On Designing a Unified Ontology for Holonic Manufacturing Networks

Giouvanni Désiré Jules, Mozafar Saadat, and Nan Li

School of Mechanical Engineering, University of Birmingham, United Kingdom
`{gdj039,m.saadat}@bham.ac.uk, nxl968@adf.bham.ac.uk`

Abstract. Small and medium enterprise (SME) manufacturers are generally better off being part of groups of integrated companies which collectively add value to an end-product. An SME is limited in two ways: by its resources and by its knowledge. In contrast, a well-created manufacturing network should have the necessary competencies in resources and knowledge it needs. While an individually owned ontology inherits the heterogeneous nature of the SMEs, the manufacturing network system integrator needs a universal knowledge base and ontology; an ontology that the SMEs would understand and 'willingly' contribute information to. This paper presents the manufacturing system ontology with a foundational framework from the Product Resource Order Staff Architecture (PROSA). With multi-agent system environment in mind, the ontology is designed for agent interpretability. The exchange and processing of production, production execution and process information need to be automated as far as possible. This paper intends to present a reusable and scalable ontology in Ontology Web Language (OWL). The paper highlights the concepts and slots that constitute the ontology and a knowledge base with a set of rules that allows selection of resources for the manufacturing of a product. The proposed ontology is finally appraised against a set of criteria and compared with a number of existing ontologies for manufacturing networks.

Keywords: Ontology, Knowledge Base, Expert System, Manufacturing System, Manufacturing Network, PROSA.

1 Introduction

Small and medium manufacturing enterprises share a unique set of characteristics that make them suitable for participation in networks. They have lean structures, oriented to a high-tech market segment, adaptable to the changes in the market segment and strive in subcontracting relations [1]. Due to their lean structure, SMEs have very few core competencies. Also being at the receiving end of outsourcing and subcontracting, they behave like independent network nodes. They also follow a set of decision making rules which are often limited to the scope of the SMEs' activities; rules in the form of relations, recommendations, directives, strategies and heuristics [2]. In contrast, a manufacturing network is designed to have all the competencies in resources and knowledge it needs, to fulfill a temporary market demand i.e. job shop production. There is a general agreement across the research field of industrial production, entrepreneurship and economics that SMEs add more value as part of a network than on their own [1-4]. SMEs within the network are held together by a sense of reliability, responsibility and commitment; in other words, trust binds a network. An empirical study was carried out to determine what makes manufacturing networks successful [5]. The study showed that, in order of importance, reliability, commitment on behalf of the network, capability and information technology are critical success factors for long term survival. A network of SMEs working together, for the first time, has to be closely monitored. During this incubation period, the network is coordinated by a system integrator. After many successful deliveries and when the reliability of the network converges towards maturity, trust is established and the intervention of the system integrator would become less critical. Trust is also a function of the length of the collaboration [6]. Referring back to the incubation stage of the network, in order to perform its role, the system integrator needs a centralized source of organized information consisting of process, capacity, performance of nodes (or SMEs), inter-node transport and a set of coordination rules [1]. The manufacturing network system integrator would benefit from a unified ontology that the SMEs would contribute information to. The ontology is one of two prerequisites for constructing an expert system. The second is the knowledge base which holds decision making rules.

Determination of the content of ontology has been the subject of much research. Manufacturing strategists take into account order information, product structures, routing data, resource information and production feedback data among others, to determine the manufacturing processes to be used. Subsequently, the manufacturing cost and leadtime of a product are derived from the manufacturing processes used [7]. In a study the need for data such as the routing, bill of materials, state of the resources, availability of resources, production schedule, priority of order and inferred permission is highlighted, when investigating the online simulation in a holonic manufacturing system [8]. In an evaluation of the reliability of network plans in cell manufacturing systems, various equations, from variables mean time between failure and mean flow time have been derived [9, 10]. An extensive case study was carried out to identify what attributes have had significant influence in the long-term success of manufacturing SMEs [11]. A strategy focusing on company orientation, price determination, production experience, product life cycle and quality control have been identified as the top five attributes for long-term survival. Moreover, it is suggested that pre-process, in-process and post-process inspection are common attributes of successful manufacturers [12].

