

**Research Space**

Journal article

**Towards decentralised job shop scheduling as a web service**

**Saeidlou, S., Saadat, M. and Jules, G. D.**



## Towards decentralised job shop scheduling as a web service

Salman Saeidlou, Mozafar Saadat & Guiovanni D. Jules |

To cite this article: Salman Saeidlou, Mozafar Saadat & Guiovanni D. Jules | (2021) Towards decentralised job shop scheduling as a web service, Cogent Engineering, 8:1, 1938795, DOI: [10.1080/23311916.2021.1938795](https://doi.org/10.1080/23311916.2021.1938795)

To link to this article: <https://doi.org/10.1080/23311916.2021.1938795>



© 2021 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.



Published online: 22 Jun 2021.



Submit your article to this journal [↗](#)



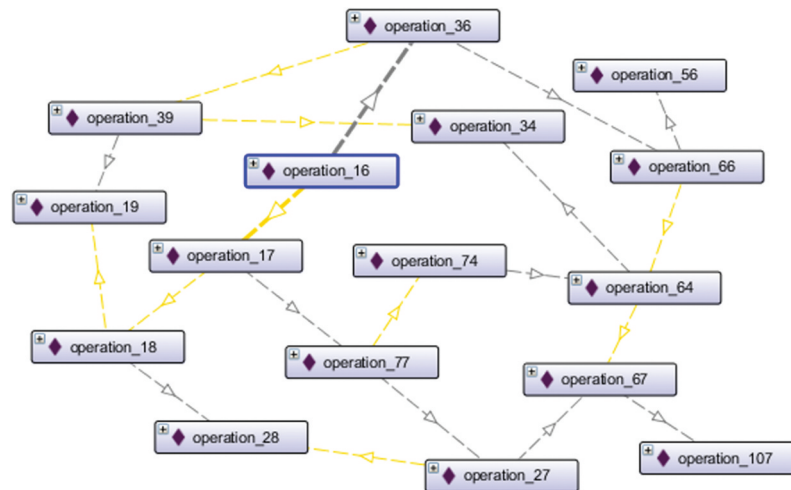
Article views: 19



View related articles [↗](#)



View Crossmark data [↗](#)



Ontological representation of interdependencies among operations

# Towards decentralised job shop scheduling as a web service

Salman Saeidlou, Mozafar Saadat and Guiovanni D. Jules

*Cogent Engineering* (2021), 8: 1938795



## PRODUCTION & MANUFACTURING | RESEARCH ARTICLE

# Towards decentralised job shop scheduling as a web service

Salman Saeidlou<sup>1\*</sup>, Mozafar Saadat<sup>2</sup> and Guiovanni D. Jules<sup>2</sup>

Received: 22 January 2021  
Accepted: 01 May 2021

\*Corresponding author: Salman Saeidlou, School of Engineering, Technology and Design, Canterbury Christ Church University, Canterbury CT1 1QU, United Kingdom  
E-mail: [salman.saeidlou@canterbury.ac.uk](mailto:salman.saeidlou@canterbury.ac.uk)

Reviewing editor:  
Wenjun Xu, Wuhan University of Technology, Wuhan, CHINA

Additional information is available at the end of the article

**Abstract:** This paper aims to investigate the fundamental requirements for a cloud-based scheduling service for manufacturing, notably manufacturer priority to scheduling service, resolution of schedule conflict, and error-proof data entry. A flow chart of an inference-based system for manufacturing scheduling is proposed and a prototype was designed using semantic web technologies. An adapted version of the Muth and Thompson  $10 \times 10$  scheduling problem (MT10) was used as a case study and two manufacturing companies represented our use cases. Using Microsoft Project, levelled manufacturer operation plans were generated. Semantic rules were proposed for constraints calculation, scheduling and verification. Pellet semantic reasoner was used to apply those rules onto the case study. The results include two main findings. First, our system effectively detected conflicts when subjected to four types of disturbances. Secondly, suggestions of conflict resolutions were effective when implemented albeit they were not efficient. Consequently, our two hypotheses were accepted which

### ABOUT THE AUTHOR



Salman Saeidlou

Salman Saeidlou received the MEng (Hons.) degree in Mechanical Engineering from the University of Birmingham, U.K., and the Ph.D. degree in the same field of study from the University of Birmingham, U.K. He is currently working as a senior lecturer in Mechanical/ Material engineering within the School of Engineering, Technology and Design at Canterbury Christ Church University, Kent, UK. His research interests include holonic manufacturing, distributed systems, agent-based modelling and intelligent manufacturing systems.

Mozafar Saadat received the B.Sc. (Hons.) degree in Mechanical Engineering from the University of Surrey, Guildford, U.K., and the Ph.D. degree in industrial automation from the University of Durham, U.K. He is currently with the Department of Mechanical Engineering, School of Engineering, University of Birmingham, U.K.

Guiovanni Jules received the BEng degree in Mechanical Engineering from the University of Birmingham, U.K., and the Ph.D. degree also from the University of Birmingham.

Professor Wenjun Xu is the reviewing editor from Cogent Engineering and he is not one of the main authors of the article. However his affiliation is Prof. Wenjun Xu, Professor, School of Information Engineering, Wuhan University of Technology, China

### PUBLIC INTEREST STATEMENT

The adoption of emerging manufacturing technologies such as the Internet of Things (IoT) and cyber physical systems (CPS) has shown promising capabilities to transform the traditional manufacturing industry into one which is more flexible and fault tolerant. These technological advancements lay the foundation for a decentralised manufacturing system that enables effective and adaptive planning and control of production systems. In light of the above, the presented work addresses the fundamental enablers of a dynamic cloud-based scheduling system, by introducing a knowledge base for one which is intelligent and inference-based. The proposed work demonstrates semantic rules and implements such a scheduling system in a case study.

gave merit for future works intended to develop scheduling as a web service. Future works will include three phases: (1) migration of our system to a graph database server, (2) a multi-agent system to automate conflict resolution and data entry, and (3) an optimisation mechanism for manufacturer prioritisation to scheduling services.

**Subjects: Relational Databases; Information & Communication Technology; ICT;; Industrial Engineering & Manufacturing; Operations Research; Production Systems; Quality Control & Reliability; Manufacturing Engineering; Supply Chain Management; Manufacturing Technology; Operations Research**

**Keywords: Ontology; semantic web; graph database; inference; multi-agent system; disturbance; conflict resolution**

## 1. Introduction

This paper aims to investigate the technical implications of job shop scheduling using semantic web technologies. The literature shows that production scheduling models are fundamentally concerned with managing priorities and capacity (Fuchigami & Rangel, 2018; Ghiyasinab et al., 2020; Lin & Gen, 2018). Capacity and priorities are mutually influenced. Capacity can be seen as a function of labour availability, raw materials and suitable utilities. Priority is driven by the need to manage production costs and production targets, as well as order due dates. Scheduling within those constraints is the main goal and meeting objectives heightens the competitiveness of schedules. Minimising makespan, earliness, tardiness and total costs, and maximising throughput and profit are common objectives. Sequencing and timing of tasks are two fundamental scheduling processes (Harjunoski et al., 2014; Kolus et al., 2020). In the following sections, the main enablers of semantic web manufacturing scheduling, which is a decentralised model in nature, are introduced. The distributed characteristics of this problem entails the use of a multi-agent system, where intelligent distributed entities co-operate with each other through the cloud, in order to end the decision-making in concord.

### 1.1. Decentralised manufacturing scheduling

Methods such as those which are decentralised and multi-agent are modular, fault tolerant and extensible (Giordani et al., 2013; Li et al., 2020). A negotiation mechanism creates a bridge between a scheduling service and a resource provider, where the former optimises user-centric targets and the latter optimises provider-centric measures (Egger et al., 2020; Prodan et al., 2011). Research on decentralised environments has been prominent. Approaches to decentralised scheduling consist of coordination mechanisms and negotiation processes between buyers and sellers in a web-based environment (Ghadimi et al., 2019; Zhao et al., 2020). Another approach consists of characterising a scheduling problem into a distributed one using a series of try-and-miss mixed integer linear programming (MILP) models, and the best model provides a solution space of job assignments. Variable neighbourhood descent (VND) is used to narrow down the solution space (Ruiz et al., 2019). In the broadest sense, approaches to decentralised scheduling can be categorised into web-based technologies and mainframe-based technologies. Due to computational and communication bottlenecks that come with the high flexibility of a system, decentralised methods of management can outperform conventional, centralised methods. The literature shows there is a gap and a manufacturing scheduling model that uses inference-based reasoning is lacking. Therefore, this research is aimed to develop a decentralised cloud-based scheduling model that uses an inferred-based system at its foundation to efficiently detect conflicts.

### 1.2. Current scheduling methods

Appropriate representation of a scheduling problem is governed by key principles associated with a solution and it is an activity driven by available computation techniques. Mixed integer programming (MIP) formulations are prominent in primary research and review literature for computing various types of scheduling models (Fuchigami & Rangel, 2018). Successful industrial implementations were enabled by interfacing technologies among MIP solvers such as CPLEX® and GAMS, relational

databases and supply chain planning software suites such as SAP-APO (Misener & Floudas, 2014). A generalised disjunctive program (GDP) is another method of formulation that formalises constraints and logic expressions to describe a scheduling model. The GDP can be solved by a MIP solver after conversion into algebraic modelling language (AML) (Rodriguez et al., 2017).

The modelling and simulation of manufacturing scheduling frequently makes use of mixed integer linear programming (MILP) although there is little to support its suitability for cloud-based application based on the literature. In manufacturing scheduling, ontology web language and disjunctive constraints written in semantic web rule language now address a gap in the use of disjunctive graphs and databases used for this purpose, such as cloud infrastructure, and are already in format and implemented.

### **1.3. Service-oriented architecture**

Service-oriented architecture (SOA) uses the internet to offer collaborative, flexible, reconfigurable and customisable services to the end user, allowing greatly incorporated human-machine manufacturing systems. It aims at creating an ecosystem of different manufacturing factors to be included in the intelligent manufacturing systems; so that the technical, managerial, and organisational levels can combine seamlessly (Morgan & O'Donnell, 2017). It defines advanced technology where the production resources are converted into smart manufacturing objects (SMOs) with the ability to interact, sense, and interlink with one another to adaptively and automatically perform the logic of manufacturing. The internet of things (IoT) environment enables the machine to machine, human to machine and human to human connections for the perception of intelligence (Eleftheriadis & Myklebust, 2016). Currently, the IoT is considered as a concept of more manufacturing under Industry 4.0; and has implemented current advances, including the cutting edge IT infrastructure for the acquisition and sharing of data which influences the productivity of the manufacturing system. This manufacturing incorporates the sharing of real-time features and real-time data gathering among different manufacturing resources, including jobs, materials, workers, and machines.

