based on set ontologies. Every organization has different ontologies as noted previously, hence the complexity of interoperability arises with horizontal collaboration and greater processing resources are required to handle the data (Babiceanu and Chen, 2006).

The Internet and cloud collectively provide the ability for the system to carry out better processing and cross-organizational collaboration; and also the horizontal collaboration in different hierarchies within an organization can be increased.

2.6.1 End-to-end digital integration (lean automation)

The optimization potential can be radically increased if the complete end-to-end integration of the whole value chain is rendered through a centrally controlled virtual network; such that the acquired information can be first changed into knowledge models and then using data analytics and visualization techniques can be shared across departments for understanding (Kolberg and Zühlke, 2015). The control and real-time information is distributed from every entity working in the supply chain to the shop-floor level and the whole organization can adapt to the need for achieving a single goal in holistic collaboration as well as collaboration in small clusters. Hence, a central premise of the Industry 4.0 concept is the integration of business process along with services and workflow which can be fully interoperable through CPS. Such end-to-end collaborations are radically adopted by the automotive industry where OEMs (original equipment manufacturer) collaborate supply networks across factories and shop floors, to enable just-in-sequence supply chains and anticipate the availability of key products to be used for better supply chain management. Such environments have a high degree of complexity given that most automobile companies have to bring together and attach 20,000 components from many different facilities (Kolberg and Zühlke, 2015).

In an item-level production as opposed to mass customization knowledge models being constructed in real-time are constantly utilized to improve historical databases; which allows for improvement and further development in the processes. Similarly, simulations are to be shared and integrated across companies for better process estimations; hence, this adds to the data which is to be constantly shared and used at both ends. To ensure complete interoperability, uniform standards for data acquisitions and transfer are required, due to which standard ontologies must be adopted in partner companies. The need for standard ontologies is increased in partner organizations where the data concerns multiple products which are to be developed simultaneously and optimization is required in the supply chains and manufacturing capabilities (Kolberg and Zühlke, 2015). Big data analytics play an instrumental role for enabling such collaborations, as exploratory tools allow users to visualize the information in an interactive manner and instantly understand the situations.

Such big data and ontology driven collaboration between companies allows better productivity and higher yields leading to competitive advantages. High demand volatility can be dealt with using long-term service contracts, as the base product can be further enhanced by partner companies in case of required customization. Entirely new remote maintenance capability can be enabled through smart sensors and control systems enabled by CPS. The streams of data acquired constantly can be used by the remote managers to understand the current status and also implement changes in the system in real time; this enables the overall system to cut delays and use raw material very efficiently, leading to automated lean manufacturing.

2.6.2 Holonic manufacturing using PROSA and CNP

PROSA (product resource order staff architecture) was introduced with an aim to present a theoretical model for intelligent modelling systems (Van Brussel et al, 1998). An intelligent manufacturing system encapsulates multiple autonomous and cooperating holons working in integration with each other through exchange of information based on standard protocols and adjusting them to achieve a higher goal of the overall system. A holon can be described as equipment in the value creation modules of a supply chain that has an inner structure based on components controlled from its own control unit, but at the same time it coordinates with the outer system to work as a constituent part of the complete system. However, a holon can be a structure housing other holons as well. A holon can be considered as an extension of a SCADA system with improved processing and communication abilities; hence allowing itself to be self-aware and self-adjusting to the external environment. The figure below shows architecture of a single holon which houses its individual information and physical processing parts.

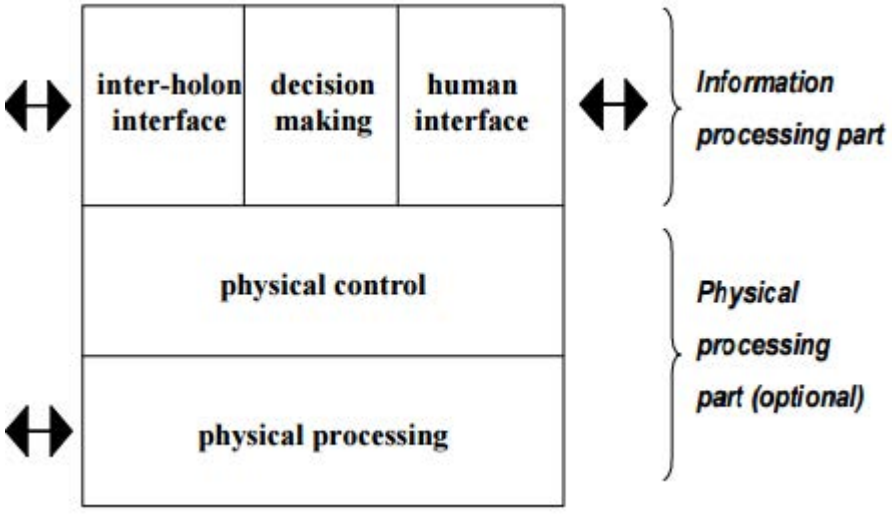

Figure 2.6 Holon architecture (McFarlane and Bussmann, 2000)

PROSA constitutes a framework or the initiation of holons and models them in an extensive system. It describes three basic holons in a manufacturing system: resource, product and order holons. A staff holon is an ad hoc holon which can be integrated with the system if manual inputs are required. Holons hence work as building blocks in a complete manufacturing system interacting with each other and defining the interactions within themselves to adjust to the overarching goal of the enterprise. As described in the other applications within this chapter, a holonic system gives modularity flexibility, robustness, scalability and agility along with decentralization ability to the manufacturing system, which has many useful prospects. In order to fulfil an order a product holon provides the basic set of knowledge models acting as a data server for the whole system (Van Leeuwen and Norrie, 1997). The information processing and the physical parts of the system are represented by resource holons. A resource control algorithm provides the blueprint for the processing of the system. The planning software, labour, machines, the shop floor etc. are the constituent parts of the resource holon. The tasks for the other holons are provided through the order holons and the control of the task flow information is also handled by them. Holons characteristically house the knowledge about themselves and other holons they interact with in the system, giving them a myopic view of their internal and external environment. Staff holons are responsible for providing other holons with the perspective of the external environment using self-adjustment and self-adaption (McFarlane and Bussmann, 2000).

PROSA also encapsulates the knowledge framework to define the interactions between the holons: product knowledge, process knowledge, and execution knowledge. The performance of a task is assessed through the process knowledge between a resource and the product holon (Van Brussel et al, 1998). The interaction between a product and order holon is monitored through the production knowledge, which constitutes ways in which the resources shall be

used to form the product. The physical progress of allocated tasks is controlled through the process execution knowledge which monitors the interactions between the order and resource holon. The *Figure 2.7* illustrates such interactions introduced by Van Brussel et al (1998).

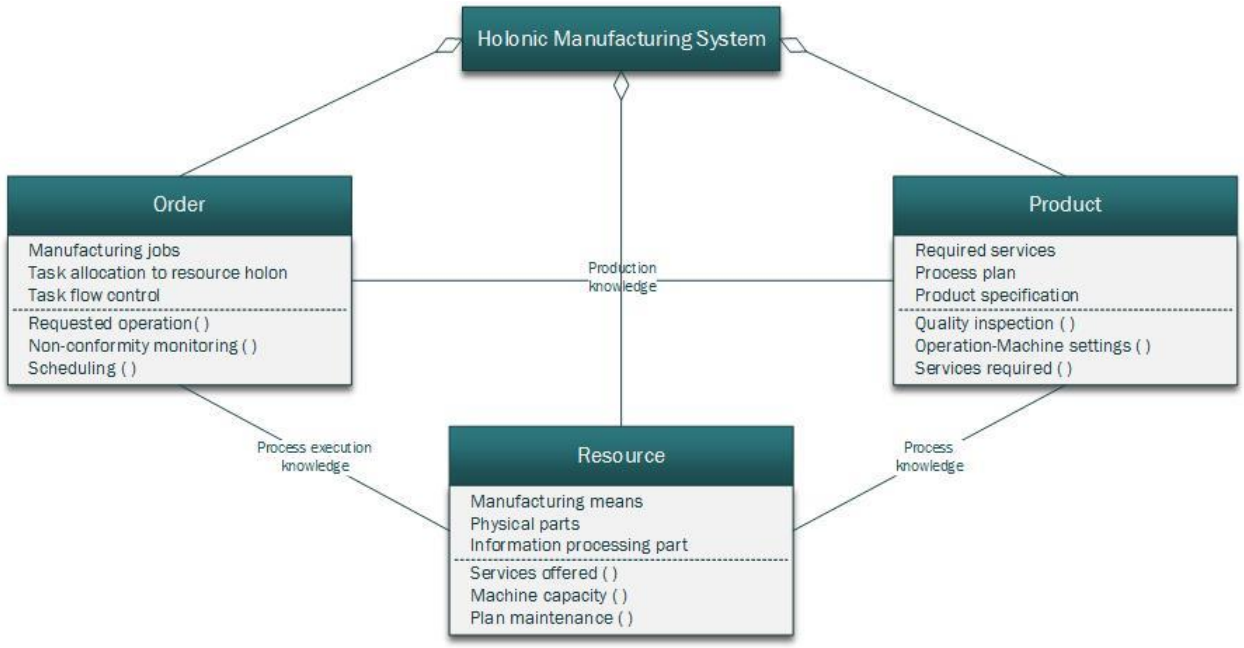

Figure 2.7 PROSA entities and their interactions

CNP (Contract-Net Protocol) is a protocol established to mechanize the interoperability through the provision of a system of interaction and negotiations in holonic manufacturing systems (FIPA, 2002). The Foundation for Intelligent Physical Agents (FIPA) is the parent body responsible for the standardization of CNP. The standard is based on the auction based interaction models: reverse auction, English auction and Dutch auction. A single order is identified as its constituent parts, each described through a set of manageable jobs to be completed in the manufacturing system through different agents. The jobs are either: parts, products or modules; such that each job can be completed by one or more manufacturers inside or outside the enterprise. The capabilities and competencies of the manufacturers are matched with the defined sets of jobs for each module; this stage is described as network

level production planning. The call for the proposal is generated by the central company which includes scheduling data, drawings, process plans and specifications for each described job. Then the bidding is opened for the manufacturers and they compete with each other to get the orders for production on the basis of their efficacy and efficiency for the jobs.

2.7 Ontology and data interoperability standards in manufacturing

To realize a “product centric” or “product driven” paradigm the integrated knowledge models become more complex, as information regarding the processes and products has to be distributed both vertically and horizontally across the value creation modules and supply chain. Hence, globally shared approaches for information operability and creation of knowledge models are required. In this regard, efforts have been made by major regulatory authorities to comprehend a system for knowledge models that can be globally accepted and shared on the basis of global standardization; which can be interoperable across organizations for collaborative partnerships in case of outsourcing of processes. Two such reliable and global standards include the ISO 10303 technical specifications (ISO/TS 10303, 2004) and the IEC 62264 set of standards (IEC 62264, 2002). These holistic models enable interoperability for end-to-end integration of an organization through heterogeneous management of information being acquired from different subsystems, through intricately aligning the knowledge in the direction of technical data for the end product. These methods have the capability to control both manufacturing and business levels of product data for an enterprise. A summary of the ontologies presented in the literature is given in the Table 2.1. The table also highlights the domain which shows the category of value creation modules for which the subsequent ontologies are built and also their purpose or the type of interoperability they offer.

Table 2.1 Literature ontologies and standards considered for the purpose of this study

Name	Year	Developer	Domain	Purpose
MASON	2006	Lemaignan et al, 2006	Manufacturing upper level ontology	Application in automatic cost estimation and semantic-aware multi agent systems
PRONTO	2011	Vegetti et al, 2011	Ontology for consistent and comprehensive representation of product information	An ontology for the for the product modelling domain, able to efficiently handle product variants
MTConnect	2008	Association for Manufacturing Technology (AMT), University of California, Berkeley (UCB), Georgia Institute of Technology (GT)	Generic data exchange standard	Allows the share of data and connectivity of manufacturing shop floor equipment
PABADIS	2002	PABADIS PROMISE (IMS Project)	Extended enterprise integration	Optimize production agility and effectiveness of cross company collaboration
STEP	1994	ISO technical committee	Product data	Computer interpretable representation and exchange of product data
PSL	1999	NIST	Neutral language for defining processes	Automated reasoning supportive neutral representation of manufacturing processes
MSE	2004	Lin et al, 2004	Extended enterprise integration	A taxonomy of concepts and terms of manufacturing to extend the reach of design knowledge across the enterprise

2.8 Distributed manufacturing scheduling

There are two key approaches to solving the job shop scheduling problem. One is the centralised approach, wherein a centralised computing unit performs the scheduling. A decentralised approach, such as the multi-agent approach, can also be used. Various heuristics, such as branch and bound, bottleneck-based heuristics and the disjunctive graph have been used to solve the job shop scheduling problem (Gabel, 2009). Branch and bound involves the use of a dynamic tree structure, which signifies the resolution space of different achievable arrangements (Gabel, 2009). Using this method, candidate solutions are frequently calculated and checked against upper and lower limits and are discarded if outside the range in order to ensure that the best suboptimal solution is achieved. Another common method is the disjunctive method, first proposed by Roy and Sussmann in 1964 (Gabel, 2009), and which can be used to formulate job shop scheduling problems as well as to represent schedules. It is a useful model that can describe instances of the job shop scheduling problem. Disjunctive graphs model a system of tasks to be scheduled and their timing constraints. This method is efficient when the objective function is regular, as there will always be an optimal solution for the problem represented.

2.8.1 Multi-agent approach to job shop scheduling

As the centralised approach requires a powerful central computing facility to handle the huge amount of data, it cannot react to machine failures (Wu, 2005). Moreover, it uses simplified theoretical models to allocate machines/resources to the jobs (Gabel, 2009). However, the multi-agent system can react to data adaptively. This is helpful to improve the stability of the system and increases its robustness and scalability. In this context, robustness suggests that the performance of the schedule can overcome disruptions, while adaptable systems are

capable of reacting to numerous unpredicted disruptions. This is important because decisions can be based on a shorter planning horizon and on limited problem knowledge (Yoo and Müller, 2002).

In contrast to centralised schedulers, an agent-based manufacturing scheduling system supports distributed scheduling, wherein each agent is assigned to one machine. An agent could be one production plant; for example, each steel production process could be represented by one agent which would be responsible for scheduling (Cowling et al, 2003). It has long been known that distributed solution approaches can be utilised in finding optimal or near optimal solutions for manufacturing industry problems because of the nature of modern day manufacturing (Wu, 2005). Modern day manufacturing requires many plants, each of which would be responsible for a certain task; and distributed scheduling enables communication between these plants in order for an optimal solution to be obtained. This communication is obtained through communication between the different agents.

In general, multi-agent systems are a useful alternative to the restrictions imposed by centralised scheduling schemes, as they are more autonomous and can react to real-time situations (Wang and Liu, 2006). Moreover, different scheduling models and methods can be used to solve the original optimisation problem; whereas their overall architecture helps with flexibility, scalability and fault tolerance (Wang and Liu, 2006).

2.8.2 Agent-based modelling in manufacturing

The well-known multi-agent system ADACOR (adaptive holonic control architecture) built on the Java development framework (JADE) reconfiguration infrastructure follows the principle of reaction upon the effects of the previous action taken place in the system. This stigmergy is set by Barbosa et al (2015) to identify the query stream in the environment in order to signal a deviation of plan and provide a chance for self-reassessment. In the research presented in this thesis however, the use of message track is not desirable since a knowledge base is a rather more scalable approach.

The development of an agent-based simulator within JADE, with a knowledge base framework capable of reasoning, was presented by Komma et al (2011) for the manufacturing shop floor domain modelling agents, namely AGV-agent, machine agent and part agent. The part-agents' dispatch algorithm in this work was produced on a first come first serve basis and lacked focus on tasks with sequence set-up limitations.

While the proposed research in this thesis focuses on operation sequencing and timing within a manufacturing network, Vrba and Marik (2010) developed a multi-agent system to comply with changes of the factory floor layout. The system reassessed the virtual map and searched for the nearest product destination using a disturbance simulated as a failed conveyor within the system of conveyors in the factory.

2.8.3 Distributed scheduling

Agent-based manufacturing scheduling has dealt with numerous issues, which were the centre of focus for the research community. Independent projects, which have mutual sets of resources, are limited in operation prioritisation and resource availability and face conflicts regarding the shared resources. Adhau et al (2012) introduced a multi-agent architecture system, which allows project agents to bid with resource agents for time spans using negotiation algorithms called virtual schedule and utility calculations, bid generation and modification through five steps of auctioning. The assessment of the final resource allocation and winner identification is carried out by the exchange agent with the aim of minimising delay, measured in average project delay and total makespan, compared to a 140 multi-project instances' set of available data.

The existing scepticism in the context of disturbance will negatively affect the solutions, leaving them valueless and impracticable. Harjunkoski et al (2014) introduced a method to allow frequent rescheduling by using deterministic scheduling in a closed-loop schedule. This will avoid unreasonable time spent on random aimless scheduling; since the required time spent on deterministic scheduling is more efficient, allowing a longer scheduling horizon at a time of uncertainty, resulting in quicker reactions to the alterations.

Researchers have also been concerned about distributed scheduling, such as scheduling where a factory layout is set with a dynamic nature, consisting of mobile robots. This dynamic nature is classified within flexible manufacturing, where the effective capacity alternates. This issue has been dealt with by a two-level scheduling algorithm introduced by Giordani et

al (2013) consisting of iterative auctioning in the first place and the Hungarian method in the final process. This allows the tasks to match up to a robot and following that, robots will randomly approach different tasks. Unlike the centralised approach, the downside of this method concerns the control policy being substandard and minimal and resources are misspent. When the resources are insufficient, a favourable partner must be found in a more cost-effective manner, as compared to the conventional tendering process. This has been achieved by a method consisting of a list of buyers and suppliers along with a practicability analyser to cooperate with a negotiation protocol. This produces a decision matrix considering the feasibility with respect to the material flow resulting from the network power and the precedence of the required qualities. The proposed method introduced by Mohebbi and Shafaei (2012) has surpassed the conventional tendering process considering the cost assessment of the backlog order, misspent capacity and total cost.

With the increasing rate of dependencies, the resulting exponential increase in the complication of each factory job allocation and unexpected incidents such as amendments of orders and lack of machinery systems, the necessity of an iterative cost adjustment method is irresistible. Alvarez (2007) has resolved this in the form of an auctioning process followed by contract net protocol. This provides the foundation of negotiating time-critical constraints in order to be utilised for bid analyses models produced by each agent; which will create a new foresight with large-scale decision making.

Finally, in order to save the nature-like quality of the agent technology, concerning the state of the art, a potential field control architecture is set to act as a vector of resource services. An indirect relation between the field intensity and the distance of resources away from the

specified job will be analysed, to be applied to a matrix of potential fields to form a decisional node. This node will be used as the reference point to identify the resource with the highest potential field for the job. Leitão et al (2012) have successfully surpassed the contract net protocol by achieving an average production time gain of 10%. Hopefully this will help to maintain a distributed nature for the agent technology alongside the applied improvements, allowing self-configuration and self-optimisation for multi- agent systems.

2.8.4 Distributed agents' interaction protocols

Each agent has partial information that can be used to solve the global optimisation problem; and through the use of a communication protocol, these agents can collaborate in order to obtain a global optimisation solution. Most agent-based manufacturing scheduling systems use negotiation protocols for resource allocation (for example, a contract net protocol or an auction protocol) (Wu, 2005). An agent communication language (ACL) is a language that forms the basis for communication between the different software agents at each manufacturing plant. The agents are connected through a local network (for example, the Ethernet). In separate research, a hybrid network for agents was presented with four types of agents (Wang and Liu, 2006). These four agents were: the resource agent (RA), which represented a resource such as a device, a tool and an automated guided vehicle (AGV); the task agent (TA), which represented a part or a part group; and the task management agent (TMA) and resource management agent (RMA), which were responsible for managing and monitoring the TAs and RAs in the system respectively. The task agent and the resource agent were equipped with a certain degree of autonomy to reason and to decide for themselves. Additionally, they were equipped with the knowledge to independently solve a part of the global problem. Such a framework is both flexible and adaptive, as agents may be

replaced or added freely. The system configuration can be continuously changed to accommodate changing requirements.

The negotiation mechanism is a direct approach to assign scheduling activities, which falls in either market-based or threshold-based methods. The latter depends on whether the agent accepts the proposed task in the event of a variety of events taking place, and it is fed with knowledge in the form of pheromones. The market-based method is specified for the agents with self-seeking objectives, who compete to be rewarded for favourable system-wide actions and to provide the required knowledge. The agents function with respect to a certain range of implicit and explicit data and rules. The advantage of the threshold-based method in comparison to the market-based approach is that it avoids the effects of a continuous change of bids, which leaves the indirect threshold approach free from communication scalability complexities (Shen, 2002; Goldingay and Van Mourik, 2013). The other component which acts alongside the negotiation mechanism is the interaction protocol, which manages the time and structure of exchanging data between each agent (Owliya et al, 2013; Jules et al, 2015).

2.8.5 Centralised versus decentralised scheduling system

A number of researchers have compared the multi-agent approach to centralised scheduling approaches for solving the job shop scheduling problem. In related research, the authors compared the performance of operational research algorithms to the multi-agent distributional algorithms (Frey et al, 2003). A simulation-based benchmarking scenario was developed in order to perform this comparison. The results showed that the disruption duration was much lower than what the operational research algorithm produced as long as vacant capacity was

available at the production unit. The same was observed for the standard deviation of the throughput time.

It was also shown that when disturbances occur, the multi-agent approach produces better results by means of shorter waiting queues. This was an expected result due to the adaptability inherent in the multi-agent framework. Moreover, it was shown that the centralised approach is unable to regulate disturbances from the original schedule and does not perform as well as the multi-agent approach in complex environments. Similar results were obtained in the studies of Aydin and Fogarty (2004) and Wong et al (2006). Constraints for the successful implementation of multi-agent scheduling were also investigated (Frey et al, 2003). Additionally, multi-agent system robustness was investigated and was shown to perform better than the operational research algorithm (Frey et al, 2003).

2.8.6 Manufacturing domain data formalization

In order for knowledge storage and inline analytics of scheduling data, this research proposes the use of open source ontology editor Protégé. The idea was highlighted by Kotulski et al (2014) who set the system of graph transformation for the agents who share knowledge. This approach has been improved to increase the pace of graph processing and to maintain the cohesion of the graph by using algorithms, which helped with the handling of the increased workload applied by the multi-agent systems.

The negative effects on the environment of moving urban goods, demonstrates the need for an improved system of decision making. This is in order to achieve efficient and

environmentally friendly transportation of goods. Anand et al (2014) achieved this by forming a model considering the stakeholder agents and their interactions, which works along with a knowledge base that illustrates the city's logistics domain.

2.8.7 Collaboration mechanism in agent-based model

In order to avoid the communication overhead in manufacturing control systems, ontology is proposed in this research as the main source of knowledge, considering its scalability quality as the result of its ability to store a different range of signals. Wang et al (2012) on the other hand, have approached this by introducing a pheromone-based coordination system that allows the agents to select signals that can act on them. This is done by providing a potential field which emits a variety of fields in terms of strength.

Another fundamental difference in the proposed method of Wang and Choi (2014) as compared to this project is the decomposition-based holonic approach (DBHA) to scheduling. This consists of a genetic algorithm control (GAC) and a minimum process time contract net protocol (SPT CNP), and it is possible to switch between them. Considering the makespan produced by the GAC in low stochastic conditions, SPT CNP prefers a high stochastic process time. Therefore, an estimation of the switch-over threshold was made using a back-propagation network with respect to a flexible flow shop problem with stochastic processing time. This project on the other hand, presents a method which uses fixed processing times in disturbance situations, which is of a lower standard than the DBHA method.

2.9 Conclusion

As shown in this study through the use of published research and applications, the standard ontology-based data driven manufacturing systems have the potential to enhance the complete value chain for manufacturers; starting from the design to the after-sales services. The benefits of the system can be unparalleled depending on the level of interoperability and if intelligence in the system is enabled. Furthermore, if the big data is used to drive intelligent insights for self-aware and modular systems, then productivity can be increased by manifold.

Integrated IT systems will optimize production processes throughout the value chain. Fully automated, integrated and modular production lines will be the result of interoperability and smart systems, replacing the insular manufacturing cells; this can greatly enhance productivity, reduce losses and downtimes. Self-aware machines, as shown previously, have the capability to be self-aware about their conditions and smart prognostics can avoid failures and increase the lifetime of a machine.

Ontology-based and a big data driven manufacturing environment has the potential to enable fully modular holonic manufacturing systems, which can self-adjust according to changes in design specifications; hence, the customization can be carried out in real time without the requirement of halting the complete production line. Smart machines and robots will work hand-in-hand to communicate with the smart products and automatically exchange information regarding the requirements; consequently, their holonic features will allow them to self-adjust and organize to achieve the set targets communicated by the products. Big data will be utilized to derive intelligent insights and artificial learning algorithms will be able to

make the system truly intelligent and self-correcting; hence, the system will be able to make decisions on the basis of factors that might even not be detectable by human involvement.

This study hence attempts to capture the current state of involvement of ontology and big data in the manufacturing industry and fully gauges its potential in the light of the literature. The study also gives perspectives for industry professionals to understand the current prevalent ontology standards for interoperability and motivates the need for adaptability by showing how such a transformation can lead to better outcomes. This project aims to develop, evaluate and demonstrate the effectiveness of a unified framework of semantic web technologies that enables a rapid and dynamic response to enterprise-wide changes and disturbances. Through the completion of this framework, a query-ready up-to-date knowledge base will be delivered for the purpose of prediction in manufacturing supply chains. This will benefit the OEMs, software houses, suppliers and academics in terms of the availability of the right data at the right time, for day-to-day production planning and control, for the creation of research case studies and for the seamless integration of the legacy and new system interfaces. This project will demonstrate three important applications of unified semantic web-enabled framework, namely, intelligent data query, distributed scheduling and prediction in manufacturing.

3 ONTOLOGY-BASED INTELLIGENT DATA QUERY

3.1 Introduction

Evolution of the web from the prevailing web technologies and web services to the semantic web technologies and semantic web services is rapidly advancing. This is as a consequence of the huge amount of data available, the increasing concern about missing out on using valuable data and the cost of using wrong data for decision making (Esposito et al, 2015; Bikakis and Sellis, 2016). The distinction between data and information is important. Data can be seen as raw, unstructured facts that seem random and useless until organised, and information is organised data presented in a given context (Chen et al, 2014). Although missing data (non-response) can be ignored, resulting in uncertainty in any system, it should also be considered as an informative output. It can be argued that a failure by some entities of the system to respond to some queries can also be considered as a response, as it provides us with some information. Therefore, in such an example, an absence of data provides information (informative non-response), and this is important in instances where there is minimal data. Therefore, data has potential for a variety of valuable but also erroneous informative outputs; while information is limited to one informative output. Manufacturing currently uses a mixture of data and information infrastructures, such as databases of raw data collected from its internal activities and a web of information for business-to-business information transfer along the supply chain.

The impact of the quality of an organization's data on their revenue is addressed by various research studies. According to an EBiz article, there is a dramatic 40% growth in business data each year, while approximately 20% of that data is irrelevant and useless (Tibbetts, 2012). On the other hand, 70% more revenue can be generated by an organization purely based on the quality of its data (SiriusDecisions, 2015). To quantify the cost of data quality, the sales and marketing research firm Sirius Decisions proposes the 1-10-100 rule as a rule of thumb. The rule states that the verification of data as it is entered would cost $1; data cleansing, removal of duplicates would cost $10; and the cost of doing nothing about the data would be $100 (SiriusDecisions, 2015). The development in semantic web technologies such as the resource description framework (RDF) instead of the traditional Hypertext Markup Language (HTML) allows data to be linked instead of linking information. There is a trend in intelligent systems for embedding RDF descriptions inside HTML pages through RDF attributes (RDFa). The embedded RDF model inside an HTML web page enables automated knowledge discovery and inferencing for intelligent agents (machines). The relationship between data is described by the RDF and the description is stored within a knowledge base instead of a database. This knowledge base has an ontology at its foundation. When data is stored with respect to an ontology, it impacts positively on data re-usability and maintenance.

Furthermore, the ontology enables semantic web services to be developed. Given a simple example of an online query, the traditional web services will support and search only for semantic-free keywords. A semantic web service version of this would be a unified and inferred list of related semantics. Likewise, in a manufacturing context, the semantic web services can increase the visibility of good data in an automated way. The semantic web and its services are scalable to accommodate virtually any knowledge in the manufacturing

environment, be it predictive production scheduling, production control or design specifications.

This chapter investigates the development of an intelligent data query framework through the use of semantic web technologies for manufacturing purposes. Better decision-making resources with intrinsic manufacturing intelligence emerges from a synchronised data sharing system for the mutual benefit of every stakeholder. In this chapter, a novel ontology-based query methodology is presented where the result of each query from the system is evaluated based on the semantic relevance and distance of the query to the local manufacturing documents. The primary objectives of the ontology-based data query were to develop an efficient and scalable data interoperability and retrieval system; in order to find the most relevant query results with minimum message cost, most hits per query and least response time. The model is built upon a real case study and the results are compared with a semantic-free random walk mechanism. By investigating an intelligent and efficient data query system in a manufacturing network, this work was able to fill a gap in the current research.

3.2 Migration into knowledge-based manufacturing

There is a trend for manufacturing companies to shift from product-oriented production to user-oriented production (Pine, 1993; Tseng et al, 1996; Tseng and Hu, 2014). The shortening of products' life cycles, global competitiveness and the economy contribute to the market volatility (Kernschmidt et al, 2015). The shortening of a product's lifecycle is caused by the ability of the competitors to quickly react to release innovative products. The global competition among companies competing on the performance dimensions of product

features, price, lead time and quality and also competing within the same market segment is becoming increasingly fierce. To stay on the leading edge, manufacturers need to address the market in new ways (Ageron et al, 2012).

Effective and efficient knowledge sharing and reuse within collaborative manufacturing enterprises is a game-changing opportunity for manufacturers to stay competitive in such a dynamic environment. A seamless flow of information enables manufacturers to increase their throughput while keeping overhead costs to a minimum (Lin et al, 2011). The manufacturing systems play a crucial role in supporting the manufacturer's efforts as they are intended to provide accurate data in a fast way (Bi et al, 2014). This data comes in a variety of formats, including documents, graphics, video files and the non-human interpretable signals produced by sensors. Data is collated within distributed databases, which are in turn kept within a variety of storage formats and programmes (Song et al, 2013). The data may be retrieved using limited key search attributes within the database, such as relational data base's (RDB) table, row, column, primary and foreign key (Astrova, 2009; Sane and Shirke, 2009; Cerbah, 2010). The power of the database's search capacity may increase exponentially if the data is linked, allowing for further attributes and for queries to be combined.

In regard to shop floor operations, the collected data may provide key metrics to the management systems on the shop floor. Such key metrics include level of asset utilisation, statistical control processes and job tracking among others. The underlying intelligence in such systems utilises all available data and provides analysis, forecasting future states and support in decision making (Khan et al, 2006). This ensures that manufacturers make better decisions at all times in supporting their operations (Lee et al, 2014). The correct application

of this intelligence gives leverage to allow manufacturers to query their manufacturing sites, to make informed decisions and in turn, perform necessary activities in an educated manner. These activities may include design experimentation, alerts within the system, optimisation, simulation, modelling, prediction and reporting more generally (Davenport et al, 2010).

3.3 Research methodology

Intelligent data query is an emerging field of research for the manufacturing supply chain community, as the need for an efficient and effective data sharing and retrieval system is more in demand than ever. This section addresses the key concepts used and the foundation for an intelligent semantic query system.

3.3.1 Manufacturing ontology

For the scope of this research, a manufacturing ontology is built with Protégé ontology editor software, which is a free and open-source leading ontological engineering tool. It provides an interface with other knowledge-based tools such as Java Expert System Shell (JESS) and is compatible with various ontology languages and formats such as eXtensible Markup Language (XML), DARPA Agent Markup Language (DAML) and Ontology Inference Layer (OIL) (Ga et al, 2009). In addition, it is particularly suitable for this research as it has deductive classifiers for validation of the ontology consistency. Furthermore, it can also export into other formats such as RDF, which is the basis of this research. When modelling in a domain, developers must be able to focus more on the concepts and relations rather than the syntax of the final results. This can be achieved through a Protégé-based editor, resulting in modelling at a conceptual level (Noy et al, 2001; Gennari et al, 2003).

The concepts of the ontology and the instances are created based on the data from the case study. The scope of the ontology was limited to the workshops' attributes, such as their core competency operations. Some of the key extracted attributes from the case study are summarised and tabulated in section 3.3.2, which is the foundation for the developed ontology. However, literature research highlighted that the existing ontologies do not cover the terminologies used in this data query system research as a result of their ‘domain specific application’ nature. For instance, the concept of ‘quality control’ is not demonstrated in the open source MASON ontology as it was specifically designed for the automatic production cost estimation. In order to widen the scope of the project, the ontology can be further developed for future work and enriched by mapping its main concepts to famous WordNet (https://wordnet.princeton.edu) lexical ontology and MASON in an offline manner. This will ensure that the implementation process is facilitated at scale. The alignments of the ontologies will lead to an improved knowledge base, which will allow more data to be queried from the system. As an example, for this case study, the concept of ‘production cost’ is not considered as it can be covered by mapping the proposed ontology to MASON. Figure 3.1 shows the core concepts of the proposed ontology where each core concept is further formalised into more sub-concepts.

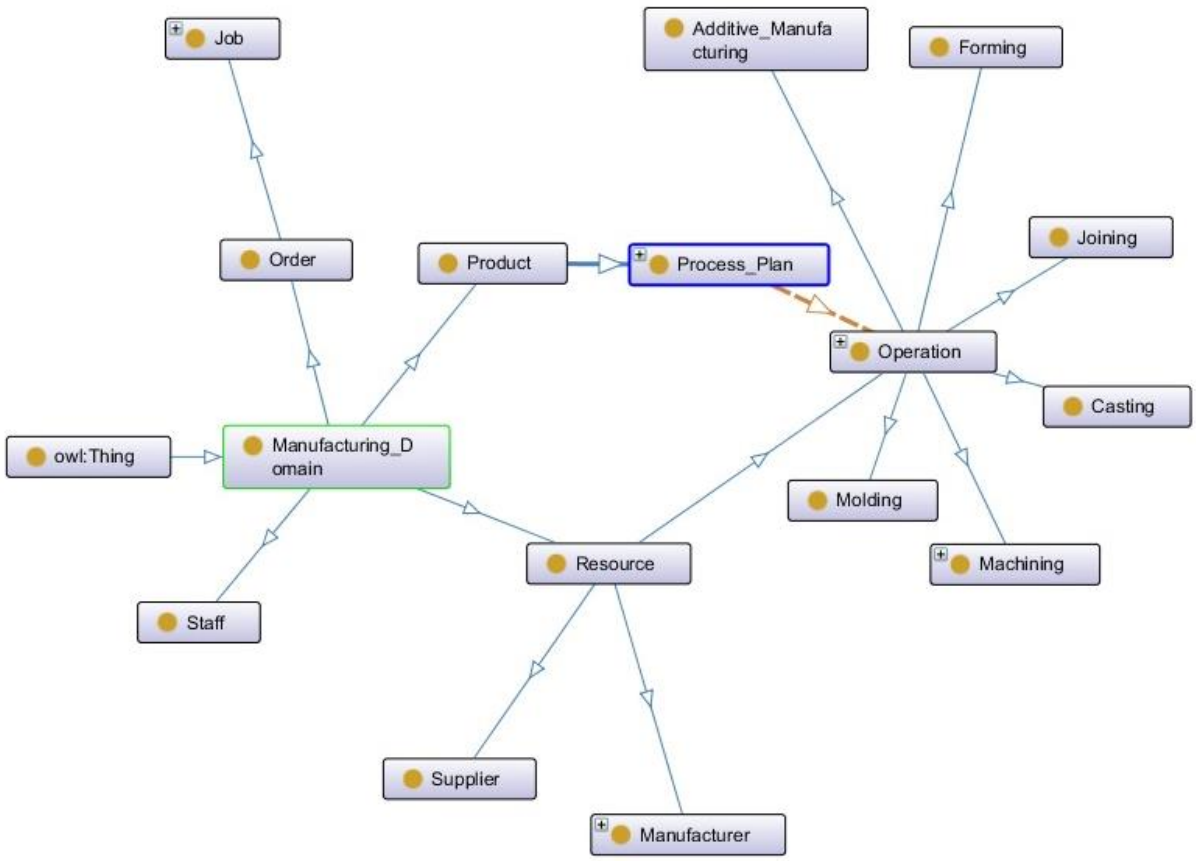

Figure 3.1 Developed ontology for manufacturing data query

For instance, the Manufacturer concept is a sub class of the Resource concept. Similarly, the Manufacturer concept is further specialised into different manufacturer instances, where each one provides a set of manufacturing operations. Each Order consists of a set of Jobs which are defined by unique process plans. These process plans outline which operations are required for each job. Therefore, isNeededBy defines which operations are needed for each job and isProvidedBy defines which manufacturer provides that particular operation. Figure 3.2 shows an example of detailed object properties within the developed ontology.