Product resource order staff architecture (PROSA) implements the concept of autonomous co-operating agents to manufacturing systems. Agent is a computer science term and the term 'holon' is its counter-part in the physical world. Holon is something that is simultaneously a part of another whole and a self-contained whole to its subordinated parts [13]. PROSA provides three basic holons of type product, resource, order and an ad-hoc holon of type ‘staff’. It is suggested that from these four types of holons only, a holonic manufacturing system (HMS) can be built. The result is a reconfigurable system, with a high degree of self-similarity, scalability and compatibility [14]. The development and application of the holonic concept in manufacturing has been widely reviewed [15]. However, the most complete holonic system so far, has been developed for the ADACOR project where a multi-agent system and a rule-based engine were utilized [16]. Based on literature review, it is understandable why manufacturing networks are likely to consist of SMEs. The paper, therefore, proposes the ontology for the domain of manufacturing network that implements the principles of PROSA. The scope of the ontology is limited to the type of information that the system integrator needs in order to carry out its function. Section 2 presents the research methodology, while section 3 explains the structure of the ontology in terms of concepts and slots. Section 4 highlights the rules used to select resources for the manufacture of a product. Section 5 explains how the multi-agent system, the ontology and the rule-based engines work together. In section 6 the proposed ontology is appraised against a set of criteria and compared with existing ontologies for manufacturing networks.

2 Research Methodology

There are several methodologies for building ontologies that have been developed over the past 15 years. This paper uses an adaptation of the Uschold and King’s method [17].

The road map, shown in Figure 1, depicts the development phases of the ontology for the unification of manufacturing system’s knowledge for manufacturing networks.

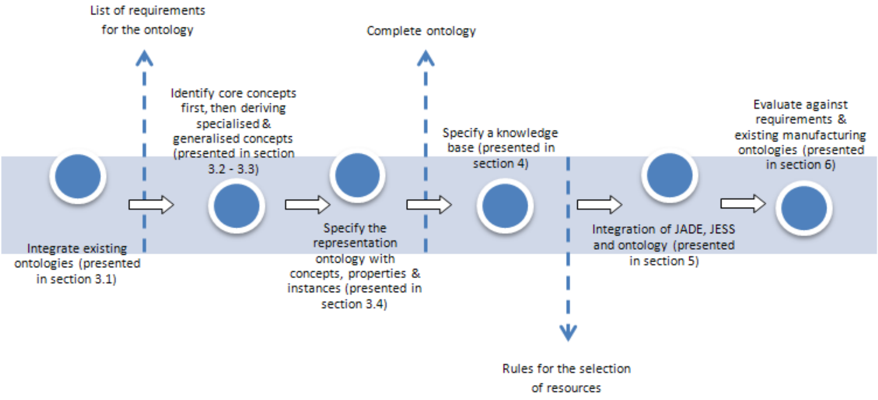


Fig. 1 Proposed methodology for ontology development

As the first step, the non-functional requirements of the ontology have been established by considering a number of fundamental points [18]. The requirements together with their explanations are given below:

S1 – Versatility of ontology to support the logistic, technical and control aspect of domain: This means that the ontology is designed to be reused by all agents involved in the system. An individual agent uses the part of the ontology that is important to its functions and ignores the irrelevant parts.

S2 – Ease of defining rules from ontology, for the knowledge-base: This means that the ontology is designed using a consistent methodology, the naming convention used is as close to the terms used in the domain of interest and the ontology is consistent with the right constraints in place. These allow the rules to be generated intuitively and to model accurately the decision-making process.

S3 – Appropriateness of ontology as communication tool for interacting agents: This means that the ontology can be used by agents to transmit objects encoded in XML or string format over a distributed network and the objects would be recognized by all agents using the same ontology.

S4 – Industrial accessibility of the data the ontology is designed to store: This means that the ontology has been designed for data that user can transfer over the network that is proprietary to the network.

S5 – Relevance of data for investigating reliability of manufacturing networks: This means that the ontology has to recognize quality, cost and delivery (QCD) data that are used to investigate reliability and process capabilities.

S6 – Accuracy of ontology to model the structures of data, used in the coordination of manufacturing networks for job shop production: This means that the ontology should model the data circulating in manufacturing networks and not the data used on manufacturers' shopfloors.