Lu and Xu (2017) proposed a knowledge-based service composition in a cloud-manufacturing environment. They used ontology to facilitate the efficient mapping between service requests and available manufacturing resources. The main focus of their research is the evaluation of the quality of service (QoS), which provides a quantitative service evaluation matrix for each mapping. This is in contrast to the research presented in this paper where the main focus is to delegate the task to intelligent agents and try to investigate the conflict resolutions during the scheduling process. On the other Shen et al. (2005), (2007) have proposed an agent-based service-oriented integration system for manufacturing scheduling. They have used the famous contract net protocol (CNP) negotiation mechanism to perform the planning process; however, they have not considered the conflict resolution for the agent-based system. Our research has used the agent-based technology in conjunction with ontology engineering and rule-based reasoning, which is more promising when it comes to scalability. In another study, Tao et al. (2011) have introduced a theoretical computing and service-oriented model called cloud manufacturing (CMfg). The key enablers of a cloud manufacturing model are discussed and the key advantages and challenges for implementing CMfg are analysed. The work demonstrated the benefits of such an intelligent system; however, it was limited to a theoretical model. T. Wang et al. (2014) also proposed a CMfg semantic model to address the matching manufacturing services' problem. Their focus was on the construction of ontology and a graph-based semantic similarity algorithm. In another study, Zhang and Roy (2018) made use of semantic web technologies to find the best dispatching rule for a job shop scheduling problem. Using a semantic similarity concept, the appropriate dispatching rules were selected based on different production objectives. Our research makes use of the current state-of-the-art service-oriented architecture and tries to further enhance the scalability by investigating the automatic conflict resolution in case of disturbances through an intelligent inference-based scheduling system.

#### **1.4. Cloud manufacturing opportunities**

A new paradigm known as Cloud Manufacturing can lead to Manufacturing as a Service by facilitating the sharing of resources across a diversity of geographical locations and exploiting computer power through web services (Helo et al., 2021; Ren et al., 2017).

Manufacturing services can develop cost-effective algorithms for problem-solving by influencing people and communities to utilise this paradigm (Sokolov et al., 2020). Furthermore, performance by cloud utility providers also covers data loss safe guards, security developments and infrastructure maintenance (Sokolov et al., 2020). Cloud services are of benefit to clients due to their flexibility and this impacts on all product aspects including manufacture, design, management and quality control (Saeidlou et al., 2014). In light of the above mentioned benefits of the cloud manufacturing paradigm, this paper has attempted to address the key concepts and enablers of such a system. It has also presented an exemplar which may serve as a systematic and practical approach to autonomous conflict resolution, which eventually will migrate this paradigm from concept to practice.

#### **1.5. Facilitating cloud-based scheduling for manufacturing**

A central aspect of cloud manufacture architecture comprising the service layer is knowledge-based scheduling (Lu & Xu, 2017; Saeidlou et al., 2019a). Tools employed currently for rescheduling in the cloud include a database and system of data capture, and a form of multi-objective optimisation: Monte Carlo simulation (Bölöni & Turgut, 2017; Guo et al., 2015). This type of manufacturing is primarily for the cloud space, benefitting from components such as flexible storage of data, rapid data query and manipulation.

##### **1.5.1. Graph database for cloud-based scheduling**

Intuitive modelling of production scheduling entities and their relationships can be represented using a disjunctive graph prior to storage in graph databases (Roy & Sussmann, 1964). Study findings have demonstrated a speed of data query 20 to 30 times greater when dealing with linked data as opposed to relational data. Moreover, restructuring of the whole schema is no longer required as these new relationships can be updated within the database by simply adding nodes and edges (Angles et al., 2017).

Factual raw data and relationships are contained in disjunctive graphs and deposited in a graph database, and bearing in mind the manufacturing process plan, this needs to be efficient in order to handle large quantities.

##### **1.5.2. Semantic reasoning algorithms**

Rule-based modelling, which permits semantic reasoners to execute inferencing within databases can also be stored in a graph database, and the reliable analysis of logic interpretation can now be carried out using modern deductive mechanisms. One such mechanism, chaining, now exposes inferred data to questioning (Pérez-Urbina et al., 2012), and this can be forward, backwards or bi-directional (as evidenced in Drools and Prolog) using Rete algorithms (Kaiser et al., 2012).

Inference engines such as Pellet and Stardog are two interesting results that have arisen from the combination of tableaux algorithms, nominal absorption and nominal-based models. Through the application of additional optimisation techniques, the two possess incremental reasoning abilities invaluable in manufacturing, subject to knowledge flux (Mourtzis et al., 2016); where ontology web language (OWL) is used to document facts and full semantic web rule language (SWRL) is adopted to define reasoning rules (Chen et al., 2020). An expert system emerges as a result of integration of a mechanism with concept descriptions and rule-based statements (Wagner, 2017).

Whilst manufacturing ontologies were highlighted in the literature review, they did not focus on the manufacturing scheduling aspect. The scope can be broadened by merging it with other existing ontologies. Additionally, there is a gap for a well-structured framework which permits

the investigation of networks such as flow shop scheduling of manufacturing, which combines agent-based modelling, ontology building and multi-agent system implementation. The development of a scheduling ontology would be supported by a framework of this nature and would allow for new algorithms for agent-based systems.

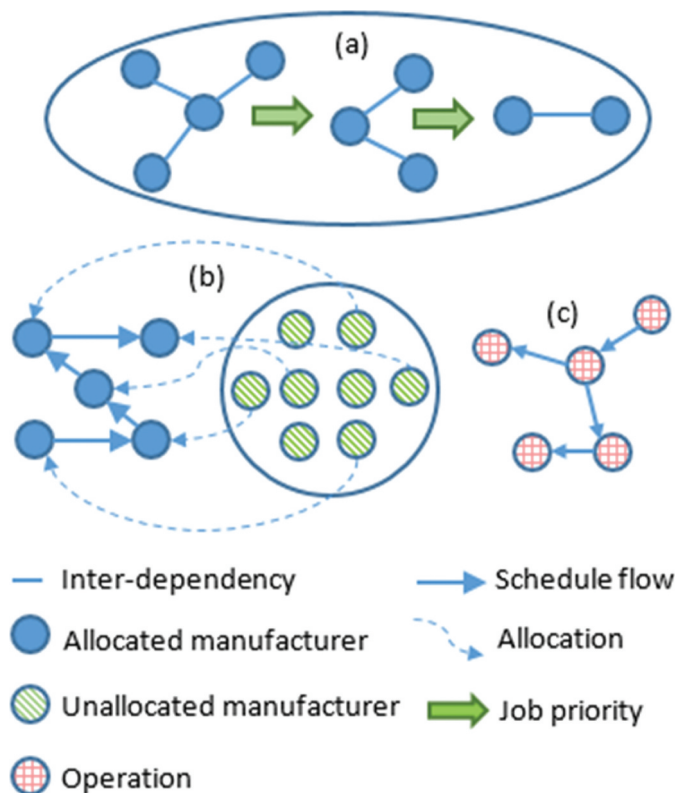
**1.6. Optimisation of a multi-agent system**

As discussed in the literature, systems of this nature requiring problem-solving capability need improvement using optimisation algorithms; these can also be multi-agent systems, or exchange models. The latter acknowledges the flow and development of information as opposed to the optimisation model which addresses the agents’ objective functions (Nedic & Ozdaglar, 2009; Saeidlou et al., 2019b). Previous work by the authors has demonstrated encouraging findings regarding decentralised GA optimisation acquired through the use of REPAST (Jules & Saadat, 2017). The functions of a multi-agent system can be defined through agent-based modelling (Abar et al., 2017), which when implemented can assist in the resolution of naturally distributed problems requiring a variety of customised computation entities (M. Wang et al., 2009).

Initial investigation into the capacity of decentralised manufacturing networks using market-based mechanisms has occurred, and there is now a requirement for research into those that facilitate the collection of simple local behaviours through the development of a multimanager operation plan. This study attempted to explore the formation of networks where manufacturers have control over their network scheduling and selection using a

market-based approach through the broader application of a multi-agent system.

**Figure 1. Previous works on (a) collaborative network organisation; (b) network formation; and current work on (c) network scheduling.**





### **1.7. Statement of purpose and rational of study approach**

In this paper, we develop scheduling rules in semantic language for technologies that can enable cloud manufacturing. Using an ontology editor Protégé 5.0 beta, a semantic incremental reasoner Pellet and Microsoft Project 2013, we investigated two hypotheses and three questions. Answers will provide pertinent support for the next phases of work, towards an efficient development of scheduling as a web service. We investigated the first hypothesis whereby we asked “When using an inference-based system (IbS) for operation scheduling, does the manufacturer’s priority matter?” If yes, is there a strong case for using optimisation to find the best manufacturer arrangement? Next, we investigated the hypothesis which asked

whether “the resolution of disturbances and the resolution of conflicts of resource-constrained scheduling are synonymous, and if so, what are the characteristics of these disturbances?” Can resource-constrained scheduling be done and conflicts be systematically resolved in an IbS? If yes, is there a strong case for using a multi-agent system to help with resolving conflicts?

### **2. Related works**

Similarities with the authors’ proposed work has been demonstrated by the decentralised Cascade Flow Shop (CFS) (Baffo et al., 2013), where job scheduling and timing can be affected by a number of decision-makers. Core elements of their model were the resolution of localised problems and downstream solution communications. Whilst this is similar to the authors’ approach regarding operation timing and conflict resolution, there are two dissimilarities. First, CFS was investigated from the perspective of machines within manufacturing cells, while IbS was about manufacturers. Second, the MILP approach was used for modelling CFS and algebraic modelling language (AML) envisioned for mainframe computing was used in its writing, with little evidence of its suitability for cloud-based application. The OWL-based disjunction graphs and SWRL rules used by the authors tapped into accessible cloud infrastructures and graph databases, although MILP optimisation capabilities will be beneficial to others.