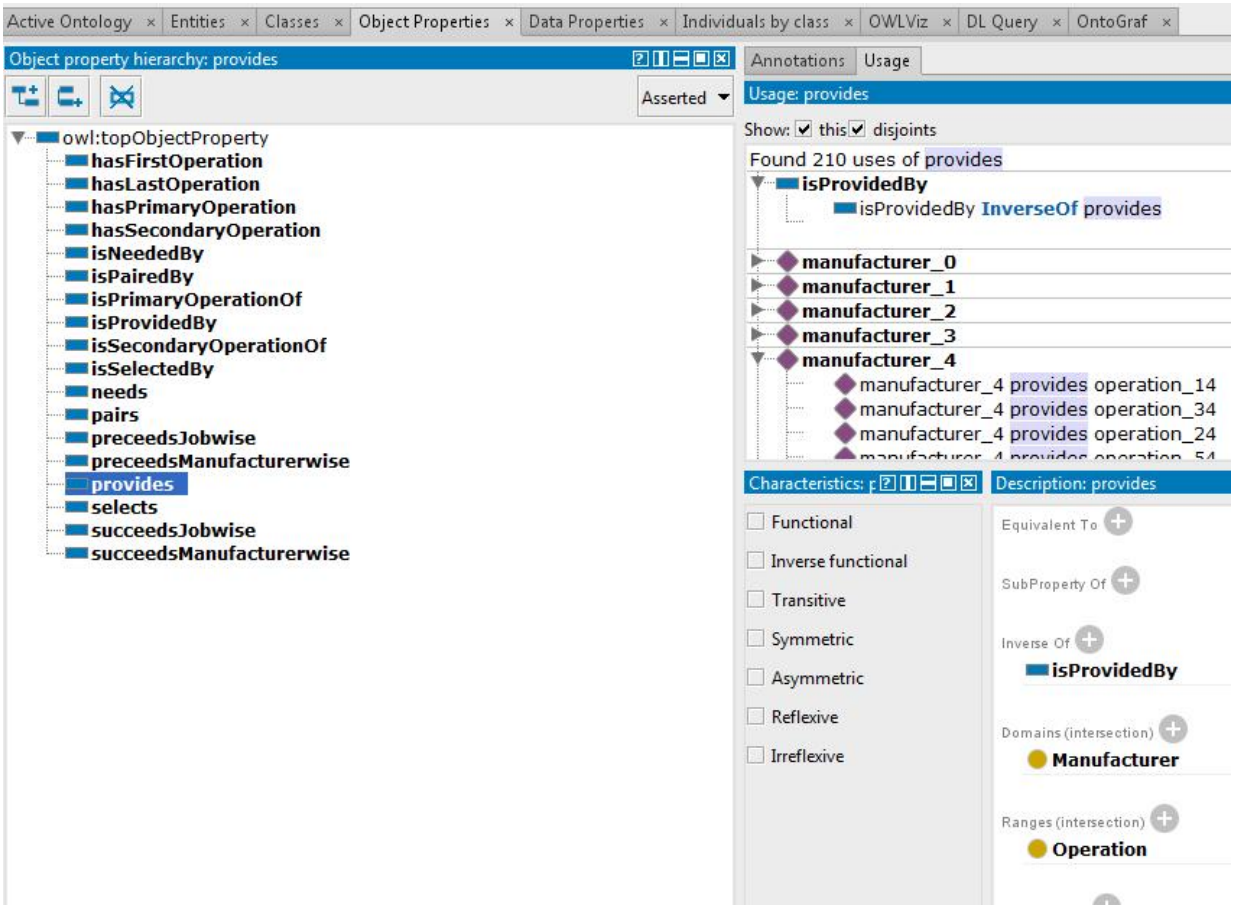

Figure 3.2 Object properties of the proposed ontology

3.3.2 Industrial case study

The case study was carried out at the Gruppo Fabricazione Meccanica (GFM) S.r.l. Group, Italy, and its objective was to use the result to help in the understanding of how the theories of semantic data query apply to real manufacturing business situations. The case study will provide information to create a model of how the company, under study, is involved in the life cycle of its products. To do this, the study will look at the activities involved when a call for a proposal is received; an offer is generated; a firm order is received; material sourcing, production, quality control and transport logistics are triggered; and finally, after-sales

service is provided. During all these phases, hundreds of queries and messages are generated and as a considerable portion of these are unnecessary and can be avoided with an intelligent query system, they are highlighted.

GFM acts as a central company to a pool of manufacturing workshops and suppliers. They collaborate with over 300 suppliers and over 40 workshops to fulfil customer production needs. Their expertise is in the production of gas turbines, steam turbines and generator parts. GFM is a single point of contact for this study, and the case study data is extracted by queries in Microsoft Structured Query Language (SQL) Server to GFM's management and information system. They have internally developed the software that monitors the supplier's performances so it can better meet their needs, and have built procedures/queries for extracting the data.

3.3.3 Simulation platform

In collaboration with GFM, the most important data about each individual supplier is gathered and imported into different nodes in the simulation software; namely, the operations provided, workpiece materials, fleet of machines and quantity, types of testing, tolerance dimensions and certifications. These are parsed into Java objects and stored in each node as its local database. A Java simulation platform is developed in NetBeans' integrated development environment (IDE) to implement the model of the network used by the manufacturing workshops and to simulate the message generation process during a data query. The simulation of the manufacturing network is built upon a primitive version of the

software presented in a previous work (Jules et al, 2015). However, the current simulation is mainly focused on the flow of data within the network and the use of ontology for data queries rather than network formation and scheduling. Each node or object represents one supplier/workshop and has the capability of communicating with other nodes. During the simulation, a query list is generated from one node (GFM). This is then circulated among the nodes until the result of the query is found. The platform makes use of two protocols for the search mechanism: namely, random search protocol and ontology search protocol. The random search setting involves randomly selected nodes; while in the ontology search protocol the query is sent to nodes with a high probability of containing the answer to the query based on the unified ontology. These two search methods are explained in more detail in the following sections. The schematic of this distributed manufacturing network is illustrated in Figure 3.3.

Table 3.1 Examples of key extracted attributes of workshops

WORKSHOP'S ATTRIBUTES	SAMPLE VALUES
SUPPLIER CODE / GFM CODE	2.3.4.10.370 / 2AE001031.00
SUPPLIER TYPE	MACHINING ; TREATMENT ; RAW MATERIAL PROVIDER
OPERATION OFFERED	TURNING ; MILLING ; HEAT TREATMENT
FLEET OF MACHINES	MAZAK NEXUS 250MY (MAX TURNING DIAMETER: 70 MM)
CERTIFICATIONS	ISO 9001:2008; CERTIFICATE NO. 6391; EXPIRY DATE: 28/04/2018
NON-DESTRUCTIVE TESTING (NDT) CONTROL	PENETRATION TEST; HARDNESS TEST; VISUAL TEST
NON-DESTRUCTIVE TEST	RADIOGRAPHIC INSPECTIONS; DYE PENETRATE INSPECTIONS; MAGNETIC PARTICLES INSPECTIONS
DESTRUCTIVE TEST	TENSILE TESTING; IMPACT TESTING; SPECTROGRAPHIC EXAMINATION; STATIONARY SPECTROMETER
MATERIALS	21CRMOV57; X15CR13; 26NICRMO146
STABILISATION	WTLV9520-01; WTLV9520.01
SOLUBILISATION	WTLV9520-01; 424W834R0; WTLV9520.01
NUMBER OF PLACED ORDERS PER MONTH/ PER YEAR	JULY 2006 / 6 ORDERS
REQUESTED DELIVERY DATE	10/10/2006
ACTUAL DELIVERY DATE	04/12/2006
DELAYS	55 DAYS
QUALITY	CONFORM; NON-CONFORM; NUMBER OF LOTS CONFORM

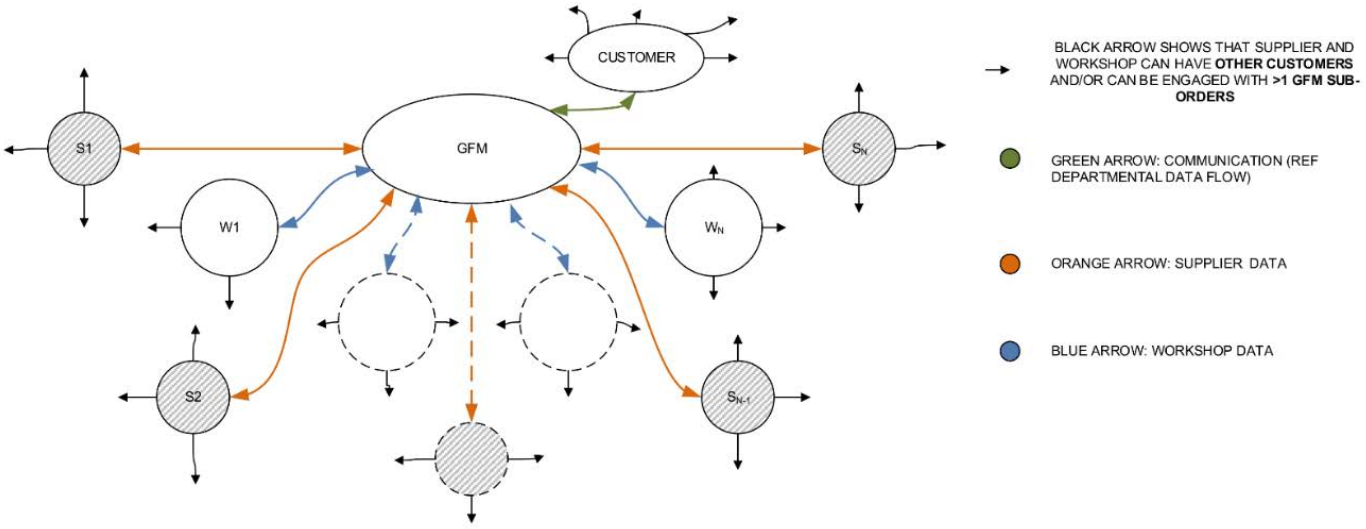

Figure 3.3 Case study network structure (data flow between network entities)

This figure illustrates the formation of a network and the data which support the network operation for 1 sub-order. This network is constituted with supplier and workshop data, who store data at every instance. The green, orange and blue arrows present the customer communication, supplier and workshop data flow within the network respectively. As the queried data cannot be accessed easily, it can be stored in any section of the network. Ontology is the most efficient solution to access any data within the network.

3.3.4 Random search protocol

In a manufacturing network, each workshop has a myopic view of the system and the data stored in different resources. Search protocols enable the finding of the best queried resources by routing the query to the right workshop. This research is trying to show the potential of ontology for the development of an intelligent and efficient query system compared to a semantic-free search method. The random search method adapted in this study is based on the Gnutella algorithm (Breadth-First Search (BFS) algorithm), which uses flooding to search all

the workshops for a given query (Yeferny et al, 2011; Kapoor et al, 2013; Arour and Yeferny, 2015). This method is well documented and it is used due to its simplicity of implementation. The aim is not to develop a random search protocol, but to modify this protocol for application within a manufacturing network and compare it with the proposed novel ontology search protocol.

The search mechanism is built upon two protocols, namely the random search protocol and the ontology protocol. During programme initialisation, an object called Message is created containing some properties such as QueryList and Hit. QueryList, as the name suggests, contains the query that is sent to the network; and Hit is a Boolean indicating whether or not the query result has been found. The simulation software is developed on a cycle-based evaluation mechanism. After each cycle, if the Message finds the answer, it triggers Hit; otherwise it is sent to a random neighbour node and also adds to the counter CrossStep. Moreover, each protocol has properties called LastMessage, IncomingMessages and ProcessList which will be used in the Observer method for the evaluation of the whole process. This performance measure's evaluation is explained later in the following sections. In each cycle, the content of the QueryList is compared with the local storage of each workshop. For each common match between the QueryList and the local storage data list, a counter is added inside the FindRelevance method and that relevance is compared with a predefined threshold. If the value of the relevance exceeds the threshold a Hit is triggered; otherwise the cycle proceeds. The algorithm for the random search protocol is demonstrated in Algorithm 1 in Figure 3.4.

Algorithm 1:

1- QueryList: Create a list from queries and send it to nodes via message
2- There is a variable in each node called LastMessage resembling the last received message
3- ProcessList: The list of operations offered by each node(workshop)
4- Relevance: The value obtained by comparing LastMessage and QueryList
5- Hit: The Boolean value indicating receive of message from a node(workshop)
6- Threshold: Hit occurs if the value of Relevance is higher that Threshold
7- CrossStep: Number of hops that a message travel until it trigger Hit

```
Procedure:
If (lastMessage != null)
{
//Compare LastMessage and QueryList and return Relevance value as the result
Relevance = 0.0;
      foreach  (queryItem in QueryList){
         if (ProcessList.contains(queryItem)) {
                  Relevance ++;
         }
      }
//Hit the message if the value of Relevance is higher than Threshold, otherwise send
the message to a random neighbor
If(Relevance> Threshold)
{
Message.hit=true;
}
Else {
Message.hit=false;
Message.crossStep++;
NeighborsPeer= getNeighbors(Random integer Index);
          SendMessage(Message, NeighborsPeer)
 }

}
```

Figure 3.4 Random search protocol

3.3.5 Ontology search protocol

As part of ontology engineering practice, there are different ways to create and represent ontologies. The common key standards are web ontology language, also known as OWL and resource description framework known as RDF. In the ontology search protocol, RDF has been used for the representation of domain information. Abstract syntaxes are used for the description and are linked with the concrete syntaxes of the formal semantics. The documentation part in this case involves XML serialisation as well as the formal semantics. Graphical representation in the form of a data model of the RDFs comprises a subject, a predicate and the target object, which are presented with the help of nodes and a directed arched diagram.

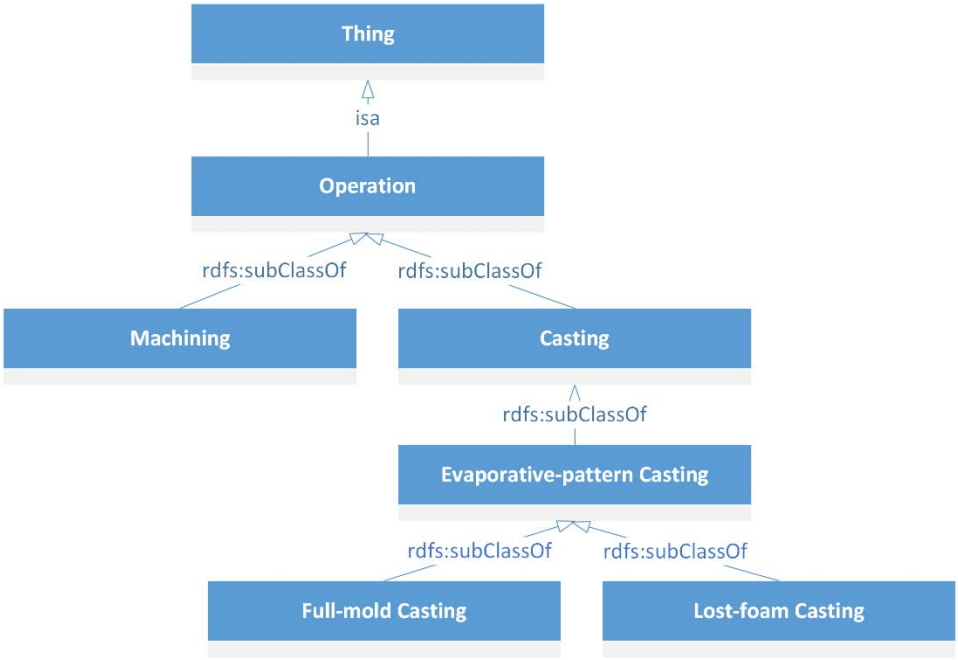

Figure 3.5 RDF representation of manufacturing ontology

In this research, the semantic distance of domain concepts is calculated from the RDF graphs and was used during the search process for the most relevant results. If the Operation concept from Figure 3.1 is further expanded to its sub-classes, the relation between existing operations can be investigated. For instance, Machining, Casting, Forming and Joining are all sub-classes of Operation. The Casting itself has sub classes such as Full-mold casting and Lost-foam casting. The RDF graph of such a hierarchy can demonstrate the relationship between these concepts in the ontology and also shows the semantic distance of these concepts. As an example, the semantic distance between Lost-foam casting and Full-mold casting is two, as it is shown in Figure 3.5. Algorithm 2 elaborates on the procedure involved when ontology protocol is in use.

Algorithm 2:

1- QueryList: Create a list from queries and send it to nodes via message
2- There is a variable in each node called LastMessage resembling the last received message
3- ProcessList: The list of operations offered by each node(workshop)
4- OntologyRelevance: The value obtained by comparing LastMessage and QueryList form RDF graph and indicating the semantic distance of two concepts
5- Hit: The Boolean value indicating receive of message from a node(workshop)
6- Threshold: Hit occurs if the value of OntologyRelevance is higher that Threshold
7- CrossStep: Number of hops that a message travel until it trigger Hit

```
Procedure:
If (lastMessage != null)
{
//Compare semantic distance of two concepts from QueryList and LastMessage and
//return the OntologyRelevance value
OntologyRelevance = 0.0;
foreach(QueryItem in QueryList) {

        foreach (ProcessItem in ProcessList) {
//The higher the distance number, the farther semantic distance of two concepts
            OntologyRelevance += 10 – getOntologyDistance(QueryItem, ProcessItem);
        }
    }
//Hit the message if the value of OntologyRelevance is higher than Threshold,
//otherwise send the message to the closest neighbor in the RDF graph with the value
more than Threshold
If(OntologyRelevance > Threshold)
{
        Message.hit=true;
}Else {
        Message.hit=false;

        While(,neighbor!=null)
        {
            if (getOntologyDistance (ListQuery, neighbor.processList) >
        Threshold) {

                Message.crossStep++;
                SendMessage(Message, neighbor)
              }else{

            Neighbor= neighbor.next();
          }
}
}
```

Figure 3.6 Ontology search protocol

The contribution of this research could be put into perspective as follows: (i) the proposed ontology search protocol presents a novel data query system, which focuses on retrieving information with respect to the semantic content of a workshop and by considering all the relevant accessible data within a workshop; (ii) this data query algorithm will evaluate the semantic richness of the database to return the most relevant results for each query, in comparison to a semantic-free based approach, even if the answer to the query does not exist within the documents with the help of reasoning; and (iii) the development of the proposed search mechanism results in a significantly more efficient data retrieval system than the other data query systems such as Random Walk and ontology-based semantic query routing

(OSQR) (Himali et al, 2012) considering the recall and precision metrics of the search quality from the literature.

3.3.6 Ontology learn protocol

Following the completion of the first phase of the project, the data query system was further optimised in the second phase in order to obtain a more intelligent and efficient system. In light of the above, a learning technique was added into system where the best results are obtained by introducing a hybrid technique of having ontology and a data repository in place. The contribution of this extension is the introduction of a hybrid model where the ontology algorithm is integrated with a repository of data. This will optimize the query process by decreasing the total cross step or the distance travelled by the messages to find the query results. Three search mechanisms have been investigated for the manufacturing data query in this section; namely, random search, ontology search and ontology learn search protocol.

The ontology learn protocol is introduced as an extension of the ontology search mechanism in this section, where it further optimizes the ontology protocol. In this protocol, the ontology framework is combined with a data repository where learning takes place. During the initialization phase, the repository is empty. As the simulation proceeds, after each Hit, data about that node is stored in this repository. Then, during each simulation cycle, the software will look at the repository first to see if the answer for the query is listed and then it will search the nodes for comparison. This method has improved the number of total CrossSteps and the simulation time, as discussed in the following sections. The ontology learn method is described in Algorithm 3.

Algorithm 3:

1- QueryList: Create a list from queries and send it to nodes via message
2- There is a variable in each node called LastMessage resembling the last received message
3- ProcessList: The list of operations offered by each node(workshop)
4- OntologyRelevance: The value obtained by comparing LastMessage and QueryList form RDF graph and indicating the semantic distance of two concepts
5- Hit: The Boolean value indicating receive of message from a node(workshop)
6- Threshold: Hit occurs if the value of OntologyRelevance is higher that Threshold
7- CrossStep: Number of hops that a message travel until it trigger Hit
8- QueryRepositoty: A storage file in CSV format for storing queries and node numbers
query1,query2,…,queryN,Node X

```
Procedure:
If (lastMessage != null)
{
//Compare semantic distance of two concepts from QueryList and LastMessage and
//return the OntologyRelevance value
OntologyRelevance = 0.0;
foreach(QueryItem in QueryList) {

        foreach (ProcessItem in ProcessList) {
//The higher the distance number, the farther semantic distance of two concepts
            OntologyRelevance += 10 – getOntologyDistance(QueryItem, ProcessItem);
        }
    }
//Hit the message if the value of OntologyRelevance is higher than Threshold,
otherwise send the //message to the closest neighbor in the RDF graph with the value
more than Threshold
If(OntologyRelevance > Threshold)
{
        Message.hit=true;
        //When the Message trigger Hit, write it inside the repository file
        Write  << QueryList in QueryRepository .csv // queryList,NodeIndex
}Else {
rproto.lastMessage.hit = false;
//If there is no relevance between the current node and the query list, find the node
//identifier
String nodeName = searchInFileAndGetNodeName(lastMessage in  QueryRepository
.csv);
        if (nodeName is not empty) {
//Send it to that specific node if it was in the file
sendToSpecificNode(Integer.valueOf(nodeName.replace("Node", "")), protocolID,
lastMessage);
} else {

sendToNearesDistancetNeighbor(node, protocolID, rproto.lastMessage);
        }
 }
}
```

Figure 3.7 Proposed ontology learn algorithm

3.3.7 Performance evaluation

Evaluation of the performance of a data retrieval system is a crucial task which entails a well-established test method and benchmark. The results of the queries from the ontology will be evaluated against the actual existing data from the manufacturing environment in order to determine the fidelity of the ontology query model and the simulation model. The key performance indicators (KPIs) involved in calculating fidelity of the proposed system are devised based on precision and recall performance criteria. Based on the literature review, these KPIs are the most prominent performance measures when dealing with information retrieval systems and ontologies (Raghavan et al, 1989; Müller et al, 2001; Euzenat, 2007), and therefore they have been chosen for the evaluation of this research. Furthermore, these KPIs have been chosen to allow a comparative approach to other similar research papers (Himali et al, 2012). In the configuration of the simulation software, an Observer method is created to evaluate the efficiency of the algorithms developed. The Observer calculates the number of Hits from all incoming messages, message cost, maximum CrossSteps and the sum of all CrossSteps, and these are the basis of the performance evaluation. The message cost is the number of all generated messages in the network and the maximum CrossStep is the longest distance travelled by the Message to trigger the Hit. The Response time is considered as the first occurring Hit. Precision is a measure of returned query relevancy, which is defined as the number of relevant retrieved instances to the total number of instances; and the other measures concern hits per query and recall. As mentioned previously, a Hit message is generated when a node or a workshop within the network has got the answer to a query. Therefore, the Hit Per Query is the average number of distinct relevant documents discovered per search query. This is shown by the equation below:

$$Hit\ Per\ Query = Sum\ Hit/Query \quad (3.1)$$

Recall is defined as the ratio of the number of relevant records retrieved to the total number of relevant records in the system. All these key performance indicators are used in this research to assess the basis of the proposed protocols.

In every information retrieval, two main metrics of precision and recall are targeted to measure the classifier output quality of the search. These metrics are also linked to their harmonic mean, F-Score, which can be obtained by the formula 3.2 and 3.3: (https://www.crosswise.com/cross-device-learning-center):

$$Recall = TP/(TP + FN) \quad (3.2)$$

$$Precision = TP/(TP + FP) \quad (3.3)$$

Precision is mainly known as the measure of result relevancy, which is dependent on the number of true positives (TP) and false positives (FP). On the other hand, recall is a metric value to evaluate the number of truly relevant results obtained and it is dependent on the number of true positives (TP) and false negatives (FN). The link between these metrics is a great illusion of the nature of the current problem in the process of information retrieval. By recalling data, all relevant documents can be accessed and finally sorted out from the junk, and this is interpreted as precision. In an information retrieval system, low precision leads to a high number of incorrectly predicted results, and low recall leads to a lower number of correctly predicted results. Therefore, the ideal system would have to demonstrate high precision and recall in order to achieve its most accurate results. As accuracy is very misleading in the evaluation of the quality of information retrieval, these metrics are taken into account together to ensure high quality of information retrieval.

3.4 Results and discussion

The results of the first phase of simulation are depicted in Figure 3.8 to Figure 3.12 and are based on the aforementioned evaluation criteria. As demonstrated in Figure 3.8 and Figure 3.9, the ontology protocol shows a considerable amount of improvement in the number of travelled steps by the Message. The random search mechanism travels more distance among the nodes to find the query result. As illustrated in Figure 3.8 and Figure 3.9, there is a maximum difference of more than 200,000 in the comparison of total cross steps, as well as a maximum change of over 1,500 maximum cross steps as a result of the new proposed ontology. Figure 3.10 and Figure 3.11 show experimental results for Hit and Hit per Query. As depicted in these figures, the ontology protocol shows a dramatic improvement of 166.67% and 7,900% in total Hit counts compared to OSQR protocol and the random protocol respectively (Himali et al, 2012). As demonstrated in Figure 3.10 and Figure 3.11, the ontology protocol has been able to reach a maximum of 7,920 Hits per cycle, while the random protocol holds only 99 Hits. The recall rate of the proposed search mechanism is drastically higher than both the OSQR and random protocols respectively. However, the OSQR protocol was based on WordNet ontology which used a wide range of sample data inputs. In order to accurately compare the results of the proposed protocol with OSQR, it is essential to simulate the constraints by equalising the number of data inputs.

Figure 3.12 shows how the ontology protocol outperforms the random search method and OSQR in regard to the precision rate. With the proposed algorithm, the possibility of an incorrect identification of data is minimised to ensure the least mismatching of the relevant data with irrelevant information to the query. The calculated improvement rate of precision resulting from this ontology is estimated at 228.57% compared to OSQR, and 604.10%

compared to the random search; which points out the efficiency of the ontology protocol when considering Hit messages per query. Table 3.2 tabulates the key performance evaluation criteria.

Table 3.2 Key performance evaluation criteria

	Recall	Precision	F-Score	Hit Count
Random	0.68	0.98	0.073	99
OSQR	1.49	2.1	0.42	2970
Ontology	12.72	6.9	0.82	7920

The only downfall of the proposed ontology protocol is the response time. For the simulation run illustrated here, the average response time for the ontology protocol is nine minutes; while the random search is done in under a minute. This is due to the semantic calculation happening in the ontology protocol and could be alleviated in future work.

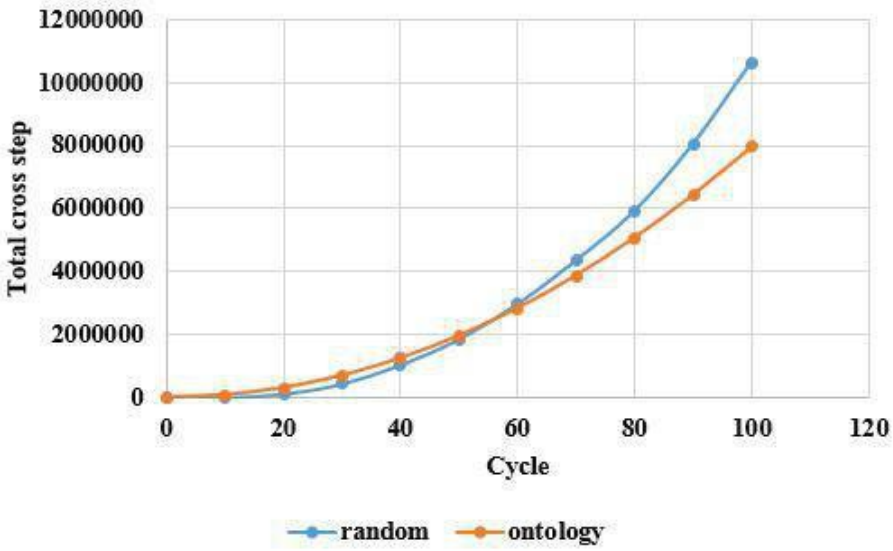

Figure 3.8 Total cross step comparison of random search and ontology

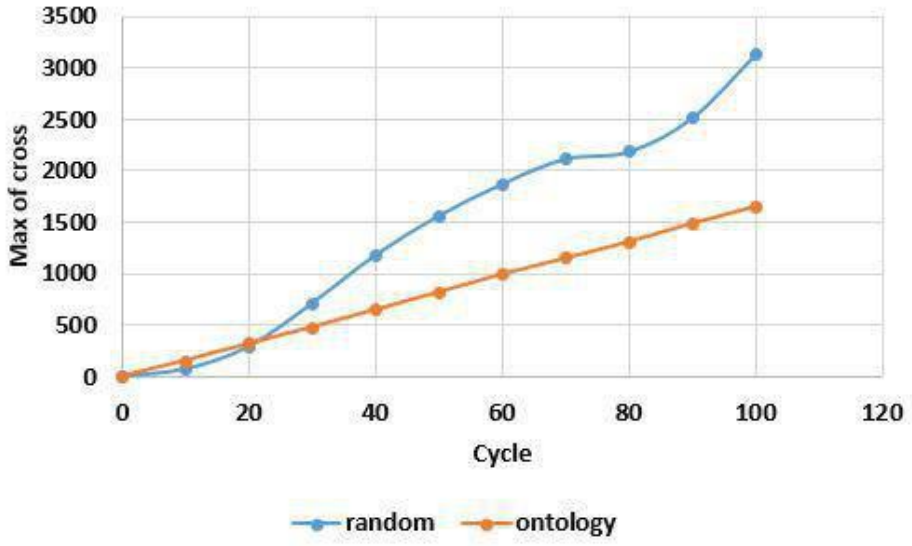

Figure 3.9 Maximum cross step comparison of random search and ontology

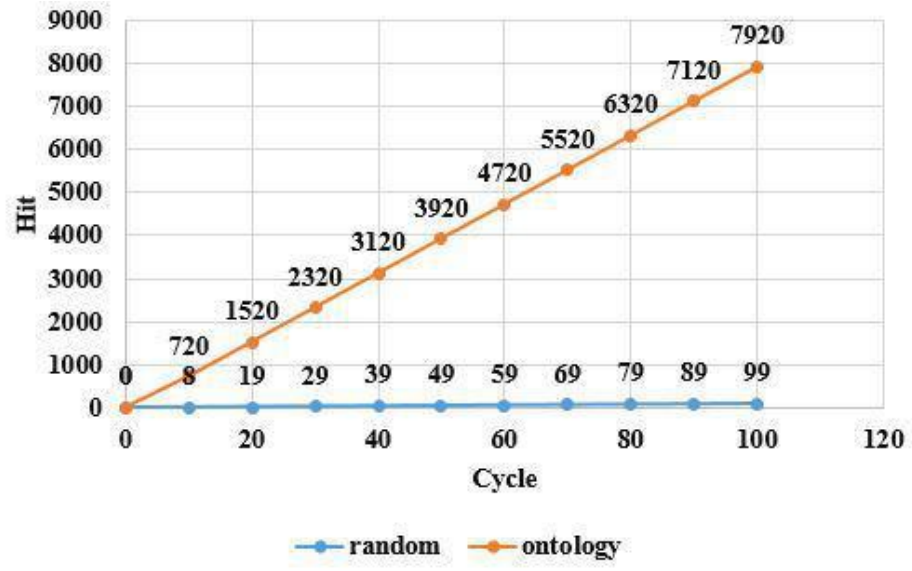

Figure 3.10 Hit rate comparison of random search and ontology

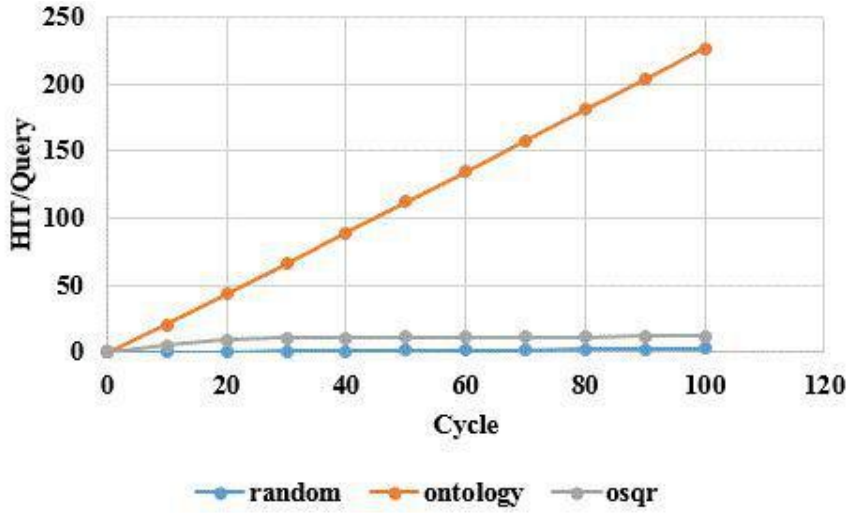

Figure 3.11 Hit per Query comparison of random, ontology and OSQR protocol

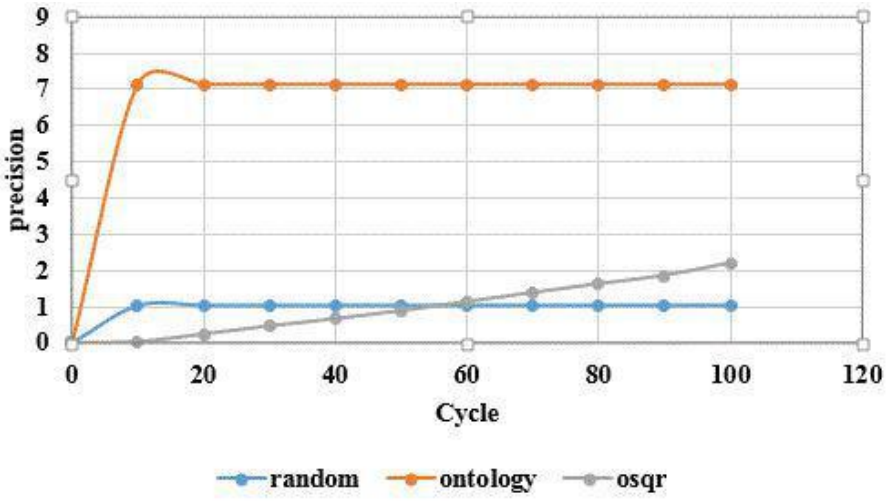

Figure 3.12 Precision rate comparison of random, ontology and OSQR protocol

The results of the second phase of the simulation which is based on the learning mechanism are illustrated in Figure 3.13 to Figure 3.15. The results are evaluated based on the number of cross steps, the number of Hits per query, recall rate and response time. For a simulation with fixed value of query size the random protocol has 43 hits and 9,857 miss messages with nearly 26 million total cross steps. The ontology has shown a significant improvement by 8,122 hits and 1,778 miss messages; while it reduced the total cross steps to nearly 16 million. However, the proposed ontology learn protocol, not only obtained 8,521 hits and 1,379 miss messages, but also dramatically improved the number of total cross steps by achieving 136,521. Also, the response time was reduced by nearly one third indicating the benefits of the proposed model.

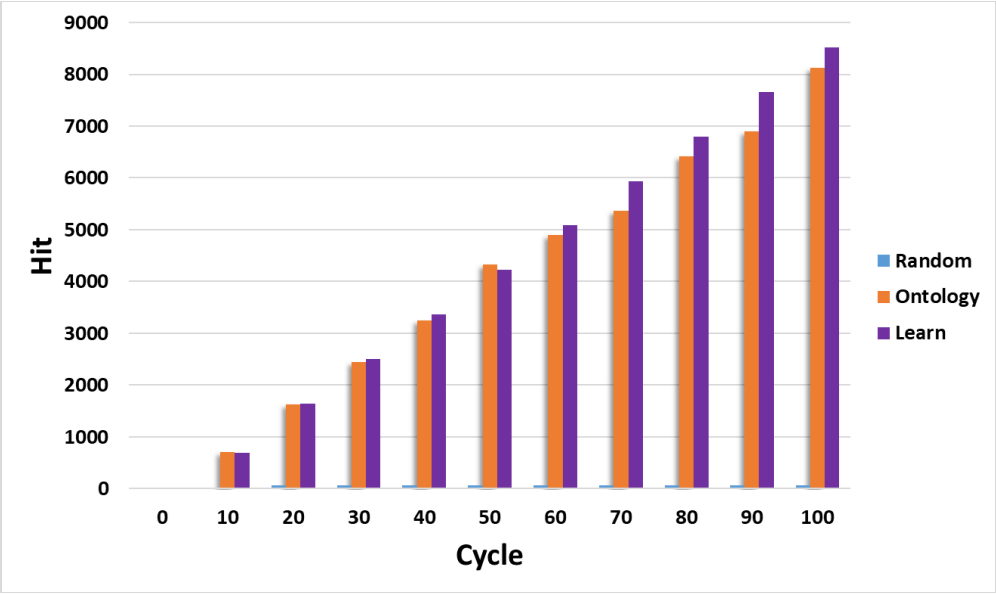

Figure 3.13 Comparison of the number of Hits

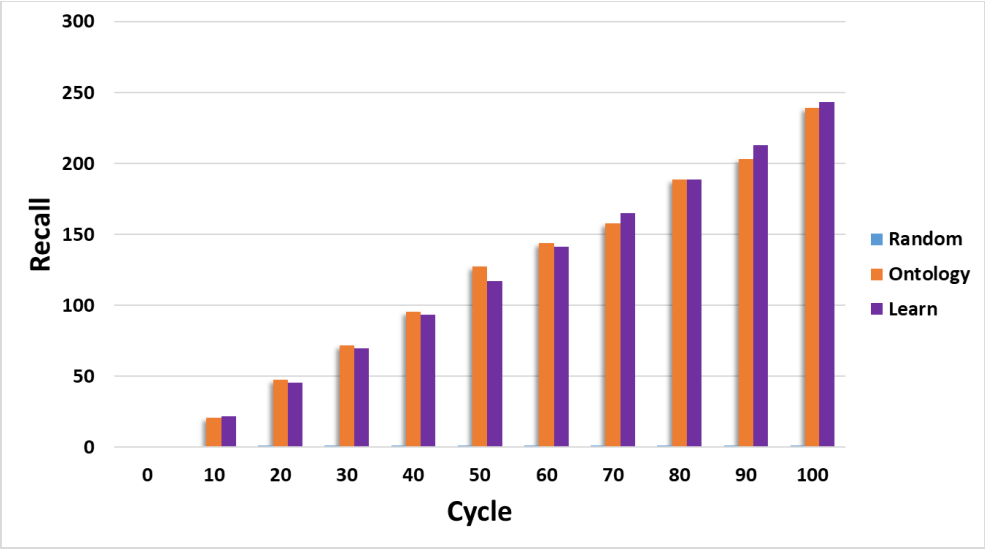

Figure 3.14 Comparison of the recall rate

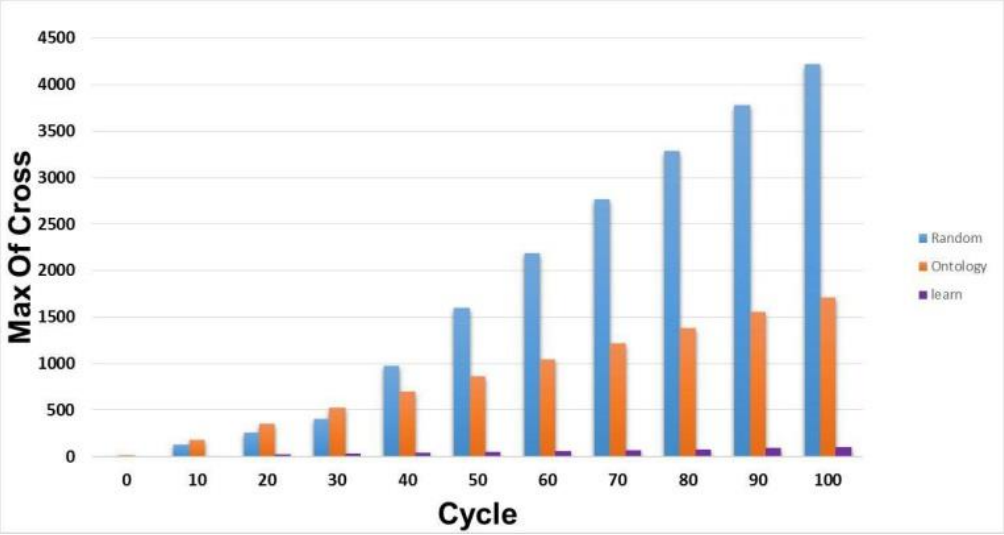

Figure 3.15 Comparison of the maximum number of cross steps

3.5 Conclusion and future work

A manufacturing ontology was created in this research to model the knowledge from a real case study. Accordingly, a simulation platform was developed to demonstrate the validity of an intelligent data query system. Based on the results, the effectiveness of the ontology protocol was shown compared to a random search mechanism and the well-established OSQR method from the literature. Using the ontology not only helped to reach the desired query results with fewer messages, but it was also useful for capturing the most relevant results per query. Moreover, it will be useful in case of the absence of query results in local storage databases. For instance, if the result for a query does not exist in the nodes, the random search will return no result with a huge number of generated messages. However, the ontology protocol will return the closest available answer to that query through the use of semantic relations. This will be beneficial especially in a dynamic manufacturing environment. In the second phase, the semantic-based technique was further improved by introducing a data repository as a means of learning. This learning process reduced the total cross steps and the processing time.