S7 – Ease of extending scope of the ontology by integrating specialized ontologies: This means that if the ontology needs to be specialized, for instance on the technical aspect of the product design, the ontology should have an extension point to integrate the specialized ontology.

To achieve these requirements, the tools available to the research community are investigated i.e. development platform, evaluation tools, extensions with inference engines and ontology generators for multi-agent systems. A number of available ontology tools that are in use today include Ontolingua Server, WebOnto, Protégé, WebODE, OntoEdit, OntoStudio, KAON, Observer, MnM, COHSE and UBOT AeroDAML. In this research the Protégé platform was chosen.

Protégé platform is an ontology-editor and a knowledge-base framework system. Protégé has relevant advantages over the other platforms. It supports two methods of modeling a domain, one of which is Protégé-OWL. It also provides a graphical user interface to develop the ontology. Moreover, Protégé supports Semantic web rule language (SWRL) which is used for developing the knowledge base. Protégé also allows the translation of SWRL rules to Java Expert System Shell (JESS) rules. JESS and SWRL will be explored in more details in Section 4.

3 Development of Ontology through an Industrial Case Study

The construction of the ontology is divided into four sections. The knowledge captured in the ontology is heavily based on literature on small and medium manufacturing enterprises, manufacturing systems in general, and holonic manufacturing systems in particular.

The appropriateness of the information to the system integrator of a manufacturing network is judged on the basis of a case study that was carried with a company acting as a system integrator to a manufacturing network. Gruppo Fabricazione Meccanica (GFM) Srl is a private company located in the province of Bergamo, Italy. GFM is a system integrator for a network of more than 30 manufacturing companies and over 500 specialized suppliers which collectively provide hundreds of processing capabilities. The company manages the production of parts and assembly equipment for gas turbines, steam turbines and electrical generators [19]. The company subcontracts orders to manufacturers based on their capability. GFM has control over the selection of manufacturers and the logistics surrounding the product i.e. collection and delivery of raw material, semi-finished and finished products, and would regularly monitor the progress of its orders to ensure that the logistics is not disturbed. The company also performs the quality inspection on the semi-finished products prior to their delivery to the next manufacturer, or on the finished product prior to its final delivery to the customer. However, GFM has no control over the manufacture of the products, which are independently managed by the manufacturers unless the manufacturers are under-performing. Thus, the system integrator acts as a coordinator and in order to perform its role, it would require the right information, which is modeled by the ontology proposed in the following sections. The ontology also captures the required information to evaluate the probability of the logistics failing during the makespan of the product i.e. the reliability of the manufacturing network.

3.1 Integrating Existing Ontologies

One of the key benefits of ontology is the opportunity to merge it with existing ontologies. Using existing ontologies not only saves time and effort but gives structure that is required for compatibility with particular applications. For instance, in the case of this paper, the ontology imports ‘OWLSimpleJADEAbstractOntology.owl’, ‘swrla.owl’ and ‘sqwrl.owl’ ontologies. The former allows our ontology to be compiled using Bean Generator tool which generates a FIPA compliant java-based ontology for the multi-agent platform JADE. The ‘swrla’ and ‘sqwrl’ ontologies allow the use of semantic web rule language (SWRL) to create the knowledge base for our ontology.

3.2 Identification of the Abstract Concepts

The proposed ontology is built using the structure of the existing ontologies. In line with the methodology given in Figure 1, the sub-classes of the class *‘Beangenerator:Concept’* are first identified. Using a middle-out strategy [17], the abstract concepts are initially identified and shown in Figure 2. The purpose of the concept is described as follows. *Products* capture the production details such as bill of material, network plan and list of capable resources. *Resources* represent capability and historical records of operation, performance, order and product. *Orders* capture the logistical aspect of production such as fixed due time, quantity and contracted resources. *Operation* captures the type of operation and technical description of operation. *Beangenerator:AID* stands for Agent ID and gives agent its name and location e-addresses in the network. *Beangenerator:AgentAction* captures the type of actions performed by agents to change its internal and environment states. *ValuePartition* is the additional information used to refine the concepts and to indicate the state of the concepts.