Task ontology, a framework for expanding cloud manufacturing (CMfg) proposed by T. Wang et al. (2014) consists of three stages. First, new task documents can be exposed to analytics which prepare CMfg ontology for new data features; second, real-time task data were embedded in an ontology template; and third, the ontology was merged into CMfg ontology via similarity analysis. The similarity with our work includes the use of semantic web technologies aimed at a cloud application. The difference is that our scope is narrower, focusing on scheduling, and has a rule base as a fifth semantic feature. The learnt outcome is that their framework can enable our scheduling ontology to merge into CMfg ontology.

Leitão et al. (2013) emphasized the holonic manufacturing system (HMS) paradigm for

software that functions autonomously, without a need for instruction, to manage unexpected situations. In our approach, local information is pushed along an edge between two nodes and a downstream node is used for localised adjustments. This partly conforms to the HMS paradigm. Our approach can also accommodate an ability for a downstream node to refuse an adjustment, consequently causing upstream nodes to reconsider. Conforming to the HMS paradigm, during conflict resolution, can improve the efficiency of the schedule timing procedure.

The framework presented in our preceding works lays the foundation for this paper. Our papers investigated an agent-based model of manufacturing network formation (Jules & Saadat, 2017) and a group formation of manufacturing networks (Jules et al., 2015). Both papers addressed aspects of decentralised manufacturing as shown in [Figure 1](#).

However, the first paper was driven by job-dispatching rules where job constraints were mostly implicit, i.e. more useful constraints could have been derived. The second paper was driven by the manufacturer priority to scheduling, where the outcome was a ring network with manufacturers in

a specific arrangement. This third paper investigated the impact of default manufacturer priority, where job constraints are explicitly defined as a set of rules. Our approach can be further refined to become a holistic framework.

### 3. Methodology

#### 3.1. Experiment overview

In an attempt to transition from lessons learnt from agent-based modelling to implementations in multi-agent system applications, we investigated one of the key transition stages, namely the knowledge base. The latter enables a multi-agent system to reason about their environment, take actions and become resilient to disturbances. The scope of the paper is limited to scheduling and based on that premise, we investigated a knowledge base as part of IbS for scheduling. Using this system provided an opportunity to revisit scheduling issues and test our research hypothesis around five research questions.

#### 3.2. Industry use case

Northern Italy is among the richest industrial districts of Europe. It has been an example of inter-firm co-operations with adoption of the concept across Europe notably in Valencia, Spain and Baden-Württemberg, Germany. The concept broadly advocates collaboration among small firms, with an emphasis on advanced technologies and innovative solutions to engineering problems for which customers are pleased to pay high prices (Aureli & Del Baldo, 2016). Our research problem is supported by a case study of Gruppo Fabbricazione Meccanica (GFM) srl which is a Northern Italian company with a strong market position in mechanical engineering services. It is established in component manufacturing for the energy industry, mainly components going into steam and gas turbines with applications in the aerospace, naval, oil and gas industries. GFM srl is a small to medium-sized enterprise (SME) that offers micro-tolerance to multi-ton machining, with ISO-certified capability through collaboration and coordination within the high concentration of SMEs of the district. It forms part of supply chains of market-leading companies, notably Siemens, Mapna and Ansaldo.

SMEs participating with GFM srl will benefit from a cloud-based scheduling service for the following reasons: (1) Customers are willing to pay a high price and therefore competitiveness is

**Table 1. Nomenclature of scheduling model properties**

Parameters	Instances
Job ( <i>j</i> )	$(o_{j,i} \square O_j)$ Set of operations required
	$(o_{j,i=1})$ First operation
	$(o_{j,i=n})$ Last operation where <i>n</i> is the total number of operations
	$(dt)$ Due time
Manufacturer ( <i>m</i> )	$(o_{m,i} \square Om)$ Set of operations offered
Operation ( <i>o</i> ) where <i>o</i> stands for either $o_{m,i}$ or $o_{j,i}$	$(ppp)$ Predecessor in process plan
	$(spp)$ Successor in process plan
	$(est)$ Earliest start time
	$(eft)$ Earliest finish time
	$(lst)$ Latest start time
	$(lft)$ Latest finish time
	$(pt)$ Processing time
	$(pst)$ Proposed start time
$(pft)$ Proposed finish time	

a function of lead time, product quality, partnership strength and resilience to customer evolving needs. (2) The decentralised production facilities aggregate to deliver a unified production service, where supply of capacity is inherently high in the context of the region. Manufacturing resilience will benefit from intelligent automated coordination which fundamentally consists of repetitive knowledge-driven triggering of simple rules. (3) Scheduling is decentralised by nature of the inter-firm operation concept, albeit an inefficient one with over-the-phone coordination. It is believed that optimisation of decentralised scheduling requires an unmanageable amount of communication and it was brought to our attention that concise information is essential for an SME. Automatically handling mundane communication in the background will leave human communication for fostering stronger partnerships. For those reasons, we believe GFM srl presents a demand for the research work presented.

### 3.3. Inference-based system overview

Object property and data property are the main constituent properties that can be defined in the formation of an ontology, where each one consists of a domain and a range. Accordingly, the domain of the ontology is defined by concepts which are the objects that possess the properties, and the range encapsulates the value of the property. As the name suggests, an object property accepts input values of an object and a data property takes data values such as an integer or a string. The entity that connects properties is termed a predicate. The appendix, illustrates data and object properties used for the scheduling model presented in this paper. As an example, the equations  $o_{j,i} = ppp_{o_{j,i+1}}$  and  $o_{j,i+1} = spp_{o_{j,i}}$  with the domain of operation  $o_{j,i}$  and the range of  $o_{j,i+1}$  characterises the predicate “precedes Jobwise”. The predicates are defined in a reciprocal way, meaning that “precedes Jobwise” inversely implies “succeeds Jobwise”. For instance, the assertion of the fact that operation 14 precedes operation 15 lead to the inference of operation 15 succeeds operation 14.

The semantic web rule language equivalent of equations  $o_{j,i} = ppp_{o_{j,i+1}}$  and  $o_{j,i+1} = spp_{o_{j,i}}$  can be formulated as (precedesJobwise ?o1 ?o2). Here, precedesJobwise denotes the object property of an operation, which in turn attains another operation as its input. Considering a job process plan, the operation o1 precedes operation o2 according to the abovementioned rule. Accordingly, the earliest possible start time of an operation ( $est_{o_{j,i}}$ ) can be formulated as (hasEarliestPossibleStartTime ?o1 ?est). Using the same logical formulation, the processing time of an operation ( $pt_{o_{j,i}}$ ) will be formatted as (hasProcessingTime ?o1 ?pt).

A potential basis for logical manipulation is modus ponens (MP), and if an MP exists whereby ‘P implies Q’, if P is true, Q must be true. This rule is represented as “P  $\square$  Q” in this case Q, can lead to another rule “Q  $\square$  R”. Within the rule base, formation of a logic chain with an inference engine such as Pellet exploits and manipulates the database of facts and rules. Provision of sound and complete reasoning is therefore done for OWL 2 facts and SWRL rules. A number of these rules are provided in the appendix, and the following example demonstrates an application of IbS.

Considering the parameters and associated attributes tabulated in Table 1, the following IbS deduction can be illustrated:

Let:

$$m = (\text{precedesJobwise } ?o1 \text{ } ?o2)$$

$$n = (\text{hasEarliestPossibleStartTime } ?o1 \text{ } ?st)$$

$$p = (\text{hasProcessingTime } ?o1 \text{ } ?pt)$$

Facts:

**Table 2. MT10 problem specification**

Job	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
1	10 (29)	11 (78)	12 (9)	13 (36)	14 (49)	15 (11)	16 (62)	17 (56)	18 (44)	19 (21)
2	20 (43)	22 (90)	24 (75)	29 (11)	23 (69)	21 (28)	26 (46)	25 (46)	27 (72)	28 (30)
3	31 (91)	30 (85)	33 (39)	32 (74)	38 (90)	35 (10)	37 (12)	36 (89)	39 (45)	34 (33)
4	41 (81)	42 (95)	40 (71)	44 (99)	46 (9)	48 (52)	47 (85)	43 (98)	49 (22)	45 (43)
5	52 (14)	50 (6)	51 (22)	55 (61)	53 (26)	54 (69)	58 (21)	57 (49)	59 (72)	56 (53)
6	62 (84)	61 (2)	65 (52)	63 (95)	68 (48)	69 (72)	60 (47)	66 (65)	64 (6)	67 (25)
7	71 (46)	70 (37)	73 (61)	72 (13)	76 (32)	75 (21)	79 (32)	78 (89)	77 (30)	74 (55)
8	82 (31)	80 (86)	81 (46)	85 (74)	84 (32)	86 (88)	88 (19)	89 (48)	87 (36)	83 (79)
9	90 (76)	91 (69)	93 (76)	95 (51)	92 (85)	99 (11)	96 (40)	97 (89)	94 (26)	98 (74)
10	101 (85)	100 (13)	102 (61)	106 (7)	108 (64)	109 (76)	105 (47)	103 (52)	104 (90)	107 (45)

$$m^1 = (\text{precedesJobwise operation}_{10} \text{ operation}_{11})$$

$$m^2 = (\text{precedesJobwise operation}_{11} \text{ operation}_{12})$$

$$n^1 = (\text{hasEarliestPossibleStartTime operation}_{10} 0)$$

$$n^2 = (\text{hasEarliestPossibleStartTime operation}_{11} \text{ null})$$

$$p^1 = (\text{hasProcessingTime operation}_{10} 29)$$

$$p^2 = (\text{hasProcessingTime operation}_{11} 78)$$

Suppose a rule in the form  $(m, n, p) \rightarrow (q, r)$  as follows:

`precedesJobwise(?o1,?o2),hasEarliestPossibleStartTime(?o1,?st),hasProcessingTime(?o1,?pt),add  
 (?ft,?st,?pt)-> hasEarliestPossibleFinishTime(?o1,?ft),hasEarliestPossibleStartTime(?o2,?ft)`

Substitution 1:

`precedesJobwise(operation_{10},operation_{11}),hasEarliestPossibleStartTime(operation_{10},0),  
 hasProcessingTime(operation_{10},29),add(29, 0, 29) = m^1, n^1, p^1`

Which yields:

$$q^1 = (\text{hasEarliestPossibleFinishTime operation}_{10} 29)$$

$$r^1 = (\text{hasEarliestPossibleStartTime operation\_11 29})$$

Substitution 2:

$$\text{precedesJobwise}(\text{operation\_11}, \text{operation\_12}), \text{hasEarliestPossibleStartTime}(\text{operation\_11}, 29), \\ \text{hasProcessingTime}(\text{operation\_11}, 78), \text{add}(107, 29, 78) = m^2, n^2, p^2$$

Which yields:

$$q^2 = (\text{hasEarliestPossibleFinishTime operation\_11 107})$$

$$r^2 = (\text{hasEarliestPossibleStartTime operation\_12 107})$$

### 3.4. Preparation of experimental data set

The first stage of preparation is to adapt the Muth and Thompson (1963)  $10 \times 10$  scheduling problem (MT10) as shown in Table 2.