The scope of this chapter was limited to two search mechanisms and a number of limited evaluation metrics. As future work, the research could focus on the enhancement of the manufacturing ontology by using well-known existing ontologies such as WordNet and MASON. Furthermore, the research will be enhanced in the following chapters by the development of a multi-agent system where autonomous agents automate the whole query process. They will communicate with each other to find the query results and also to populate

the ontology in an automatic way for knowledge reuse. In addition, a novel methodology will be introduced in section 5.3.5 of this thesis, where the ontology is created by mapping data from spreadsheets automatically. This method allows the ontology-based data query system to be easily applied on larger data sets, which can considerably improve the scalability of the whole model.

4 AGENT-BASED DISTRIBUTED MANUFACTURING SCHEDULING

4.1 Introduction

Manufacturing companies need to be very efficient in modern competitive economies. Production plants currently have a requirement for greater flexibility and productivity in order to be able to compete. To increase efficiency, optimal scheduling algorithms are important (Liu and Sycara, 1996). Scheduling in a manufacturing context refers to the allocation of resources to plants in a way that will minimize the time taken to produce products.

A common problem in manufacturing scheduling is the job shop scheduling problem (JSSP), which provides an obstacle to increasing the efficiency of a manufacturing plant and is applicable to numerous manufacturing industries, where resources need to be efficiently assigned (Yoo and Müller, 2002). Job shop scheduling refers to a set of jobs which need to be processed by a set of machines; each job consists of a chain of operations which must be processed during a specific time period on a given machine (Jules and Saadat, 2016). Each job is composed of a set of partially ordered operations; where each operation has a deterministic processing time and pre-assigned materials that need to be minimised, subject to certain constraints (Jules and Saadat, 2016). The constraints can relate to material quantity, job prioritisation and the capacity of the operational centres, as well as deadlines (Gabel, 2009).

It is therefore an NP (non-polynomial time) complete combinatorial optimisation problem. Combinatorial optimisation problems are optimisation problems with a finite number of feasible solutions that are difficult to solve in real time (Wu and Durfee, 2006). The job shop scheduling problem is NP hard; its algorithmic complexity increases exponentially with the number of operators and the number of machines defining the job shop scheduling problem, dynamic nature and practical interest for industrial applications. This means that unless P = NP, there exists no polynomial time algorithm that finds the optimal solution of a given JSSP instance (Wu and Durfee, 2006). Therefore, a heuristic or metaheuristic optimisation algorithm is essential to tackle these kinds of problems.

A large number of algorithms have been developed in order to solve general scheduling problems, including the job shop scheduling problem (Hasan et al, 2009; Martin et al, 2016). Those include the genetic algorithm, particle swarm optimisation, simulated annealing, ant colony optimisation, artificial bee colony, and bee colony optimisation (Gabel, 2009; Hasan et al, 2009; Martin et al, 2016).

This research proposes an approach to a combined, shared knowledge base, which leads to the development of a decentralised manufacturing network. It is an attempt to help small and medium manufacturing enterprises (SMEs) to achieve an informal nature of leadership amongst their firm by sharing knowledge and resources (Müller-Seitz and Sydow, 2012). In order to end the decision making in concord, a shared knowledge base is required to act as a medium in which the members of a firm can share their complementary data (Artto et al, 2011; Li et al, 2013).

The manufacturing industry is now more focused on open organisational structures and decentralised operations as well as the ability of market predictability. This has indeed led to the development of intelligent virtual systems within a software architecture also known as a multi-agent system. This distributed artificial intelligence allows the representation of complex and dynamic real-life systems with the help of its internal agents perceiving their environments and acting accordingly, with respect to their nature of computational entities (Komma et al, 2011).

In this context, simple algorithms are developed for each existing agent, which can also enable new agents to be placed in the software network. This multi-agent system's framework can easily identify and comprehend its environment and act accordingly by semantically enriching any data and meta-data (Luck et al, 2003; Guo and Zhang, 2009). This data will need to be produced and managed independently. Ontology, as a well-known knowledge base, will openly signify structures and enable them to hold the domain data to allow automated queries, reasoning and coalition within the agents with a common communication language and mutual area of focus (Monostori et al, 2006; Guo and Zhang, 2009; Lin et al, 2011).

The proposed network paradigm is a combination of ontology and a multi-agent system, designed to prevail over each system by optimising scheduling objectives on its own and to form an advanced innovatory decentralised scheduling system.

In this chapter, a decentralised scheduling system was developed that enables intelligent agents to interact with each other in order to find an optimum solution. The structure and interaction protocol of the agents are presented in detail in the following sections. For the scope of this study, a number of job shop scheduling instances were used; the ABZ6 is shown in Table 4.1 for illustration. These problem instances are the basis for research evaluation and validation within the operational research community (Lawrence, 1984).

Table 4.1 JOB PROCESS PLANS FOR ABZ6 10 X 10 PROBLEM

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
J1	17 (62)	18 (24)	15 (25)	13 (84)	14 (47)	16 (38)	12 (82)	10 (93)	19 (24)	11 (66)
J2	25 (47)	22 (97)	28 (92)	29 (22)	21 (93)	24 (29)	27 (56)	23 (80)	20 (78)	26 (67)
J3	31 (45)	37 (46)	36 (22)	32 (26)	39 (38)	30 (69)	34 (40)	33 (33)	38 (75)	35 (96)
J4	44 (85)	48 (76)	45 (68)	49 (88)	43 (36)	46 (75)	42 (56)	41 (35)	40 (77)	47 (85)
J5	58 (60)	59 (20)	57 (25)	53 (63)	54 (81)	50 (52)	51 (30)	55 (98)	56 (54)	52 (86)
J6	63 (87)	69 (73)	65 (51)	62 (95)	64 (65)	61 (86)	66 (22)	68 (58)	60 (80)	67 (65)
J7	75 (81)	72 (53)	77 (57)	76 (71)	79 (81)	70 (43)	74 (26)	78 (54)	73 (58)	71 (69)
J8	84 (20)	86 (86)	85 (21)	88 (79)	89 (62)	82 (34)	80 (27)	81 (81)	87 (30)	83 (46)
J9	99 (68)	96 (66)	95 (98)	98 (86)	97 (66)	90 (56)	93 (82)	91 (95)	94 (47)	92 (78)
J10	100 (30)	103 (50)	107 (34)	102 (58)	101 (77)	105 (34)	108 (84)	104 (40)	109 (46)	106 (44)

P1 = first process, 17 = needed by Job 1, provided by Manufacturer 7,
(62) = operation processing time

4.2 Problem definition

This chapter concerns job shop scheduling for instances with multiple routings' associations. Job shop scheduling is the problem associated with scheduling of n jobs on a set of m machines, where individual jobs contain equal operations' numbers to be processed via unique machine routes. The job shop scheduling problem may be formalised as an instance of a problem P = (M, O, J) in the scheduling of a job shop, which is made of: a set of machines (M machines); O set of operations each having an association with a machine over a given duration of time; J set of jobs with one occurrence ability for each operation; some schedule S for P which gives an assignment to all operations; some starting time T(o) where T(o) is greater or equal to 0. The processing time for each operation is P (o) = T (o) + d (o) with some precedent constraint (Bai and Tang, 2013). The overall aim is to minimise the total completion time of all jobs; this is also called the makespan (Cmax). The mathematical parameters for the job shop scheduling problems are defined as follows:

n	total number of jobs
m	total number of machines (manufacturers)
o_{ji}	operation ith required by job j
t_i	processing time of operation i
c_j	completion time of job j
C_{max}	completion time of last operation of all jobs

Table 4.1 shows the original process plan, where in each row (each job) the sequence of operations is fixed and the order of operations cannot be changed. In contrast, when all operations presented in Table 4.1 are rearranged based on the machine number, the outcome would be the sequence of operations, which could be processed in any order for each row (each machine). In other words, each job must be processed in the predefined order, while

each machine or manufacturer is free to process its operations in any preferable order that satisfies its own objective. This notion is illustrated in Table 4.2.

Various techniques and approaches to solving these problems have been devised previously, just as there are many pieces of research directed along this line (Owiti et al, 2014). This research will mainly focus on the multi-agent solution approach to the job shops scheduling problem. The multi-agent system will be a system that comprises of agents who are autonomous entities with the ability to cooperate with each other in order to fulfil a common goal. Using a multi-agent system will enable the decentralised scheduling of tasks within the manufacturing supply chain. Case studies of scheduling problems were used to test the proposed scheduling algorithm.

Table 4.2 MACHINE PROCESS PLANS FOR ABZ6 10 X 10 PROBLEM

	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
M0										
M1	31 (45)	91 (95)	101 (77)	61 (86)	21 (93)	51 (30)	41 (35)	81 (81)	11 (66)	71 (69)
M2										
M3										
M4										
M5										
M6										
M7	17 (62)	37 (46)	77 (57)	57 (25)	107 (34)	97 (66)	27 (56)	67 (65)	87 (30)	47 (85)
M8										
M9										

O1 = first operation, 17 = needed by Job 1, provided by Manufacturer 7,
(62) = operation processing time

4.3 Preliminary model

As mentioned previously, there exists various methods and models to solve job shop scheduling problems, such as heuristics or metaheuristic methods. The aim here is to develop an agent-based model so it can demonstrate the feasibility of having autonomous collaborating agents within a manufacturing supply chain. However, an optimisation technique such as genetic algorithm, particle swarm optimisation or bee algorithm is needed due to the nature of the problem. In recent years, to optimize the traditionally used classical dispatching and greedy heuristic algorithms, metaheuristics including simulated annealing, tabu search and genetic algorithm were implemented, causing improvement on the overall outcome (Pezzella et al, 2008). GA has been applied successfully in many scheduling problems, in particular job shop scheduling (Hasan et al, 2009). The superiority of GA over alternative local search algorithms is found to be the result of its overarching ability in the simultaneous adoption of multiple strategies for finding individuals to add to the mating pool. This will function during both phases of the initial population and dynamic generation respectively; and it allows investigation in a higher variability of search space at each step of an algorithm. Therefore, the GA was chosen so the work could be comparable to other studies and also because the algorithm could be optimised where necessary. In order to develop an agent-based scheduling model, an understanding of the problem and the existing methodologies to solve that problem is essential. Therefore, a preliminary platform was developed based on the work of Kobayashi et al (1995); Zalzala and Fleming (1997); Sharma and Yao (2004). A genetic algorithm was used as the evolutionary algorithm to build the model and also a preference list based representation (job sequence matrix) was used for solution representation in order to encode the JSSP for the GA. Based on the aforementioned problem definition, there are n jobs to be processed on m machines. Each job consists of m

operations (i.e. number of operations is equal to number of machines). Each operation must be done according to a predefined sequence of machines and the allocated processing time; which means that each operation of a particular job must be done in a fixed order of machines one by one.

The matrix J which is an n-by-m size, represents machine sequences for each job where i^{th} row is the required sequence of machines for i^{th} job. Matrix J in the GUI input captures the sequence of machines. For example:

2 3 1 4

4 3 2 1

4 1 2 3

1 4 3 2

means that for job 1 the machine sequence is as follows:

2 3 1 4

Therefore, job 1 must be processed on machine 2 (1st operation) then on machine 3 (2nd operation) then on machine 1 (3rd operation) and finally on machine 4 (4th operation).

The same rule applies for the rest of the jobs:

job 2 list: 4 3 2 1

job 3 list: 4 1 2 3

job 4 list: 1 4 3 2

The matrix P which is again an n-by-m size, represents processing times of each operation accordingly. Matrix P in the GUI input captures processing time.

4.3.1 Schedule coding with job sequence matrix

The active schedules are considered for constructing the preliminary model. In order to encode a schedule for the GA, job sequence matrix M is used. Its size is m-by-n and i^{th}-row of matrix m represents jobs' sequences on machine i^{th}. For example:

1 2 3

3 1 2

2 1 3

means that the first machine must process the first job, then after that the second job and then the third job. Or for instance, 1 2 in the first row means "some operation of job2 must be processed after some operation of job1". Or 1 3 in the 3rd row means "some operation of job3 must be processed after some operation of job1".

In the developed Java platform, the method makespan is designed to calculate the total time of the schedule (makespan) from given matrix M and it works in the following way:

In the makespan method in the while loop, there are counters in rows of matrix J and counters in rows of matrix M. Accordingly, the process moves the counters from left to right and builds the schedule; while it also moves from row to row of matrix M in a downward direction repeatedly, until all the schedules are built or it encounters a deadlock.

For example, consider the following:

Matrix M:

M1 1 2 3

M2 3 1 2

M3 2 1 3

Matrix J :

J1 1 2 3

J2 1 3 2

J3 2 1 3

Let's start with the 1st element of the first row:

M1 1 2 3

M2 3 1 2

M3 2 1 3

The first element in matrix M indicates job 1. Now considering Matrix J, the first job is represented as: J1 1 2 3. As the two operations match then it is possible to build the schedule as follows:

Figure 4.1 Preliminary model for schedule creation- Job 1 Operation 1

The compiler proceeds with the same logic:

M1 ~~1~~ 2 3

M2 3 1 2

M3 2 1 3

J1 ~~1~~ 2 3

J2 1 3 2

J3 2 1 3

Therefore, the next row of matrix M is considered:

M2 3 1 2

The first element indicates the 3rd job, which is equivalent to: J3 2 1 3 in the matrix J. It starts with machine number 2 (J3 2 1 3), therefore the schedule could be built as follows:

Figure 4.2 Preliminary model for schedule creation- Job 3 Operation 1

The complier processes this section and moves to the next row:

M1 ~~1~~ 2 3

M2 ~~3~~ 1 2

M3 2 1 3

J1 ~~1~~ 2 3

J2 1 3 2

J3 ~~2~~ 1 3

The compiler moves to the 3rd row: M3 2 1 3. It indicates job 2 which is represented in matrix J as : J2 1 3 2. However, because it needs to be processed at machine number 1 and this doesn't match the machine sequence (machine number 3), the compiler does nothing at this stage and moves to the first row again: M1 ~~1~~ 2 3. The sequence designates job 2 which is represented in Matrix J as: J2 1 3 2. As the two numbers match, the platform starts to update the schedule as follows:

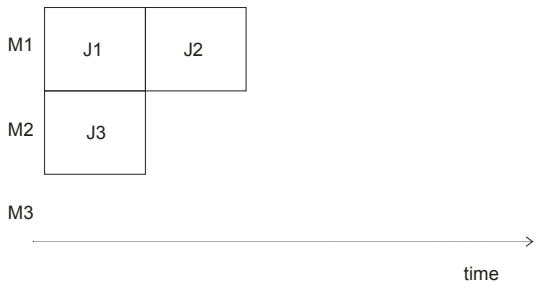

Figure 4.3 Preliminary model for schedule creation- Job 2 Operation 1

The completion of the model in Figure 4.3 will lead to:

M1 ~~1~~ ~~2~~ 3

M2 ~~3~~ 1 2

M3 2 1 3

J1 ~~1~~ 2 3

J2 ~~1~~ 3 2

J3 ~~2~~ 1 3

and it continues until all numbers in matrices J and M are processed. It is important to note that a particular operation of a job can only start just after its precedence in the process plan. This constraint is illustrated in Figure 4.4 as follows:

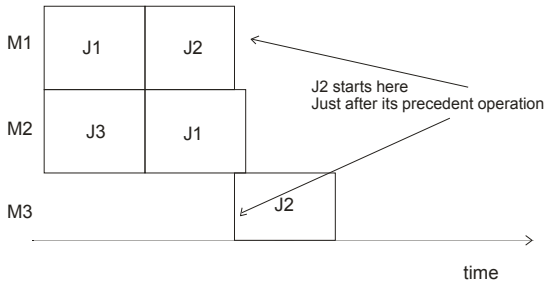

Figure 4.4 Preliminary model for schedule creation- Process plan constraint

4.3.2 Genetic algorithm

Each matrix M represents each individual. All individuals stored in a 3D array called P. Indexes in P are formulated as P[individual_index][M_row_index][M_column_index] with ranges: P[0..population_size-1][0..m-1][0..n-1]. The individuals are selected to crossover

with random_indexes method that return random indexes in ii array. Indexes in ii array can be repeated and the indexes with higher fitness values repeated more frequently.

Therefore the individuals are presented for crossover as follows:

P[ii[0]][..][..]

P[ii[1]][..][..]

…

P[ii[population_size-1]][..][..]

The fitness is inverse makespan for all feasible schedules:

fitness=1/makespan

For unfeasible schedules it is:

fitness=1/maximal_possible_makespan

Where the maximal_possible_makespan is sum of all elements of matrix P, which is the worst case, estimated as is when all jobs made on one machine one by another.

In the crossover operator, two parents randomly exchange their rows in P[ii[individual_index]][..][..] matrices.

For example, two parents are considered as follows:

Parent 1:

M1 1 2 3 4

M2 4 1 2 1

M3 2 1 3 4

M4 2 1 4 3

Parent 2:

M1 4 1 2 3

M2 4 3 1 2

M3 2 1 3 4

M4 1 2 4 3

Random numbers are chosen to select which rows are to be used for separation between two children. Each two parents always make two children. For instance, child 1 can be randomly chosen to consist of 1 row of 1 parent or 2 row of 2 parents or 3 row of 1 parent or 4 row of 2 parents etc.

The outcome would be two children as follows:

Child 1:

M1 1 2 3 4

M2 4 3 1 2

M3 2 1 3 4

M4 1 2 4 3

Child 2:

M1 4 1 2 3

M2 4 1 2 1

M3 2 1 3 4

M4 2 1 4 3

This crossover keeps permutations in row because it works row-wise.

Three mutation operators are applied in the platform:

- Exchange mutation: which swap two random elements in a random row of matrix M:

Before:

M1 1 2 3

M2 3 1 2

M3 2 1 3

After:

M1 3 2 1

M2 3 1 2

M3 2 1 3

- Exchange as general mutation: which split matrix M with vertical line and swap 2 pieces. The vertical line is selected in a random place.

Before:

M1 1 2 3

M2 3 1 2

M3 2 1 3

After:

M1 2 3 1

M2 1 2 3

M3 1 3 2

- Mirror mutation: which makes vertical mirror transform

Before:

M1 1 2 3

M2 3 1 2

M3 2 1 3

After:

M1 3 2 1

M2 2 1 3

M3 3 1 2

Crossover and mutation are executed one by another in repeats in a loop called generations. There is a permanent elitism concept applied in the platform. The elitism in a genetic algorithm is the option to increase the chance of finding better solutions in the least number of generations. The elitism happens when the best individuals are kept unchanged to the next generation. The simulation platform and its associated Gantt chart is presented in Figure 4.5.

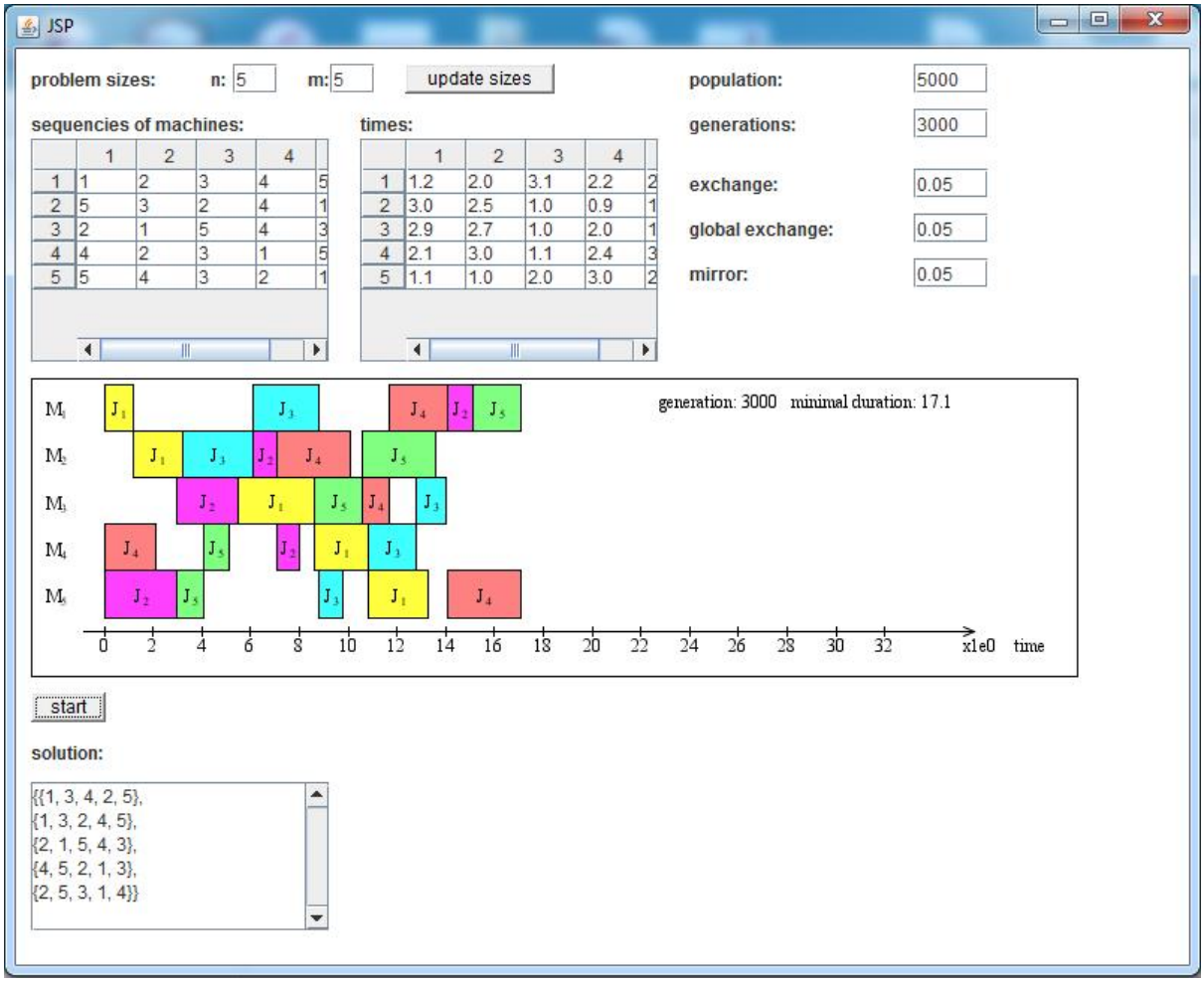

Figure 4.5 Platform GUI for a random 5 x 5 JSS case study

The aim here is not to develop a preliminary platform for solving the JSSP, however, by creating this model and replicating operators of a classical GA in this prototype, the author would understand the basics of solving combinatorial optimisation problems; and hence, this could help with the development of an agent-based model in the following sections.

4.4 Agent-based decentralised scheduling approach

In this section, a new hybrid method is proposed based on a GA and MAS to solve the JSS problem. Since the JSS is a NP-complete problem, there is no exact way to find an optimized

solution and for the same reason, it is a suitable candidate for applying a GA algorithm. If the problem is encoded properly using a GA, then it would be much easier to parallelize computations by distributing it over several computational nodes in a network. Now suppose that instead of a simple dummy computational node, which simply runs a GA algorithm and evolves the generations, an autonomous independent agent is used (as a portion of a multi-agent framework) that is able to apply different techniques to improve its final makespan. In addition, these agents will cooperate to achieve the best result. By putting several agents together, a new multi-agent framework for solving the JSS problem is introduced.

4.4.1 Overview of the multi-agent scheduling platform

The primary solution is in the form of a number of (n) jobs, consisting of numerous consecutive operations performed in a particular sequence which must be followed precisely. Every operation is scheduled to be applied to (m) machines, each with a specific duration of performance. There can only be one specific operation of a job applied to a machine at a time with no breaks, which means that switching or stopping the application of a task is not possible once a supposed task ‘O’ has started its operation. Each job is considered completed when all its tasks are accomplished.

In this project, we aimed to achieve the most efficient solution to reach the minimum possible makespan with the help of genetic algorithms. Considering the multi-agent tasking capability of job shop scheduling problems, the context of multi-agent systems can be of great help to maintain efficiency in a system by distributing the operations between different agents and hence reducing the overall duration of operations. (In other words, we simply distribute the

workload among multiple agents across a network to reduce processing time). Another advantage to the multi-agent tasking is creating the opportunity to use a kind of divide and conquer technique, in order to breakdown the global problem into some smaller problems and solve each of them separately with an independent agent, and finally collect and merge the outcome data from all agents to form the final solution. This can be achieved by using the genetics algorithm to allow simultaneous data processing with the breaking down of the initial population to a number of sub-populations, for further distribution between agents. This distributed agent-based method will improve the search diversification and also avoid the premature convergence. Each agent will independently apply the specific algorithm on its sub-population simultaneously to the other agents. The optimum solution is either selected as a result of data synchronization within the agents, or by considering an alternative agent to indicate the optimum solution from the rest of the results.

The development of a multi-agent system requires convenient underpinning knowledge and tools. In this project Java programming language and JADE framework are nominated, considering the object-oriented nature of Java which makes the ideal choice for the development of simulation and academic applications. There are advantages with Java, including its high speed of development and its comprehensive tools and the numerous libraries it offers; whilst avoiding unnecessary complexities in alternative programming languages such as C++ (allocation and management of memory, or the pointer handling in C and C++ programming languages, which are more suitable for functional environments focusing on ‘performance’ rather than the simplicity, readability and speed of development and variations for which this project aims).

The JADE framework is an open-source academic implementation of FIPA standard, fully written in Java language, which enables fast and easy development of a Java-friendly multi-agent system leading to the production of an ontology of the intelligent agents and their correlation with each other. For storing the intermediate results for further analysis (graphs of data), this project has selected a Java-friendly relational database management system called Derby which was accompanied by a reporting tool named JasperReports for the production of the graphical content of the project. Finally, the required IDE of this project was Eclipse. All the stated tools and software are available to download free for Linux and Windows systems.

4.4.2 Chromosome representation

The first step in using the genetic algorithms is to formulate the problem in the first instance. In other words, the chromosomes must be represented in a simple and flexible form to perform the operations. One of the simplest solutions is as follows. If the problem had N jobs each with M operations, each chromosome is represented as a sequence of N x M characters of numbers between 1 to M (a sequence of genes). The first M in the sequence is indicative of the first operation of a job M, likewise, the n^{th} character in the sequence represents the n^{th} operation of job M. For instance, considering a sample problem with three jobs each consisting of three operations, one suggested chromosome could be 132213123.

The first number indicates the first operation of job 1, similarly, the second number (3) represents the first operation of job 3 and finally, the last number (3) represents the third operation of job 3. Therefore, for N jobs consisting of M operations, a sequence of N x M of numbers ranging from 1 to M (Chromosome Genes) with N repetitions, indicate a valid chromosome enabling the calculation and analysis of the makespan. This allows the

accomplishment of the most efficient sequence (minimum makespan) which was aimed for at the beginning.

This method is advantageous in several ways. The chromosomes are represented as a simple string in Java language where each character is indicative of one gene (or in the form of a character array where each cell is in fact, one gene of the chromosome). This leads to the possibility of serialization for exchange between the agents within the JADE framework. Furthermore, the required operations for genetic algorithms such as cross over and mutation of the chromosomes and also the validation of each chromosome after the application of changes are applied with more simplicity and rapidity. This is of great importance in reducing the overall time of operation in each simulation.

It should also be noted that this solution carries some disadvantages considering that it is limited for problems with less than ten operations for each job. This matter is taken into account in this project, which limits the number of operations to ten. Furthermore, the starting index of the operations is 1 instead of 0. The other fundamental limitation of this solution might be its inability of direct application in the problem. There is a need for primary calculations and analysis of the overall duration of operations by summing the time of operation of each machine.

In this project, the stated cons of this method are negligible, considering the fact that its advantages fulfil our requirements and purpose by reducing the operation time whilst adding simplicity to the codes of programming and maintaining better understanding of the codes. The previously stated genetic algorithm operates in tasks which are directly relative to the

project's proposed multi-agent system. Therefore, in the following section, key agents will be introduced and their operation will be briefly discussed.

4.4.3 Multi-agent architecture

The programme will begin by creating a primary population of the chromosomes by the order agent. This agent will then break the primary population down to a specific number of sub-populations and distributes them between resource agents. Resource agents are responsible for providing the main sources of processing which will help with resolving the problem. Each of the resource agents will independently operate to optimize their sub-populations. For this cause, the key parameters of the genetic algorithm including the cross-over rate and mutation rate will be communicated to each resource agent as a constant identical value. Unlike the job sequence matrix representation, which was presented in the preliminary model, coding chromosomes for the agent-based model is modified to be an operation-based representation (Tsujimura and Gen, 1996), as it was described in the previous section. This is due to the simplicity of the coding when using this method and also it is more convenient when applying the problem constraints.

The concept of product, resource, order, and staff architecture was used as a foundation for the development of the platform. Creating the initial population, distributing it among other agents and controlling the overall process is carried out by an order agent. Each resource agent may use a different method to evolve its sub-population and optimize its chromosomes and its final makespan. After pre-defined generations of evolving, each agent will stop for a while to optimize its achievement up to that moment with the help of another agent, which is

an optimizer agent; and will leverage some of optimization techniques that are described in the literature (Hasan et al, 2009; Asadzadeh and Zamanifar, 2010; Kurdi, 2016) to promote the final makespan for the best chromosomes of the resource agent. After trying optimization, the resource agent will exchange its worst chromosomes with its neighbouring agents, hoping these low-quality chromosomes may find a chance to promote and survive in a different environment (e.g. with applying a different algorithm). This method is proposed and completely described in the work of (Kurdi, 2016). In order to accomplish this chromosome migration among agents, there is a need for a synchronization mechanism which is applied and controlled by a synchronizer agent. The process will be finalized by the synchronizer agent responsible for the synchronization of the agents along with an analysis of the results produced by each agent. The synchronizer will then conduct a comparative analysis between all the produced results and identifies the best chromosome as the best solution. This will be the chromosome with the least makespan that will simplify the identification of the operating sequence of the agents for different jobs and their associated machines. Finally, the synchronizer agent will store the results from each simulation in a Derby database for further sensitivity analysis and for sketching the graphs. The sequence of activities taking place is demonstrated in Figure 4.6.

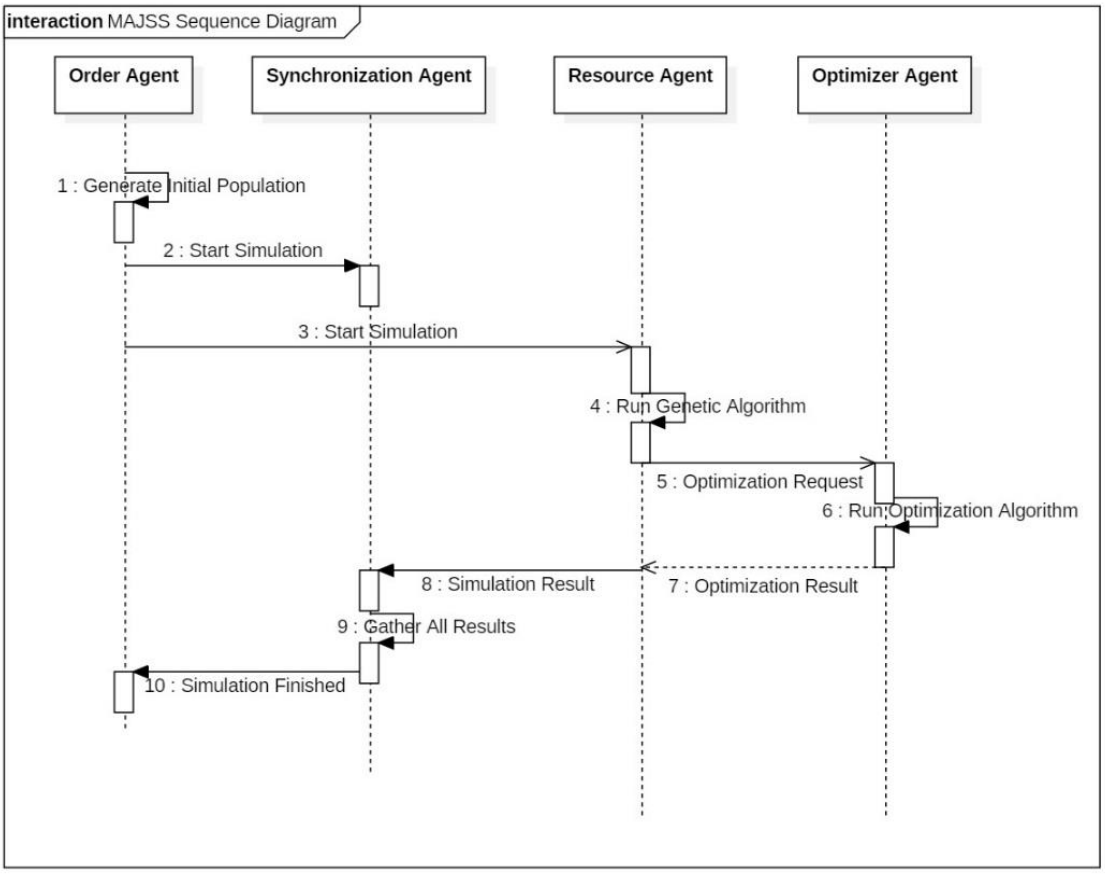

Figure 4.6 Sequence diagram of interactions between agents in UML

In order to run the proposed multi-agent system stated in this project, JADE will be used as a Java library (framework) operating on a Java virtual machine (JVM). Every typical MAS simulation in JADE consists of a main container consisting of any arbitrary number of containers, each operating on a separate machine and can interact with each other via a network connection. Each container holds an arbitrary number of agents which are capable of communicating with the other agents within their own container or other containers of a system.

All the phases of creation, management, message transfer and other operations are controlled and managed by the JADE-based multi-agent system framework, leading to improved productions by the programmers. The directory facilitator ‘DF’ is a context of JADE available in public frameworks which enables the agents to offer registry services. This allows the agents to search for their desired service through this registry and access the address of the service provider for communication purposes.

The multi-agent system proposed in this project contains a number of agents as follows:

4.4.3.1 Order agent

This agent is responsible for receiving the necessary primary information and settings for a simulation process of a multi-agent system. The data relative to the JSS problem (list of jobs and duration of each operation and associated machines) is loaded from the entry file to create the primary population of chromosomes respectively. The primary population is then broken down into identical sub-populations with convenient sizes to distribute them to the resource agents. The order agent will then collect the optimum outcomes from the resource agents, store the results from each simulation process and identify the optimum result from the final solutions. For this purpose, a graphical user interface (GUI) was created to assist with data collection and display. As this agent has only a supervisory role, it must be added to the main container prior to other agents. Also it can be run on a machine with lower processing power because it is not participating in the optimisation algorithms.

The behaviours of this agent are as follows:

- CreateInitialPopulationBehaviour: this behaviour is a subclass of OneShotBehavior in JADE framework. The agent will only operate once to identify the primary population of the chromosomes.
- Check4ResourceAgentsBehavior: this is a subclass of JADE's TickerBehaviour and operates frequently to distribute the sub populations to the accessible resource agents. The resource agents can join the process at any random time; therefore, this behaviour will ensure that each resource agent will receive its sub population immediately after registering in the DF and will be ready to operate instantly. The number of resource agents is dependant to the size of the sub populations. For instance, if a population of 1000 chromosomes is broken down into a sub population of 100 chromosomes, 10 resource agents will be required. Any extra resource agent will be left without work since this behaviour will be stopped immediately after identifying the number of required resource agents.
- Check4OptimizerAgentsBehavior: this is another TickerBehaviour which will evaluate the optimizer agents' registration frequently and checks for new agents to provide them with their required information regarding the job to which they are entitled. (Each optimizer agent will require a comprehensive statement of the problem prior to the start of its operation). Like the previous behaviour, this behaviour will stop after identifying the required number of optimizer agents and transferring the necessary information to them.

4.4.3.2 Synchronization agent

This agent is mainly responsible for the coordination of the resource agents. Each resource agent will switch to a waiting phase after the completion of its first round of the genetic

algorithm on its subpopulation. It will then send a message to confirm the end of its operation with its best chromosome to the synchronization agent. The synchronization agent announces the start of the chromosome exchange phase to all the agents upon receiving confirmation from all of them. As mentioned previously, this allows the exchange of the worst chromosomes within the agents, to provide them with the chance of improvement and survival in another agent, leading to improved results. Storing the outcome of each round of simulation in a database and restarting the simulation is accomplished by the synchronization agent as well.