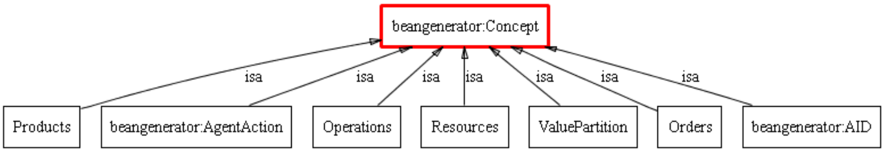


Fig. 2 Main concepts of the ontology

3.3 Identification of the Specialized Concepts

The *'Beangenerator:Concept'* are specialized into more specific concepts. The ontology must maintain a good balance between its usability and reusability. The scope of the ontology is also limited to the information that the system integrator needs, to project-manage manufacturing networks. For example, it may be tempting to specialize the concept 'Resources' into 'manufacturer', 'inspector', 'haulier', 'warehouse' and 'packager'. However, apart from their difference in the services they provide, they all have the same property types such as 'name', 'product history', 'order history', 'operation history' and 'performance history'. Figures 3 – 8 show the proposed taxonomy for a manufacturing network.

Assembly, Subassembly, Component and *RawMaterial* are specializations of the *Products* concept as shown in Figure 3a. *BoughtStockOrder* and *MakeToOrderOrder* are specializations of *Orders* as shown in Figure 3b. *BoughtStockOrder* contains the order name, quantity, arrival time and due time. *MakeToOrderOrder* contains the order name, quantity, price, due time and a checklist for delivery on time, quality and external assistance required.

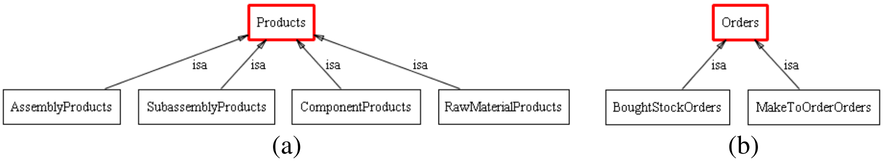


Fig. 3 Products and orders concepts

ProductHolonAID, ResourceHolonAID, OrderHolonAID and *StaffHolonAID* are the specializations of *Beangenerator:AID* as illustrated in Figure 4a. *ProductHolonAID* consists of the product managed and the actions for managing the product. *OrderHolonAID* consists of the order managed and the actions for managing the order. *ResourceHolonAID* consists of the resource managed and the actions for managing the resource. It can consist of other *ResourceHolonAID*. *StaffHolonAID* consists of adhoc holons for the scheduling and sequencing of order. *Beangenerator:AgentAction* specializes into *OrderHolonAction, ResourceHolonAction and ProductHolonAction* as Figure 4b shows.

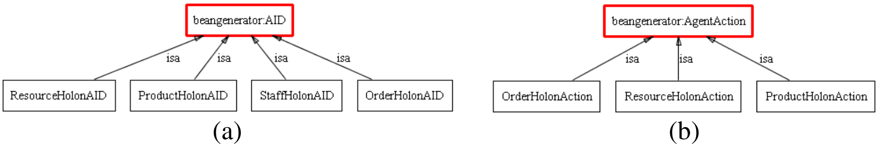
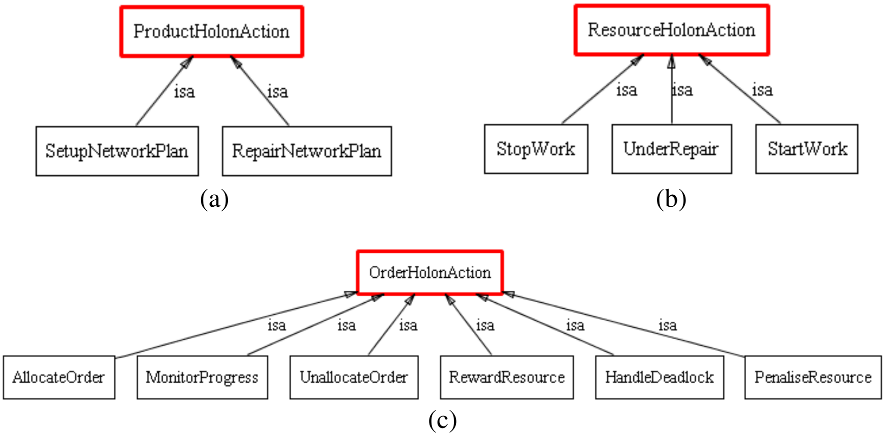