O1 represents operation 1; 10 is correlated with job 1 and provided by manufacturer 0; (29) indicates manufacturing processing time

The adapted version slightly differs from the original MT10 on the basis of the IDs of the manufacturing operations. For instance, the process plan of a Job 1 was originally presented as {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} with digits representing individual numbers by ID respectively. In our version, the need to mention Job1 is defunct as {10, 11, 12, 13, 14, 15, 16, 17, 18, 19} indicate the sequence of operations of Job 1 and is further demonstrated by the process plan of Job 2 {20, 22, 23, 24, 25, 26, 27, 28}. With the MT10, we are also given the processing times of operations and the known optimal lead time is 930 (Yen & Ivers, 2009). In the second stage of preparation where Microsoft Project was used, MT10 was loaded in, levelling was performed and an operation plan for every manufacturer was generated as shown in Table 3. Levelling is a function in Microsoft Project that resolves resource conflicts by delaying tasks. Consequently, the latest lead time achieved, among the ten jobs, was 1040. The resulting data set in Table 3 is the finished version of the input to the ontology.

M represents manufacturer; O1 indicates operation 1; 90 is correlated with job 9 and processed by manufacturer 0

### 3.5. Preparation of experimental apparatus

In designing OWL2, ontology editor Protégé 5.0.beta was employed with classes of “Operation”, “Job”, “Verification” and “Manufacture”. The SWRL editor embedded within the ontology editor facilitates the calculation and development of constraints, scheduling and verification rules. Figure 2 shows the apparatus framework and verification rules to be developed, and the related job process requirements and plans and manufacturer services originating from the MT10 scheduling problem. The formation of this ontology came primarily from data object instantiation and linking. Further developments can automate current manual processes if our hypotheses are proved acceptable. The manufacturer service plans came from Ms Project, as presented in Table 3, to complete the ontology. Further work can replace Ms Project with a closed-loop feedback system to optimise manufacturer service plans if our hypotheses are proved acceptable. Additional information about the ontology design is presented in the appendix.

Constraints’ calculation is autonomous; however “plan scheduling” is currently semi-autonomous because several judgment calls are required to perform time adjustments effectively. Abuse of constraints, origins of conflicts and size and direction of necessary adjustments are incorporated into a set of time adjustments and implemented through a range of rules. Upon proving that our hypotheses are acceptable, we aim to automate plan scheduling to increase

**Table 3. Manufacturer operation plans generated by microsoft project**

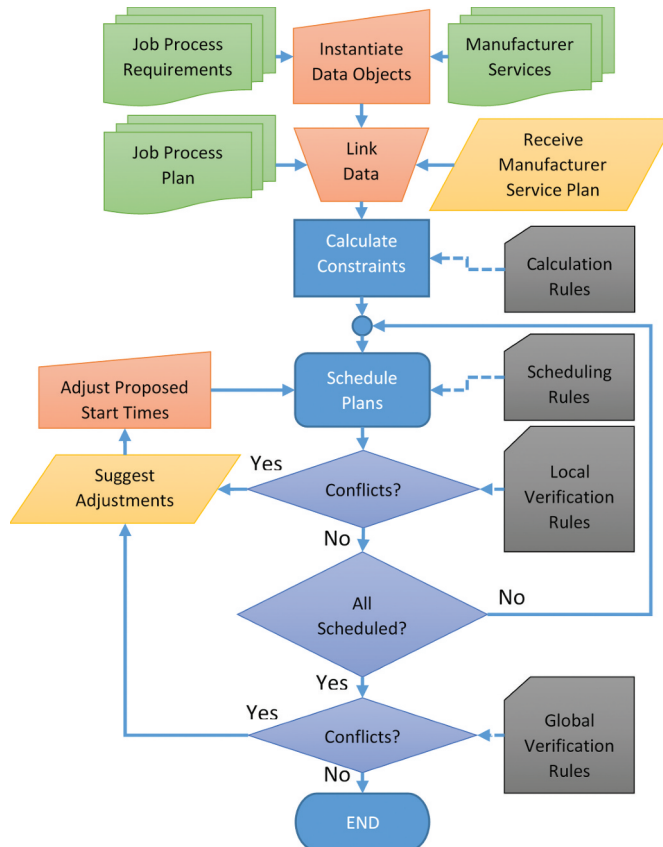
M	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
0	90	20	80	40	100	30	50	10	70	60
1	41	31	101	91	61	81	71	51	11	21
2	82	52	42	62	22	102	32	12	92	72
3	93	33	63	73	53	13	23	103	43	83
4	44	24	84	54	14	104	94	74	64	34
5	65	95	85	55	105	75	35	15	25	45
6	46	106	86	76	96	26	16	36	66	56
7	47	37	57	97	87	17	77	27	67	107
8	108	48	68	38	88	58	78	98	18	28
9	29	109	99	69	79	89	59	49	39	19

efficiency. Detailed scheduling, calculations and verification rules are presented in the appendix using SWRL format.

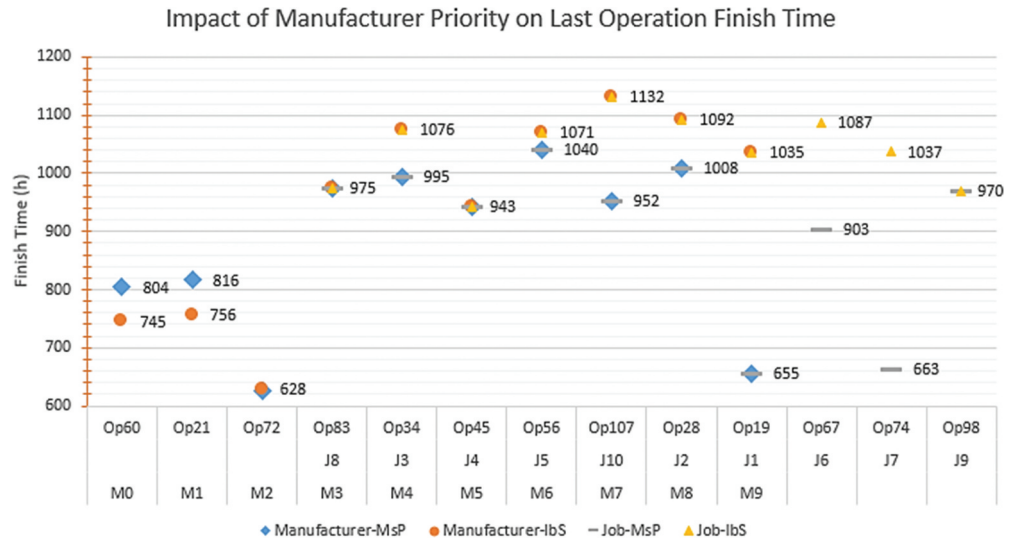
**4. Results**

We investigated the first hypothesis whereby we asked: “when using a rule-based chaining system for operation scheduling, does the manufacturer’s priority matter?” The manufacturer priority used in Protégé was {M0-M1-M2-M3-M4-M5-M6-M7-M8-M9} and that generated in Microsoft Project is unknown. The plot in Figure 3 revealed a few characteristics of the data set as well as those of the

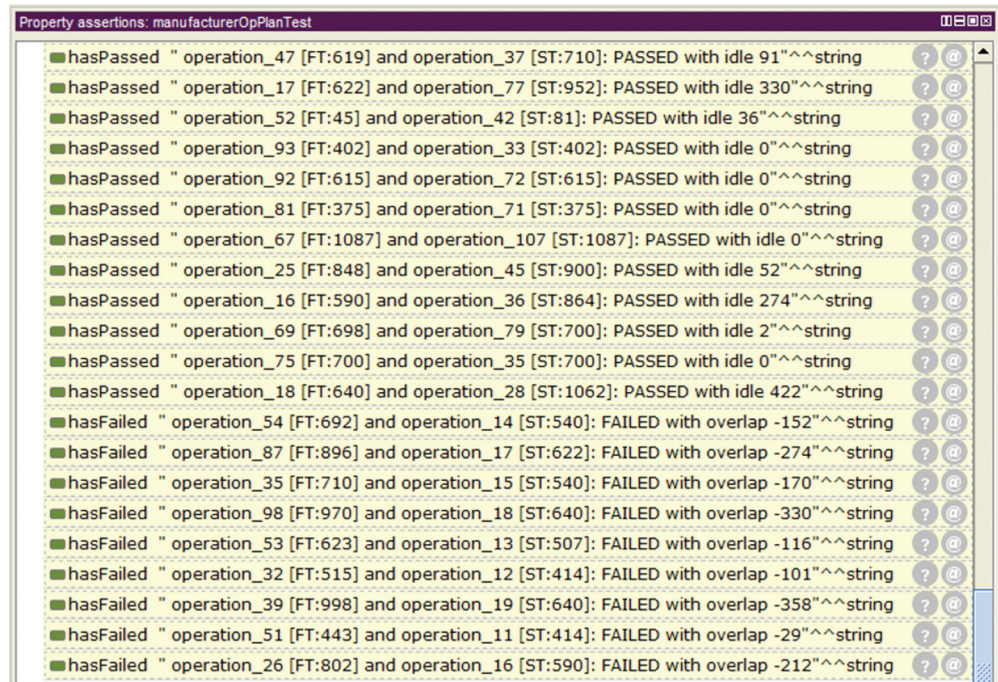
**Figure 2. Proposed inference-based system scheduling framework.**



**Figure 3. Lead time of jobs and manufacturers scheduled in microsoft project and protégé.**



**Figure 4. Manufacturer operation plan test performed on a schedule involving a cancelled job.**



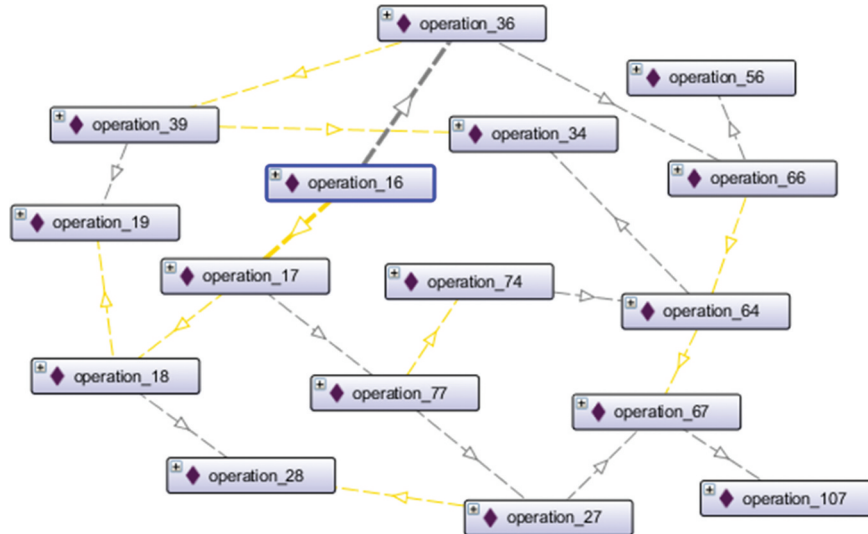
results. It represents finish times of last operations when scheduled by Microsoft Project (MsP) and Protégé Inference-based System (IbS) for all jobs as well as for all manufacturers.