4.4.3.3 Resource agent

This agent is the main processing unit in the proposed method of this project. A simulation system can hold one or more of these agents (the minimum number required for solving a problem is one, however in the proposed method, which is based on the resource agents exchange of chromosomes, the minimum required number of resource agents is two). Each resource agent can use a variety of techniques in identifying the optimum results. In this project, the focus is solely on genetic algorithms; however, the results can also be optimized using alternative techniques rather than genetic algorithms, which sets the foundation for further research in this matter.

The behaviours of this agent are as follows:

- GABehavior: this is a subclass of CyclicBehaviour in the JADE framework, which will apply the genetic algorithm on the primary sub population for a specific number of generations. In fact, this behaviour controls the evolving of generations and

manages the production of new chromosomes, along with the adding and exchanging of them in a primary subpopulation.

- OptimizeFinalResultBehavior: this subclass of OneShotBehavior in the JADE framework will start its work immediately after the evolution process is completed by the GABehavior and it is responsible for transferring a number of the best chromosomes of the current subpopulation to the optimizer agent for final optimization. This behaviour will only succeed if the optimizer agent is found in the system and if it is capable of optimizing the transferred chromosomes.
- SendCompletionNotificationBehaviour: this OneShotBehavior will announce the completion of the operation to the synchronization agent after the completion of all the genetic algorithm and the probable optimization. It will also identify and transfer the best chromosome to the synchronizer.

4.4.3.4 Optimizer agent

This agent is responsible for optimizing the outcome results of the genetic algorithm using a variety of techniques. In this project, four optimisation techniques from the literature, namely partial reordering (PR) (Hasan et al, 2009), inversion mutation (INV), insertion mutation (INS) and reciprocal exchange mutation (REX) (Kurdi, 2016) are used; however, other techniques can also be considered for future projects. The PR method is based on the possibility of exchanging the sequence of operations in job shop scheduling problems to improve the final makespan. This will only be possible if all other aspects of the problem remain valid after the exchange. In this method the bottleneck machine is identified at the first stage. Then the optimizer will realise which job is the last job to be processed on that machine. Consequently, it will shift the first operation of that job to the beginning of the

corresponding machine associated for the first operation. Therefore, the technique will result in a reduction in the operation's gap for the other machines, hence improving the overall duration of the task.

The inversion, insertion and reciprocal exchange mutation are three mutation operators, which have shown superior results, compared to a random mutation method. Therefore, these are selected for the main optimizer agents. However as mentioned previously, due to the distributed nature of the system, several techniques can be integrated to the system at any time for performance improvement.

4.4.3.5 Multi-agent ontology

In this section, the ontology of the proposed multi-agent system will be explained in detail. In this project, the JADE framework has been used for the formation of an ontology consisting of a variety of sections, each in the form of a class in Java.

- Machine: this class is the implementation of the machine concept in the JSS which includes all the supporting methods relative to a machine in a problem; such as the method for the identification of the current operating task at any time.
- Job: this class is the implementation of the job concept in the JSS which includes all the supporting methods relative to jobs in the problem; such as the methods of scheduling, operating time and other methods of identifying jobs and their corresponding machines.

- Operation: this class implements the concept of the operation of each job in the JSS; which consists of the related methods of operation in the problem, such as the specification of the required job and its collaborative machine.
- Chromosome: this class is in charge of the chromosome concept in the JSS including the methods, which includes a list of chromosomes, supporting methods for verification of feasible chromosomes, cross over, mutation etc.
- SetupJSS: this is a class for the AgentAction which is in charge of the data transfer between the agents of a system (JADE ontology's action). As mentioned previously, the optimizer agents require a comprehensive data set of the problem to help them with a better understanding of the problem. For this purpose, the order agent will transfer all the relative information regarding the list of jobs along with their operations and specific duration to the agent. The optimizer agent will receive the information and use it in its internal processes by using a method called ServeSetupJSSRequest.
- BeginProcessing: this is a subclass of the JADE ontology's action, which is sent to each resource agent by the order agent for the start of the process operation. The resource agent will receive the notice by the serveBeginProcessingRequest method and begins the operation.
- FinishProcessing: this AgentAction class is responsible for announcing the completion of an operation from the resource agent to the synchronization agent. The synchronizer will receive the message by serveFinishProcessingRequest and follow the next step of the procedure.
- BeginOptimization: this AgentAction is used by the resource agent to send the results of a genetic algorithm to the optimizer agent to optimize the results. Each optimizer

agent will receive the announcement by serveBeginOptimizationRequest method and start the optimization process.

- FinishOptimization: this AgentAction will help the optimizer agent to announce the completion of its optimization process to the resource agent which requested the optimization. The resource agent will use serveFinishOptimizationRequest method to receive this message and follow the next step.
- BeginMigration: this AgentAction class is used by the synchronization agent to announce the completion of a round of operation from the resource agents to order the start of the chromosome exchange phase within resource agents. Each resource agent will receive the notice by the serveBeginMigrationRequest method and start exchanging its chromosome with the neighbouring agents.

4.4.3.6 Simulation platform

As mentioned previously, the simulation platform is developed using the JADE environment. The GUI of the simulation contains some parameters both for the MAS and for the GA algorithm. The parameters for the GA algorithm include population size, crossover rate, mutation rate and generation span. The MAS parameters include migration frequency, migration rate, total resource agents, total optimization agents and simulation repeats.

The first step in running the simulation is to define a case study problem. This is done by loading a scheduling case study instance into the system via a text file. The figure below shows an example of such an instance.

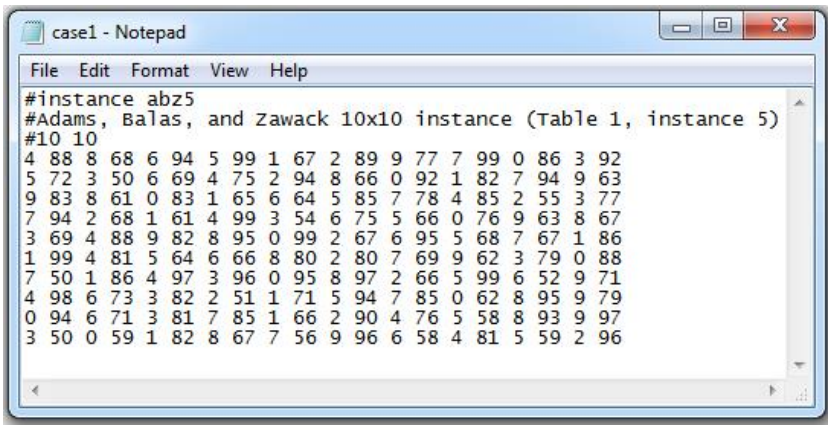

Figure 4.7 ABZ5 scheduling instance as an input for the simulation

The second step is to define the GA parameters such as population size, crossover rate and mutation rate. After defining the GA parameters, there is a need to instantiate the required agents. The simulation starts the process as soon as all the resource agents and optimizer agents are defined. There is an ad hoc feature added into the system which helps for the visualisation and analysis of the results. The results for each run of the simulation are imported into a relational database and it could be queried by a Derby management system in order to draw the total makespan versus the generation's graph. Figure 4.8 shows the GUI of the simulation software. Figure 4.9 demonstrates all the participating agents in the JADE environment in a sample run of the simulation, with three resource agents and one optimizer agent. Figure 4.10 shows the sequence of messages for a random resource agent during the simulation; and finally, Figure 4.11 illustrates the results for one run of the ABZ5 instance.

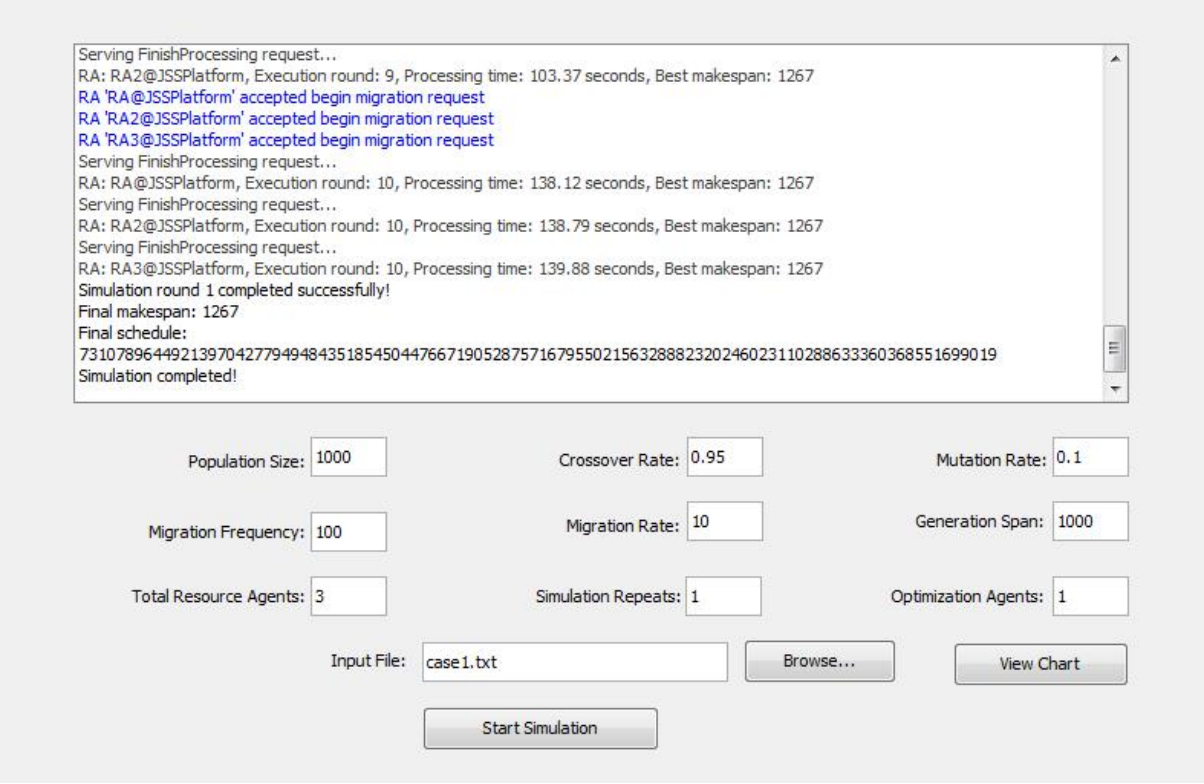

Figure 4.8 GUI of the simulation platform with its associated MAS and GA parameters

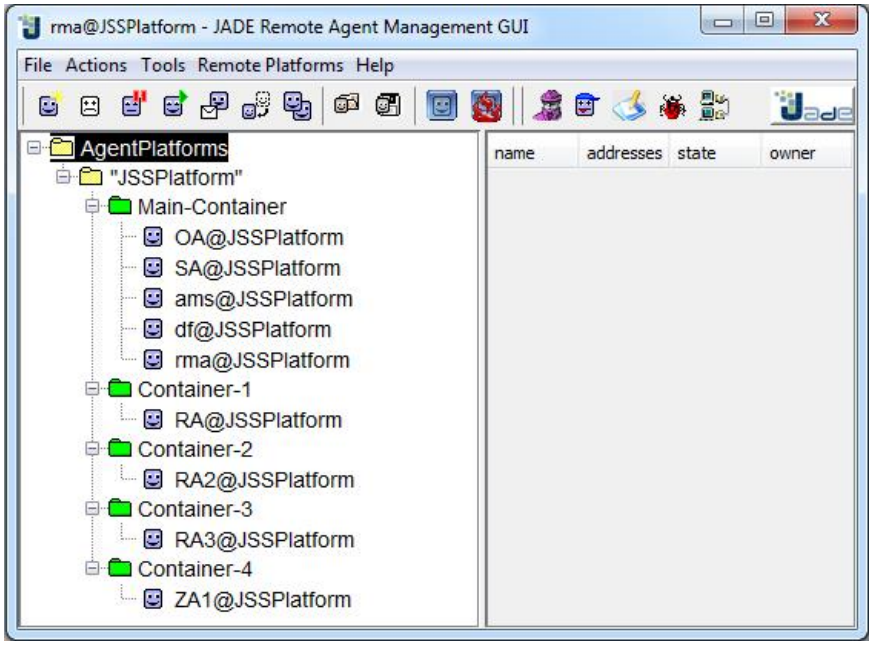

Figure 4.9 JADE environment for a run with three resources and one optimizer agent

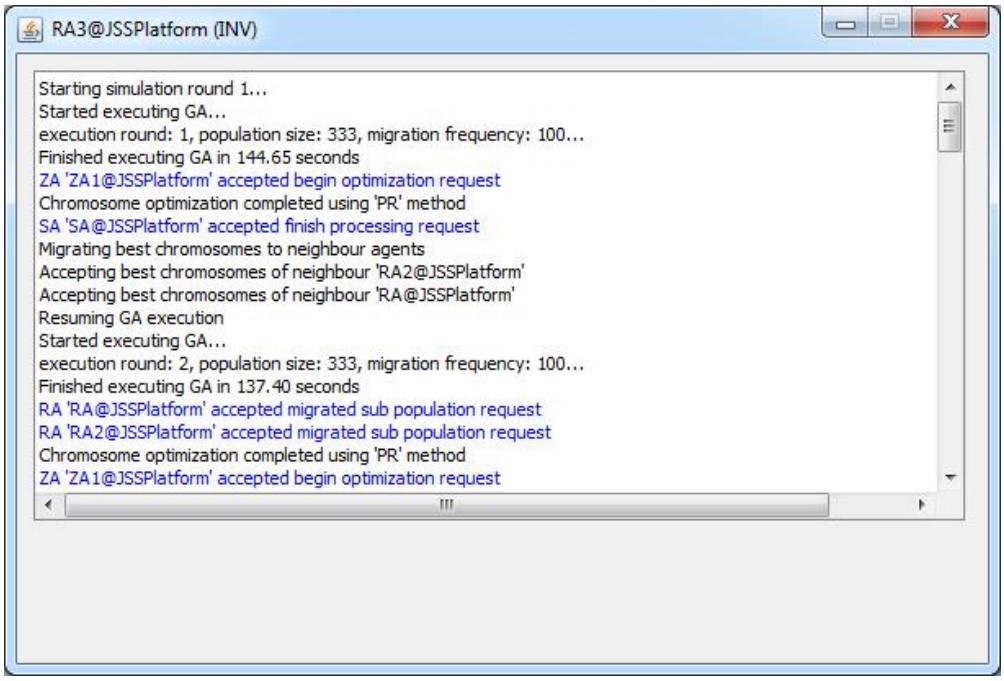

Figure 4.10 Sample sequence of messages for resource agent 3

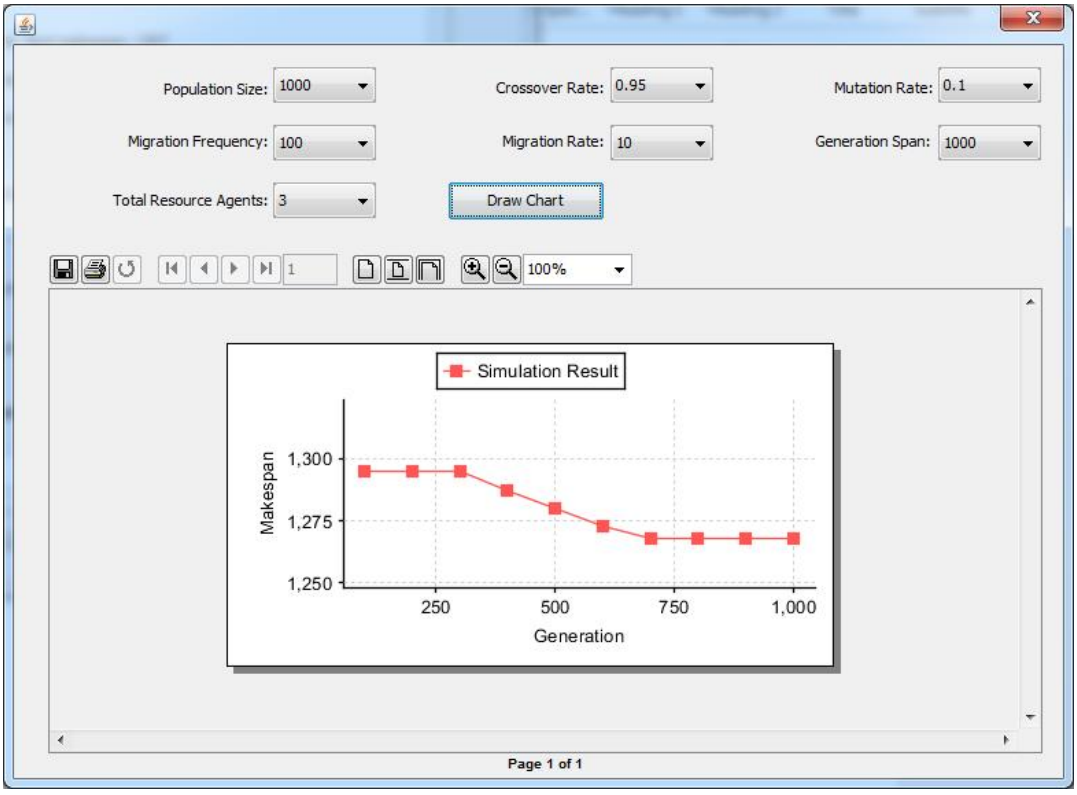

Figure 4.11 Makespan vs generation graph for one run on ABZ5 case study with five agents

4.5 Hybrid agent-based approach for distributed manufacturing scheduling

The fundamental need for self-sequencing operation plans in autonomous agents has led this research to the development of algorithms. These allow the resolution of combinatorial optimisation of a global schedule, which consists of the fixed process planned jobs; it also requires operations offered by manufacturers. The proposed agent-based approach is adapted from the bio-inspired metaheuristic - particle swarm optimisation (PSO) where agents move towards the schedule with the best global makespan. The main difference with the conventional PSO is that there is no velocity for movement, and in each iteration, all the agents adapt the best schedule plan and try to perform a genetic algorithm individually to minimize their own objective functions. This section has achieved a novel hybrid optimisation algorithm to allow agents to schedule operations whilst cutting down on the duration of the computational analysis as well as improving the performance extensibility amongst others. The multi-agent platform is built upon the Java Agent Development Environment (JADE) framework. The operation research case studies were used as benchmarks for the evaluation of the proposed model. The proposed approach shows a practical use case of a decentralised manufacturing system.

4.5.1 Solution representation

The proposed hybrid agent-based architecture consists of two main functions for solving job shop scheduling problems, namely: a multi-agent function, which is adapted from classical PSO with some modifications; and a genetic algorithm (GA) function for further solution optimisation. When solving a JSSP, the coding mechanism and solution representation are

extremely important regarding the efficiency and complexity of the platform. In general, PSO is suitable for linear and continuous search spaces; and in contrast, JSSP is a discontinuous problem with distinctive solutions. Therefore, an encoding method was needed that could transform the feasible schedules into manageable arrays of solutions both for the PSO and GA.

As explained in the previous section, solutions of an n jobs and m machines the JSSP problem can be represented with an array of $n\ x\ m$ real numbers. These arrays are similar to chromosomes in a GA, albeit they can be used for PSO operators. In this method, the solution space is represented with an $n\ x\ m$ array. For instance, for the ABZ6, which is a 10 x 10 problem, the solution schedules are arrays with 100 cells. Each job from Table 4.1 appears 10 times in the solution array. For example, number 4 appears in the array 10 times which represents job 4. Each occurring number 4 in the array represents the corresponding operation for that job in turn. To be more precise, the first number 4 represents the first operation of job 4, then the second number 4 in the array denotes the second operation of job 4, and this will continue 10 times to fulfil all operations of job number 4.

As explained previously, the sequence of operations is important and should be done with respect to the predefined process plan. For instance, job 2 must start with operation 25 and then operation 22 and then operation 28 and so on. Operations 25 and 22 are predecessors for operation 28 and therefore, operation 28 is not allowed to start before operations 22 and 25 are finished. The unique and simple aforementioned technique for solution representation will efficiently satisfy this sequence constraint. Figure 4.12 shows an example solution to a 3 x 3 problem.

3	1	2	1	3	2	3	2	1

Figure 4.12 Sample solution array for a 3 x 3 job shop scheduling problem

In the above example, 3 means job number 3 and the first number 3 represents the first operation of job 3. Consequently, the second 3 means the second operation of job 3, and the third number 3 means the last operation of job 3. The same logic is applied to the rest of the jobs.

4.5.2 Evolutionary algorithms for combinatorial optimization

The interaction protocol for the agent's behaviours is adapted from the particle swarm optimisation (PSO) algorithm for this research. The PSO is a computational search technique, which can be used for optimisation of continuous linear problems (Lin et al, 2010). It was developed by Eberhart and Kennedy (1995). Central to the working of the PSO is the concept of swarming theory; where organisms which are normally considered simple and not very intelligent individually, like ants, bees or termites, are able to group together or 'swarm' to collectively perform complex functions towards a common goal. This might be gathering foods or building sophisticated nest structures and filling them with food (Bonabeau et al, 2010). In doing this, they demonstrate behaviour and functions as a 'swarm', which is well above the ability of any individual organism within the group. These self-organised and decentralised problem-solving systems result in 'swarm intelligence', used to solve the complex problems that the organisms encounter daily. It provides flexibility and efficiency in adapting quickly to changes in the environment during the process of problem solving (Deepa and Senthilkumar, 2016). One of the key advantages of PSO is the fact that it is problem-independent, meaning that only the fitness function for the problem domain is needed. This makes this algorithm more scalable and flexible, compared to many other methods. As explained previously, the aim here is to develop a decentralised multi-agent scheduling system; and PSO is a reasonable candidate for this research, due to its swarm intelligence

nature. The collaboration of particles resembles the activities of agents in our simulation. Therefore, this algorithm is chosen as the foundation for the development of the multi-agent scheduling system. It is worth mentioning that PSO is an efficient algorithm for a global search, but when it comes to local search ability, it is a weak option. Therefore, some optimised search schemes should be integrated with PSO to produce an overall optimised system. In this research, the GA is integrated with PSO to form an optimised hybrid model in light of the above statement.

The PSO is based on a simulation of this social behaviour of individual agents in a swarm, such as a bird flock; they adapt to their environment using a combination of their own individual experience and the benefits of collective knowledge and information gathered from the swarm as a whole. This social sharing provides an evolutionary advantage for solving complex problems (Eberhart and Kennedy, 1995), such as an individual fish in a school swimming away from a threat not directly observed. Similar to this, the PSO algorithm is modelled towards the behaviour of a bird flock searching for food. The best condition is arrived at by the swarm through communication with members who have better information and others are informed so that they all move simultaneously towards the food (Rini et al, 2011). This process is repeated continuously until optimal conditions are reached or the food is found.

The algorithm starts by generating a random set of potential solutions (particles) which can fly through the search space, just like a bird (Lin et al, 2010), looking for optimal conditions and concentrating the search around beneficial areas. Each swarm particle *sp* iterates an optimum solution search based on its own best experience, global best experience of the

swarm, and using newly generated velocity. This is in addition to its previous position to determine the position change as shown in the equations (Lin et al, 2010) below:

Assuming X_{sp} is the position of the *spth* particle:

$$X_{sp} = X_{sp} + V_{sp} \quad (4.1)$$

$$V_{sp} = \omega \times V_{sp} + C_1 \times Rand(\ \ ) \times \left(X_{sp}^{best} - X_{sp}\right) + C_2 \times Rand(\ \ ) \times \left(X_{sg}^{best} - X_{sp}\right) \quad (4.2)$$

Where X_{sp} is the moving distance of the *spth* particle in one iteration; ω is defined as the inertial weight for one step movement distance for a particle; $C_1$is the self-learning factor; C_2 is the social learning factor; X_{sp}^{best}represents the personal best position of the *spth* particle and X_{sg}^{best}indicates the best global position.

4.5.3 Proposed decentralized scheduling architecture

The proposed multi agent based system consists of two main functions and an ontology at its foundation. During the initial phase, a set of agents are created and randomly distributed across the search space. Each one of these agents has a schedule solution represented by an array, as explained in the previous sections. After the initialisation phase, all agents evaluate their fitness values (makespan) in order to determine the *lbest* (local best, the best makespan so far for each agent) and the *gbest* (global best, the minimum makespan) in each iteration. The agents then try to move towards the best solution by updating their position and velocity as the method is adapted from the PSO.

One key factor to consider is that the PSO is originally developed for linear and continuous problems. However, the JSSP is a problem with discrete instances. Therefore, by mapping the schedule solutions into the proposed arrays and parsing them into individual agents, the system is modified and is compatible to perform the PSO's procedure on a discontinuous problem.

An ontology with scheduling concepts has been developed to cater for agents' communications. Agents extract machine and job information from the ontology for their reasoning and evaluation and also assert their operation plans into the ontology for data sharing. This knowledge representation facilitates automated data query, data extraction and agents' collaboration by holding domain information in a logic-based language. The proposed architecture is illustrated in Figure 4.13 below.

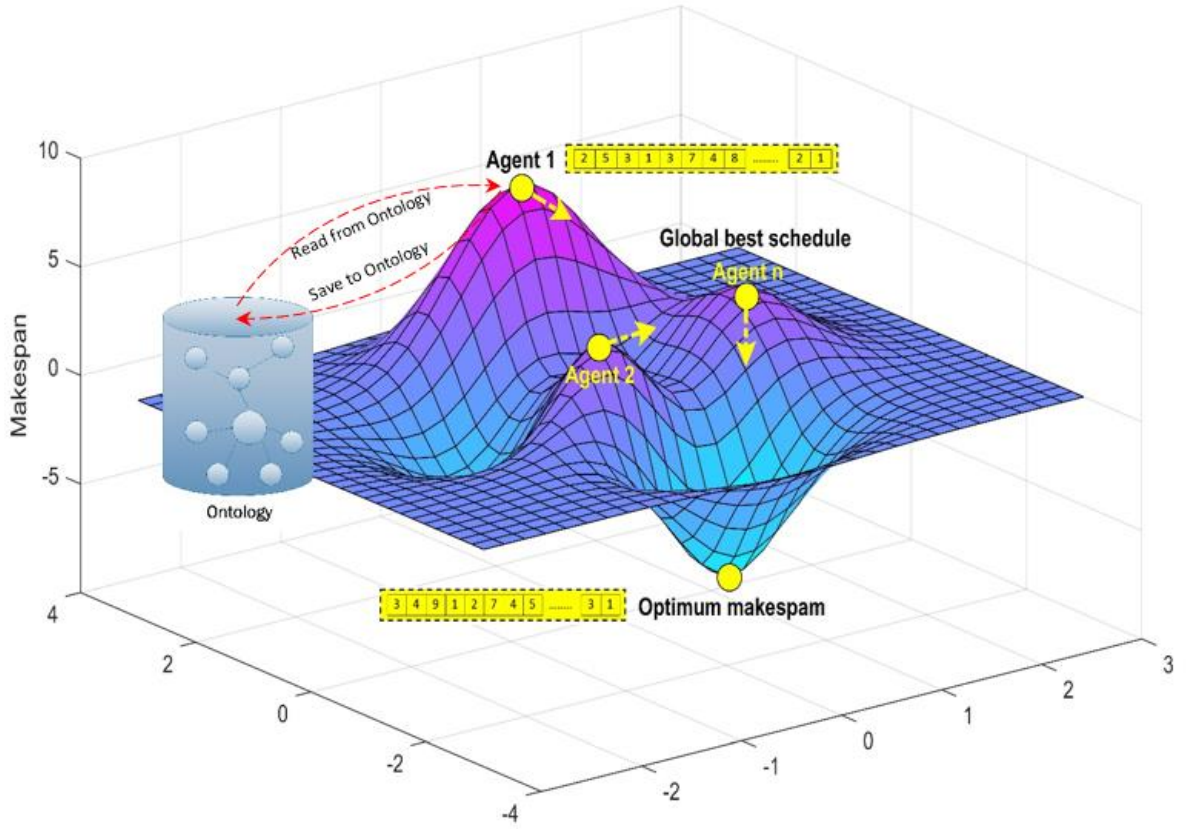

Figure 4.13 Proposed multi-agent system architecture

As shown in Figure 4.13, all agents are scattered in the search space at the initial phase. During each iteration, all agents evaluate their fitness functions to find their best makespan found so far. This is called the local best makespan for each agent. After the execution of each iteration, all agents share their knowledge about the local best makespan through the common ontology and the shortest makespan amongst all agents will be chosen as the global best makespan. As the global best solution is chosen, all agents move towards the global best and try to perform their fitness function evaluation again. This process continues until the optimum makespan is found, such as that presented at the bottom of Figure 4.13. The process can be terminated by the number of iterations or by a time-elapsed threshold.

The velocity of the agents is defined as the rate of change of permutation. As an example, if the solution in Figure 4.12 represents the global best for a problem instance, all agents will try to make themselves similar to that solution arrangement. For the ABZ6, where we have a 10 x 10 problem, the velocity will range from 1 to 100. A low velocity such as 1 will ask the moving agent towards the *gbest* to change the first cell of its array to the similar cell from the global best. In the case of Figure 4.12, the moving agent changes its first cell to number 3. By the same logic, velocity 2 means changing the first cell to number 3 and the second cell to number 1. As the agents move towards the best solution, the velocity increases and that means imitating more cells like that in the best global position.

For the purpose of this research, there are two possible settings for the movements of the agents towards the best solution:

1) Gradual approach of all agents towards the optimum solution, where position and velocity changes with respect to the *gbest* and velocity increases as agents get closer to the optimum.

2) Having zero velocity, where in each iteration all agents jump to the position of the best-known agent and then try to perform a GA to optimise the solutions and to find the new *gbest*.

The GA operator performs optimisation on both settings. The only difference is that in the latter case, when all agents are generated randomly and the best-known is calculated, all agents locate themselves in the position of the best-known solution and then they try to perform the GA individually. During the GA optimisation process, if an agent finds a better makespan, it will ask all agents to move to that position with a thread in the software; otherwise, it will go back to the previous mode and try to perform the GA on the previous generation. It is worth mentioning that agents are not moving in the space in the same manner as the classical PSO. When we are saying that agents are moving towards a better solution, it means that they try to change their arrays and make them more similar to the better solutions and the actual movement does not take place. Figure 4.14 shows the flow diagram of the sequence of operations.

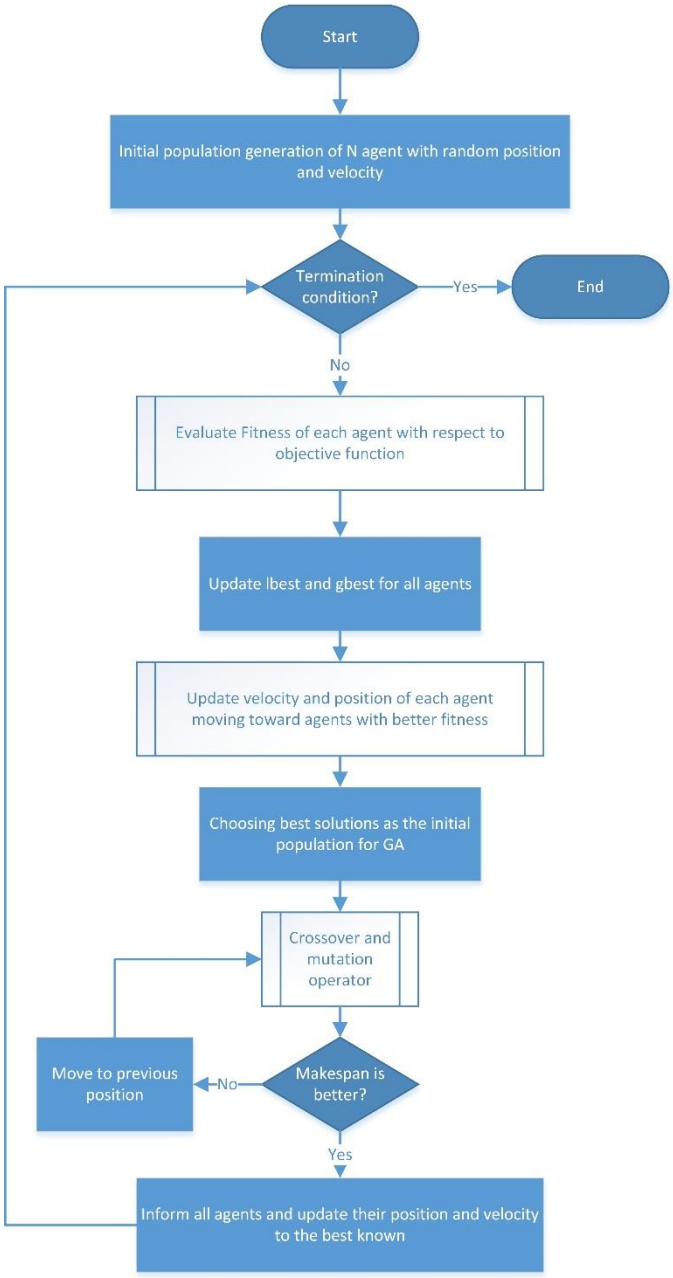

Figure 4.14 Flow diagram of the main functions

4.5.4 Ontology for intelligent manufacturing scheduling

The literature on the subject shows a trend towards the provision of developing globally accepted ontology based on semantics' agreements, in order to align the terminologies for interoperable applications. This is described as being the backbone for an integrated manufacturing system. The ontology gives the appropriate meaning to stored data through

constructing databases attached to standard terminologies. These can then be utilised to automate the process and create intelligent self-regulating systems. Such a semantic web is used for connectivity of machines working in a facility using the standard protocols.

Knowledge management includes three main components, i.e. acquisition, presentation and examination. Knowledge management in the manufacturing ontology domain deals with the rules and ontology defined for interoperability. After acquisition of knowledge regarding a product's design and manufacture through multiple sources of human induction and literature, the defined ontology is used to present the acquired knowledge in a standard manner. Ontology web language (OWL)/resource description framework (RDF) coding standards are used by the experts to deploy knowledge bases in semantic web rule language (SWRL). The creation and maintenance of ontologies is carried out through additional tools such as Protégé. Researchers have used Protégé to further create APIs (application programming interfaces) based on Java.

C.J. van Aart from the University of Amsterdam has introduced a novel concept, namely the Bean Generator, in order to implement a system which is capable of transforming the 'manual method' of creating ontology definition classes. These include schemas, along with predicates, concept classes and agent actions with an 'automated method' (Bellifemine et al, 2007). The Bean Generator minimises the time required to produce the Java files, showing an ontology compatible with the JADE toolkit. It does this by acting as a plug-in for Protégé, whilst producing JADE compliant ontologies and Protégé projects, along with the import and export ability of RDF and RDFS.

Using ontologies to store data enables data re-usability and maintenance, which is a massive

advantage as compared to normal databases; it also provides the opportunity to merge it with available ontologies for knowledge enhancement. This ontology alignment gives structure that is essential for compatibility with domain specific applications. This research made use of 'OWLSimpleJADEAbstractOntology.owl' in order to generate FIPA compliant ontology for the multi-agent platform JADE, which is compiled using the bean generator tool. The 'swrla.owl' and 'sqwrl.owl' ontologies enable realisation of the knowledge base through the use of semantic web rule language. Figure 4.15 shows the main structure of the aforementioned ontologies.

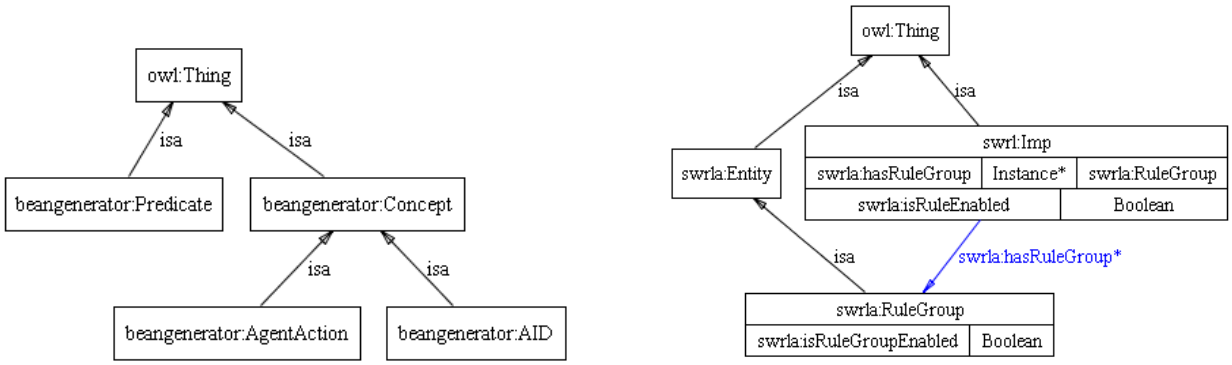

Figure 4.15 Bean Generator OWL ontology and SWRL ontology

Under the structure of the existing ontologies, the proposed ontology is developed. First, the sub-classes of the class 'Beangenerator:Concept' are identified using a top-down strategy. These abstract concepts are identified as *Products,* which contain product information such as process plans; *Resources,* which include properties such as machines; *Orders,* which contain order details such as quantity or due time; *Beangenerator:AID* which represents Agent ID and indicates the agent's details, such as its name and location in the network; and *Beangenerator:AgentAction,* which relates to the type of actions executed by agents in order

to change the state of the environment. Then the main concepts are specialised into more specific concepts making sure that the ontology is not overspecialised.

The proposed ontology is developed in a systematic and scalable way in order to be re-usable for further development and implementation in a real case study of a manufacturing network. For the scope of this research, only the operation research case studies are considered. However, scheduling data from the GFM Srl Company is extracted to further realise the application of the proposed ontology-based system on a real industrial case study for further research.

4.6 Results and discussion

The experimental results were obtained by executing the proposed multi-agent system platform based on 10 runs for each case study instance. The number of agents was set at 30; while the number of iterations was set at 100 and 5000 for the PSO and GA respectively. The results obtained from the proposed model were compared with benchmark instances from the operation research library (Lawrence, 1984). In order to evaluate the quality of the proposed model, two separate algorithms were chosen from the literature to act as the benchmark. The first algorithm is extracted from the research on the use of a GA for solving the JSSP (Hasan et al, 2009), and the second algorithm uses PSO optimisation for the same application (Lin et al, 2010).