Fig. 4 AID and AgentAction concepts

ProductHolonAction further specializes into *SetupNetworkPlan* and *RepairNetworkPlan* as shown in Figure 5a. *SetupNetworksPlan* creates many alternative network plans for a product and finds potential resources to form the networks. A network plan is equivalent to a process plan. *RepairNetworkPlan* finds an alternative network plan to a faulty network. *StartWork, StopWork* and *UnderRepair* are specializations of *ResourceHolonAction* as shown in Figure 5b. *StartWork* indicates the the resource has started processing an order. *StopWork* indicates that the resource is idle. *UnderRepair* indicates that the resource is affected by a breakdown. *OrderHolonAction* is specialized into *AllocateOrder, HandleDeadlock, MonitorProgress, PenaliseResource, RewardResource* and *UnallocateOrder* as illustrated by Figure 5c. *AllocateOrder* contracts a resource with a product via an order agreement. *HandleDeadlock* resolves the conflicts between order holons needing the same resource. *MonitorProgress* monitors the tardiness, progress and status of an order. *PenalizeResource* penalizes the resource for breaching order agreement. *RewardResource* rewards the resource for a well delivered order. *UnallocateOrder* voids the contract with a resource.

Beangenerator:Predicate specializes into *Deadlock, FaultyNetworkPlan, OrderAllocation, OrderPriority, OrderProgress, OrderStatus, Performance, NetworkPlan, ResourceStatus, ScheduledStartTime* and *ScheduledFinishTime* as shown by Figure 6. *Deadlock indicates* the conflicting orders and the target resource. *FaultyNetworkPlan* indicates the faulty resources affecting the network. *OrderAllocation* shows the list of potential resources for the order. *OrderPriority* indicates the priority assigned to order. *OrderProgress* indicates the percentage of order completion. *OrderStatus* indicates that an order has started, finished, has been accepted or has been rejected. *Performance* indicates the resource-operation reliability, usage and the mean time between failures. *NetworkPlan* shows the sequence of operations and the list of resources having at least one operation required by a product. *ResourceStatus* indicates the work state of the resource. *ScheduledStartTime* shows the start time of the order and *ScheduledFinishTime* shows the finish time of the order.



(a) (b)

(c)

Fig. 5 ProductHolonAction, ResourceHolonAction and OrderHolonAction concepts

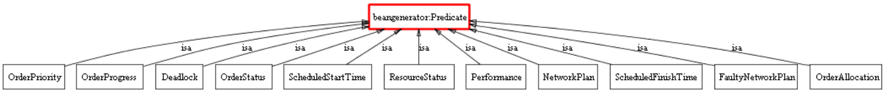


Fig. 6 Predicate concept

3.4 Identification of Slots of Concepts

A slot is an attribute which defines the characteristics of a concept. The property of a slot is called a facet. A facet represents the cardinality of a slot, the type of a slot and default values [20]. Table 1 presents all the slots that define the concepts described in section two of this paper. In the context of a multi-agent system, the slots with the dynamic data will be monitored at regular intervals by the agents. Slots having static data will be monitored by the agents only when a static data has been modified, which would be a rare occurrence.

Table 1 All the slots that define the concepts of the ontology

Slot names			
Slots that contain static data			
canBeContractedWith	hasOrderHistory	onOperation	hasDueTime
hasActions	hasPerformanceHistory	requiresOperations	hasName
hasContractWith	hasProduct	hasActionID	hasNotRequiredExternalSupervision
hasInputComponent	hasProductHistory	hasArrivalTime	hasPassedQuality
hasInputRawMaterial	hasProductSpecification	hasBeenDelivered	hasPriceRatio
hasInputSubassembly	hasResource	hasBestLeadtime	hasQuantity
hasOperationHistory	hasSubordinates	hasBestPrice	
hasOrder	isType	hasDescription	
Slots that belong to subclasses of beangenerator:Predicate are slots that hold dynamic data			
approvedResourceHolons	hasWorkStatus	hasOrderProgress	hasStateDuration
conflictingOrderHolons	resourceHolonsForNetworkPlan	hasPriority	hasUsageFrequency
faultyResourceHolons	hasFinishTime	hasReliabilityScore	hasTardines
hasOrderStatus	hasMeanTimeBetweenFailure	hasStartTime	

4 Development of Knowledge Base

The ontology enables the representation of concepts and their slots. However in order to develop an expert system, the ability for decision making needs to be implemented in the form of a knowledge base. Semantic web rule language (SWRL) is an expressive OWL-based rule language [21] that is used to define the relationship between individual concepts. An inference engine such as java expert system shell (JESS) [22] interprets the relationship and carries out the decisions made.