**4.1. Observations about data set**

Inherently, operations 21, 60 and 72 can be last operations of their respective manufacturers; however, they will never be last operations of their respective jobs.

Operations 83, 34, 45, 56, 107, 28, 19, 67, 74 and 98 are last operations of their respective jobs and may also be last operations of their respective manufacturers.

**Figure 5. Operational interdependencies emerging from operation 16 (created in OntoGraf).**



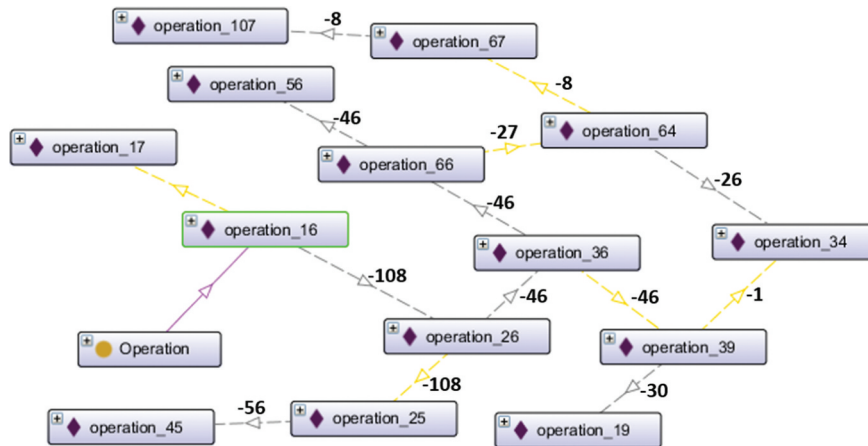
In the prepared data set, operations 83, 34, 45, 56, 107, 28 and 19 are the last operations of both jobs and manufacturers, which is why the graph lines overlap in Figure 3.

**4.2. Observations about results without disturbance**

For the first two manufacturers, M0 and M1, Manufacturer-IbS, outperforms Manufacturer-MsP, with finish time differences of 59 and 60 hours respectively. Thereafter, Manufacturer-MsP outperforms Manufacturer-IbS by an average finish time difference of 94 hours, as we move down the priority ladder.

Operation 28 is the last operation of Manufacturer 8 and the finish time of the preceding Operation 98 has an effect. A similar effect takes place for Operation 34 and the preceding Operation 74 of Manufacturer 4, as well as Operation 107 of Manufacturer 7 and the preceding Operation 67.

**Figure 6. Disturbance propagation paths showing adjustment edges following rush scenario (created in OntoGraf).**





**Table 4. The effect of rush operation on job lead times**

Job	Lead time (h)	
	With Rush Operation	Without Rush Operation
1	1065	1035
2	1092	1092
3	1103	1076
4	999	943
5	1117	1071
6	1095	1087
7	1037	1037
8	975	975
9	970	970
10	1140	1132

#### 4.3. Observations about results with disturbance

The next stage concerned our inquiry into the second hypothesis: “whether the resolution of disturbances and the resolution of conflicts of resource-constrained scheduling are synonymous, and if so, what are the characteristics of these disturbances?” This required four types of disturbances in Protégé and observations from inferred data emerging from three tests. Types of disturbance used were: i) order cancellation, ii) operation delay, iii) manufacturer collapse, and iv) operation rush. The tests applied focused on a job process plan conflict, a manufacturer operation plan conflict test, and due time. A manufacturer operation plan test is presented in Figure 4.

##### 4.3.1. Job cancellation disturbance

This was carried out by the simulation of job cancellation by deleting the processing times of all Job 1 operations. As shown in Figure 4, all those except for Operation 10 were flagged as a fail. The collapse of Operation 14 (measured by an overlap of -152 hours) and Operation 17 (with an overlap of -274 hours) affect Manufacturers M4 and M7, respectively.

##### 4.3.2. Manufacturer collapse disturbance

An undefined schedule, M0, applied to operations indicates the simulation of a manufacturer collapse, or zero processing times. With verification tests flagging disturbances, findings demonstrated failure by almost one-third of tests as a negative impact on all jobs related to the manufacturer.

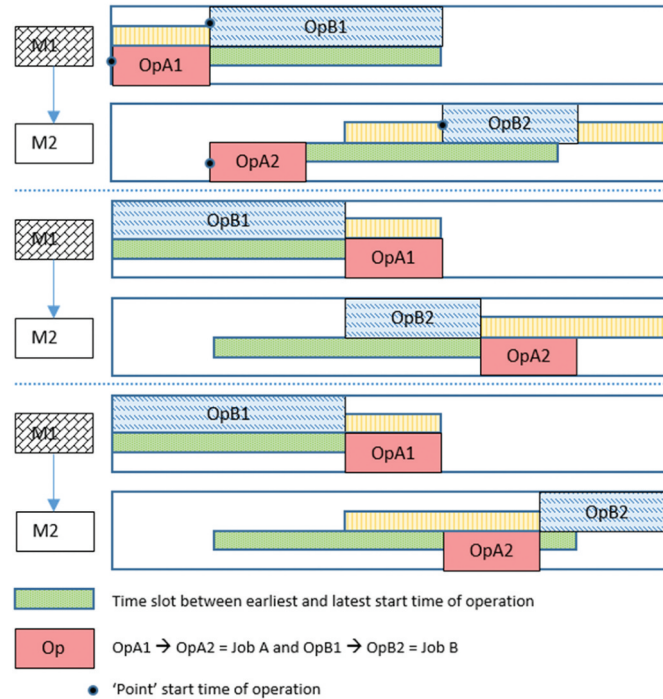
##### 4.3.3. Operation delay disturbance

Operation 16’s processing time was tripled in order to replicate a delay and led to a due time evaluation of on-time delivery relating to Job 1 with an accompanying lateness of 30 hours. Furthermore, increased finish times in respect of Operations 17, 18, 36, and 77 resulted in an overlap average of 108 hours. Whilst other operation finish times seemed unaffected, Jobs 2, 3, 5, 6, 7, and 10 were seen to be affected following exposure to a process of conflict resolution. This is shown in Figure 5, along with various pathways illustrating delays escalating to 15 additional operations. As an example, Operation 16 simultaneously disturbs Operation 17 and also Operation 36.

##### 4.3.4. Rush operation disturbance

This scenario is introduced by switching Operations 16 and 26 within the manufacturer M6 operation plan. Flagging of conflicts within this plan and the ‘job process plan’ test allows for their adjustment. Figure 6 highlights how adjustment needs spread to other operations and jobs. With regards to jobs, Operation 16 has an impact on Operation 17, and as regards

**Figure 7. Three feasible operation plans for job A and job B.**



manufacturer, the same operation influences Operation 26. A similar scenario is applied for further operations in this figure and jobs 1, 3, 4, 5, 6, and 10 failed their ‘due time’ test as illustrated in Table 4.

## 5. Discussion

### 5.1. Last operation heuristic rule

Given a scenario of two jobs, two manufacturers and four operations, the final operation of a job may also be the final operation of a manufacturer. Therefore, if jobs and manufacturers share the same final operations, reducing manufacturer lead times will reduce job lead times. It seems to be a heuristic rule used by Microsoft Project that the final operation of a job becomes the final operation of a manufacturer, as far as the job process plans allow it. Therefore, for MT10, jobs J1, J2, J3, J4, J5, J8 and J10 share last operations of manufacturers M3, M4, M5, M6, M7, M8 and M9 as depicted in Figure 3.

### 5.2. Manufacturer priority matters

In this paper, the sequence in which manufacturers execute their scheduling activities is called manufacturer priority. We loaded the same manufacturing operation plans generated by Microsoft Project (MsP) into the inference-based system (IbS). It was expected that finish times of the last operations, produced in either MsP or IbS would be similar. Figure 3 showed that this was not the case where the only exceptions were operations Op72, Op83 and Op45. Our deduction is that the manufacturer priority factor caused the deviation from the expected results as explained next.

Considering the timeslot between the earliest and latest possible start time of an operation, the “point” start time is adjusted to lie between the time slot between the finish times of preceding and succeeding operations, as shown in Figure 7. Operation plans 1 and 2 are probable, but plan 3

is unlikely, as we believe MsP produced optimised plans. As the scenario is scaled up into MT10, the importance of manufacturer priority probably increases for the following reasons.

At an operation level, the instant of the start time adjustment relative to the instants of other operation adjustments may matter. At a manufacturer level, the sequence in which operations are scheduled may affect these instants at which adjustments occur, at the operation level. Knowing that the decentralised paradigm restricts control over how manufacturers schedule, we can however dictate when manufacturers should schedule. Therefore, we accept the hypothesis that the priority of manufacturers' scheduling does matter.

To reinforce our hypothesis, the results of MsP and IbS in [Figure 3](#) showed that most significant finish time differences occurred in operations Op19 and Op74 of manufacturers M9 and M4 where MsP outperformed. In operations Op60 and Op21, IbS outperformed because manufacturers M0 and M1 were scheduled first.