Based on the extracted algorithms, three different models were developed and compared. The first model is designed to solve the JSSP by implementing a GA as the only algorithm. The second model is designed to solve the same problem, but only using PSO; and the third model is the proposed hybrid model. Ten runs for each model were executed and the results

were depicted as the average of all these tests. The optimum solution for the ABZ5 instance from the OR library is found to be 1234 (Applegate and Cook, 1991). The PSO solver achieved 1424.4; while the results for the GA and hybrid model were 1296.8 and 1240.5 respectively.

For the ABZ6 case study, the optimum solution was found to be 943 according to the literature (Applegate and Cook, 1991). Accordingly, the PSO algorithm showed 1088.9; the GA model got 988.8; and the hybrid model achieved 947.9 units of time. Although the results from the hybrid agent-based system are not the same as the best known from the literature, the proposed model obtained near optimum solutions and it also outperforms individual GA and PSO models. The performance of the PSO over the GA shows that the PSO algorithm, which is mainly developed to solve continuous problems, cannot perform as efficiently as the GA for solving a JSSP; which is a combinatorial problem with discrete solution space. The aim of this study is not to optimise the PSO algorithm, but to demonstrate the feasibility of implementing multi-agent systems for intelligent decentralised scheduling. However, there are relevant studies on the improvement of PSO for the JSSP in the literature (Lin et al, 2010) which can be further investigated. The comparison results for these three models are illustrated in Figure 4.16.

Figure 4.16 Achieved makespan for PSO, GA and hybrid models

Furthermore, results from the proposed model are compared to some relevant benchmarks from the literature in order to evaluate the quality of our work. For this reason, results for ABZ5, ABZ6, FT10, LA16 - LA20 and ORB1 case studies were extracted from the work done by Aydin and Fogarty (2004), Meloni et al (2004), and also from Kurdi (2017). In the first research study, the authors have considered several algorithms to solve a JSSP. This includes simulated annealing (SA), and chaotic simulated annealing (CSA) models; tabu search (TS), convergent tabu search (CTS), and neighbouring task (NT) based on tabu search; hill climbing (HC), and constrained hill climbing (CHC); GA based on a genetic algorithm; and asynchronous teams' (ATeams) model based on the collaboration of distributed agents.

The proposed model is compared with various alternative solutions, all shown in the table below. The SA and CSA solutions have similar modus operandi with minimal differences. Their functionality is based on simulated annealing, starting with reading a solution and setting a primary temperature. In each phase of the work, a task is randomly chosen and the orientation of its source arc will be targeted for reversion. The simulated annealing will set the acceptance criteria for the computed make-span. The minimal difference between the two solutions is found to be that the CSA focuses on the arcs located on the highest path length.

Alternative solutions of TS and CTS are set on the foundation of tabu search. The TS and CTS both function through machine solutions, hitting all the swaps of the neighbouring tasks in each cycle of action; albeit, the CTS only considers the arc located on the highest path length. The TS and CTS both accept the reverse of an action to be non-tabu and hence, whilst attempting all swaps, the best non-tabu move is chosen. The algorithms of HC and CHC are known as the hill-climbing solutions: choosing a task and reversing the orientation of the arc

leaving the task; although, the CHC will always target the arc on the longest path. The GA solution is developed on the basis of a genetic algorithm. Its starting point is the reading of two solutions and building up the respective machine orders. Tasks are randomly selected from the top of the queues, from either parent. A task at a further point from the head of the queue will later be chosen to avoid functional loops, if no predecessor for the head's job is decided yet. Lastly, the NT is a similar solution to the CTS, which improves on the functionality of the CTS by including reverse actions in its tabu list of 50 in length.

In the second research study, the work was focused on three heuristic algorithms, namely avoid maximum current completion time (AMCC); select most critical pair (SMCP); and select max sum pair (SMSP). All the three models are based on the idea of rollout metaheuristics represented by the alternative graph formulation.

Finally, in the third study, an improved island model memetic algorithm (IIMMA) was investigated. The model introduces a new cooperation mechanism for knowledge sharing in an evolutionary algorithm, aimed for multi-objective job shop scheduling problems. The comparison of the results is tabulated below in Table 4.3.

Table 4.3 Result of the proposed model versus existing optimal solutions

	Proposed Model	*SA*	*CSA*	*NT*	*CHC*	*GA*	*ATeams*	*AMCC*	*SMSP*	*SMCP*	*IIMMA*
ABZ5	1240.5	1274.25	1234	1241.60	1263	1254.33	1240.33	1346	1402	1541	-
ABZ6	947.9	980	946	945.00	948	958.33	946.33	985	978	1253	-
FT10	983.7	1041.33	954.74	967.20	1036.33	998.33	963.40	985	1013	1369	952.10
LA16	954.3	998.33	946.33	967.67	988	988.33	954.5	979	1060	1144	953.20
LA17	793.75	821.67	784	784.33	808.33	794	784.75	800	892	1041	784.50
LA18	855.50	904	848	857	880	874	855.75	916	976	1127	854.5
LA19	845.70	878.67	847	853	878.75	865.67	842.5	846	932	991	845.90
LA20	908	954.33	910	904.33	948.67	922.33	905.75	930	964	1053	909.30
ORB1	1087.33	1194.33	1102	1128.5	1167.33	1157	1112	1213	1264	1444	1094.20

The output data of the proposed model benchmarked and compared to the literature referenced above, illustrates a noticeable improvement in several cases, whilst competing with some of the other data with a close proximity.

Figure 4.17 demonstrates how the use of multi agents leads to the convergence of solutions to near optimum as the time elapses. As the number of iterations increases, more solution spaces are investigated by the agents and they move towards the best makespan. By the rate of convergence, it is visible that agents are randomly scattered within the search space and they tend to quickly approach the best solution. Consequently, as the simulation reaches its final generations, the improvement is achieved with a slight rate.

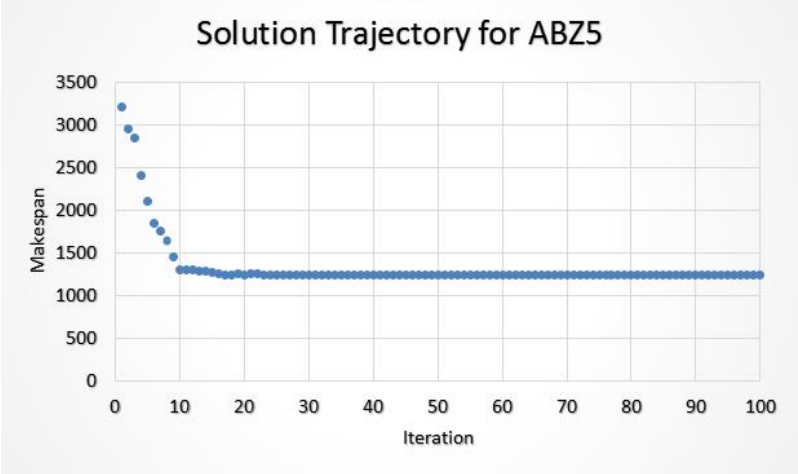

Figure 4.17 Convergence of the solution over 100 iterations for ABZ5

The effect of swapping operations within a schedule on the improvement of the makespan can be investigated thoroughly by reviewing two adjacent solutions from the software. For this purpose, the generated solutions for a random agent are extracted and the results are illustrated in Figure 4.18. Each line in the Output section of Figure 4.18 demonstrates a feasible scheduling solution, where the final value of best fitness shows the makespan for that

particular schedule solution. Each number in the line represents one operation of each job. For instance, the first number 7 in the very first line indicates the first operation of job 7 and the following number 9 represents the first operation of job 9. If the two consequent solutions are considered from Figure 4.18, it is noticeable that by swapping four operations the total makespan is reduced by 13 units of time. To be more precise, by changing the 1st operation of job 3 with the 4th operation of job 2, and also by changing the 8th operation of job 9 with the last operation of job 4, the makespan has changed from 1255 to 1242. This is demonstrated in Figure 4.19, which shows the contribution of each agent in the overall optimisation.

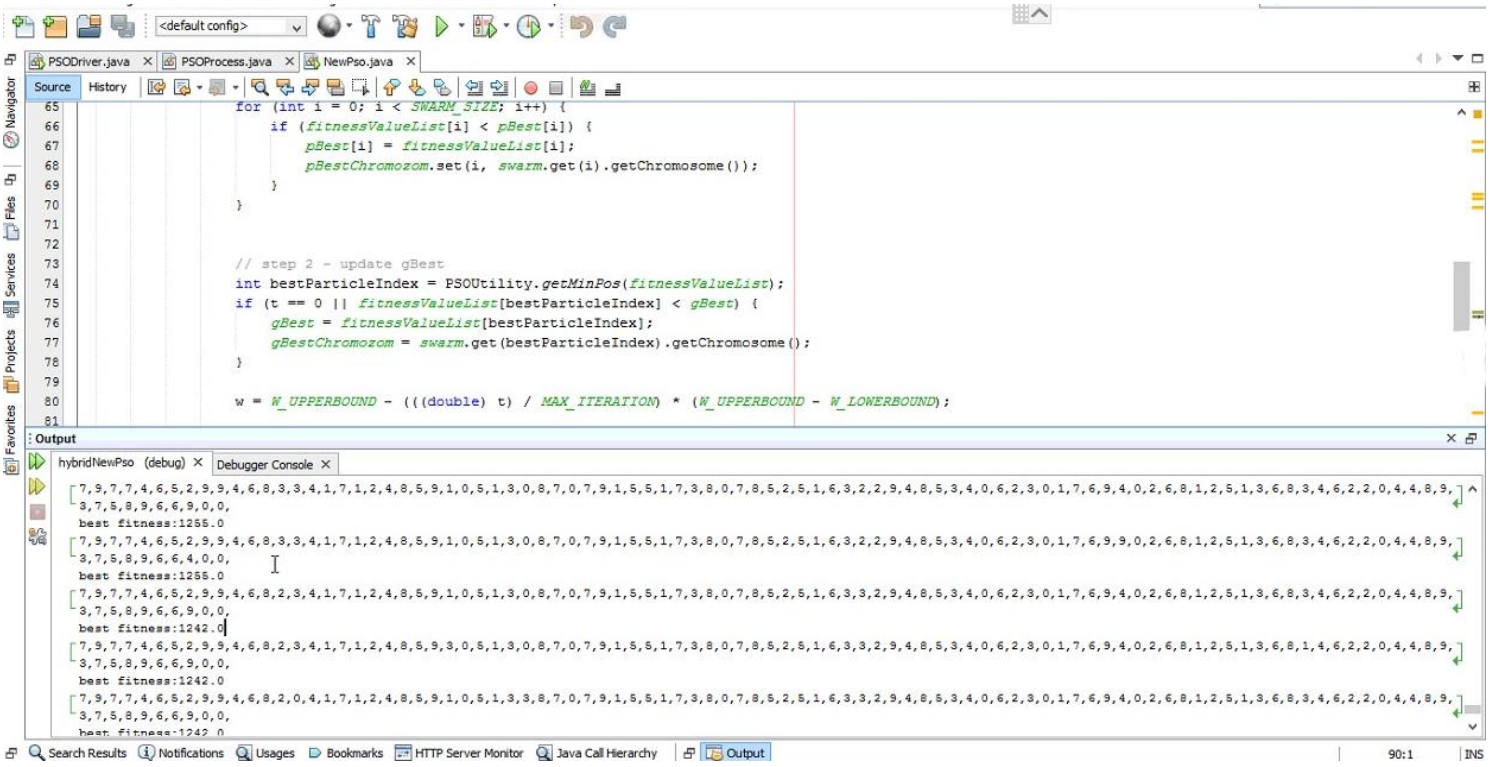

Figure 4.18 Generated solutions for an individual agent

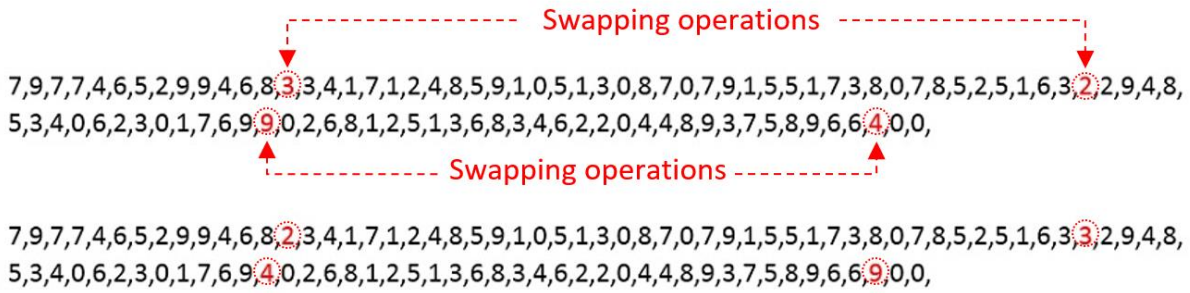

Figure 4.19 Effect of swapping 4 operations within one agent's solutions

4.7 Conclusion

The physical dispersing nature of job scheduling is fundamental, as the manufacturing companies enlarge and expand with the increase of suppliers, customers and partners. This is followed by the contrast in the job shop scheduling of the manufacturing industry with the theoretical problems. This can be resolved by a decentralised system, which allows each individual company to decide and schedule plans with respect to their suppliers and sub-suppliers.

Further research could be conducted in combining machine learning with multi-agent systems, as by "teaching" the machines how to react in certain situations, this could further enhance their performance and bring the optimal solution closer. Additionally, heuristics can be combined with the multi-agent approach in order to obtain solutions that are close to the optimal. These concepts could also be applied to the flexible job shop scheduling problem, where customised goods are developed at the manufacturing plants. They are useful because the development of customised goods requires a high degree of adaptability. Machine learning or heuristics in combination with multi-agent systems will add further adaptability

and flexibility to the manufacturing process. In this research, a decentralised multi-agent system for scheduling of manufacturing operations was presented. The use of simple algorithms combined with distributed agents and an ontology as their core communication mechanism leads to an intelligent manufacturing system. The underlying intelligence in such systems enables agents to sequence their own operation and acts autonomously while cooperating to achieve a common goal. In this research, we investigated the development and implementation of an intelligent scheduling system, which provides flexibility and scalability through the use of heterogeneous agents while obtaining near optimum solutions.

5 PREDICTIVE MANUFACTURING

5.1 Introduction

Current trends in manufacturing show the need for visibility and reliability in decision making. The underlying intelligence of a predictive manufacturing system will adapt to disturbances by exploiting the seamless flow of relevant available data. Chaotic data from upstream and downstream of a manufacturing supply chain can be logically interpreted by predictive manufacturing through an enhanced fusion of information. A sophisticated expert system which stores machine-interpretable data enables effective reasoning and hence better decision making throughout the whole supply chain. This project extends the use of the data to predictive manufacturing. The motivating scenarios or examples of the role of prediction in manufacturing are gathered through the data from the project case study provider. The next step is to define a list of retrievable KPI terminologies under which raw data will be stored and identified. These enable retrievable web resources to be linked and merged into what is known as ontologies for predictive manufacturing. The ontology is a semantic tool which will be used to express the concepts and properties of the manufacturing domain in a standardised formal language such as XML and OWL because of its share-ability and interoperability. The ontologies are then parsed into Java objects that can be implemented into an agent-based model. In the remaining sections of this chapter, the prediction scenarios and examples which form the scope of this research are introduced. Then the proposed solutions are discussed alongside the achieved results.

5.2 Motivating scenarios and questions

As discussed previously, raw data are extracted from GFM for the development of a unified ontology. Key semantics are defined and extracted alongside their relationship within the manufacturing enterprise domain, which are then stored in a knowledge base and will undergo the next step: defining the predictive questions that the system would like to answer. The preliminary work for the development of a predictive manufacturing model is to address which questions or scenarios are desirable for the system. The framework of the system is dependent upon the nature of the scenarios and questions mentioned in the first stage. These questions will form the scope of the proposed predictive model. A variety of machine learning and data mining methods is used in solving these questions, considering the nature of each scenario, which consequently leads to the formation of the proposed model. The questions which are the backbone of this chapter are as follows:

1) How can the prompt delivery of an order with GFM code X, lot value of Y and an expected time of Z be predicted?
2) How can the conformity of a placed order with GFM's supplier upon delivery be predicted?
3) Would there be a potential for the relationship between the quantity, delay and quality of a placed order? (For instance, if an order of a particular supplier is placed with more than X lot (quantity), would it be delivered late/non-conforming with a Y percentage?)
4) How to can the percentage of delay/non-conformity of orders for a specific supplier in year X be forecast?

5) Which suppliers will offer the delivery of orders with a Y-day delay by a probability of X%? (For instance, which suppliers will deliver the orders with a 70% probability of a maximum 10-day delay?)
6) How can suppliers with X quantity of conformed orders without any delays be selected?

Based on the aforementioned prediction and classification scenarios, some machine learning and data mining techniques are considered to answer these questions. These methods are elaborated in detail in the following sections with the obtained results.

5.3 Machine learning and data mining for predictive manufacturing

In the recent past, intensive research has been done to improve the effectiveness of machine learning. Consequently, cutting-edge machine learning algorithms have been produced. Some of the algorithms brought forward include decision trees, neural networks, regularized logistic regression, random forests and support vector machines (SVM). A support vector machine is a type of learning algorithm that supports supervised learning. The machine learning algorithm has found numerous applications in both regression and classification challenges, though it (SVM) is mainly used in prediction and classification challenges. The SVM has gained popularity due to its ability for generalization and finding nonlinear solutions. The development of the support vector machine is attributed to Chervonenkis and Vapnik (1963). Vapnik and Chervonenkis proposed a methodology for creating nonlinear classifiers whereby the kernel trick was applied to the greatest margin hyperplane.

5.3.1 Working mechanism of SVM algorithm

The process of classification focuses on building a single model using discrete classes of data. The development of the single model entails using a set of training data to build class labels from which an algorithm learns. As earlier connoted, one of the main learning algorithms is the support vector machine algorithm. The SVM algorithm builds its model from learning by example. Fundamentally, the SVM works by determining a hyperplane that divides data set classes into two. The data points closer to the hyperplane are the support vectors. The support vectors are the critical elements of the data set since their absence, or position change leads to alteration of the hyperplane position. The position of the hyperplane is chosen such that there is the greatest possible margin between any support vector and the hyperplane. The greatest possible margin is determined from the data used for training (Statnikov et al, 2011). Consider a training data set in an X–matrix with a range$(x_1 : x_m)$. The data set is divided into two separable classes that have class labels of -1 and +1 and the storage is in a Y-vector of the range $(y_1 : y_m)$. The margin plane of the data set can be determined using equation 5.1 shown below:

$$\left||W|\right|_2 = w.w = \sum_{i=1}^{d} w_i^2 \qquad (5.1)$$

The constraints that satisfy equation 5.1 above are as defined below:

$$y_i(w.x_i + b) \geq 1$$

Where $i = 1 : m$ and $b \in R, x_i \in R^d$. The generated function i.e. the decision function has the form $f(x) = \mp(w > x + b)$. The function is negative if the argument is less than zero and positive if the argument is greater than zero. For a non-separable case, a slack variable (s_i) is introduced to minimize the decision function as described by equation 5.2.

$\frac{1}{2} w.w + C \sum_{1=1}^{m} s_i$ (5.2)

The above function is within the constraints defined as shown below:

$y_i(w.x_i + b) \geq 1 - s_i \; and \; s_i \geq 0$

Introducing Lagrange multipliers and finding partial derivatives appropriately in equation 5.1 and equation 5.2, the decision function used in the SVM classification is given by equation 5.3 below:

$f(x) = \pm(\sum_{sv} \alpha_i y_i K(x_i, x) + b)$ (5.3)

The value α_i is a non-zero parameter and it is a subset of the training points. The parameters $y_i, x_i, and \; K$ are the training points. The parameter x is a test point.

Summary of the SVM algorithm

Step	Action
Step 1	Training the initial SVM using the training data to produce support vectors that correspond to the decision function.
Step 2	For every support vector highlighted in step 1, its projection on the hyper surface is determined along the decision function gradient.
Step 3	Compute the hyper surface generalized curvature for each of the support vectors.
Step 4	In descending order, sort the support vectors from the training data.
Step 5	Using the data point pairs, train the SVM.

5.3.2 Comparison with other prediction and classification algorithms

Logistic Regression

Logistic regression is a linear classification machine learning algorithm that classifies data by using the probability of occurrence of a given event. The logistic regression model aims at maximizing the probability of the data. This is unlike the SVM which focuses on finding a hyperplane that maximizes the distance between points close to each other. Logistic regression compares all points in determining the most probable point, while the SVM only compares the vector support points (Hsu and Lin, 2002). A study done by Musa (2013) shows that both the linear regression and SVM have almost equal performance for both unbalanced and balanced data. Furthermore, the study found out that the SVM has a better performance on unbalanced data. It also pointed out that the SVM is a black box predictor while the logistic regression boosts of high interpretability. Another study done by Salazar et al (2012) discovered that SVM models were feasible for predicting a data set with new observations based on single variables. Salazar further opines that the SVM performs better than linear regression when there is a need for high correlation structures. Also, the study discovered that less variables are required to effectively implement the SVM as compared to linear regression (Cristianini and Shawe-Taylor, 2000).

Decision Tree

A decision tree is a flowchart-like algorithm that entails representation of learned functions as decision trees. The learned functions can be represented using if-then rules. Classification using the decision tree algorithm entails sorting out instances down the tree to the leaf node from the root node. For each node in the tree, an attribute of an instance is represented; each

branch from the node contains possible attribute options (Smola and Schölkopf, 2004). A decision tree is characterised with either two or more branches. The leaf nodes on the branches show a decision or a classification.

The development of a tree entails data partitioning to form subsets that have similar instances. For homogenous data subsets, the target variables' entropy is zero, while the entropy of the subject is a 50-50 mixture. The comparison between the entropy before splitting the data set and entropy after the splitting gives the entropy i.e. the quantification of the amount of information obtained during the split.

A decision tree algorithm is a type of supervised learning and is applicable in finding solutions to classification and regression problems. The pseudo code for the decision tree algorithm is as highlighted below:

(a) The best attribute in the data set is at the tree root.

(b) The training data set is then split into subsets whereby each of the subsets has data with similar attributes.

(c) Steps (a) and (b) are repeated till all leaf nodes are represented on all the branches.

In developing a decision tree, it is assumed that the whole of the training data set is the root and the records in the data set are recursively distributed. There are various measures used in the selection of the attributes. The two most popular methods for selecting attribute measures

are the Gini index and the information gain. The information gain attribute selection criterion aims at estimating information from each of the attributes.

The randomness of any given variable within the data set is defined by entropy, which is the quantification of the amount of information obtained during the split. The calculation of the entropy measures helps to estimate the information gain. The entropy measure can be approximated using the following formula:

$$H(X) = E_X[I(x)] = -\sum_{x \in X} p(x) \log p(x). \qquad (5.4)$$

Using the estimated information gain, the expected entropy reduction resulting from the sorting attributes can be estimated.

One of the main advantages of the decision tree algorithm, as compared to the SVM, is that decision tree algorithms require less data cleaning. This is based on the fact that outliers and missing values have little influence on the learning process. In an SVM, since each point is a vector point, an outlier and missing value affect the position of the hyperplane, thereby affecting the learning process.

One major disadvantage of the decision tree algorithm is its inapplicability on continuous variables. Decision tree algorithms are only suitable for categorical inputs and discrete events. The SVM has capabilities of analyzing both discrete and continuous data sets (Smola and Schölkopf, 2004).

Time Series

A time series algorithm is a type of supervised machine learning algorithm. A time series is a historical measurement sequence for given equal time intervals of an observable variable. In this research, the aim is to use time series analysis to predict the future values of conformity and delays based on historical data of a real industrial case study. This concept is presented in section 5.4.3 in detail. Time series analysis was chosen for this purpose as the data presented in section 5.4.3 show a natural temporal ordering. Analysis of time series data entails determining the most efficient model of the described pattern of a given data set. The analysis involves determining the seasonality, trend, long-run cycle, outliers, abrupt changes and constant variance within a given data set (Box et al, 2015). Time series models can be classified into two time-domain categories: the autoregressive integrated moving average model i.e. ARIMA; and the ordinary regression models (Chen et al, 2015; Weigend, 2018). These models are used in the following sections to infer a logical pattern from conformity and delay data points. A comparison of time series with an SVM shows that the SVM requires a small amount of training data as compared to the time series.

K-Nearest Neighbour

In the k-nearest neighbour, learning is based on analogy whereby a given tuple is compared to similar training tuples. The training data set contains n-attributes, and each tuple is stored in n-dimensional space. The k-NN classifier searches the training data to determine values close to the unknown as the nearest neighbour. The closeness of the unknown to the training data is determined using a given distance metric. Various studies have been done to compare

the SVM and k-NN. A study conducted by Danades et al (2016) to compare the accuracy of the SVM and k-NN, determined that the SVM was more accurate in making predictions. Palaniappan et al (2014), in their comparative study of the k-NN and SVM, found out that the SVM is more robust and accurate.

Random Forest

The random forest is a type of supervised algorithm that applies to both prediction and classification. The algorithm utilizes randomization in the production of a diversity of pools with individual classifiers. For instance, to build a random forest, a sample feature subset for the decision tree is selected randomly and for each of the decision trees from the training data subset. For the random forest to work on randomization, large instances of the training data set are required. A random forest uses soft linear boundaries which produce inferior results as compared to the maximum margin used in a support vector machine learning algorithm (Hamel, 2011).

In this section, a comparison of the support vector machine learning algorithm has been carried out. Each of the stated algorithms has strengths and limitations which are dependent on the type of data set and the implementation mechanism. However, it can be shown that the support vector machine learning method produces an outlier-free and clean data set in almost all the scenarios. The strength of the support vector machine learning algorithm lies in its sparseness and its generalization. This tends to make the SVM superior to the other algorithms, such as the artificial neural network algorithm.

5.3.3 Selection of prediction and classification models

In light of the above, there are several machine learning techniques suitable for use in the project. Considering the nature of the prediction scenarios presented earlier in this chapter, the previous comparison indicated that the support vector machine mechanism, decision tree and time series methods give a better performance as compared to the other algorithms; and furthermore, are suitable for solving the selected prediction questions or scenarios. Other numerous advantages that inspired the selection of the SVM algorithm for the project are as highlighted below.

The kernels in a SVM make the application of the SVM flexible in the selection of the separation of threshold parameters. The output of the SVM has an out of the sample generalization making the algorithm robust. Due to the complex nature of the optimality problem, the support vector machine produces a unique solution. Another important advantage of the SVM is its speed in evaluating the function that has been learned. Furthermore, there is a high prediction accuracy when SVMs are used in predictions and classifications' problems (Wang, 2005).

Also, for the classification problem presented in section 5.4.1, it is realised that due to the nature of the provided real industrial case study data, we are dealing with a classification problem with heavily overlapping data points. Therefore, the SVM was chosen for the work presented in section 5.4.1 as it enables the classification of the overlapping data in higher dimensions.

For the conformity prediction problem presented in section 5.4.2, a decision tree model was chosen for several reasons. First of all, the decision tree requires reasonably little data preparation and data cleansing effort. This is crucial in the presented case study, as the amount of data provided for this prediction is relatively large. Secondly, the decision tree allows an implicit feature selection which is well-established and logical. Thirdly, the non-linear relationship that exists between the presented parameters in section 5.4.1 does not affect the prediction performance. Finally, it is very easy to implement and interpret the data using this model. Therefore, a decision tree model was chosen for section 5.4.2.

For the prediction scenario in section 5.4.3, the selection of a time series analysis is plausible, as the aim is to predict some features with respect to time and the temporal effects are of utmost importance. Moreover, it enables the prediction of seasonal and trend patterns which can be replicated or extrapolated for other predictions.

5.3.4 Programming language implementation

The SVM can be applied in various programming languages such as Python, C++, JAVA, and MATLAB. Each of the stated programming languages has libraries specifically developed to facilitate the implementation of the SVM.

The SVM implementation has been done using R language; it is an open-source programming language which has gained a reputation for its strong computational capabilities when dealing with statistical analysis and machine learning problems. Since it has a huge collection of machine learning libraries, R is considered better than the other

programming languages in the implementation of SVM. As compared to the other programming languages, R is faster than C++, JAVA, PYTHON, and MATLAB. Furthermore, R language has numerous and easy to use tools for the visualization of data. The broad ranging capabilities of R in performing various functions including data manipulation, statistical modelling and graphics have previously been recognised by many users. Its extensible nature has allowed the developers to design their personalised software and take advantage of an add-on package for its distribution. It is compatible with many add-on packages, including the RODBC package that connects R with the Open Database Connectivity protocol (ODBC); as well as the ROracle package that allows R to read from the Oracle databases. For the decision tree and time series prediction, Microsoft Excel was used due to its simplicity and also because of its add-in and plugins for decision tree and time series analysis.

5.3.5 Automating ontology and SWRL rules' generation

In order to show the scalability factor and to demonstrate the real capabilities of the proposed ontology model when dealing with more complex problems, there is a need for an automated ontology and rule generation process, where human intervention is reduced to a minimum. Introducing an intelligent model will enable the generation of an ontology, the creation of rules and a reasoning process from any data sample or from any case study, no matter how complex or how large the data set. Therefore, an intelligent system is proposed which has two main components. The first stage of the model, uses the Cellfie plugin of Protégé to import all available data from an Excel file into our ontology editor. Cellfie is an open-source plugin for Protégé that enables flexible mapping of spreadsheets to OWL ontologies (Kola and Rector, 2007; Hardi, 2015). This plugin is well-established and updated more frequently

compared to similar plugins, such as OntoBase; and therefore, is a reasonable candidate for such an implementation. Thus, the Cellfie plugin was used to import axioms from spreadsheets to Protégé where the transformation rules were used to facilitate the mapping process.

The second phase of the model deals with the generation of SWRL rules for the reasoning process. The main challenge here is to generate the rules in an automatic way so it can be extrapolated to other problems. In order to demonstrate the feasibility of this method and to show the scalability of the system, SWRL rule generation from the decision tree predictive model is implemented. First, the decision tree model is imported into CSV format. Then a program was written in MATLAB which can read the lines in the CSV file, extract the SWRL rules accordingly, and save them into another CSV file. The syntax below shows the decision tree model imported into CSV format for a simple case of PRODUCT_VALUE and LOT_VALUE.

PRODUCT_VALUE <= 1.128

LOT_VALUE_in_Euros <= 88.55: s2341064 (1601.0/59.0)

This means that if PRODUCT VALUE < = 1.128 && LOT VALUE in Euros < = 88.55 then allocate the order to s2341064 which is the code for a supplier.

The SWRL rule for the above notion is as follows:

Order(?od) ^ PRODUCT_VALUE(?od, ?P) ^ swrlb:lessThanOrEqual(?P, "1.128"^^xsd:decimal) ^ LOT_VALUE_in_Euros(?od, ?L) ^ swrlb:lessThanOrEqual(?L, "88.55"^^xsd:decimal) - > s2341064(?od)

When the ontology and the SWRL rules are in place, then a reasoner such as the Pellet could be used to infer knowledge and perform activities such as prediction. The complete example of the imported decision tree in CSV format alongside the extracted SWRL rules and the source code of the MATLAB program are presented in Appendix B.

The development of this novel mechanism, shows that more complex problems or larger datasets can be mapped into an ontology automatically. Consequently, data can be manipulated or queried as it is stored in an ontology, like the work presented in Chapter 3 of this thesis; or it can be used for other applications, such as predictive manufacturing (the concept as discussed in this chapter). In either case, the model demonstrates the flexibility and scalability, where there is no limitation in the size of the instance problems. Therefore, it can be extrapolated for larger and complex problems or even for other applications with minimum alterations. This developed concept is also presented in the following publication (Zhong et al, 2018).

5.3.6 Application of SVM in manufacturing domain

Although the manufacturing industry is well established, efforts are being made to increase efficiency and hence productivity. Various manufacturing companies have adopted machine learning in processing of large amounts of complex data. The SVM has been used in various

industries for the derivation of patterns from pre-existing data sets to project approximations about the future behaviour of a given system. The projected information is handy in directly improving the productivity of the system and also, helps the management to make informed decisions.

In general, the SVM has been implemented by various companies for process monitoring, process control, and process optimization. There are various implementations of SVMs in the manufacturing domain. Most of the manufacturing processes are automated due to the high volume of production. For efficiency in production, close monitoring of the process is paramount. Consequently, the support vector machine algorithm has been implemented to affect the monitoring process. The systems developed using the SVM are responsible for diagnosis of faults, monitoring of the condition of tools and machines, and reporting about wear and tear of the manufacturing systems. Another important process where the support vector machine has been successfully implemented in the manufacturing industry is in the monitoring of the quality of products.

Another critical application of the SVM in the manufacturing domain is its use in the development of image recognition systems. Such image recognition systems developed using SVMs are used in the detection of products that have been damaged along the production line.

5.4 Results and discussion

For the scope of this research, six scenarios were selected as the foundation for predictive manufacturing and they were presented in section 5.2. By thoroughly investigating all six scenarios, it was noted that three machine learning and data mining techniques could be implemented in order to address all of them. Therefore, the SVM, decision tree and time series methods were selected to solve these questions. Three scenarios were selected respectively among the aforementioned scenarios as the case study and they are studied in the following sections.

5.4.1 Predicting the prompt delivery of orders

The pre-processing of data is an essential part of any data mining and classification problem. In the context of big data, a large volume of data is generated every second under different headings. Likewise, in the GFM case, there are several available data with different features.

Table 5.1 Sample extracted data from industrial case study showing different features

PO_YEAR	PO_DATE	SUPPLIER CODE	PO NU	PO POSITIO	GFM CODE	QTY	REQUESTED_ DELIVERY	ACTUAL DELIVERY	LOT VALUE	DELAY DAYS	OTD	QUALITY
2006	06/07/2006	2.3.4.10.370	2	10	2AE001031.00	2	10/10/2006	04/12/2006	€ 4,000.00	55	DELAY	CONFORM
2006	11/07/2006	2.3.4.10.370	3		2AEPT00001.00	1	10/07/2012	13/07/2012	€ -	3	DELAY	CONFORM
2006	11/07/2006	2.3.4.10.370	3	10	6AE000199.00	2	06/10/2006	06/11/2006	€ 14,400.00	31	DELAY	CONFORM
2006	11/07/2006	2.3.4.10.370	3	20	6AE000199.00	2	06/11/2006	30/11/2006	€ 14,400.00	24	DELAY	CONFORM
2006	11/07/2006	2.3.4.10.370	3	30	6AE000199.00	2	06/12/2006	29/01/2007	€ 14,400.00	54	DELAY	CONFORM
2006	11/07/2006	2.3.4.10.7	4	20	2AE000385.00	8	24/07/2006	01/08/2006	€ 38.80	8	DELAY	CONFORM
2006	11/07/2006	2.3.4.10.7	4	40	2AE000380.00	8	24/07/2006	26/07/2006	€ 60.80	2	DELAY	CONFORM
2006	11/07/2006	2.3.4.10.258	5	30	2AE000367.00	2	24/07/2006	12/07/2006	€ 24.00	-12	ON TIME	CONFORM
2006	11/07/2006	2.3.4.10.453	6		CR00002016	8	24/07/2006	26/07/2006	€ 25.74	2	DELAY	CONFORM
2006	13/07/2006	2.3.4.10.113	7	10	2AE001145.00	2	17/07/2006	26/07/2006	€ 120.00	9	DELAY	CONFORM
2006	14/07/2006	2.3.4.10.31	8	10	GANSGCR037002	10	29/09/2006	17/10/2006	€ 36,500.00	18	DELAY	CONFORM
2006	14/07/2006	2.3.4.10.31	8	20	GANSGCR037002	10	24/11/2006	01/12/2006	€ 36,500.00	7	DELAY	CONFORM
2006	14/07/2006	2.3.4.10.31	8	50	GANSGCR037001	10	29/09/2006	17/10/2006	€ 35,000.00	18	DELAY	CONFORM
2006	14/07/2006	2.3.4.10.31	8	60	GANSGCR037001	10	24/11/2006	01/12/2006	€ 35,000.00	7	DELAY	CONFORM
2006	17/07/2006	2.3.4.10.370	9	820	2AE001108.00	1	04/01/2007	08/11/2006	€ 600.00	-57	ON TIME	CONFORM
2006	24/07/2006	2.3.4.10.7	10	10	2AE000849.00	44	06/11/2006	06/11/2006	€ 568.04	0	DELAY	CONFORM
2006	24/07/2006	2.3.4.10.7	10	20	2AE000848.00	36	06/11/2006	06/11/2006	€ 648.00	0	DELAY	CONFORM
2006	24/07/2006	2.3.4.10.7	10	30	2AE000827.00	40	06/11/2006	23/10/2006	€ 516.40	-14	ON TIME	CONFORM
2006	24/07/2006	2.3.4.10.7	10	40	2AE000850.00	8	06/11/2006	13/11/2006	€ 144.00	7	DELAY	CONFORM
2006	24/07/2006	2.3.4.10.7	10	50	2AE000783.00	4	06/11/2006	23/10/2006	€ 51.64	-14	ON TIME	CONFORM
2006	24/07/2006	2.3.4.10.7	10	80	2AE001030.00	24	06/11/2006	22/11/2006	€ 360.00	16	DELAY	CONFORM
2006	24/07/2006	2.3.4.10.7	10	90	2AE000881.00	4	06/11/2006	23/10/2006	€ 72.00	-14	ON TIME	CONFORM
2006	24/07/2006	2.3.4.10.7	10	100	2AE000872.00	4	06/11/2006	23/10/2006	€ 52.00	-14	ON TIME	CONFORM
2006	24/07/2006	2.3.4.10.7	10	110	2AE000885.00	4	06/11/2006	06/11/2006	€ 52.00	0	DELAY	CONFORM
2006	18/07/2006	2.3.4.10.62	11	10	CS01003001	1	18/07/2006	28/07/2006	€ 1,512.00	10	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.313	12	10	CI00001003	1500	03/08/2006	01/08/2006	€ 1,395.00	-2	ON TIME	CONFORM
2006	19/07/2006	2.3.4.10.64	13	10	CV02005002	540	24/07/2006	01/08/2006	€ 16.20	8	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.64	13	20	CV02002008	36	24/07/2006	01/08/2006	€ 23.76	8	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.64	13	30	CD00001008	36	24/07/2006	01/08/2006	€ 0.59	8	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.64	13	40	CV02005004	24	24/07/2006	01/08/2006	€ 1.28	8	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.64	13	50	CR00001007	36	24/07/2006	01/08/2006	€ 0.59	8	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.150	14	10	2AE000857.00	36	24/07/2006	27/07/2006	€ 90.00	3	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.150	14	20	2AE001231.00	6	24/07/2006	27/07/2006	€ 36.00	3	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.135	16	10	CE00002029	50	31/07/2006	21/09/2006	€ 249.20	52	DELAY	CONFORM
2006	19/07/2006	2.3.4.10.135	16	20	CE00002058	50	31/07/2006	21/09/2006	€ 62.67	52	DELAY	CONFORM
2006	19/07/2006	2.3.4.20.14	17	10	GLALANI102001.0.0	263	10/10/2006	16/10/2006	€ 18,444.19	6	DELAY	CONFORM

Therefore, feature selection plays an important role in our classification problem. Selecting the most important features saves time and reduces computational complexities without missing important and relevant information. Accordingly, an evaluation function is needed in order to compare different features. In this study, the correlation evaluation function was chosen due to its ease of implementation and accuracy. Using this function, the effect of each feature on the prediction of the desired class can be calculated and eventually, features with highest correlation values will be chosen among available features for final prediction. Table 5.1 shows a sample of extracted data from GFM.

According to the introduced feature selection function, the correlation values for presented features were calculated with respect to the desired prediction class. It is worth mentioning that the desired prediction class is the on-time/delay for scenario 1 in section 5.2. For the illustration, the results for the GFM code 2.3.4.10.370 are calculated and presented below. Table 5.2 to Table 5.9 demonstrate the results for the feature selection calculation.

- Correlation between quantity of orders (QTY) and label

  The allocated label values are: delay = -1 and on-time = 1

Table 5.2 Correlation between quantity and delay/on-time

Correlation	QTY	Lable
QTY	1	-
Lable	0.023451398	1

- Correlation between lot value of orders and label

Table 5.3 Correlation between lot value and delay/on-time

Correlation	LOT Value	Lable
LOT Value	1	-
Lable	0.178319033	1

- Correlation between waiting time of delivery of and label

The waiting time is the difference between expected time and actual delivery time

Table 5.4 Correlation between waiting time and delay/on-time

Correlation	Waiting Time	Lable
Waiting Time	1	-
Lable	0.048077899	1

- Correlation between total value of orders and label

The total value is calculated as multiplication of QTY and lot value

Table 5.5 Correlation between total value and delay/on-time

Correlation	Total Value	Lable
Total Value	1	-
Lable	0.058392101	1

According to the obtained results for the correlation values between features of supplier 2.3.4.10.370, lot value feature and waiting time are the most suitable features for our prediction because they have higher correlation values with respect to the label. The reason that the waiting time was chosen over the total value is because the total value is obtained from the multiplication of lot value and the lot value has already been selected.

The correlation between all features and the delay days

- Correlation between quantity of orders (QTY) and delay days

Table 5.6 Correlation between quantity and delays

Correlation	QTY	Delays
QTY	1	-
Delays	-0.05266725	1

- Correlation between lot value of orders and delays

Table 5.7 Correlation between lot value and delays

Correlation	LOT Value	Delays
LOT Value	1	-
Delays	0.209599798	1

- Correlation between waiting time of delivery and delays

  The waiting time is the difference between expected time and actual delivery time

Table 5.8 Correlation between waiting time and delays

Correlation	Waiting Time	Delays
Waiting Time	1	-
Delays	-0.091887151	1

- Correlation between total value of orders and delays

  The total value is calculated as multiplication of QTY and lot value

Table 5.9 Correlation between total value and delays

Correlation	Total Value	Delays
Total Value	1	-
Delays	0.072583975	1

Based on the results from Table 5.2 to Table 5.9, the lot value feature and waiting time are the most suitable features for our prediction because they have higher correlation values.

The outcome of the feature selection procedure suggested the selection of lot value and expected time as the main variables affecting the prediction of prompt delivery. Based on the selected features, the support vector machine (SVM) was adapted to implement the classification problem. The SVM method not only can handle linear and non-linear

classification problems but also it is efficient when dealing with overlapping data sets. The kernel trick enables this method to project the overlapping problem into other dimensions and to try to solve them in a linear manner; which makes it a suitable candidate for this research due to the overlapping nature of the extracted data.

This method uses sparse solutions for solving the classification problem meaning that it is not using all the training data sets. Instead, it uses only a subset of data which are called support vector (SV). This is another advantage of this method compared to other classification techniques, because it dramatically reduces the calculation time due to its training efficiency.

In this simulation, features (lot value) and (expected) time were chosen and named as x1 and x2 respectively. Having two classes of (on-time) and (delay) in this problem, the classes were labelled as +1 and -1.

In the learning phase, it is necessary to choose a set of data as the training set in order to define the classifier function. Accordingly, from 661 available data, 114 data belong to label +1 or on-time and 547 data belong to label -1 or delay. It was noted that the (lot value) was missing for some data. Therefore, six data from label +1 and four data from label -1 were designated and deleted as the missing data. Hence, after this data cleansing, there were 543 data from label -1 and 108 data from label +1 where they are 0.83 and 0.17 percent of the total available data. These fractions were used for the proper selection of the training data from each individual label. Therefore, from the total 400 training data, 80 data (0.2) were selected from the positive label and 320 data (0.8) were selected from the negative label.

In the implementation phase in the R language, the kernel function (radial basis) or rbf with parameters c = 1 and γ = 100 was used. The value of parameters cost = 1 and gamma = 100 were selected with the help of tune function from the intervals of 1, 10, 100, 1000 in order to increase the efficiency of the classification.

Finally, by using 20 data from the positive label and 20 data from negative label as the test data, the prediction and classification problem was evaluated. Figure 5.1 and Figure 5.2 demonstrate the training data and test data respectively.

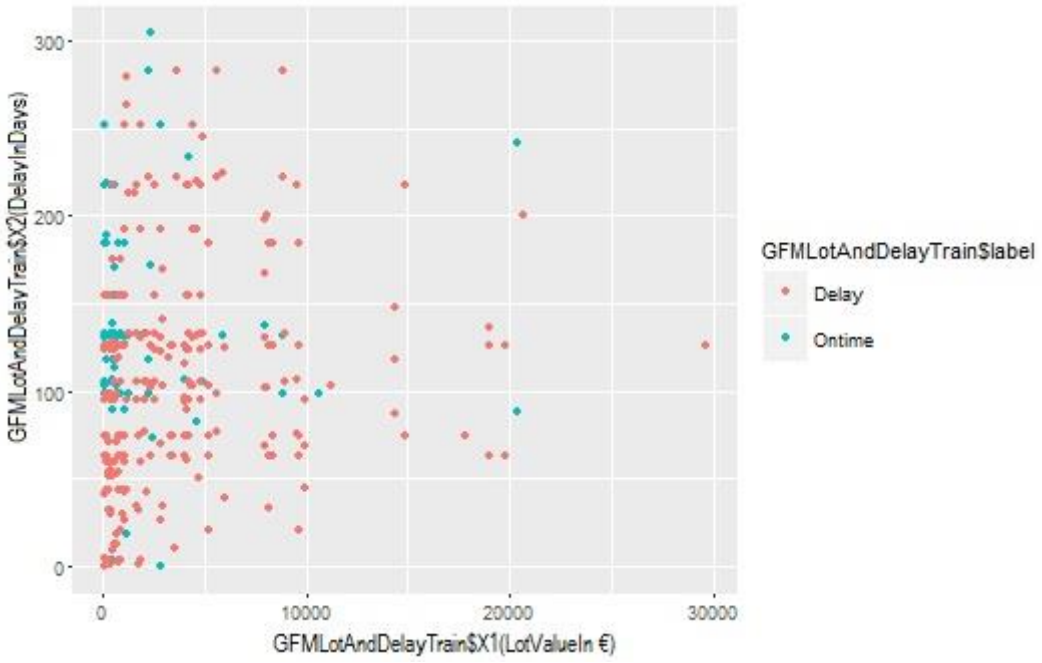

Figure 5.1 Training data for the on-time/delay prediction

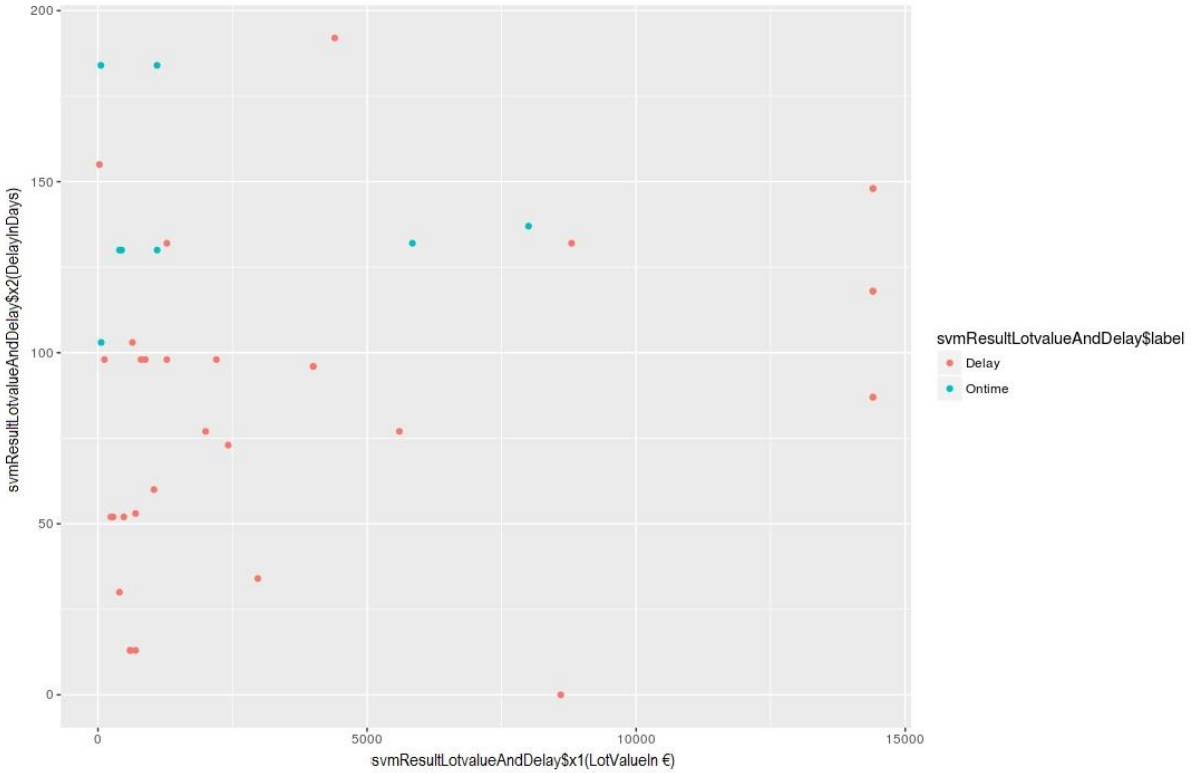

Figure 5.2 Test data for the on-time/delay prediction

Based on the simulation results, 8 data were predicted correctly from the positive label and 12 data were predicted incorrectly. All the 20 data from the negative label were predicted correctly. Finally, for the evaluation of the whole prediction process, the ACcuracy Criterion (ACC) evaluation method is used; where it is calculated from the fraction of the correctly predicted data over the total number of test data.

$$ACC = \frac{28}{40} = 0.7 \qquad (5.5)$$

This shows that the prediction model can predict the prompt delivery of an order with GFM code X, lot value of Y and an expected time of Z with 70% accuracy. Consequently, it will be possible to obtain better results with higher confidence levels by repeating the simulation

with more training data sets and also by choosing different kernel functions with better optimised parameters.

5.4.2 Predicting the conformity of a placed order

In order to solve the conformity prediction problem, a preliminary procedure similar to the previous scenario is needed, with the difference that the effect of features on quality criteria is of upmost importance here. This is similar to the feature selection problem where quality, quantity, delay and lot value features were considered. For solving the conformity prediction problem, the decision tree method was chosen. This method, like the SVM, is a classification technique but with the difference that there is a flowchart mechanism instead of a decision margin or hyperplane.

The most important factor in constructing the decision tree is to find out where to start, or in other words, to decide which feature is the best choice for the topmost decision node. There are three important elements in any decision tree: 1) the root node (decision node) where the tree begins; 2) the leaf node where the flowchart finishes; and 3) the branch (child) which is the connection between the different nodes. The most common way to choose the root node is the ID3 algorithm introduced by Quinlan (1986), where it uses the entropy and information gain concepts as its foundation. Entropy represents the homogeneity of a sample data and its formulation was introduced in section 5.3.2. Information gain is the difference between two entropies. Identifying the specific feature, which results in the highest information gain, is an essential step towards the building of a decision tree. In this research, the entropy was calculated for the target feature (quality) in the first stage. Then, the entropy of all other features is calculated relative to quality in the second phase. Table 5.10 shows the entropy

calculation for the target feature (quality); while Figure 5.3 illustrates the frequency diagram of feature quality.

Table 5.10 Quality entropy calculation

Bin	*Frequency*	Probability	Entropy	
0	8	0.13114754	-0.266417	
1	53	0.86885246	-0.122145	
More	0		0.3885623	Total Entropy
	61			

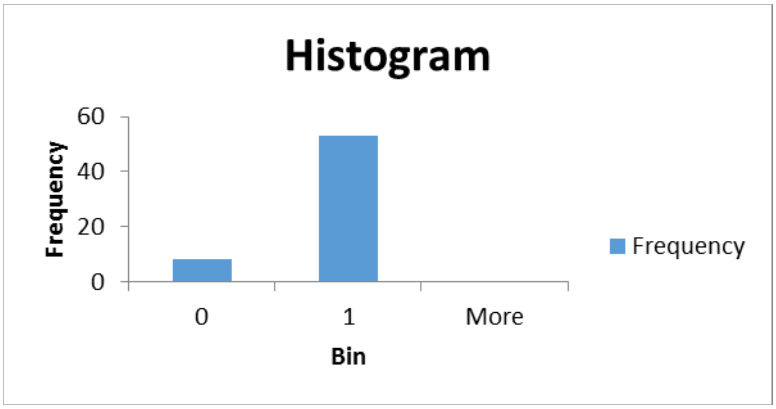

Figure 5.3 Frequency diagram for quality entropy calculation

Here, the excel histogram function was used to draw the frequency diagram of the quality feature. Referencing to equation 5.4, it was mentioned that the entropy is calculated based on probability $p(x)$. Therefore, the probability column was established based on the frequency diagram (GFM's data) in order to calculate entropy. The Bin column represents the number of ranks and their associated values. In the quality entropy it is clear that there are only two ranks which are 0 and 1 or conform and non-conform. However, this is not the case for more complex features such as lot value or quantity. In such scattered data, the ranks could be calculated based on available formulations using minimum, maximum and the range of available data. The entropy calculation for quantity, lot value and on-time delivery is demonstrated as below:

Table 5.11 Quantity entropy calculation

Bin	*Frequency*	probability	entropy	
1	28	0.459016393	-0.357421999	
4.58	27	0.442622951	-0.360754081	
8.16	1	0.016393443	-0.067391375	
11.74	2	0.032786885	-0.112056613	
15.32	1	0.016393443	-0.067391375	
18.9	1	0.016393443	-0.067391375	
22.48	0	0	0	
26.06	1	0.016393443	-0.067391375	
More	0		1.099798192	QTY entropy
	61			

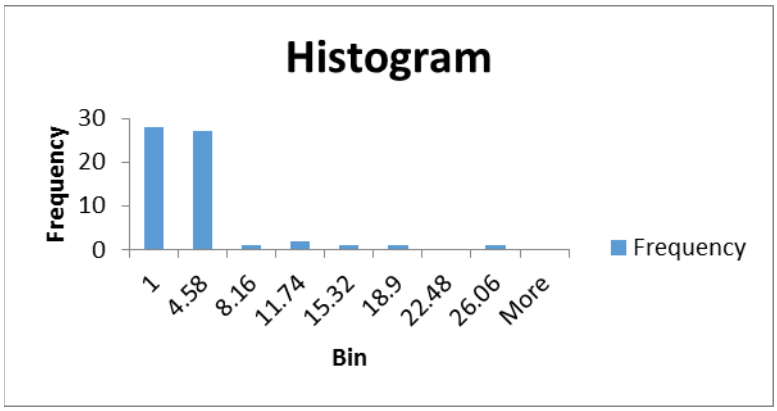

Figure 5.4 Frequency diagram for quantity entropy calculation

Table 5.12 Lot value entropy calculation

Bin	*Frequency*	probability	entropy	
€ 69.00	1	0.016393443	-0.067391375	
€ 573.43	28	0.459016393	-0.357421999	
€ 1,077.86	13	0.213114754	-0.329459321	
€ 1,582.29	9	0.147540984	-0.282341698	
€ 2,086.71	6	0.098360656	-0.228109613	
€ 2,591.14	2	0.032786885	-0.112056613	
€ 3,095.57	1	0.016393443	-0.067391375	
€ 3,600.00	1	0.016393443	-0.067391375	
More	0		1.511563367	total entropy
	61			

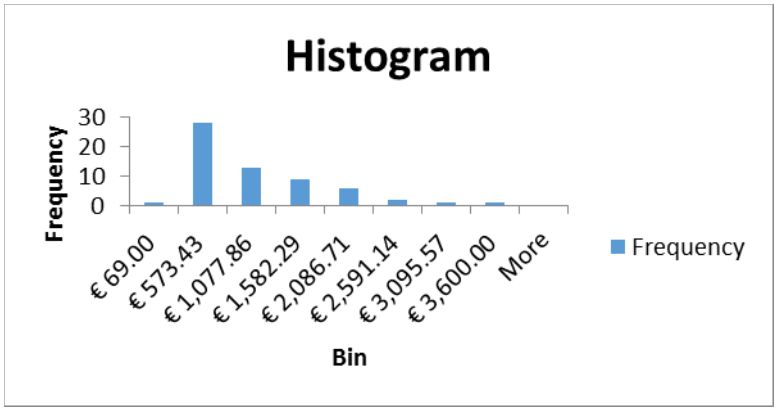

Figure 5.5 Frequency diagram for lot value entropy calculation

Table 5.13 OTD entropy calculation

Bin	*Frequency*	probability	entropy	
0	50	0.819672131	-0.16299	
1	11	0.180327869	-0.3089	
More	0		0.47189	total entropy
	61			

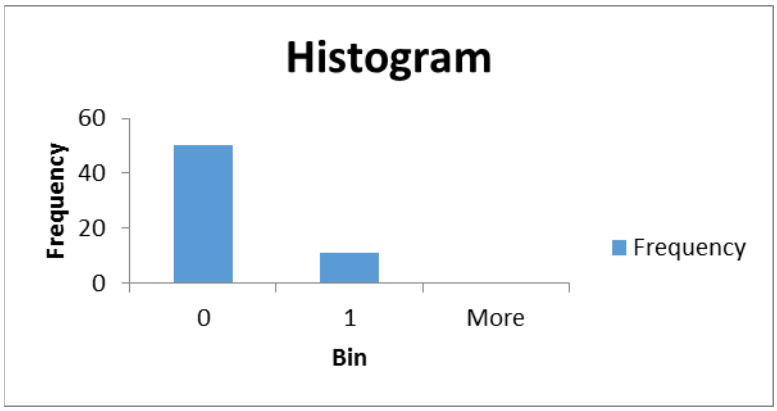

Figure 5.6 Frequency diagram for on-time delivery entropy calculation

As mentioned previously, the second phase of tree construction is the calculation of the entropy of other features with respect to the target feature (quality). This notion is

demonstrated in the following tables; E(T, X) is the entropy of each individual feature with respect to the target feature (quality) while Entropy (T) is the quality entropy. Finally, information gain is the difference between E(T) and E(T, X). Based on the obtained results, the feature with the highest information gain would be selected as the best choice for the root node of the decision tree.

Table 5.14 Frequency distribution of lot value with respect to quality

		Quality		
		conform	non-conform	total
	69-1246	40	2	42
lot value	1247-2424	9	5	14
	2425-3600	2	1	3
				59

Table 5.15 Information gain for lot value feature

p-lotvalue	conform entropy	nonconf entropy	total entropy	two attribute entropy	
0.7118644	-0.046466823	-0.144977259	0.191444082	0.136282228	
0.2372881	-0.284035341	-0.36772122	0.651756561	0.154654099	
0.0508475	-0.270310072	-0.366204096	0.636514168	0.032365127	
				0.323301454	E(T,X)
				0.3885623	Entropy(T)
				0.065260846	Gain

Table 5.16 Frequency distribution of quantity with respect to quality

		Quality		
		conform	non-conform	total
	1-9	48	8	56
QTY	10-18	4	0	4
	19-26	1	0	1
				61

Table 5.17 Information gain for quantity feature

p-lotvalue	conform entropy	nonconf entropy	total entropy	two attribute entropy	
0.9180328	-0.132129154	-0.277987164	0.410116318	0.376500227	
0.0655738	0	0	0	0	
0.0163934	0	0	0	0	
				0.376500227	E(T,X)
				0.3885623	Entropy(T)
				0.012062073	Gain

Table 5.18 Frequency distribution of on-time delivery with respect to quality

		Quality		
		conform	non-conform	total
OTD	ontime	10	1	11
	delay	43	7	50
				61

Table 5.19 Information gain for on-time delivery feature

p-lotvalue	conform entropy	nonconf entropy	total entropy	two attribute entropy	
0.1803279	-0.086645618	-0.217990479	0.304636097	0.054934378	
0.8196721	-0.129707685	-0.2752558	0.404963485	0.331937283	
				0.386871661	E(T,X)
				0.3885623	Entropy(T)
				0.001690639	Gain

Based on the information gain obtained from Table 5.15, Table 5.17 and Table 5.19, it is realised that lot value has the highest information gain and therefore it is the best choice for the topmost decision node; QTY obtained the second place; and finally OTD has the least information gain. This ranking is demonstrated in Table 5.20.

Table 5.20 Ranking of features based on information gain for decision node selection

selection of decision node			
feature	Information Gain		
lot value	0.065260846		
Qty	0.012062073		
OTD	0.001690639		
	0.065260846	decision node= lot value	

Based on the aforementioned results, the decision tree can be constructed for the conformity prediction problem. As an example, one leaf node of the constructed decision tree is explained here. Based on Table 5.20 it was concluded that the best decision node is lot value. Therefore, the effect of other features on lot value is important. Referring to Table 5.14, the lot value variables were categorized into three regions (Bins). The second Bin of lot value is chosen from Table 5.14 and the rest of the features are compared with that specific Bin. This leaf node construction is shown in Table 5.21. As demonstrated in the table, the first column represents the second Bin of lot value. The second column shows the corresponding quantities in three different colours (Bins), the third column represents the on-time delivery and finally the last column which is the leaf node is allocated for quality (conformity). This is just a section of the constructed decision tree while the rest of the model can be created with the same logic.

Table 5.21 Sample leaf node of constructed model

LOT VALUE	QTY	OTD	QUALITY
€ 1,250.00	1	0	1
€ 1,250.00	1	0	1
€ 1,288.00	1	1	1
€ 1,510.00	1	0	1
€ 1,510.00	1	0	1
€ 1,890.00	1	1	1
€ 2,000.00	1	0	0
€ 1,290.00	2	0	0
€ 1,560.00	2	0	1
€ 1,754.00	2	0	0
€ 1,754.00	2	0	1
€ 1,970.00	2	0	0
€ 1,330.00	4	0	0
€ 1,800.00	4	0	1
€ 1,328.00	16	0	1
€ 2,158.00	26	0	1

One of the major advantages of the constructed decision tree in this example is that we can predict with 100 percent confidence, unlike the SVM method where the accuracy was around 70% probability. It is also worth mentioning that as the number of data and features increases, the accuracy of the designed prediction model improves. Therefore, constructing the model with a larger amount of data and variables is recommended for further optimisation. Also, the decision tree can be mapped into a set of rules. This implies the similarity of the model with ontology and rule-based reasoning where we can infer knowledge from a knowledge base with the help of devised rules.

5.4.3 Forecasting delay/conformity of orders with respect to time

The third category of the proposed prediction model based on extracted GFM scenarios is the time series prediction analysis. In order to perform the time series prediction, it is essential to analyse the relation between qualities, quantities and on-time delivery with time. Time series prediction is a complex concept built upon several elements, each with unique characteristics. Therefore, the scope here is not to fully cover all aspects of time series prediction but to create a model to address the proposed GFM scenario.

The main approach to a precise analysis of time series data takes account of the most proficient model which outlines the pattern of a data set. The method is broken down into the investigation of several factors such as the seasonality, trend, long-term cycle, outliers, abrupt changes and constant variance within the data set. The most common models of time series are categorized in three domains, namely the autoregressive, moving average model and autoregressive integrated moving average which is the combination of the above (ARIMA).

The very first step before creating any of these models is to extract and analyse elements of the available time series such as trends, seasonality or abrupt properties.

In the implementation phase, two different suppliers with GFM's code of 2.3.4.10.10 and 2.3.4.10.10.16 from Table 5.1 were selected. Based on the extracted data, it was observed that all the orders were delivered as conformed for the first supplier. Therefore, only quantity and OTD time series diagrams are of most concern for the code 2.3.4.10.10. For the quantity variable, there was more than one data for each specific time value. Therefore, in order to draw the time series graph all quantity values for each individual date were combined under a single value and formed a new data structure. Also for the OTD variable, instead of having 0 and 1 conditions, all the values for each specific date were aggregated and expressed as a percentage. Table 5.22 tabulated the constructed data for code 2.3.4.10.10 while Figure 5.7 and

Figure 5.8 demonstrate the time series prediction for quantity and on-time delivery respectively. Here, polynomial interpolation is used for curve fitting in order to construct the prediction curve.

Table 5.22 Quantity and on-time delivery data for GFM code 2.3.4.10.10

new date	new quantity		new date	on-time percent
17/10/2006	3000		17/10/2006	0.334
31/01/2007	3000		31/01/2007	0.667
09/05/2007	2000		09/05/2007	0.5
08/10/2007	2000		08/10/2007	0
13/11/2008	6000		13/11/2008	1
30/06/2009	1000		30/06/2009	1
01/10/2009	8000		01/10/2009	0.25
04/02/2010	3600		04/02/2010	1
16/07/2010	1600		16/07/2010	0.6
09/11/2010	6000		09/11/2010	0.667
15/02/2011	525		15/02/2011	1
25/07/2011	601		25/07/2011	0
02/11/2011	1000		02/11/2011	0
28/11/2011	1601		28/11/2011	0.601
01/03/2012	1000		01/03/2012	1
02/05/2012	1000		02/05/2012	1
18/06/2012	601		18/06/2012	1
02/07/2012	1000		02/07/2012	0
10/07/2012	1500		10/07/2012	0
26/10/2012	1000		26/10/2012	1
07/02/2013	2500		07/02/2013	1
14/03/2013	3000		14/03/2013	0
22/10/2013	5600		22/10/2013	0.357
23/07/2014	4600		23/07/2014	0
26/02/2015	1000		26/02/2015	0
01/04/2015	8000		01/04/2015	0.75
16/10/2015	2000		16/10/2015	0

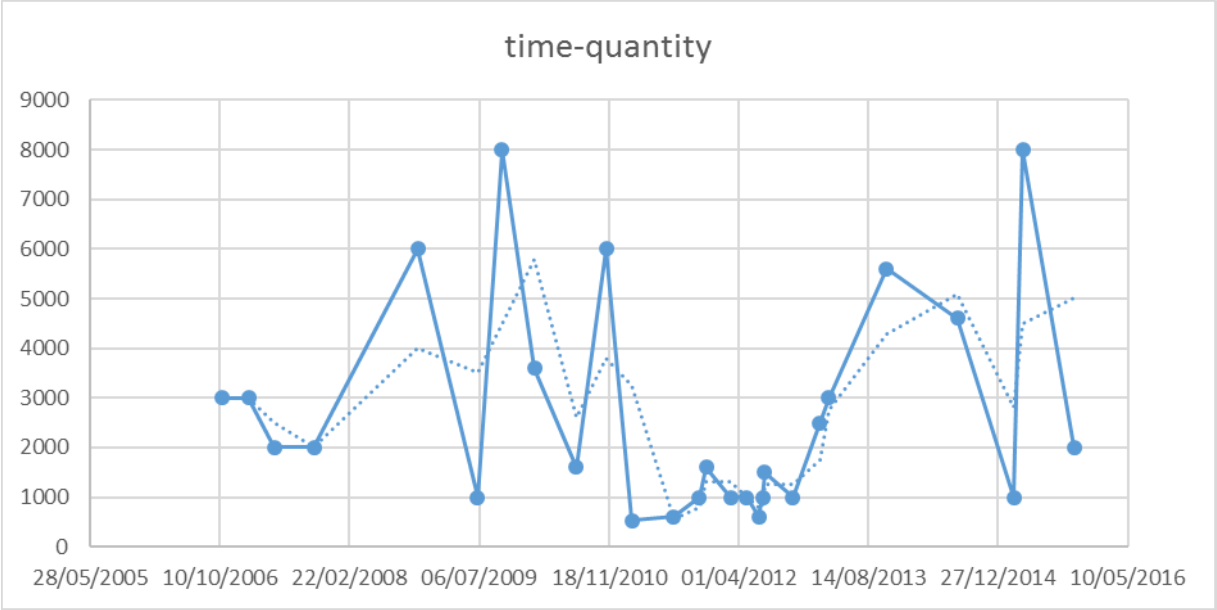

Figure 5.7 Time series line chart for prediction of quantity over time for GFM code 2.3.4.10.10

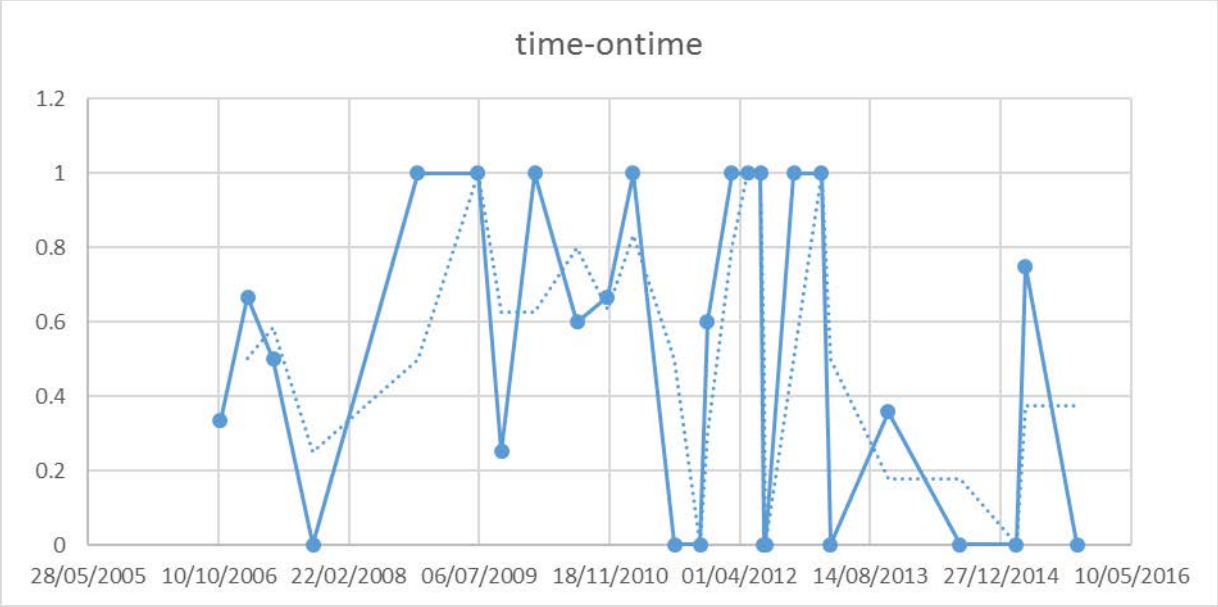

Figure 5.8 Time series line chart for prediction of on-time delivery over time for GFM code 2.3.4.10.10

In the second model (2.3.4.10.10.16), it is observed that there is trend factor for the quantity chart but there is no seasonality factor. As mentioned previously, the major effort in constructing a time series model is allocated to the extraction of entities such as trend, seasonality, abrupt changes from the model. Therefore, the fewer features that exist, the faster the construction will be obtained. In the first model (2.3.4.10.10), the simple polynomial interpolation is used because there was neither trend, nor seasonality. The only existing feature was the abrupt changes, which was the main factor in building the model. However, in the second model (2.3.4.10.10.16) there are both trends and abrupt changes and that makes it more difficult to create a prediction model due to its complexity. Results for the second model are demonstrated below.

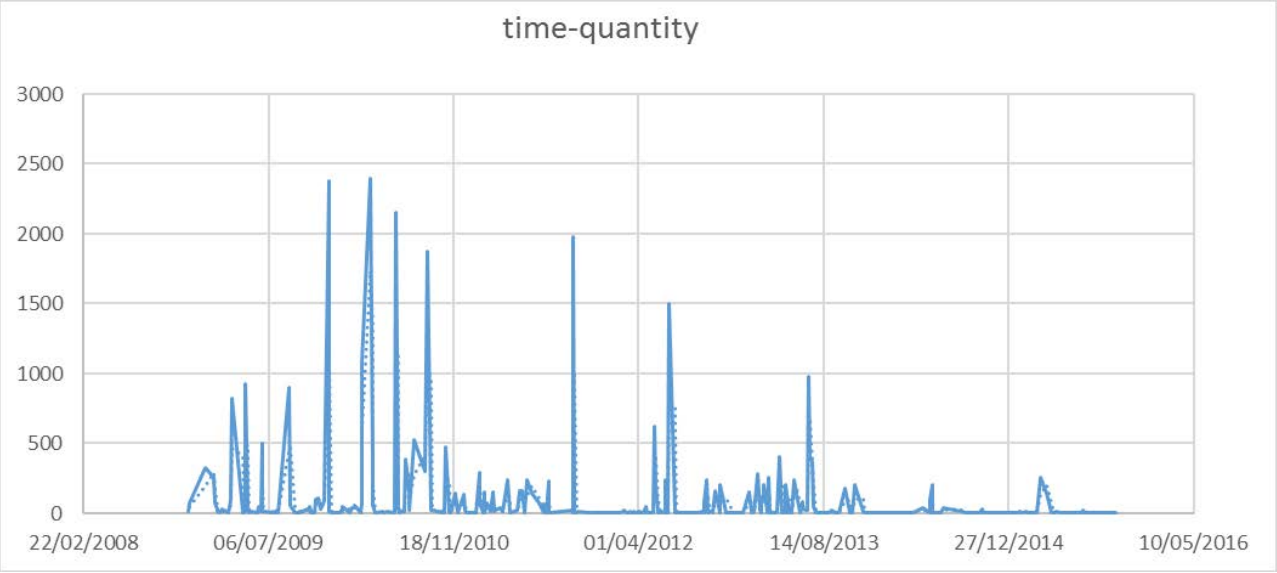

Figure 5.9 Time series line chart for prediction of quantity over time for GFM code 2.3.4.10.10.16

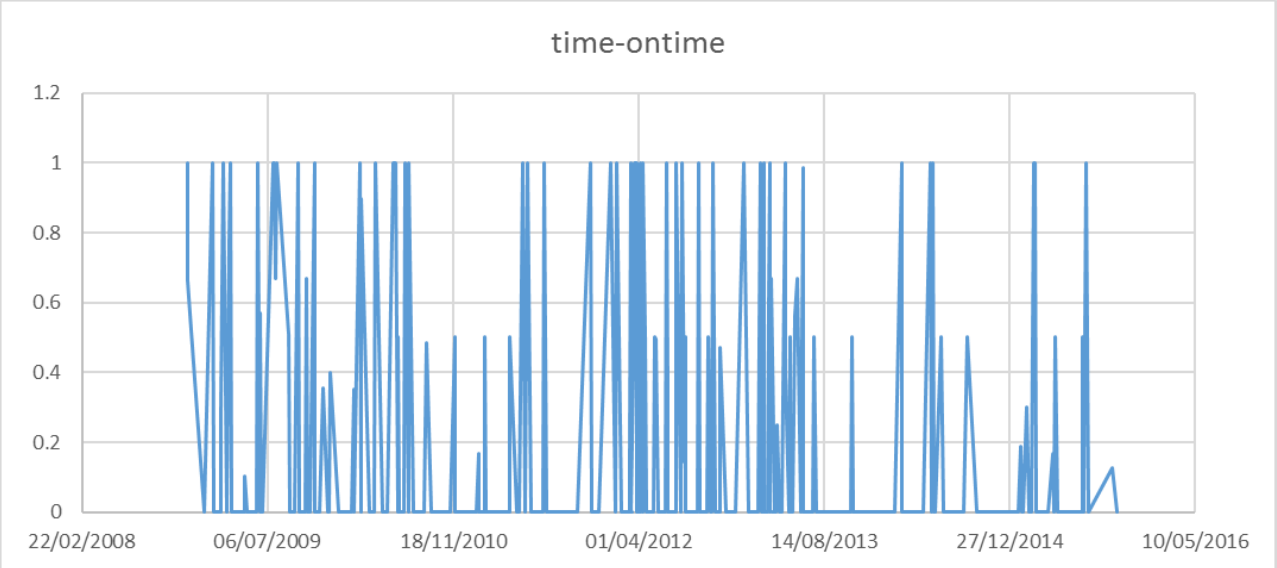

Figure 5.10 Time series line chart for prediction of on-time delivery over time for GFM code 2.3.4.10.10.16

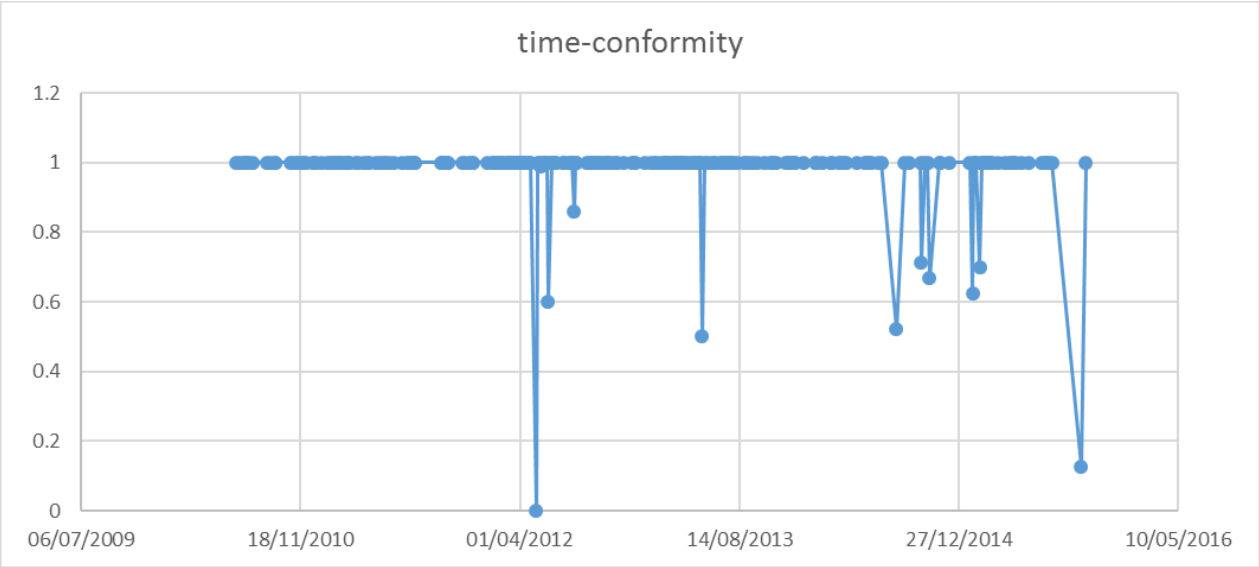

Figure 5.11 Time series line chart for prediction of conformity over time for GFM code 2.3.4.10.10.16

5.5 Conclusion

In this chapter, the scope of prediction was defined for our manufacturing case study with the help of six questions or scenarios. Based on the nature of the scenarios, three of them were selected each with a unique solving method. The support vector machine algorithm alongside the decision tree, time series and other machine learning and data mining techniques have been reviewed. The review entailed the working mechanism of the algorithms as well as a comparison between the SVM and other machine learning algorithms such as a k-nearest neighbour, random forest and linear regression. The comparison established that implementation of machine learning using the SVM for the first scenario; a decision tree for the second case study; and finally time series models for the last case has more advantages over the other machine learning algorithms. It has also been determined that the strength of each of the mention algorithms are dependent on the application environment.