### **5.3. Disturbance induced conflict**

Schedule conflict takes place when three sets of rules are broken and consequently the response is a set of start time adjustments suggested by Rule C2. The first set of rules ensures that every operation conforms within a time slot that is determined from its job process plan and job due time. This first set consists of rules B1, B3 to B6 as presented in the appendix. The second set of rules verifies that no operation overlaps a preceding operation from its job process plan and its manufacturing operation plan. The second set consists of rules B2 and C1. The final set of rules ensures that all scheduled operations are conflict free, in case conflicts were missed by previous rules. This set of rules is made up of E1 to E6. This case can happen when data is wrongly entered by the user. A conflict may be resolved too late, and it causes inefficiencies in scheduling.

For the job cancellation scenario, it was thought that zeroing processing times would not affect the schedule but the scenario flagged up conflicts in the manufacturer operation plans. There are three reasons for the flagging. First, Rule A2 to A6 and D1 use processing times. Second, Pellet Reasoner is incremental i.e. every data change triggers relevant rules into action. Finally, nine new inferred data have conflicts with about 90 other unchanged inferred data. Therefore, it is important that a cancelled job be completely removed and linked data in manufacturer operation plans be updated. However, if this process is performed manually, it is prone to data errors.

For the manufacturer collapse scenario, the effect was simulated by zeroing out processing times of operations offered. The observations supported our expectation of major conflicts. A pervasive overhaul of a data update and start time adjustments are required to remove conflicts. Consequently, the probability of error during manual data entry is increased.

An operation delay was introduced in operation Op16 by tripling its processing time. A swapping of operations Op16 and Op26 can simulate a rush scenario. Both scenarios require dependent operations to undergo pervasive adjustments as shown in [Figures 5 and 6](#). These adjustments suggested by rules and schedules are manually adjusted accordingly. This process can be made more efficient through automation. Moreover, a heuristic rule can be developed to pick up on inferred idle times between operations and take effective decisions about them to boost schedule efficiency.

In light of the discussion, the second hypothesis can be accepted, suggesting that disturbance resolution can be performed in a similar manner to conflict resolution. With inferred data such as idle times, adjustments, current time budgets, earliness and lateness, a well-informed decision can be taken in resolving disturbances. Therefore, systematic conflict resolution is the contribution of the proposed inference-based system (IbS). Taking into account the known capabilities of multi-agent systems, such systems can perform conflict resolution in a fast, accurate and tractable manner. This is strongly thought to be a pre-requisite for resilient manufacturing scheduling as a web service.

## 6. Conclusion

There are web-based and mainframe-based technologies to support decentralised scheduling. The chosen computation technique determines how a scheduling problem is modelled. Cloud manufacturing can enable geographically distributed manufacturing resources to be shared as a service. Manufacturing scheduling can benefit from expandable data storage, faster data queries and the analytics that the cloud space offers. Raw data in constant flux and ever-growing relationships associated with scheduling can be better stored in a graph database. Graph databases enable data storage in the form of disjunctive graphs and incremental reasoning via inference engines. This representation technique has been used in this paper to model entities involved in scheduling, their relationships and logic rules for inferences. These technologies form the basis of an advanced expert system that can be augmented with optimisation algorithms and decentralised implementation based on a multi-agent system. A scheduling knowledge base is proposed in this paper for an inference-based system (IbS). Ontology web language (OWL) was used to document facts consisting of raw data and relationships; while the logical rules of scheduling were modelled using semantic web rule language (SWRL). The facts come from the modified Muth and Thompson  $10 \times 10$  job shop scheduling problem and Microsoft Project-derived manufacturer operation plans. The results from the IbS led us to the following conclusions. First, the deviation between expected results and actual results is due to the use of default manufacturer priority to scheduling. Manufacturer scheduling priority matters. Second, conflict may be resolved too late causing inefficiencies in scheduling. Conflict resolution timing matters. During a disturbance, conflict resolution necessitates a pervasive overhaul of data relationships and start time adjustments. However, it is a manual data entry activity which is prone to error. Further works were identified including the important role of a multi-agent system in automating conflict resolution.

Future opportunities include the exploration of the migration of OWL ontology and SWRL rules into a graph database server from a legacy Protégé platform using SWRL reasoning provision, to facilitate development of a framework for knowledge bases regarding distributed creation, efficient deployment, contained maintenance and scheduling updates. This will be implemented via a multi-agency system (MAS) in three phases. First, a MAS will be developed to augment manufacturers' functioning plans and scheduling priorities in order to outperform MsP. Second, the update of a graph database using a MAS-based query will be key to an efficient scheduling process. Finally, the MAS will automate conflict resolution in a completely tractable and humanly readable manner.

### Acknowledgements

The authors would like to show our gratitude to GFM S.r.l. and Mapna (TUGA) P.J.S. for supplying the use case data for this research paper.

### Funding

The authors received no direct funding for this research.

### Author details

Salman Saeidlou<sup>1</sup>  
E-mail: [salman.saeidlou@canterbury.ac.uk](mailto:salman.saeidlou@canterbury.ac.uk)  
ORCID ID: <http://orcid.org/0000-0002-3235-6592>  
Mozafar Saadat<sup>2</sup>  
Guiovanni D. Jules<sup>2</sup>

<sup>1</sup> School of Engineering, Technology and Design, Canterbury Christ Church University, Canterbury CT1 1QU, UK.

<sup>2</sup> School of Mechanical Engineering, University of Birmingham, Birmingham B15 2TT, UK.

### Cover Image

Source: Author.

### Citation information

Cite this article as: Towards decentralised job shop scheduling as a web service, Salman Saeidlou, Mozafar Saadat & Guiovanni D. Jules, *Cogent Engineering* (2021), 8: 1938795.

### References

- Abar, S., Theodoropoulos, G. K., Lemarini, P., & O'Hare, G. M. (2017). Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24, 13–33. <https://doi.org/10.1016/j.cosrev.2017.03.001>
- Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., & Vrgoč, D. (2017). Foundations of modern query languages for graph databases. *ACM Computing Surveys (CSUR)*, 50(5), 68. <https://doi.org/10.1145/3104031>
- Aureli, S., & Del Baldo, M. (2016). Formal inter-firm cooperation and international expansion: How Italian SMEs are using the network contract. In *The Changing Global Economy and its Impact on International Entrepreneurship*. Edward Elgar Publishing. (pp. 157).
- Baffo, I., Confessore, G., & Stecca, G. (2013). A decentralized model for flow shop production with flexible transportation system. *Journal of Manufacturing Systems*, 32(1), 68–77. <https://doi.org/10.1016/j.jmsy.2012.10.002>
- Böloni, L., & Turgut, D. (2017). Value of information based scheduling of cloud computing resources. *Future Generation Computer Systems*, 71, 212–220. <https://doi.org/10.1016/j.future.2016.10.024>
- Chen, G., Jiang, T., Wang, M., Tang, X., & Ji, W. (2020). Modeling and reasoning of IoT architecture in semantic ontology dimension. *Computer*

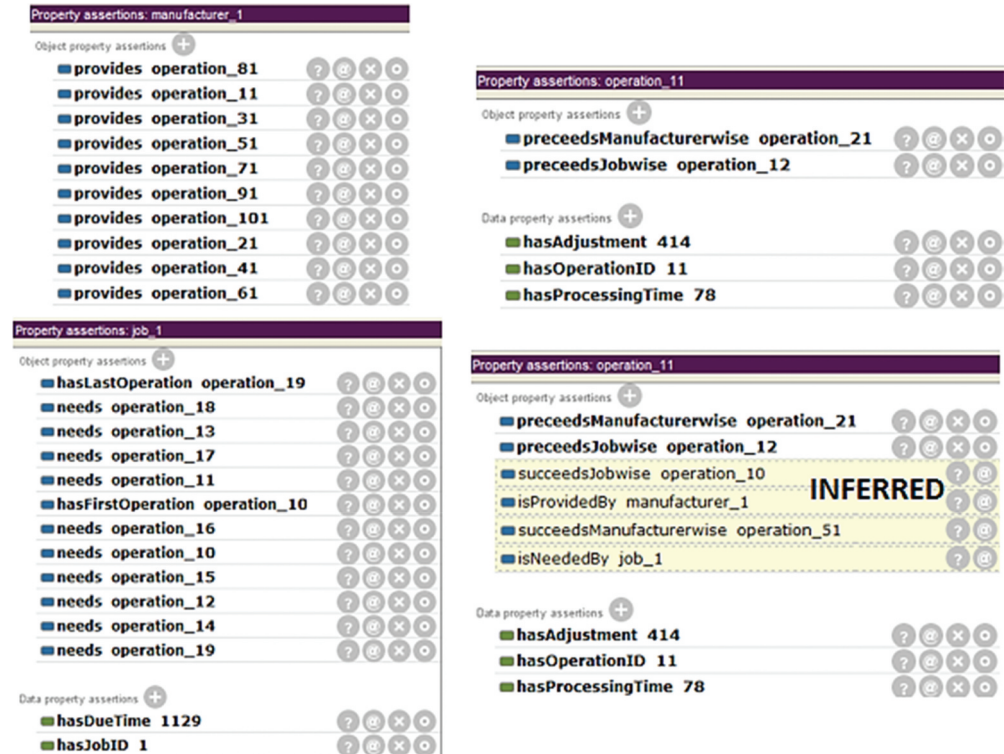
- Communications*, 153, 580–594. <https://doi.org/10.1016/j.comcom.2020.02.006>
- Egger, G., Chaltsev, D., Giusti, A., & Matt, D. T. (2020). A deployment-friendly decentralized scheduling approach for cooperative multi-agent systems in production systems. *Procedia Manufacturing*, 52, 127–132. <https://doi.org/10.1016/j.promfg.2020.11.023>
- Eleftheriadis, M. S. R. J., & Myklebust, O. (2016). A guideline of quality steps towards zero defect manufacturing in industry. In *Proc. Int. Conf. Ind. Eng. Oper. Manag.* IEOM Society International. (pp. 332–340).
- Fuchigami, H. Y., & Rangel, S. (2018). A survey of case studies in production scheduling: Analysis and perspectives. *Journal of Computational Science*, 25, 425–436. <https://doi.org/10.1016/j.jocs.2017.06.004>
- Ghadimi, P., Wang, C., Lim, M. K., & Heavey, C. (2019). Intelligent sustainable supplier selection using multi-agent technology: Theory and application for Industry 4.0 supply chains. *Computers & Industrial Engineering*, 127, 588–600. <https://doi.org/10.1016/j.cie.2018.10.050>
- Ghiyasinab, M., Lehoux, N., Ménard, S., & Cloutier, C. (2020). Production planning and project scheduling for engineer-to-order systems-case study for engineered wood production. *International Journal of Production Research*, 59(4), 1–20. <https://doi.org/10.1080/00207543.2020.1717009>
- Giordani, S., Lujak, M., & Martinelli, F. (2013). A distributed multi-agent production planning and scheduling framework for mobile robots. *Computers & Industrial Engineering*, 64(1), 19–30. <https://doi.org/10.1016/j.cie.2012.09.004>
- Guo, Z. X., Ngai, E. W. T., Yang, C., & Liang, X. (2015). An RFID-based intelligent decision support system architecture for production monitoring and scheduling in a distributed manufacturing environment. *International Journal of Production Economics*, 159, 16–28. <https://doi.org/10.1016/j.ijpe.2014.09.004>
- Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., & Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 62, 161–193. <https://doi.org/10.1016/j.compchemeng.2013.12.001>
- Helo, P., Hao, Y., Toshev, R., & Boldosova, V. (2021). Cloud manufacturing ecosystem analysis and design. *Robotics and Computer-Integrated Manufacturing*, 67, 102050. <https://doi.org/10.1016/j.rcim.2020.102050>
- Jules, G., & Saadat, M. (2017). Agent cooperation mechanism for decentralized manufacturing scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(12), 3351–3362. <https://doi.org/10.1109/TSMC.2016.2578879>
- 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. <https://doi.org/10.1109/TSMC.2014.2387099>
- Kaiser, R., Weiss, W., Falelakis, M., Michalakopoulos, S., & Ursu, M. F. (2012, July). A rule-based virtual director enhancing group communication. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on* (pp. 187–192). Melbourne, VIC, Australia: IEEE.
- Kolus, A., El-Khalifa, A., Al-Turki, U. M., & Duffuaa, S. O. (2020). An integrated mathematical model for production scheduling and preventive maintenance planning. *International Journal of Quality & Reliability Management*, 37(6/7), 925–937. <https://doi.org/10.1108/IJQRM-10-2019-0335>
- Leitão, P., Mařík, V., & Vrba, P. (2013). Past, present, and future of industrial agent applications. *IEEE Transactions on Industrial Informatics*, 9(4), 2360–2372. <https://doi.org/10.1109/TII.2012.2222034>
- Li, H., Wu, Y., & Chen, M. (2020). Adaptive fault-tolerant tracking control for discrete-time multiagent systems via reinforcement learning algorithm. *IEEE Transactions on Cybernetics*, 51(3), 1163–1174. <https://doi.org/10.1109/TCYB.2020.2982168>
- Lin, L., & Gen, M. (2018). Hybrid evolutionary optimisation with learning for production scheduling: State-of-the-art survey on algorithms and applications. *International Journal of Production Research*, 56(1–2), 193–223. <https://doi.org/10.1080/00207543.2018.1437288>
- Lu, Y., & Xu, X. (2017). A semantic web-based framework for service composition in a cloud manufacturing environment. *Journal of Manufacturing Systems*, 42, 69–81. <https://doi.org/10.1016/j.jmsy.2016.11.004>
- Misener, R., & Floudas, C. A. (2014). ANTIGONE: Algorithms for continuous/integer global optimization of nonlinear equations. *Journal of Global Optimization*, 59(2–3), 503–526. <https://doi.org/10.1007/s10898-014-0166-2>
- Morgan, J., & O'Donnell, G. E. (2017). Enabling a ubiquitous and cloud manufacturing foundation with field-level service-oriented architecture. *International Journal of Computer Integrated Manufacturing*, 30(4–5), 442–458. <https://doi.org/10.1080/0951192X.2015.1032355>
- Mourtzis, D., Doukas, M., & Giannoulis, C. (2016). An inference-based knowledge reuse framework for historical product and production information retrieval. *Procedia CIRP*, 41, 472–477. <https://doi.org/10.1016/j.procir.2015.12.026>
- Muth, J. F., & Thompson, G. L. (Eds.). (1963). *Industrial scheduling*. Prentice-Hall.
- Nedic, A., & Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1), 48–61. <https://doi.org/10.1109/TAC.2008.2009515>
- Pérez-Urbina, H., Rodriguez-Diaz, E., Grove, M., Konstantinidis, G., & Sirin, E. (2012). Evaluation of query rewriting approaches for OWL 2. In *Proc. of the Joint Workshop on Scalable and High-Performance Semantic Web Systems (SSWS+ HPCSW 2012)* (Vol. 943). Boston, USA.
- Prodan, R., Wiecezorek, M., & Fard, H. M. (2011). Double auction-based scheduling of scientific applications in distributed grid and cloud environments. *Journal of Grid Computing*, 9(4), 531–548. <https://doi.org/10.1007/s10723-011-9196-x>
- Ren, L., Zhang, L., Wang, L., Tao, F., & Chai, X. (2017). Cloud manufacturing: Key characteristics and applications. *International Journal of Computer Integrated Manufacturing*, 30(6), 501–515. <https://doi.org/10.1080/0951192X.2014.902105>
- Rodríguez, M. A., Montagna, J. M., Vecchiotti, A., & Corsano, G. (2017). Generalized disjunctive programming model for the multi-period production planning optimization: An application in a polyurethane foam manufacturing plant. *Computers & Chemical Engineering*, 103, 69–80. <https://doi.org/10.1016/j.compchemeng.2017.03.006>
- Roy, B., & Sussmann, B. (1964). Les problèmes d'ordonnement avec contraintes disjonctives. In *Note ds* (pp. 9). SEMA.

- Ruiz, R., Pan, Q. K., & Naderi, B. (2019). Iterated greedy methods for the distributed permutation flowshop scheduling problem. *Omega*, 83, 213–222. <https://doi.org/10.1016/j.omega.2018.03.004>
- Saeidlou, S., Saadat, M., & Jules, G. D. (2014, October). Cloud manufacturing analysis based on ontology mapping and multi agent systems. In *2014 IEEE International Conference on Systems, Man, and Cybernetics* (Vol.10, pp. 5–8). San Diego, CA, USA.
- Saeidlou, S., Saadat, M., Jules, G. D., & Lou, P. (2019a). Knowledge and agent-based system for decentralized scheduling in manufacturing. *Cogent Engineering*, 6(1), 1582309. <https://doi.org/10.1080/23311916.2019.1582309>
- Saeidlou, S., Saadat, M., Sharifi, E. A., Jules, G. D., & Peng, T. (2019b). Agent-based distributed manufacturing scheduling: An ontological approach. *Cogent Engineering*, 6(1), 1565630. <https://doi.org/10.1080/23311916.2019.1565630>
- Shen, W., Hao, Q., Wang, S., Li, Y., & Ghenniwa, H. (2007). An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. *Robotics and Computer-Integrated Manufacturing*, 23(3), 315–325. <https://doi.org/10.1016/j.rcim.2006.02.009>
- Shen, W., Li, Y., Hao, Q., Wang, S., & Ghenniwa, H. (2005, September). Implementing collaborative manufacturing with intelligent web services. In *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on* (pp. 1063–1069). Shanghai, China: IEEE.
- Sokolov, B., Ivanov, D., & Dolgui, A. (2020). *Scheduling in industry 4.0 and cloud manufacturing*. Springer.
- Tao, F., Zhang, L., Venkatesh, V. C., Luo, Y., & Cheng, Y. (2011). Cloud manufacturing: A computing and service-oriented manufacturing model. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 225(10), 1969–1976. <https://doi.org/10.1177/0954405411405575>
- Wagner, W. P. (2017). Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies. *Expert Systems with Applications*, 76, 85–96. <https://doi.org/10.1016/j.eswa.2017.01.028>
- Wang, M., Wang, H., Vogel, D., Kumar, K., & Chiu, D. K. (2009). Agent-based negotiation and decision making for dynamic supply chain formation. *Engineering Applications of Artificial Intelligence*, 22(7), 1046–1055. <https://doi.org/10.1016/j.engappai.2008.09.001>
- Wang, T., Guo, S., & Lee, C. G. (2014). Manufacturing task semantic modeling and description in cloud manufacturing system. *The International Journal of Advanced Manufacturing Technology*, 71(9–12), 2017–2031. <https://doi.org/10.1007/s00170-014-5607-z>
- Yen, G. G., & Ivers, B. (2009). Job shop scheduling optimization through multiple independent particle swarms. *International Journal of Intelligent Computing and Cybernetics*, 2(1), 5–33. <https://doi.org/10.1108/17563780910939237>
- Zhang, H., & Roy, U. (2018). A semantics-based dispatching rule selection approach for job shop scheduling. *Journal of Intelligent Manufacturing*, 30, 2759–2779. <https://doi.org/10.1007/s10845-018-1421-z>
- Zhao, C., Luo, X., & Zhang, L. (2020). Modeling of service agents for simulation in cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, 64, 101910. <https://doi.org/10.1016/j.rcim.2019.101910>

## Appendix

### (A) Instantiating data objects and linking data

Figure 1. Input data required by pellet reasoner to perform inferences.



Operations are instantiated with their respective properties. Jobs and manufacturers are related to operations via “provides”, “needs”, “hasFirstOperation” and “hasLastOperation” predicates. Process plans of jobs are set up using predicates “precedesJobWise”, “precedesManufacturerWise”. Predicates “succeedsJobWise”, “succeedsManufacturerWise”, “isProvidedBy” and “isNeededBy” follow the transitive rule and their attributes are inferred by the Pellet Reasoner.