In addition, the various programming languages for implementing the SVM and other techniques have been reviewed in this chapter. Out of the various programming languages capable of implementing the SVM, preference has been given to R over the others due to a large collection of machine learning libraries and the ease of implementation of the SVM. Microsoft Excel was chosen for second and third scenarios due to its simplicity of implementation. The application of machine learning and data mining in manufacturing industries was reviewed. It has been determined that the SVM has various applications in manufacturing including monitoring, image processing, quality control, process control, among other applications.

Lastly, three prediction models were developed and investigated for all three selected scenarios. For the prompt delivery prediction scenario, a feature selection algorithm was used for the selection of the main features followed by an SVM implementation in R. It was concluded that lot value and expected time are the most important features in the classification model; and finally a classification model with nearly 70% confidence level accuracy was developed and tested. For the conformity prediction problem, a similar preliminary feature selection procedure was performed but with the help of entropy and information gain. The feature selection revealed the order of importance of features for the construction of a decision tree model. Ultimately, the conformity prediction was enabled by the constructed decision tree. Last but not least, a time series forecasting model was developed for two randomly selected suppliers. The models forecast the variation of quality, quantity and on-time delivery with respect to time. However, more work is needed to further develop the model due to the complexity of the time series analysis. Detailed analysis of trends, seasonality, abrupt changes and other entities of a time series model is essential for a well-developed and advanced forecasting framework.

6 CONCLUSION

6.1 Introduction

As the complexity of products, machines, parts and processes for manufacturing is increasing there is a growing need for more stable and reliable ontology standards and intelligent systems, that can enable not only interoperability between the entities in the manufacturing enterprise but also beyond its borders. Traditional standardization bodies are constantly making efforts to reach a certain level of alignment for all the manufacturing systems (as discussed in this study). There is still more research required for a full realisation of an intelligent and interoperable manufacturing system. The distribution of decision making can be fully realized through the use of completely interoperable systems, set on globally accepted ontology standards. More adaptable and flexible equipment integration can be enabled through the interconnected portals for collaborative partner relationships and the availability of shared software. Currently, industrial automation vendors and manufacturers of machine tools have significantly adapted to software for interoperability and software is being improved to meet the high demands of interoperability. IT players are also making a radical shift towards developing production related applications, shop-floor applications and data driven services; as the need is steadily rising. In the light of the above, this thesis tried to address interoperability, automation and prediction issues.

The overall aim of this research was to demonstrate efficient creation and re-configuration of intelligent manufacturing supply chain by facilitating data exchange and interoperability

among distributed manufacturing entities through the use of ontology. This was achieved through the completion of the following objectives:

- A literature review of intelligent manufacturing system and its enablers
- Identification of the problem, extraction of industrial case study data and development of a semantic web of manufacturing data through ontology
- Development of an intelligent query system for creating the initial data conditions for modelling, scheduling and production prediction
- Design of algorithms for an intelligent decision support system and the application of Artificial Intelligence (AI) and a multi-agent system (MAS) for simulating a decentralised manufacturing environment

6.2 Summary of the work presented in the thesis

A detailed literature review was presented in Chapter 2 of this thesis. It tried to elaborate on the notion of Industry 4.0 and big data in the manufacturing domain. The definition of ontology was presented as an enabler of interoperability alongside its manufacturing applications. In addition, the paradigm of predictive manufacturing was discussed with its trending industrial and academic application. Finally, the concept of a multi-agent system for distributed manufacturing scheduling was presented.

The proposed unified framework was built upon three modules namely: intelligent data interoperability and query, agent-based manufacturing for automation and machine learning for prediction. Therefore, Chapter 3 concentrates on the ontology-based intelligent data query. In this chapter, manufacturing data was extracted from a real industrial case study and an ontology was developed in Protégé. The ontology was presented in RDF for calculating the semantic distance of two concepts. For the evaluation, the proposed ontology search protocol was compared with a semantic-free method and the advantages of using ontology for

manufacturing data queries was shown based on multiple criteria. Finally, a learning mechanism was integrated into the ontology search protocol in order to enhance its performance. Through the completion of this framework, a query-ready up-to-date knowledge base was delivered for the purpose of scheduling and prediction in manufacturing supply chains. This will benefit the manufacturers in terms of the availability of the right data at the right time, for day-to-day production planning and control, for the predictive data analytics and for the seamless integration of the legacy and new system interfaces. The proposed ontology will enable the transparency of communication and data sharing as well as reducing ambiguity of product specification. Additionally, the ontology will facilitate the development of the intelligent agents (similar to the concept in Chapter 4) that will scout the ontology for specific data retrieval. This data can be used as production key performance indicators and can inform execution agents in the control aspect of production. These data can be used as a data set for machine learning for all agents who are increasingly more reliable in the decision-making process.

Chapter 4 addresses the agent-based distributed manufacturing scheduling. Three different models were developed as part of this chapter. The preliminary model was a classical model based on a centralised paradigm, which helped towards the understanding of the problem domain. Then two distinct multi-agent scheduling platforms were developed. The first model was based on the combination of a GA and a MAS; where resource agents are responsible for parallel optimisation with the capability of exchanging their chromosomes (solutions) in a migration phase, while any optimiser agents could be integrated into the system to further optimise the solutions. In contrast, the second model was inspired from a metaheuristic particle swarm optimisation method, where agents move towards the best solution by exchanging their understanding from the environment with the help of an ontology. The

proposed multi-agent scheduling system not only achieved near optimal solutions when compared with operational research benchmarks, but also demonstrated the feasibility of the realisation of distributed manufacturing systems through heterogeneous intelligent agents. The scheduling system could be further optimised by introducing more combinatorial optimisation techniques and also agents could be further developed for automated data queries. The welfare of each individual agent can be further investigated in order to develop a more sophisticated and reliable distributed system.

The data, which was extracted from GFM, was the basis for the prediction model in Chapter 5. In the initial stage, a list of questions was presented to fulfil the prediction purposes. The aim was to achieve a predictive model by answering these questions. In the second phase, the feature selection was performed to find the suitable data for prediction and classification. Based on the data pre-processing in the second phase, a predictive model was developed using a support vector machine, decision tree and time series methods. The model was able to answer the shortlisted questions and predict the supply chain behaviour. The successful progress of the research leads to an improvement of the study of supply chain informatics, enterprise resource planning and the operational research community. A number of solutions to important technical challenges will have to be engineered which will be of major benefit and interest to the research community. As an example, one such challenge can be data query engineering for predictive manufacturing. Another important aspect of the proposed model, is its superior scalability and efficiency compared to classical decision support systems such as decision tree. This is because of the use of ontology and rule-based reasoning instead of a simple database and some limited if-then rules. Also to show the scalability of the presented work, an intelligent system was developed where ontology creation and SWRL rule generation are performed automatically. This would lead to implementation of the proposed

model in larger and more complex problems, and it can even be further applied in other fields of research concerning data interoperability and knowledge extraction.

6.3 Contribution of the thesis

The contributions of this study within the intelligent manufacturing domain are as follows:

- Construction of a novel ontology-based data query system with the application in the manufacturing supply chain

  The proposed ontology-based data query system introduced a novel and efficient information retrieval system within the manufacturing supply chain. Additionally, the integration of the proposed ontology-based framework with agent-based modelling will help to automate the data query and also to automate the development of the ontology itself.

- Development of an agent-based combinatorial manufacturing routing with an ability to provide independent planning and scheduling

  Product-resource-order-staff architecture was used for modelling the entities and their interactions on the shop floor within the context of routing formation and routing re-configuration. The planning involving job allocation and scheduling process is decentralised so that the computation capability of the machines can be summoned to carry out the planning process

- Creation of a new predictive manufacturing model aligned with the notion of knowledge-base and rule-based reasoning.

A predictive manufacturing model was developed based on a real manufacturing case study. The model attempts to answer tangible industrial questions within the concept of predictive manufacturing by implementing machine learning and data mining techniques. The model has the capability of automatic ontology and rule generation for reasoning purposes. The proposed system benefits from the application of ontology and rule-based reasoning, which is original and unique within the predictive manufacturing paradigm.

6.4 Future works

This thesis, is a demonstration of the development of an intelligent ontology-based data query system; the development of heuristics and optimisation of manufacturing systems through the use of agent-based technologies; the application of artificial intelligence techniques for simulating the manufacturing environment; and the use of intelligent software agents and linked data for predictive manufacturing. As for the application of semantic web technologies, the ontology can be further developed and enriched by mapping its main concepts to WordNet lexical ontology and MASON for future studies. This will ensure that the implementation process is facilitated at scale and the alignments of the ontologies will lead to an improved knowledge base, which will allow more data to be queried from the system. Also the introduction of a multi-agent system for the automation of the query process is highly recommended. The agents find the answers to queries in concord while populating and enhancing the ontology automatically.

Future research work can be built upon the agent-based scheduling chapter, where the conflict resolution routine during the planning and scheduling will be further investigated in

order to find the optimum architecture for the proposed scheduling model. Furthermore, the development of various sets of optimized rules for controlling the disturbance during the scheduling process can enhance the performance of the propsed platform.

With regards to the prediction model, the research was limited to few machine learning algorithms. In order to widen the scope of the project, further research can investigate other approaches to improve the reliability of the prediction model as well as its cohesion.

Appendices

Appendix A

SUPPLEMENTARY MATERIALS FOR CHAPTER 3: ONTOLOGY-BASED INTELLIGENT DATA QUERY

A.1 Sample data of GFM's suppliers

Table A.1 Sample data of a workshop with machining processes competency

FLEET OF MACHINES	DETAILS
MAZAK	MAX TURNING DIAMETER: 250 mm
MULTIPLEX 6250	MAX TURNING LENGTH 150 mm
OKUMA LB3000EX	MAX TURNING DIAMETER: 370 mm MAX TURNING LENGTH 1250 mm
OKUMA LB300M	MAX TURNING DIAMETER:370 mm MAX TURNING LENGTH 1250 mm
OKUMA LB25II	MAX TURNING DIAMETER: 460 mm MAX TURNING LENGTH 650 mm
OKUMA LB15II	MAX TURNING DIAMETER: 370 mm MAX TURNING LENGTH 1000 mm
MAZAK VARIAXIS 500	DIMENSIONE TAVOLA 500X405 ALTEZZA UTILE 460
MAZAK NEXUS 510	TABLE SIZE 1000X700 HEIGHT 460
BRIGEPORT WMC1000	TABLE SIZE 1000X500 HEIGHT 400
BRIGEPORT WMC760	TABLE SIZE 760X450 HEIGHT 400
TAKISAWA MACV1-F	TABLE SIZE 430X350 HEIGHT 460
MAZAK NEXUS 250MSY	MAX TURNING DIAMETER: 70 mm
MAZAK NEXUS 200M	MAX TURNING DIAMETER:60 mm
MAZAK SQT200M	MAX TURNING DIAMETER: 50 mm
OKUMA LB10 M	MAX TURNING DIAMETER: 50 mm
OKUMA LB3000EX	MAX TURNING DIAMETER: 80 mm
MAZAK NEXUS 250M	MAX TURNING DIAMETER: 70 mm
MAZAK NEXUS 250MY	MAX TURNING DIAMETER: 70 mm
OKUMA LB15	MAX TURNING LENGTH 500 mm
OKUMA LB12	MAX TURNING LENGTH 370 mm
MAZAK integrex j400	MAX TURNING LENGTH 600 mm
MAZAK integrex i300	MAX TURNING LENGTH 350 mm
CERTIFICATIONS	ISO 9001:2008; CERTIFICATE NO. 6391; EXPIRY DATE: 28/04/2018
NDT CONTROLS	PENETRATION TEST, HARDNESS TEST; VISUAL TEST
MATERIALS	21CRMOV57; X15CR13 (X12CR13); X25CRMOWV12; X30Cr13; X5CRNI18-10+1E; 1.0038; 1.0481; 1.0737; 1.22.EK.01; 1.22.GC.65; 1.2764; 1.4435; 1.4541; 1.4550; 1.4923; 1.4923 + QT-2; 1.7711; 10.9; 10269; 11SMN30 + C; 11SMNPB30; 11SMNPB37; 12CRMO19.5; 13CRMO4-5; 13CRMO4-5 Z15; 15MO3; 15MO3 - Z15; 16MNCR5+A; 16MNCR5G; 16MO3; 16MO3 - Z15; 2.4668; 20MN5; 20MNCR5; 20MNCR5 & E6; 20MNCR5G; 21CrMoV511; 21CRMOV57; 21CRMOV5-7; 21CRMOV5-7+QT; 24CRMO4+QT; 24CRMO5; 25CRMO4; 25CRMO4+QT; 26NICRMO146; 304; 30CRMOV9; 30CRMOV9 & BN; 30CRMOV9+QT; 316; 316TI; 321; 347;

Table A.2 Sample data of a raw material supplier

MATERIALS	21CRMOV57; 30CRMOV9; AISI 410; EN 10250-4; EN 10269; INCONEL 625; NICR22MO9NB; S235J2G3+N / S235JR+N; S235JR; S235JR+N/S235J2G3+N; S235JRG2+N; X12CR13+QT650; X20CR13/QT700; X20CR137QT800; X22CRMOV121; X22CRMOV12-1; X22CRMOV121 + QT1; X22CRMOV121
NON DESTRUCTIVE TEST	RADIOGRAPHIC INSPECTIONS DYE PENETRATE INSPECTIONS MAGNETIC PARTICLES INSPECTIONS ULTRASONIC INSPECTIONS HARDNESS TESTING FERRITE TESTING POSITIVE MATERIAL IDENTIFICATION MOBILE SPECTROTESTS
DESTRUCTIVE TEST	MECHANICAL TEST: TENSILE TESTING, IMPACT TESTING, BENDING, ETC. CHEMICAL ANALYSIS: SPECTROGRAPHIC EXAMINATION, STATIONARY SPECTROMETER METALLOGRAPHIC EXAMINATIONS
MAGNETIC TEST	EN10228-1 CL.4
SPECIFICS	426W642RA; EN 10088-3; 424W805RA/DGTLV804442001RB; DGTLV824831001RA; TLV9248.02-16 (2012-07) + 424W739RA; 601W141RA; DGTLV804443001RB; DGTLV-804443001RB; VA-PS-08-99999-A-03 REV.01; 423W846R0; DGTLV804442001RB/424W805RA; 424W795R0; 424W795R0/DGTLV817041001RA; 601W141RA/; TLV9248.02-16RI; DGTLV-817041001RC; TLV924802-16RI (2012-07); 423W480R0 - ESCLUSO IL NICHEL; 424W739R0; 427W278R0; EN 10250-2; TLV924802 (REV.10-2003) WTLV9248.02; DGTLV824831001RA; EN10025-2 2005; EN 10025-2 2005 MUN500.8/1 (REV.08-1988)
DIMENSIONS	530 MM; 495 MM; 485 0/+1 MM; 450 0/+2 MM; 445 (+1/-1) MM; 414 (0/+2) MM; 365 0/+2 MM; 365 (0/+2) MM; 364 (0/+2) MM; 345 (0/+2) MM; 495 MM; 2000 MM; 1800 MM; 2200 MM; 1950 MM; 2400 MM;
MATERIALS	X22CRMOV121 MULTIPLA X CONI; S235JR; X12CR13+QT650; X22CRMOV121 + QT1; NICR22MO9NB; X20CR13/QT700; S235JR+N/S235J2G3+N; X20CR137QT800; 16MO3; S235J2G3+N / S235JR+N; 30CRMOV9; X22CRMOV121; X22CRMOV12-1; 21CRMOV57; S235JRG2+N
ULTRASONIC TESTS	EN 10228-3 CLASSE A4; EN 10228-3 CL.4 (100%); EN 10228-3 CL.3; 602W443R0; EN 10228-3 CL.3 MIN.; EN10228-3 CL.3; DIN EN 10228-3 CL.3 (3b,3c); EN 10228-3 CL.3 (3b-3c); GN-TP-00-99999-A-234 CL.2
TREATMENTS	VF200291/01; FV200291/01
CERTIFIATION	UN EN ISO 9001:2008

Table A.3 Sample data of a workshop with treatment processes competency

STABILISATION	WTLV9520-01; WTLV9520.01
HEAT TREATMENT	TLV9254.14 2005; WTLV8520.04RA1; WTLV9520.01; 424W834R0; WTLV9520-01; 424W834R0 P. 3.3; ASTM SB-637;423W937R0; 424W948R0; 423W937RA
HARDNESS TEST	424W834R0; EN ISO 6506-1 2006 HB-331-450
MATERIALS	NICR20TIAL (NIMONIC80); NICR19FE19NB5MO3; S355J2+N; GX4CRNI13.4+QT2; 16MO3; 2.4668; INCONE 718; INCONEL 718; NICR19FE19NB5MO3; NICR20TIAL; NICR20TIAL (NIMONIC80 A); NICR20TIAL (NIMONIC80); NIMONIC80; NIMONIC80A; SB-637; X22CRMOV121
SOLUBILIZATION	WTLV9520-01; 424W834R0; WTLV9520.01
SEPCIFICS	ASTM SB-637
CERTIFICATIONS:	ISO 9001:2008; CERTIFICATE NO. 9813; EXPIRY DATE: 14/09/2018 EN 9100:2009; CERTIFICATE NO. 00220-N; EXPIRY DATE: 25/11/2017

Appendix B

SUPPLEMENTARY MATERIALS FOR CHAPTER 5: PREDICTIVE MANUFACTURING

B.1 Decision tree sample model in CSV format

```
PRODUCT_VALUE <= 1.128
|   LOT_VALUE_in_Euros <= 88.55: s2341064 (1601.0/59.0)
|   LOT_VALUE_in_Euros > 88.55
|   |   PRODUCT_VALUE <= 0.8048: s2341064 (120.0/35.0)
|   |   PRODUCT_VALUE > 0.8048: s23410277 (50.0/13.0)
PRODUCT_VALUE > 1.128
|   PRODUCT_VALUE <= 4.69
|   |   QTY <= 50
|   |   |   PRODUCT_VALUE <= 4.016
|   |   |   |   REQUESTED_DELIVERY_DAYS <= 77
|   |   |   |   |   OTD = DELAY: s23410277 (322.0/141.0)
|   |   |   |   |   OTD = ON_TIME
|   |   |   |   |   |   LOT_VALUE_in_Euros <= 90.4: s2341064 (197.0/108.0)
|   |   |   |   |   |   LOT_VALUE_in_Euros > 90.4: s23410277 (52.0/27.0)
|   |   |   |   REQUESTED_DELIVERY_DAYS > 77
|   |   |   |   |   OTD = DELAY: s23410277 (195.0/41.0)
|   |   |   |   |   OTD = ON_TIME
|   |   |   |   |   |   PRODUCT_VALUE <= 2.5802: s23410277 (156.0/57.0)
|   |   |   |   |   |   PRODUCT_VALUE > 2.5802
|   |   |   |   |   |   |   PRODUCT_VALUE <= 3.325: s234107 (89.0/25.0)
|   |   |   |   |   |   |   PRODUCT_VALUE > 3.325
|   |   |   |   |   |   |   |   PRODUCT_VALUE <= 3.7: s23410277 (58.0/12.0)
|   |   |   |   |   |   |   |   PRODUCT_VALUE > 3.7: s234107 (62.0/20.0)
|   |   |   PRODUCT_VALUE > 4.016: s234107 (300.0/98.0)
|   |   QTY > 50: s23410277 (3028.0/795.0)
|   PRODUCT_VALUE > 4.69
|   |   QTY <= 9
|   |   |   REQUESTED_DELIVERY_DAYS <= 80
|   |   |   |   PRODUCT_VALUE <= 415
|   |   |   |   |   REQUESTED_DELIVERY_DAYS <= 32
|   |   |   |   |   |   REQUESTED_DELIVERY_DAYS <= 21
|   |   |   |   |   |   |   LOT_VALUE_in_Euros <= 107.5: s23410277
(357.0/174.0)
|   |   |   |   |   |   |   LOT_VALUE_in_Euros > 107.5
|   |   |   |   |   |   |   |   PRODUCT_VALUE <= 136: s234107 (129.0/79.0)
|   |   |   |   |   |   |   |   PRODUCT_VALUE > 136: s23410277 (57.0/33.0)
|   |   |   |   |   |   REQUESTED_DELIVERY_DAYS > 21
|   |   |   |   |   |   |   REQUESTED_DELIVERY_DAYS <= 24: s23410420
(86.0/40.0)
|   |   |   |   |   |   |   REQUESTED_DELIVERY_DAYS > 24: s23410277
(175.0/62.0)
|   |   |   |   |   REQUESTED_DELIVERY_DAYS > 32: s234107 (710.0/362.0)
```

```
|   |   |   |   PRODUCT_VALUE > 415: s23410225 (55.0/30.0)
|   |   |   REQUESTED_DELIVERY_DAYS > 80
|   |   |   |   PRODUCT_VALUE <= 55
|   |   |   |   |   LOT_VALUE_in_Euros <= 27.6
|   |   |   |   |   |   LOT_VALUE_in_Euros <= 21.315: s234107 (85.0/18.0)
|   |   |   |   |   |   LOT_VALUE_in_Euros > 21.315: s23410277 (92.0/34.0)
|   |   |   |   |   LOT_VALUE_in_Euros > 27.6
|   |   |   |   |   |   PRODUCT_VALUE <= 15
|   |   |   |   |   |   |   PRODUCT_VALUE <= 10.6312: s234107 (409.0/92.0)
|   |   |   |   |   |   |   PRODUCT_VALUE > 10.6312
|   |   |   |   |   |   |   |   REQUESTED_DELIVERY_DAYS <= 168: s234107
(114.0/62.0)
|   |   |   |   |   |   |   |   REQUESTED_DELIVERY_DAYS > 168: s23410258
(115.0/30.0)
|   |   |   |   |   |   PRODUCT_VALUE > 15: s234107 (617.0/112.0)
|   |   |   |   PRODUCT_VALUE > 55
|   |   |   |   |   PRODUCT_VALUE <= 109
|   |   |   |   |   |   PRODUCT_VALUE <= 60.9: s23410258 (94.0/1.0)
|   |   |   |   |   |   PRODUCT_VALUE > 60.9: s234107 (112.0/25.0)
|   |   |   |   |   PRODUCT_VALUE > 109
|   |   |   |   |   |   LOT_VALUE_in_Euros <= 729.44: s23410277 (55.0/26.0)
|   |   |   |   |   |   LOT_VALUE_in_Euros > 729.44: s23410258 (66.0/3.0)
|   |   QTY > 9
|   |   |   LOT_VALUE_in_Euros <= 225.12
|   |   |   |   QTY <= 10
|   |   |   |   |   REQUESTED_DELIVERY_DAYS <= 35: s234107 (65.0/32.0)
|   |   |   |   |   REQUESTED_DELIVERY_DAYS > 35: s23410277 (100.0/36.0)
|   |   |   |   QTY > 10
|   |   |   |   |   REQUESTED_DELIVERY_DAYS <= 18: s234107 (130.0/75.0)
|   |   |   |   |   REQUESTED_DELIVERY_DAYS > 18
|   |   |   |   |   |   REQUESTED_DELIVERY_DAYS <= 31: s23410277
(105.0/41.0)
|   |   |   |   |   |   REQUESTED_DELIVERY_DAYS > 31: s234107 (663.0/186.0)
|   |   |   LOT_VALUE_in_Euros > 225.12
|   |   |   |   PRODUCT_VALUE <= 7.3625
|   |   |   |   |   PRODUCT_VALUE <= 7.178
|   |   |   |   |   |   PRODUCT_VALUE <= 4.9
|   |   |   |   |   |   |   QTY <= 163: s2341033 (98.0/37.0)
|   |   |   |   |   |   |   QTY > 163: s23410277 (107.0/31.0)
|   |   |   |   |   |   PRODUCT_VALUE > 4.9
|   |   |   |   |   |   |   PRODUCT_VALUE <= 5.335: s234107 (322.0/114.0)
|   |   |   |   |   |   |   PRODUCT_VALUE > 5.335
|   |   |   |   |   |   |   |   PRODUCT_VALUE <= 6: s23410277 (162.0/39.0)
|   |   |   |   |   |   |   |   PRODUCT_VALUE > 6: s234107 (213.0/57.0)
|   |   |   |   |   PRODUCT_VALUE > 7.178
|   |   |   |   |   |   PRODUCT_VALUE <= 7.2: s2341033 (115.0/3.0)
|   |   |   |   |   |   PRODUCT_VALUE > 7.2: s23410277 (97.0/31.0)
|   |   |   |   PRODUCT_VALUE > 7.3625
|   |   |   |   |   PRODUCT_VALUE <= 9.9: s234107 (561.0/46.0)
|   |   |   |   |   PRODUCT_VALUE > 9.9
|   |   |   |   |   |   PRODUCT_VALUE <= 10.05: s23410277 (78.0/22.0)
|   |   |   |   |   |   PRODUCT_VALUE > 10.05: s234107 (1887.0/714.0)
```

B.2 SWRL rules for training set in the prediction model

Rule 1: Conformity is positive one when order is 'conform'.

details(?od) ^ hasConfirmity(?od, ?oc) ^ swrlb:equal(?oc, "conform") -> hasOrderConfirmity(?od, 1)

Rule 2: Conformity is zero when order is non conform.

details(?od) ^ hasConfirmity(?od, ?oc) ^ swrlb:equal(?oc, "non conform") -> hasOrderConfirmity(?od, 0)

Rule 3: Order value is positive one when order value is greater than average value.

details(?od) ^ hasValue(?od, ?ov) ^ hasAverageValue(?od, ?av) ^ swrlb:greaterThan(?ov, ?av) -> hasOrderValue(?od, 1)

Rule 4: Order value is positive one when value is equal to average.

details(?od) ^ hasValue(?od, ?ov) ^ hasAverageValue(?od, ?av) ^ swrlb:equal(?ov, ?av) -> hasOrderValue(?od, 1)

Rule 5: Order value is zero when order value is below average value.

details(?od) ^ hasValue(?od, ?ov) ^ hasAverageValue(?od, ?av) ^ swrlb:lessThan(?ov, ?av) -> hasOrderValue(?od, 0)

Rule 6: Quantity is positive one when order quantity is greater than average quantity.

details(?od) ^ hasQuantity(?od, ?oq) ^ hasAverageQuantity(?od, ?aq) ^ swrlb:greaterThan(?oq, ?aq) -> hasOrderQuantity(?od, 1)

Rule 7: Quantity is zero when order quantity is below average quantity.

details(?od) ^ hasQuantity(?od, ?oq) ^ hasAverageQuantity(?od, ?aq) ^ swrlb:lessThan(?oq, ?aq) -> hasOrderQuantity(?od, 0)

Rule 8: Order delay days is zero when Delay is positive.

details(?od) ^ swrlb:greaterThan(?dl, 0) ^ hasDelay(?od, ?dl) -> hasOrderDelayDays(?od, 0)

Rule 9: Order delay days is zero when Delay is equal to 0.

details(?od) ^ swrlb:equal(?dl, 0) ^ hasDelay(?od, ?dl) -> hasOrderDelayDays(?od, 0)

Rule 10: Order delay days is one when delay is below zero.

details(?od) ^ hasDelay(?od, ?dl) ^ hasZero(?od, ?zero) ^ swrlb:lessThan(?dl, ?zero) -> hasOrderDelayDays(?od, 1)

Rule 11: Pass Large Quantity Test when Order Confirmity is one and Order Quantity is one.

details(?od) ^ hasOrderConfirmity(?od, ?oc)^ swrlb:equal(?oq, 1) ^ verification(?test)^ swrlb:equal(?testID, "LargeQuantityTest") ^ hasOrderQuantity(?od, ?oq) ^ hasTestID(?test, ?testID) ^swrlb:equal(?oc, 1) -> hasPassed(?od, ?test)

Rule 12: Fail Large Quantity Test when Order Confirmity is zero and Order Quantity is one.

details(?od) ^ hasOrderConfirmity(?od, ?oc) ^ swrlb:equal(?oc, 0) ^ hasOrderQuantity(?od, ?oq) ^ swrlb:equal(?oq, 1) ^ verification(?test) ^ hasTestID(?test, ?testID) ^ swrlb:equal(?testID, "LargeQuantityTest") -> hasFailed(?od, ?test)

Rule 13: Pass Small Quantity Test when Order Confirmity is one and Order Quantity is zero.

details(?od) ^ hasOrderConfirmity(?od, ?oc) ^ swrlb:equal(?oc, 1) ^ hasOrderQuantity(?od, ?oq) ^ swrlb:equal(?oq, 0) ^ verification(?test) ^ hasTestID(?test, ?testID) ^ swrlb:equal(?testID, "SmallQuantityTest") -> hasPassed(?od, ?test)

Rule 14: Fail Small Quantity Test when Order Confirmity is zero and Order Quantity is zero.

details(?od) ^ hasOrderConfirmity(?od, ?oc) ^ hasOrderQuantity(?od, ?oq)^swrlb:equal(?oq, 0) ^ swrlb:equal(?testID, "SmallQuantityTest") ^ hasTestID(?test, ?testID) ^ verification(?test) ^ swrlb:equal(?oc, 0) -> hasFailed(?od, ?test)

Rule 15: Pass High Value Test when Order Confirmity is one and Order Value is one.

swrlb:equal(?testID, "HighValueTest") ^ hasOrderValue(?od, ?ov) ^ hasTestID(?test, ?testID) ^ details(?od) ^ swrlb:equal(?oc, 1) ^ verification(?test) ^ hasOrderConfirmity(?od, ?oc) ^ swrlb:equal(?ov, 1) -> hasPassed(?od, ?test)

Rule 16: Fail High Value Test when Order Confirmity is zero and Order Value is one.

details(?od) ^ hasOrderConfirmity(?od, ?oc) ^ swrlb:equal(?oc, 0) ^ hasOrderValue(?od, ?ov) ^ swrlb:equal(?ov, 1) ^ verification(?test) ^ hasTestID(?test, ?testID) ^ swrlb:equal(?testID, "HighValueTest") -> hasFailed(?od, ?test)

Rule 17: Pass Low Value Test when Order Confirmity is one and Order Value is zero.

details(?od) ^ hasOrderConfirmity(?od, ?oc) ^ swrlb:equal(?oc, 1) ^ hasOrderValue(?od, ?ov) ^ swrlb:equal(?ov, 0) ^ verification(?test) ^ hasTestID(?test, ?testID) ^ swrlb:equal(?testID, "LowValueTest") -> hasPassed(?od, ?test)

Rule 18: Fail Low Value Test when Order Confirmity is zero and Order Value is zero.

hasOrderValue(?od, ?ov) ^ swrlb:equal(?ov, 0) ^ hasTestID(?test, ?testID) ^ swrlb:equal(?testID, "LowValueTest") ^ details(?od) ^ verification(?test) ^ swrlb:equal(?oc, 0) ^ hasOrderConfirmity(?od, ?oc) -> hasFailed(?od, ?test)

Rule 19: Pass On Time Test if Order Confirmity is one and Order Delay Day is one.

hasOrderDelayDays(?od, ?odl) ^ swrlb:equal(?odl, 1) ^ hasTestID(?test, ?testID) ^ details(?od) ^ swrlb:equal(?oc, 1) ^ verification(?test) ^ swrlb:equal(?testID, "OnTimeTest") ^ hasOrderConfirmity(?od, ?oc) -> hasPassed(?od, ?test)

Rule 20: Fail On Time Test if Order Confirmity is zero and Order Delay Days is one.

details(?od) ^ hasOrderConfirmity(?od, ?oc) ^ swrlb:equal(?oc, 0) ^ hasOrderDelayDays(?od, ?odl) ^ swrlb:equal(?odl, 1) ^ verification(?test) ^ hasTestID(?test, ?testID) ^ swrlb:equal(?testID, "OnTimeTest") -> hasFailed(?od, ?test)

Rule 21: Pass Delayed Test when Order Confirmity is one and Order Delay Days is zero.

details(?od) ^ hasOrderConfirmity(?od, ?oc) ^ swrlb:equal(?oc, 1) ^ hasOrderDelayDays(?od, ?odl) ^ swrlb:equal(?odl, 0) ^ verification(?test) ^ hasTestID(?test, ?testID) ^ swrlb:equal(?testID, "DelayedTest") -> hasPassed(?od, ?test)

Rule 22: Fail Delayed Test when Order Confirmity is zero and Order Delay Days is zero.

hasOrderDelayDays(?od, ?odl) ^ swrlb:equal(?testID, "DelayedTest") ^ swrlb:equal(?odl, 0) ^ hasTestID(?test, ?testID) ^ details(?od) ^ verification(?test) ^ swrlb:equal(?oc, 0) ^ hasOrderConfirmity(?od, ?oc) -> hasFailed(?od, ?test)

B.3 SWRL rules for test set in the prediction model

Rule 1: Quantity is one when Order Quantity is greater than Quantity Boundary.

details(?o)^hasOrderQuantity(?o,?oq)^hasQuantityBoundary(?o,?qb)^swrlb:greaterThan(?oq, ?qb)->hasQuantity(?o,1)

Rule 2: Quantity is zero when Order Quantity is less than Quantity Boundary.

details(?o)^hasOrderQuantity(?o,?oq)^hasQuantityBoundary(?o,?qb)^swrlb:lessThan(?oq,?qb)->hasQuantity(?o,0)

Rule 3: Delay is zero when Order Delay Days is less than zero.

details(?o)^hasOrderDelayDays(?o,?odd)^hasZero(?o,?zero)^swrlb:lessThan(?odd,?zero)

->hasDelay(?o,0)

Rule 4: Delay is one when Order Delay Days is greater than zero.

details(?o)^hasOrderDelayDays(?o,?odd)^hasZero(?o,?zero)^swrlb:greaterThan(?odd,?zero)

->hasDelay(?o,1)

Rule 5: Delay is one when Order Delay Days is equal to zero.

details(?o)^hasOrderDelayDays(?o,?odd)^hasZero(?o,?zero)^swrlb:equal(?odd,?zero)

->hasDelay(?o,1)

Rule 6: Value is zero when order Value is less than 100.

details(?o)^hasOrderValue(?o,?ov)^swrlb:lessThan(?ov,100)-> hasValue(?o, 0)

Rule 7: Value is zero when order Value is equal to 100.

details(?o)^hasOrderValue(?o,?ov)^swrlb:equal(?ov,100)-> hasValue(?o, 0)

Rule 8: Value is one when order Value is between 100 and 400.

details(?o)^hasOrderValue(?o,?ov)^swrlb:greaterThan(?ov,100)^swrlb:lessThan(?ov,400)->hasValue(?o,1)

Rule 9: Value is two when order Value is between 400 and 1154.

details(?o)^hasOrderValue(?o,?ov)^swrlb:greaterThan(?ov,400)^swrlb:lessThan(?ov,1154)->hasValue(?o,2)

Rule 10: Value is three when order Value is between 1154 and 1500.

details(?o)^hasOrderValue(?o,?ov)^swrlb:greaterThan(?ov,1154)^swrlb:lessThan(?ov, 1500)->hasValue(?o,3)

Rule 11: Value is four when order Value is between 1500 and 3700.

details(?o)^hasOrderValue(?o,?ov)^swrlb:greaterThan(?ov,1500)^swrlb:lessThan(?ov, 3700)->hasValue(?o,4)

Rule 12: Value is five when order Value is between 3700 and 3799.

details(?o)^hasOrderValue(?o,?ov)^swrlb:greaterThan(?ov,3700)^swrlb:lessThan(?ov, 3799)->hasValue(?o,5)

Rule 13: Value is six when order Value is greater than 3799.

details(?o)^hasOrderValue(?o,?ov)^swrlb:greaterThan(?ov,3799)^ ->hasValue(?o,6)

Rule 14: Fail conformity test when Quantity is 1, Delay is 1, Value is 5.