(A) Constraints calculation rules

(B) Local verification rules

(C) Adjustment suggestion rules

(D) Plan scheduling rules

(E) Global verification rules

**Table 1. SWRL constraint calculation rules**

Rule A1	<p>Earliest possible start time for the FIRST operation is zero</p> <p>hasFirstOperation(?j,?o) □ hasEarliestPossibleStartTime(?o,0)</p>
Rule A2	<p>Earliest possible finish time for the FIRST operation is its processing time which is also the earliest possible start time for the NEXT operation</p> <p>hasFirstOperation(?j,?o1),hasProcessingTime(?o1,?pt),precedesJobwise(?o1,?o2) □ hasEarliestPossibleFinishTime(?o1,?pt), hasEarliestPossibleStartTime(?o2,?pt)</p>
Rule A3	<p>Earliest possible finish time for ANY operation is its start time plus its processing time which is also the earliest possible start time for the NEXT operation</p> <p>hasEarliestPossibleStartTime(?o1,?est),add(?eft,?est,?pt),precedesJobwise(?o1,?o2),hasProcessingTime(?o1,?pt) □ hasEarliestPossibleFinishTime(?o1,?eft), hasEarliestPossibleStartTime(?o2,?eft)</p>
Rule A4	<p>Earliest possible finish time for the LAST operation is its start time plus its processing time</p> <p>hasLastOperation(?j, ?o), hasEarliestPossibleStartTime(?o, ?est), hasProcessingTime(?o, ?pt), add(?eft, ?est, ?pt) □ hasEarliestPossibleFinishTime(?o, ?eft)</p>
Rule A5	<p>Latest possible finish time for the LAST operation is the due time of the job and latest possible start time for that same operation is the same due time minus the operation's processing time</p> <p>hasLastOperation(?j,?o),hasDueTime(?j,?dt),hasProcessingTime(?o,?pt),subtract(?lst,?dt,?pt) □ hasLatestPossibleFinishTime(?o,?dt), hasLatestPossibleStartTime(?o,?lst)</p>
Rule A6	<p>Latest possible start time for ANY operation is its latest possible finish time minus its processing time. Its latest possible finish time is the latest possible start time of the succeeding operation</p> <p>hasLatestPossibleStartTime(?o1, ?lst1), subtract(?lst2, ?lst1, ?pt), hasProcessingTime(?o2, ?pt), succeedsJobwise(?o1, ?o2) □ hasLatestPossibleStartTime(?o2,?lst2), hasLatestPossibleFinishTime(?o2, ?lst1)</p>



**Table 2. SWRL local verification rules**

<p>Rule B1</p>	<p>Compatibility is positive one when proposed start time of an operation lies in the interval of its earliest and latest possible start time</p> <p>hasProposedStartTime(?o,?pst),  hasEarliestPossibleStartTime(?o,?est),  hasLatestPossibleStartTime(?o,?lst),  greaterThanOrEqualTo(?pst,?est),lessThanOrEqualTo(?pst,?lst)⊔hasCompatibility(?o,1)</p>
<p>Rule B2</p>	<p>Compatibility is zero when the schedules for a pair of operations overlap</p> <p>precedesJobwise(?o1,?o2),hasProposedFinishTime(?o1,?ft), hasProposedStartTime(?o2,?st),greaterThan(?ft,?st)⊔hasCompatibility(?o1,0), hasCompatibility(?o2,0)</p>
<p>Rule B3</p>	<p>Compatibility is negative one when an operation violates Rule D1 and proposed start time of operation is earlier than the earliest permissible start time</p> <p>hasProposedStartTime(?o,?pst),  hasEarliestPossibleStartTime(?o,?est), lessThan(?pst,?est) ⊔ hasCompatibility(?o,-1)</p>
<p>Rule B4</p>	<p>Compatibility is negative two when an operation violates Rule D1 and proposed start time of operation is later than the latest permissible start time</p> <p>hasProposedStartTime(?o,?pst),  hasLatestPossibleStartTime(?o,?lst), greaterThan(?pst,?lst) ⊔ hasCompatibility(?o,-2)</p>
<p>Rule B5</p>	<p>Operation earliness message when compatibility is negative one</p> <p>hasEarliestPossibleStartTime(?o,?est),subtract(?overlap,?est,?st),equal(?c,-1),hasProposedStartTime(?o,?st), hasOperationID(?o,?id),hasCompatibility(?o,?c), stringConcat(?str," operation_",?id," is too early by ","?overlap)⊔ hasOperationConflict(?o,?str)</p>
<p>Rule B6</p>	<p>Operation lateness message when compatibility is negative two</p> <p>equal(?c,-2),subtract(?delay,?st,?lst),  hasProposedStartTime (?o,?st),hasCompatibility(?o,?c),  hasOperationID(?o,?id),stringConcat(?str," operation_",?id," is too late by ","?delay),  hasLatestPossibleStartTime(?o,?lst)  ⊔hasOperationConflict(?o,?str)</p>

**Table 3. SWRL adjustment suggestion rules**

Rule C1	<p>Operation start time adjustment message when compatibility is zero</p> <pre>equal(?c,0),subtract(?overlap,?ft,?st),precedesJobwise(?o1,?o2),hasCompatibility(?o2,?c),hasProposedFinishTime(?o1,?ft),hasProposedStartTime(?o2,?st),hasOperationID(?o1,?id1),hasOperationID(?o2,?id2),hasCurrentTimeBudget(?o2,?ct),subtract(?diff,?ct,?overlap),stringConcat(?str,"operation_",?id1," finish time overlaps operation_",?id2," start time, adjust operation_",?id2," [", ?diff, "]" by ":",?overlap)∩hasOperationConflict(?o1,?str),hasOperationConflict(?o2,?str)</pre>
Rule C2	<p>Succeeding operation start time adjustment messages for all possible successors of current operation</p> <pre>stringConcat(?str, "If operation_",?ID2," [RB", ?diff, "PT", ?pt, "]", " succeeds operation_",?ID1, " adjust operation_", ?ID2, " by ":", ?overlap), hasOperationID(?o2,?ID2), hasProcessingTime(?o2,?pt),provides(?m,?o1),hasProposedStartTime(?o2,?pst2),provides(?m,?o2),hasOperationID(?o1,?ID1),subtract(?overlap,?pft1,?pst2),hasProposedFinishTime(?o1, ?pft1),hasCurrentTimeBudget(?o2,?ct), notEqual(?ID1,?ID2),Manufacturer(?m),subtract(?diff,?ct,?overlap)∩hasOperationConflict(?o1,?str)</pre>

**Table 4. SWRL plan scheduling rules**

Rule D1	<p>Proposed start time of ANY operation is its earliest possible start time plus time adjustment required for conflict resolution and proposed finish time is its proposed start time plus its processing time</p> <pre>hasEarliestPossibleStartTime(?o,?est),hasAdjustment(?o,?adj), hasProcessingTime(?o,?pt),add(?pst,?est,?adj),add(?pft,?pst,?pt)∩hasProposedStartTime(?o,?pst),hasProposedFinishTime(?o, ?pft)</pre>
Rule D2	<p>Current time budget is the time left before a proposed finish time creates a conflict with an operation latest possible finish time</p> <pre>hasProposedFinishTime(?o,?pft),subtract(?diff,?lft,?pft),hasLatestPossibleFinishTime(?o,?lft)∩hasCurrentTimeBudget(?o, ?diff)</pre>

**Table 5. SWRL global verification rules**

Rule E1	<p>Passes test if finish time of last operation of job process plan is not greater than due time of job and displays the earliness</p> <p>hasTestID(?test,?testId),equal(?testId,"Due Time Test"), hasLastOperation(?j,?o),hasOperationID(?o,?opId),hasDueTime(?j,?dt),hasProposedFinishTime(?o,?ft),lessThanOrEqual(?ft,?dt),hasJobID(?j,?id),subtract(?earliness,?dt,?ft),stringConcat(?str," job_",?id," [DT: ",?dt,") and last operation_",?opId," [FT: ",?ft,"): PASSED with earliness ",?earliness) □ hasPassed(?test,?str)</p>
Rule E2	<p>Passes test if operations in a job process plan do not overlap and displays idle time between adjacent operations</p> <p>hasTestID(?test,?testId), equal(?testId, "Process Plan Test"), preceedsJobwise(?o1,?o2),hasOperationID(?o1,?id1),hasOperationID(?o2,?id2), hasProposedFinishTime(?o1,?ft), hasProposedStartTime(?o2,?st),lessThanOrEqual(?ft,?st),subtract(?idle,?st,?ft), stringConcat(?str," operation_",?id1, " [FT: ",?ft, " ] and operation_",?id2, " [ST: ",?st, " ]: PASSED with idle ",?idle) □ hasPassed(?test,?str)</p>
Rule E3	<p>Passes test if operations in a manufacturer operation plan do not overlap and displays idle time between adjacent operations</p> <p>hasTestID(?test,?testId), equal(?testId,"Operation Plan Test"), preceedsManufacturerwise(?o1,?o2), hasOperationID(?o1,?id1), hasOperationID(?o2,?id2), hasProposedFinishTime(?o1,?ft), hasProposedStartTime(?o2,?st),lessThanOrEqual(?ft,?st),subtract(?idle,?st,?ft), stringConcat(?str," operation_",?id1, " [FT: ",?ft, " ] and operation_",?id2, " [ST: ",?st, " ]: PASSED with idle ",?idle) □ hasPassed(?test,?str)</p>
Rule E4	<p>Fails test if operations in manufacturer operation plan overlap and displays overlap time</p> <p>hasTestID(?test,?testId), equal(?testId, "Operation Plan Test"), preceedsManufacturerwise(?o1,?o2), hasOperationID(?o1,?id1), hasOperationID(?o2,?id2), hasProposedFinishTime(?o1,?ft), hasProposedStartTime(?o2,?st),greaterThan(?ft,?st), subtract(?overlap,?st,?ft),stringConcat(?str," operation_",?id1, " [FT: ",?ft, " ] and operation_",?id2, " [ST: ",?st, " ]: FAILED with overlap ",?overlap) □ hasFailed(?test,?str)</p>
Rule E5	<p>Fails test if finish time of last operation of job process plan is greater than job due time</p> <p>hasTestID(?test,?testId),equal(?testId,"Due Time Test"), hasLastOperation(?j,?o),hasOperationID(?o,?opId),hasDueTime(?j,?dt),hasProposedFinishTime(?o,?ft),greaterThan(?ft,?dt),hasJobID(?j,?id),subtract(?lateness,?ft,?dt), stringConcat(?str," job_",?id, " [DT: ",?dt, " ] and last operation_",?opId, " [FT: ",?ft, " ]: FAILED with lateness ",?lateness) □ hasFailed(?test,?str)</p>

(Continued)

**Table 5. (Continued)**

Rule E6	Fails test if operations in job process plan overlap and displays overlap time
	<pre>hasTestID(?test,?testId),equal(?testId,"Process Plan Test"),precedesJobwise(?o1,?o2),hasOperationID(?o1,?id1),hasOperationID(?o2,?id2),hasProposedFinishTime(?o1,?ft),hasProposedStartTime(?o2,?st),greaterThan(?ft,?st),subtract(?overlap,?st,?ft),stringConcat(?str,"operation_",?id1," [FT:?",?ft,"] and operation_",?id2," [ST:?",?st,"]: FAILED with overlap ",?overlap) □ hasFailed(?test,?str)</pre>



© 2021 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

You are free to:

Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.

You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



**Cogent Engineering (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.**

**Publishing with Cogent OA ensures:**

- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

**Submit your manuscript to a Cogent OA journal at [www.CogentOA.com](http://www.CogentOA.com)**