details(?o)^hasQuantity(?o,?q)^swrlb:equal(?q,1)^hasDelay(?o,?d)^swrlb:equal(?d,1)^

hasValue(?o,?v)^swrlb:equal(?v,5)^verifications(?test)^hasTestID(?test,?testID)^swrlb:equal(?testID,”ConformityTest”)->isFailedBy(?test,?o)

Rule 15: Pass conformity test when Quantity is 1, Delay is 1, Value is not 5.

details(?o)^hasQuantity(?o,?q)^swrlb:equal(?q,1)^hasDelay(?o,?d)^swrlb:equal(?d,1)^

hasValue(?o,?v)^swrlb:notEqual(?v,5)^verifications(?test)^hasTestID(?test,?testID)^swrlb:equal(?testID,”ConformityTest”)->isPassedBy(?test,?o)

Rule 16: Pass conformity test when Quantity is 1, Delay is 0.

details(?o)^hasQuantity(?o,?q)^swrlb:equal(?q,1)^hasDelay(?o,?d)^swrlb:equal(?d,0)

^verifications(?test)^hasTestID(?test,?testID)^swrlb:equal(?testID,”ConformityTest”)->isPassedBy(?test,?o)

Rule 17: Fail conformity test when Quantity is 0, Delay is 1, Value is 3.

details(?o)^hasQuantity(?o,?q)^swrlb:equal(?q,0)^hasDelay(?o,?d)^swrlb:equal(?d,1)^

hasValue(?o,?v)^swrlb:equal(?v,3)^verifications(?test)^hasTestID(?test,?testID)^swrlb:equal(?testID,”ConformityTest”)->isFailedBy(?test,?o)

Rule 18: Pass conformity test when Quantity is 0, Delay is 1, Value is not 3.

details(?o)^hasQuantity(?o,?q)^swrlb:equal(?q,0)^hasDelay(?o,?d)^swrlb:equal(?d,1)^

hasValue(?o,?v)^swrlb:notEqual(?v,3)^verifications(?test)^hasTestID(?test,?testID)^swrlb:equal(?testID,”ConformityTest”)->isPassedBy(?test,?o)

Rule 19: Fail conformity test when Quantity is 0, Delay is 0, Value is 1.

details(?o)^hasQuantity(?o,?q)^swrlb:equal(?q,0)^hasDelay(?o,?d)^swrlb:equal(?d,0)^

hasValue(?o,?v)^swrlb:equal(?v,1)^verifications(?test)^hasTestID(?test,?testID)^swrlb:equal(?testID,”ConformityTest”)->isFailedBy(?test,?o)

Rule 20: Pass conformity test when Quantity is 0, Delay is 0, Value is not 1.

details(?o)^hasQuantity(?o,?q)^swrlb:equal(?q,0)^hasDelay(?o,?d)^swrlb:equal(?d,0)^

hasValue(?o,?v)^swrlb:notEqual(?v,1)^verifications(?test)^hasTestID(?test,?testID)^swrlb:equal(?testID,”ConformityTest”)->isPassedBy(?test,?o)

B.4 MATLAB program for SWRL rule generation

```
clc;
swrl = string(); % going to store rules in this array
rulesCount = 1;  % going to count number of rules
temp = cell(2,size(DecisionTree,2));

% Scanning through the Decision Tree cell which is imported from the Text
file
for i = 1:size(DecisionTree,1)

    for j = 1:size(DecisionTree,2)
        cellVal = DecisionTree{i,j};
        switch cellVal
            case {'|'}
                % do nothing
            case {''}
                break % break the column scanning and move to new row
            otherwise
                v = char(DecisionTree{i,j});
                v = v(1,1); % storing the first char of the cell

                if(~strcmp(DecisionTree{i,j},temp{1,j}))
                    temp{1,j} = DecisionTree{i,j};
                    if(~contains(DecisionTree{i,j+2},':'))
                        if(contains(DecisionTree{i,j+1},'<='))
                            format = '^ %s(?od, ?%c) ^
swrlb:lessThanOrEqual(?%c, "%s"^^xsd:decimal) ';
                            c =
sprintf(format,DecisionTree{i,j},v,v,DecisionTree{i,j+2});
                            swrl(rulesCount,1) = swrl(rulesCount,1) + c;
                            temp{2,j} = swrl(rulesCount,1);

                        elseif(contains(DecisionTree{i,j+1},'>'))

                            format = '^ %s(?od, ?%c) ^
swrlb:greaterThanOrEqual(?%c, "%s"^^xsd:decimal) ';
                            c =
sprintf(format,DecisionTree{i,j},v,v,DecisionTree{i,j+2});
                            swrl(rulesCount,1) = swrl(rulesCount,1) + c;
                            temp{2,j} = swrl(rulesCount,1);

                        elseif(contains(DecisionTree{i,j+1},'='))


                            format = '^ %s(?od, ?%c) ^ swrlb:equal(?%c,
"%s") ';
                            c =
sprintf(format,DecisionTree{i,j},v,v,DecisionTree{i,j+2});
                            swrl(rulesCount,1) = swrl(rulesCount,1) + c;
                            temp{2,j} = swrl(rulesCount,1);
                        end

                    elseif(contains(DecisionTree{i,j+2},':'))
                        d = char(DecisionTree{i,j+2});
```

```
                        d = d(1:end-1);
                            if(contains(DecisionTree{i,j+1},'<='))

                                format = '^ %s(?od, ?%c) ^
swrlb:lessThanOrEqual(?%c, "%s"^^xsd:decimal) -> %s(?od)';
                                c =
sprintf(format,DecisionTree{i,j},v,v,d,DecisionTree{i,j+3});
                                temp{2,j} = swrl(rulesCount,1);
                                swrl(rulesCount,1) = swrl(rulesCount,1) +
c;

                                rulesCount = rulesCount + 1;
                                swrl(rulesCount,1) = "";
                                break
                            elseif(contains(DecisionTree{i,j+1},'>'))

                                format = '^ %s(?od, ?%c) ^
swrlb:greaterThanOrEqual(?%c, "%s"^^xsd:decimal) -> %s(?od)';
                                c =
sprintf(format,DecisionTree{i,j},v,v,d,DecisionTree{i,j+3});
                                swrl(rulesCount,1) = temp{2,j} + c;

                                rulesCount = rulesCount + 1;
                                swrl(rulesCount,1) = "";
                                break
                            elseif(contains(DecisionTree{i,j+1},'='))


                                format = '^ %s(?od, ?%c) ^ swrlb:equal(?%c,
"%s") -> %s(?od)';
                                c = sprintf(format,DecisionTree{i,j},v,v,d,
DecisionTree{i,j+3});
                                temp{2,j} = swrl(rulesCount,1);
                                swrl(rulesCount,1) = temp{2,j} + c;

                                rulesCount = rulesCount + 1;
                                swrl(rulesCount,1) = "";
                                break
                            end
                        end
                    elseif(strcmp(DecisionTree{i,j},temp{1,j}))
                        if(~contains(DecisionTree{i,j+2},':'))

                            if(contains(DecisionTree{i,j+1},'>'))

                                format = '^ %s(?od, ?%c) ^
swrlb:greaterThanOrEqual(?%c, "%s"^^xsd:decimal) ';
                                c =
sprintf(format,DecisionTree{i,j},v,v,DecisionTree{i,j+2});
                                if(j == 1)
                                    swrl(rulesCount,1) = c;
                                elseif(j~= 1)
                                    swrl(rulesCount,1) = temp{2,j-1} + c;
                                end
                                temp{2,j} = swrl(rulesCount,1);

                            elseif(contains(DecisionTree{i,j+1},'='))
```

```
                            format = '^ %s(?od, ?%c) ^ swrlb:equal(?%c,
"%s") ';
                            c =
sprintf(format,DecisionTree{i,j},v,v,DecisionTree{i,j+2});
                                if(j == 1)
                                    swrl(rulesCount,1) = c;
                                elseif(j~= 1)
                                    swrl(rulesCount,1) = temp{2,j-1} + c;
                                end
                            temp{2,j} = swrl(rulesCount,1);
                        end

                    elseif(contains(DecisionTree{i,j+2},':'))
                        d = char(DecisionTree{i,j+2});
                        d = d(1:end-1);
                            if(contains(DecisionTree{i,j+1},'>'))

                                format = '^ %s(?od, ?%c) ^
swrlb:greaterThanOrEqual(?%c, "%s"^^xsd:decimal) -> %s(?od)';
                                c =
sprintf(format,DecisionTree{i,j},v,v,d,DecisionTree{i,j+3});
                                if(j == 1)
                                    swrl(rulesCount,1) = c;
                                elseif(j~= 1)
                                    swrl(rulesCount,1) = temp{2,j-1} + c;
                                end
                                rulesCount = rulesCount + 1;
                                swrl(rulesCount,1) = "";
                                break
                            elseif(contains(DecisionTree{i,j+1},'='))


                                format = '^ %s(?od, ?%c) ^ swrlb:equal(?%c,
"%s") -> %s(?od)';
                                c = sprintf(format,DecisionTree{i,j},v,v,d,
DecisionTree{i,j+3});
                                if(j == 1)
                                    swrl(rulesCount,1) = c;
                                elseif(j~= 1)
                                    swrl(rulesCount,1) = temp{2,j-1} + c;
                                end
                                rulesCount = rulesCount + 1;
                                swrl(rulesCount,1) = "";
                                break
                            end
                    end

                end
            end
        end
    end

swrl1 = string();
for i = 1:rulesCount-1
    swrl1(i,1) = 'Order(?od) ' + swrl(i,1);
end
% writing the data to a Text file
file = fopen('SwrlRules.txt','w');
fprintf(file,'%s \r\n',swrl1);
fclose('all');
```

List of References

Adhau, S., M. Mittal and A. Mittal (2012). "A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach." Engineering Applications of Artificial Intelligence **25**(8): 1738-1751.

Ageron, B., A. Gunasekaran and A. Spalanzani (2012). "Sustainable supply management: An empirical study." International Journal of Production Economics **140**(1): 168-182.

Ahmed, S., S. Kim and K. M. Wallace (2007). "A methodology for creating ontologies for engineering design." Journal of computing and information science in engineering **7**(2): 132-140.

Alvarez, E. (2007). "Multi-plant production scheduling in SMEs." Robotics and Computer-Integrated Manufacturing **23**(6): 608-613.

Anand, N., R. van Duin and L. Tavasszy (2014). "Ontology-based multi-agent system for urban freight transportation." International Journal of Urban Sciences **18**(2): 133-153.

Applegate, D. and W. Cook (1991). "A computational study of the job-shop scheduling problem." **3**(2): 149-156.

Arour, K. and T. Yeferny (2015). "Learning model for efficient query routing in P2P information retrieval systems." Peer-to-Peer Networking and Applications **8**(5): 741-757.

Artto, K., I. Kulvik, J. Poskela and V. Turkulainen (2011). "The integrative role of the project management office in the front end of innovation." International Journal of Project Management **29**(4): 408-421.

Asadzadeh, L. and K. Zamanifar (2010). "An agent-based parallel approach for the job shop scheduling problem with genetic algorithms." Mathematical and Computer Modelling **52**(11): 1957-1965.

Astrova, I. (2009). "Rules for mapping SQL relational databases to OWL ontologies." Metadata and Semantics: 415-424.

Auschitzky, E., M. Hammer and A. Rajagopaul (2014). "How big data can improve manufacturing." McKinsey & Company.

Aydin, M. E. and T. C. Fogarty (2004). "A simulated annealing algorithm for multi-agent systems: a job-shop scheduling application." Journal of Intelligent Manufacturing **15**(6): 805-814.

Aydin, M. E. and T. C. Fogarty (2004). "Teams of autonomous agents for job-shop scheduling problems: An Experimental Study." Journal of Intelligent Manufacturing **15**(4): 455-462.

Babiceanu, R. F. and F. F. Chen (2006). "Development and applications of holonic manufacturing systems: a survey." Journal of Intelligent Manufacturing **17**(1): 111-131.

Bagheri, B., S. Yang, H.-A. Kao and J. Lee (2015). "Cyber-physical systems architecture for self-aware machines in industry 4.0 environment." IFAC-PapersOnLine **48**(3): 1622-1627.

Bai, D. and L. Tang (2013). "Open shop scheduling problem to minimize makespan with release dates." Applied Mathematical Modelling **37**(4): 2008-2015.

Barbosa, J., P. Leitão, E. Adam and D. Trentesaux (2015). "Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution." Computers in Industry **66**: 99-111.

Bellifemine, F. L., G. Caire and D. Greenwood (2007). Developing multi-agent systems with JADE, John Wiley & Sons.

Bi, Z., L. Da Xu and C. Wang (2014). "Internet of things for enterprise systems of modern manufacturing." IEEE Transactions on industrial informatics **10**(2): 1537-1546.

Bikakis, N. and T. Sellis (2016). "Exploration and visualization in the web of big linked data: A survey of the state of the art." arXiv preprint arXiv:1601.08059.

Bonabeau, E., D. Corne, J. Knowles and R. Poli (2010). "Swarm intelligence theory: A snapshot of the state of the art." Theoretical Computer Science **411**(21): 2081-2083.

Box, G. E., G. M. Jenkins, G. C. Reinsel and G. M. Ljung (2015). Time series analysis: forecasting and control, John Wiley & Sons.

Bughin, J., M. Chui and J. Manyika (2010). "Clouds, big data, and smart assets: Ten tech-enabled business trends to watch." McKinsey Quarterly **56**(1): 75-86.

Cai, H., L. Da Xu, B. Xu, C. Xie, S. Qin and L. Jiang (2014). "IoT-based configurable information service platform for product lifecycle management." IEEE Transactions on Industrial Informatics **10**(2): 1558-1567.

Cai, M., W. Zhang and K. Zhang (2011). "ManuHub: a semantic web system for ontology-based service management in distributed manufacturing environments." IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **41**(3): 574-582.

Cerbah, F. (2010). "Learning ontologies with deep class hierarchies by mining the content of relational databases." Advances in knowledge discovery and management: 271-286.

Chen, F., P. Deng, J. Wan, D. Zhang, A. V. Vasilakos and X. J. I. J. o. D. S. N. Rong (2015). "Data mining for the internet of things: literature review and challenges." **11**(8): 431047.

Chen, M., S. Mao and Y. Liu (2014). "Big data: A survey." Mobile Networks and Applications **19**(2): 171-209.

Cho, J., S. Han and H. Kim (2006). "Meta-ontology for automated information integration of parts libraries." Computer-Aided Design **38**(7): 713-725.

Cowling, P. I., D. Ouelhadj and S. Petrovic (2003). "A multi-agent architecture for dynamic scheduling of steel hot rolling." Journal of Intelligent Manufacturing **14**(5): 457-470.

Cristianini, N. and J. Shawe-Taylor (2000). An introduction to support vector machines and other kernel-based learning methods, Cambridge university press.

Danades, A., D. Pratama, D. Anggraini and D. Anggriani (2016). Comparison of accuracy level K-Nearest Neighbor algorithm and Support Vector Machine algorithm in classification water quality status. System Engineering and Technology (ICSET), 2016 6th International Conference on, IEEE.

Davenport, T. H., J. G. Harris and R. Morison (2010). Analytics at work: Smarter decisions, better results, Harvard Business Press.

Deen, S. M. (2013). Agent-based manufacturing: advances in the holonic approach, Springer.

Deepa, O. and A. Senthilkumar (2016). "SWARM INTELLIGENCE FROM NATURAL TO ARTIFICIAL SYSTEMS: ANT COLONY OPTIMIZATION." Networks (GRAPH-HOC) **8**(1).

Eberhart, R. and J. Kennedy (1995). A new optimizer using particle swarm theory. Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, IEEE.

Esposito, C., M. Ficco, F. Palmieri and A. Castiglione (2015). "A knowledge-based platform for Big Data analytics based on publish/subscribe services and stream processing." Knowledge-Based Systems **79**: 3-17.

Euzenat, J. (2007). Semantic Precision and Recall for Ontology Alignment Evaluation. IJCAI.

FIPA. (2002). "Foundation for Intelligent Physical Agents. Contract Net Interaction Protocol Specification." Retrieved July, 2017, from http://www.fipa.org.

Frey, D., J. Nimis, H. Wörn and P. Lockemann (2003). "Benchmarking and robust multi-agent-based production planning and control." Engineering Applications of Artificial Intelligence **16**(4): 307-320.

Ga, D., D. Djuric and V. Deved (2009). Model driven engineering and ontology development, Springer Science & Business Media.

Gabel, T. (2009). "Multi-agent reinforcement learning approaches for distributed job-shop scheduling problems."

Gao, J., L. Zhang and W. Jiang (2007). Procuring requirements for ERP software based on semantic similarity. Semantic Computing, 2007. ICSC 2007. International Conference on, IEEE.

Gennari, J. H., M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy and S. W. Tu (2003). "The evolution of Protégé: an environment for knowledge-based systems development." International Journal of Human-computer studies **58**(1): 89-123.

Giordani, S., M. Lujak and F. Martinelli (2013). "A distributed multi-agent production planning and scheduling framework for mobile robots." Computers & Industrial Engineering **64**(1): 19-30.

Goldingay, H. and J. Van Mourik (2013). "The effect of load on agent-based algorithms for distributed task allocation." Information sciences **222**: 66-80.

Gruber, T. (1993). "What is an Ontology." WWW Site http://www-ksl. stanford. edu/kst/whatis-an-ontology. html (accessed on 07-09-2004).

Guarino, N., D. Oberle and S. Staab (2009). What is an Ontology? Handbook on ontologies, Springer**:** 1-17.

Guo, Q. and M. Zhang (2009). "A novel approach for multi-agent-based intelligent manufacturing system." Information Sciences **179**(18): 3079-3090.

Hamel, L. H. (2011). Knowledge discovery with support vector machines, John Wiley & Sons.

Hardi, J. (2015). "Protege project/cellfie-plugin." 2018, from https://github.com/protegeproject/cellfie-plugin/wiki.

Harjunkoski, I., C. T. Maravelias, P. Bongers, P. M. Castro, S. Engell, I. E. Grossmann, J. Hooker, C. Méndez, G. Sand and J. Wassick (2014). "Scope for industrial applications of production scheduling models and solution methods." Computers & Chemical Engineering **62**: 161-193.

Hasan, S. K., R. Sarker, D. Essam and D. Cornforth (2009). A genetic algorithm with priority rules for solving job-shop scheduling problems. Natural Intelligence for Scheduling, Planning and Packing Problems, Springer**:** 55-88.

Hellinger, A. and H. Seeger (2011). "Cyber-Physical Systems. Driving force for innovation in mobility, health, energy and production." Acatech Position Paper, National Academy of Science and Engineering(0): 2.

Himali, D. R., S. B. Navathe and S. K. Prasad (2012). OSQR: A framework for ontology-based semantic query routing in unstructured P2P networks. High Performance Computing (HiPC), 2012 19th International Conference on, IEEE.

Hsu, C.-W. and C.-J. Lin (2002). "A comparison of methods for multiclass support vector machines." IEEE transactions on Neural Networks **13**(2): 415-425.

IIC. (2015). "Industrial Internet Consortium: Manufacturing." Retrieved February, 2017, from http://www.iiconsortium.org/vertical-markets/manufacturing.htm.

Jules, G. and M. Saadat (2016). "Agent Cooperation Mechanism for Decentralized Manufacturing Scheduling." IEEE Transactions on Systems, Man, and Cybernetics: Systems.

Jules, G. D., M. Saadat and S. Saeidlou (2013). Ontological extension of PROSA for manufacturing network formation. International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Springer.

Jules, G. D., M. Saadat and S. Saeidlou (2015). "Holonic ontology and interaction protocol for manufacturing network organization." IEEE Transactions on Systems, Man, and Cybernetics: Systems **45**(5): 819-830.

Kagermann, H. (2015). Change through digitization—Value creation in the age of Industry 4.0. Management of permanent change, Springer**:** 23-45.

Kagermann, H., J. Helbig, A. Hellinger and W. Wahlster (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group, Forschungsunion.

Kapoor, H., K. Mehta, D. Puri and S. Saxena (2013). "Survey of various search mechanisms in unstructured peer-to-peer networks." International Journal of Computer Applications **68**(6).

Kernschmidt, K., S. Preißner, C. Raasch and B. Vogel-Heuser (2015). From Selling Products to Providing User Oriented Product-Service Systems–Exploring Service Orientation in the German Machine and Plant Manufacturing Industry. IFIP International Conference on Product Lifecycle Management, Springer.

Khan, S., A. R. Ganguly and A. Gupta (2006). Creating knowledge for Business decision making. Encyclopedia of Knowledge Management, IGI Global**:** 81-89.

Khilwani, N., J. A. Harding and A. K. Choudhary (2009). "Semantic web in manufacturing." Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture **223**(7): 905-924.

Kim, K.-Y., D. G. Manley and H. Yang (2006). "Ontology-based assembly design and information sharing for collaborative product development." Computer-Aided Design **38**(12): 1233-1250.

Kobayashi, S., I. Ono and M. Yamamura (1995). An efficient genetic algorithm for job shop scheduling problems. Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc.

Kola, J. and A. Rector (2007). Importing Spreadsheet data into Protégé: Spreadsheet plug-in. Proc. of the Intl. Protege Conference.

Kolberg, D. and D. Zühlke (2015). "Lean automation enabled by industry 4.0 technologies." IFAC-PapersOnLine **48**(3): 1870-1875.

Komma, V. R., P. K. Jain and N. K. Mehta (2011). "An approach for agent modeling in manufacturing on JADE™ reactive architecture." The International Journal of Advanced Manufacturing Technology **52**(9): 1079-1090.

Kotulski, L., A. Sędziwy and B. Strug (2014). "Translation of graph-based knowledge representation in multi-agent system." Procedia Computer Science **29**: 1048-1056.

Kulon, J., P. Broomhead and D. Mynors (2006). "Applying knowledge-based engineering to traditional manufacturing design." The International Journal of Advanced Manufacturing Technology **30**(9-10): 945-951.

Kurdi, M. (2016). "An effective new island model genetic algorithm for job shop scheduling problem." Computers & operations research **67**: 132-142.

Kurdi, M. (2017). "An improved island model memetic algorithm with a new cooperation phase for multi-objective job shop scheduling problem." **111**: 183-201.

Lapira, E. R. (2012). Fault detection in a network of similar machines using clustering approach, University of Cincinnati.

Lawrence, S. (1984). "Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (supplement)." Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Lee, C., K. Choy, K. Law and G. Ho (2014). "Application of intelligent data management in resource allocation for effective operation of manufacturing systems." Journal of Manufacturing Systems **33**(3): 412-422.

Lee, J., E. Lapira, B. Bagheri and H.-a. Kao (2013). "Recent advances and trends in predictive manufacturing systems in big data environment." Manufacturing Letters **1**(1): 38-41.

Leitão, P., J. Barbosa and D. Trentesaux (2012). "Bio-inspired multi-agent systems for reconfigurable manufacturing systems." Engineering Applications of Artificial Intelligence **25**(5): 934-944.

Lemaignan, S., A. Siadat, J.-Y. Dantan and A. Semenenko (2006). MASON: A proposal for an ontology of manufacturing domain. Distributed Intelligent Systems:

Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on, IEEE.

Li, W., R. Veliyath and J. Tan (2013). "Network characteristics and firm performance: An examination of the relationships in the context of a cluster." Journal of Small Business Management **51**(1): 1-22.

Lin, H. and J. A. Harding (2007). "A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration." Computers in Industry **58**(5): 428-437.

Lin, L., W. Zhang, Y. Lou, C. Chu and M. Cai (2011). "Developing manufacturing ontologies for knowledge reuse in distributed manufacturing environment." International Journal of Production Research **49**(2): 343-359.

Lin, T.-L., S.-J. Horng, T.-W. Kao, Y.-H. Chen, R.-S. Run, R.-J. Chen, J.-L. Lai and I.-H. Kuo (2010). "An efficient job-shop scheduling algorithm based on particle swarm optimization." Expert Systems with Applications **37**(3): 2629-2636.

Liu, J.-S. and K. P. Sycara (1996). Multiagent coordination in tightly coupled task scheduling. Proceedings of the Second International Conference on Multi-Agent Systems.

Luck, M., P. McBurney and C. Preist (2003). Agent technology: enabling next generation computing (a roadmap for agent based computing), AgentLink.

Martin, S., D. Ouelhadj, P. Beullens, E. Ozcan, A. A. Juan and E. K. Burke (2016). "A multi-agent based cooperative approach to scheduling and routing." European Journal of Operational Research **254**(1): 169-178.

McFarlane, D. C. and S. Bussmann (2000). "Developments in holonic production planning and control." Production Planning & Control **11**(6): 522-536.

Meloni, C., D. Pacciarelli and M. Pranzo (2004). "A rollout metaheuristic for job shop scheduling problems." Annals of Operations Research **131**(1): 215-235.

Mohebbi, S. and R. Shafaei (2012). "e-Supply network coordination: the design of intelligent agents for buyer-supplier dynamic negotiations." Journal of intelligent Manufacturing **23**(3): 375-391.

Monostori, L., J. Váncza and S. R. Kumara (2006). "Agent-based systems for manufacturing." CIRP Annals-Manufacturing Technology **55**(2): 697-720.

Mrugalska, B. and M. K. Wyrwicka (2017). "Towards Lean Production in Industry 4.0." Procedia Engineering **182**: 466-473.

Müller-Seitz, G. and J. Sydow (2012). "Maneuvering between networks to lead–A longitudinal case study in the semiconductor industry." Long Range Planning **45**(2): 105-135.

Müller, H., W. Müller, D. M. Squire, S. Marchand-Maillet and T. Pun (2001). "Performance evaluation in content-based image retrieval: overview and proposals." Pattern Recognition Letters **22**(5): 593-601.

Musa, A. B. (2013). "Comparative study on classification performance between support vector machine and logistic regression." International Journal of Machine Learning and Cybernetics **4**(1): 13-24.

Nawijn, M., M. Van Tooren, J. Berends and P. Arendsen (2006). Automated finite element analysis in a knowledge based engineering environment. 44th AIAA Aerospace Science Meeting and Exhibit.

NIST. (2013, January 2013). "Workshop report on foundations for innovation in cyber-physical systems." from https://www.nist.gov/el/upload/CPS-WorkshopReport-%201-30-13-Final.pdf.

Noy, N. F., M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson and M. A. Musen (2001). "Creating semantic web contents with protege-2000." IEEE intelligent systems **16**(2): 60-71.

Owiti, O. H., E. T. O. Omulo, W. Okelo-Odongo and B. Manderick (2014). "Game Theoretic Multi-Agent Algorithms for the Job Shop Scheduling Problem." International Journal of Computer and Information Technology **3**(06).

Owliya, M., M. Saadat, G. G. Jules, M. Goharian and R. Anane (2013). "Agent-based interaction protocols and topologies for manufacturing task allocation." IEEE Transactions on Systems, Man, and Cybernetics: Systems **43**(1): 38-52.

Palaniappan, R., K. Sundaraj and S. Sundaraj (2014). "A comparative study of the svm and k-nn machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals." BMC bioinformatics **15**(1): 223.

Patil, L., D. Dutta and R. Sriram (2005). "Ontology-based exchange of product data semantics." IEEE Transactions on automation science and engineering **2**(3): 213-225.

Pezzella, F., G. Morganti, G. J. C. Ciaschetti and O. Research (2008). "A genetic algorithm for the flexible job-shop scheduling problem." **35**(10): 3202-3212.

Pine, B. J. (1993). Mass customization: the new frontier in business competition, Harvard Business Press.

Quinlan, J. R. (1986). "Induction of decision trees." Machine learning **1**(1): 81-106.

Raghavan, V., P. Bollmann and G. S. Jung (1989). "A critical investigation of recall and precision as measures of retrieval system performance." ACM Transactions on Information Systems (TOIS) **7**(3): 205-229.

Rini, D. P., S. M. Shamsuddin and S. S. Yuhaniz (2011). "Particle swarm optimization: technique, system and challenges." International Journal of Computer Applications **14**(1): 19-26.

Saeidlou, S., M. Saadat and E. Aminisharifi (2016). Intelligent Semantic Query for Manufacturing Supply Chain. Advances in Manufacturing Technology XXX: Proceedings of the 14th International Conference on Manufacturing Research, Incorporating the 31st National Conference on Manufacturing Research, September 6–8, 2016, Loughborough University, UK, IOS Press.

Salazar, D. A., J. I. Vélez and J. C. Salazar (2012). "Comparison between SVM and logistic regression: Which one is better to discriminate?" Revista Colombiana de Estadística **35**(SPE2): 223-237.

Sane, S. S. and A. Shirke (2009). Generating OWL ontologies from a relational databases for the semantic web. Proceedings of the International Conference on Advances in Computing, Communication and Control, ACM.

Sanfilippo, E. M. and S. Borgo (2016). "What are features? An ontology-based review of the literature." Computer-Aided Design **80**: 9-18.

Schalkoff, R. J. (2011). Intelligent systems: Principles, paradigms and pragmatics, Jones & Bartlett Publishers.

Schlenoff, C., P. Denno, R. Ivester, D. Libes and S. Szykman (2000). "An analysis and approach to using existing ontological systems for applications in manufacturing." AI EDAM **14**(4): 257-270.

Sharma, B. and X. Yao (2004). Characterizing genetic algorithm approaches to job shop scheduling. UK Workshop on Computational Intelligence, Loughborough University.

Shen, W. (2002). "Distributed manufacturing scheduling using intelligent agents." IEEE intelligent systems **17**(1): 88-94.

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

SiriusDecisions. (2015). "The Impact of Bad Data on Demand Creation." Retrieved June 2017, from https://www.siriusdecisions.com/TheImpactofBadDataonDemandCreation.

Smola, A. J. and B. Schölkopf (2004). "A tutorial on support vector regression." Statistics and computing **14**(3): 199-222.

Song, F., G. Zacharewicz and D. Chen (2013). "An ontology-driven framework towards building enterprise semantic information layer." Advanced Engineering Informatics **27**(1): 38-50.

Statnikov, A., C. F. Aliferis, D. P. Hardin and I. Guyon (2011). A Gentle Introduction To Support Vector Machines In Biomedicine: Volume 1: Theory and Methods, World Scientific Publishing Co Inc.

Stock, T. and G. Seliger (2016). "Opportunities of sustainable manufacturing in industry 4.0." Procedia Cirp **40**: 536-541.

Sun, W., C.-f. Yuan, T.-y. Gao and Q. Ma (2007). "Research on approach of knowledge acquirement based on artificial neural network and supporting design process." JOURNAL-DALIAN UNIVERSITY OF TECHNOLOGY **47**(2): 190.

Tibbetts, H. (2012). "Integration on the Edge: Data Explosion & Next-Gen Integration." Retrieved June 2017, 2017, from http://www.ebizq.net/blogs/integrationedge/2012/01/.

Tseng, M. M. and S. J. Hu (2014). Mass customization. CIRP Encyclopedia of Production Engineering, Springer**:** 836-843.

Tseng, M. M., J. Jiao and M. E. Merchant (1996). "Design for mass customization." CIRP Annals-Manufacturing Technology **45**(1): 153-156.

Tsujimura, Y. and M. Gen (1996). Genetic algorithms for solving multiprocessor scheduling problems. Asia-Pacific Conference on Simulated Evolution and Learning, Springer.

Uschold, M., M. King, S. Moralee and Y. Zorgios (1998). "The enterprise ontology." The knowledge engineering review **13**(1): 31-89.

Usman, Z., R. Young, N. Chungoora, C. Palmer, K. Case and J. Harding (2013). "Towards a formal manufacturing reference ontology." International Journal of Production Research **51**(22): 6553-6572.

Van Brussel, H., J. Wyns, P. Valckenaers, L. Bongaerts and P. Peeters (1998). "Reference architecture for holonic manufacturing systems: PROSA." Computers in industry **37**(3): 255-274.

Van Leeuwen, E. and D. Norrie (1997). "Holons and holarchies~ intelligent manufacturing systems." manufacturing Engineer **76**(2): 86-88.

Vijayaraghavan, A., W. Sobel, A. Fox, D. Dornfeld and P. Warndorf (2008). "Improving machine tool interoperability using standardized interface protocols: MT connect." Laboratory for Manufacturing and Sustainability.

Vrba, P. and V. Marik (2010). "Capabilities of dynamic reconfiguration of multiagent-based industrial control systems." IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **40**(2): 213-223.

Wang, K. and S. Choi (2014). "A holonic approach to flexible flow shop scheduling under stochastic processing times." Computers & Operations Research **43**: 157-168.

Wang, L. (2005). Support vector machines: theory and applications, Springer Science & Business Media.

Wang, L., D.-B. Tang, W.-B. Gu, K. Zheng, W.-D. Yuan and D.-S. Tang (2012). "Pheromone-based coordination for manufacturing system control." Journal of Intelligent Manufacturing **23**(3): 747-757.

Wang, L., M. Törngren and M. Onori (2015). "Current status and advancement of cyber-physical systems in manufacturing." Journal of Manufacturing Systems **37**(Part 2): 517-527.

Wang, L. and G. Wang (2016). "Big Data in Cyber-Physical Systems, Digital Manufacturing and Industry 4.0." IJEM-International Journal of Engineering and Manufacturing (IJEM) **6**(4): 1.

Wang, Z. and Y. Liu (2006). A multi-agent agile scheduling system for job-shop problem. Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on, IEEE.

Wei, S., M. Qin-Yi and G. Tian-Yi (2009). "An ontology-based manufacturing design system." Information Technology Journal **8**(5): 643-656.

Weigend, A. S. (2018). Time series prediction: forecasting the future and understanding the past, Routledge.

Wong, T., C. Leung, K.-L. Mak and R. Y. Fung (2006). "Dynamic shopfloor scheduling in multi-agent manufacturing systems." Expert Systems with Applications **31**(3): 486-494.

Wu, J. and E. H. Durfee (2006). Mixed-integer linear programming for transition-independent decentralized MDPs. Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM.

Wu, Z. (2005). "Multi-agent workload control and flexible job shop scheduling."

Yeferny, T., A. Bouzeghoub and K. Arour (2011). "A query learning routing approach based on semantic clusters." International Journal of Advanced Information Technology **1**(6): 1.

Yoo, M.-J. and J. P. Müller (2002). "Using multi-agent system for dynamic job shop scheduling."

Zalzala, A. M. and P. J. Fleming (1997). Genetic algorithms in engineering systems, Iet.

Zhao, W. and J. Liu (2008). "OWL/SWRL representation methodology for EXPRESS-driven product information model: Part II: Practice." Computers in Industry **59**(6): 590-600.

Zhong, Z., S. Saeidlou, M. Saadat and A. Abukar (2018). Ontology and Rule-ased Reasoning for Intelligent Predictive Manufacturing. Advances in Manufacturing Technology XXXII: Proceedings of the 16th International Conference on Manufacturing Research, incorporating the 33rd National Conference on Manufacturing Research, September 11–13, 2018, University of Skövde, Sweden, IOS Press.

Zhou, J. and R. Dieng-Kuntz (2004). Manufacturing ontology analysis and design: towards excellent manufacturing. Industrial Informatics, 2004. INDIN'04. 2004 2nd IEEE International Conference on, IEEE.

Publications by the Author

Peer-reviewed journal papers

Saeidlou, S., Saadat, M., Amini Sharifi, E., & Jules, G. D. (2019). Agent-based distributed manufacturing scheduling: An ontological approach. *Cogent Engineering*.

Saeidlou, S., Saadat, M., & Jules, G. D. (2019). Knowledge and Agent-Based System for Decentralised Scheduling in Manufacturing. *Cogent Engineering*.

Saeidlou, S., Saadat, M., Amini Sharifi, E., & Jules, G. D. (2017). An ontology-based intelligent data query system in manufacturing networks. *Production & Manufacturing Research*, *5*(1), 250-267.

Saeidlou, S., Saadat, M., & Jules, G. D., "Towards Decentralised Job Shop Scheduling as a Web Service" *International Journal of Production Research*, Under review.

Saadat, M., **Saeidlou, S.**, & Tan, M. C. (2017). A holonic workforce allocation model for labour-intensive manufacturing. *Cogent Engineering*, *4*(1), 1324747.

Jules, G. D., Saadat, M., & **Saeidlou, S.** (2015). Holonic ontology and interaction protocol for manufacturing network organization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(5), 819-830.

Khan, Z., **Saeidlou, S.**, Saadat, M., "Ontology-based decision tree model for prediction in a manufacturing netwrok" *Proceedings of the Institute of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Under review.

Peer-reviewed conference papers

Saeidlou, S., Saadat, M., & Aminisharifi, E. (2016, August). Intelligent Semantic Query for Manufacturing Supply Chain. In *Advances in Manufacturing Technology XXX: Proceedings of the 14th International Conference on Manufacturing Research, Incorporating the 31st National Conference on Manufacturing Research, September 6–8, 2016, Loughborough University, UK* (Vol. 3, p. 419). IOS Press.

Saeidlou, S., Saadat, M., & Jules, G. (2014, October). Cloud manufacturing analysis based on ontology mapping and multi agent systems. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on* (pp. 2019-2024). IEEE.

Zhong, Z., **Saeidlou, S.**, Saadat, M., & Abukar, A. (2018, August). Ontology and Rule-ased Reasoning for Intelligent Predictive Manufacturing. In *Advances in Manufacturing Technology XXXII: Proceedings of the 16th International Conference on Manufacturing Research, incorporating the 33rd National Conference on Manufacturing Research, September 11–13, 2018, University of Skövde, Sweden* (Vol. 8, p. 355). IOS Press.

Zhen, H. A. O., Abukar, A., Saadat, M., & **Saeidlou, S.** (2018, August). A Neural Network Approach to Predict Deliverability in Manufacturing. In *Advances in Manufacturing Technology XXXII: Proceedings of the 16th International Conference on Manufacturing Research, incorporating the 33rd National Conference on Manufacturing Research, September 11–13, 2018, University of Skövde, Sweden* (Vol. 8, p. 361). IOS Press.

Jules, G., Saadat, M., & **Saeidlou, S.** (2013, October). Holonic goal-driven scheduling model for manufacturing networks. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on* (pp. 1235-1240). IEEE.

Jules, G. D., Saadat, M., & **Saeidlou, S.** (2013, August). Ontological extension of PROSA for manufacturing network formation. In *International Conference on Industrial Applications of Holonic and Multi-Agent Systems* (pp. 140-151). Springer, Berlin, Heidelberg.