Below are examples of two sets of rules that enable the selection of suitable service providers and manufacturers for products *p*.

(Rule 1) ResourceHolonAID (?rh)∧hasProductHistory(?rh, ? p)∧hasProduct(?ph, ?p)∧ProductHolonAID(?ph) → canBeContractedWith(?ph, ?rh)

(Rule 2a) ProductHolonAID(?ph)∧ProductHolonAction(?networkplan) → hasActions(?ph, ?networkplan)

(Rule 2b) ProductHolonAID(?ph)∧hasProduct(?ph, ?p)∧hasActions(?ph, ?action)∧requiresOperations(?p, ?operation)∧ swrlx:makeOWLThing(?networkplan, ?ph) → NetworkPlan(?networkplan)∧problemSolvingAction(?networkplan, ?action)

(Rule 2c) ProductHolonAID(?ph)∧hasProduct(?ph, ?p)∧requiresOperations(?p, ?operation)∧hasOperationHistory(?rh, ?operation)∧ ResourceHolonAID(?rh)∧hasnetworkplan(?ph, ?networkplan)∧NetworkPlan(?networkplan) → resourceHolonsForNetworkPlan(?networkplan, ?rh)

(Rule 2d) NetworkPlan(?networkplan)∧swrlx:makeOWLThing(?rh, ?networkplan) → ResourceHolonAID(?rh)∧ hasmegaresourceholon(?networkplan, ?rh)

(Rule 2e) NetworkPlan(?networkplan)∧hasmegaresourceholon(?networkplan, ?megaresourceholon)∧ ResourceHolonAID(?megaresourceholon)∧resourceHolonsForProcessPlan(?networkplan, ?rh)∧hasnetworkplan(?p, ?networkplan)∧ Products(?p) → hasSubordinates(?megaresourceholon, ?rh)∧hasProductHistory(?rh, ?p)

Rule 1 establishes a relationship between the product history of the resource holons *rh* and product *p* of the product holons *ph*. In other words, the rule matches product holons with resource holons which have worked on the same products before.

Rules 2a-2e are for the scenario where a product has never been produced before and has no process plan. The rules are followed sequentially in response to the 'networkplan' action of the product holons. The inference engine queries for all the resource holons that have the operation capability to satisfy all the operation requirements of product holons. The rule enables the rule-based engine to form networks of resource holons that collectively provide the required operation capability. It must be noted that resource holons can consist of other resource holons by definition. The *'hasProductHistory'* slot of the new resource holons is then updated. Finally by re-using rule 1, these resource holons potentially can be directly contracted with the product holon, when the next order is placed.

5 Integration of JADE, JESS and Ontology

JESS is an instrument that can be used to add artificial intelligence to multi-agent systems that was built using JADE. JADE is a JAVA based software agent middleware and it provides an environment and the services that the agents need, to work. In contrast to other development framework such as JACK®, JADE does not provide the tools for developing intelligence in its agents while JESS can be used to implement a rule-based type of intelligence in individual agents. The ontology is essential to enable accurate and effective communication in the multi-agent system and for JESS to work.

JADE provides the communication method which enables decentralized agents to transmit data objects. The ontology plays a vital role in communication. The sender agents transmit the data objects in an XML-based language and the receiver agents convert the XML-based messages back into data objects. The XML-schema that is used to convert an object into an XML-based message and vice versa is stored in the ontology. This communication method removes the need for the serialization of data objects resulting in a faster transmission performance. Moreover, XML-based messages have a low memory utilization footprint. Also, the communication method is effective even when the agents are distributed on devices with different operating systems as long as the device has a Java Virtual Machine (JVM).

Once the data objects are received, the agents need to use the data and make decisions. This intelligence can be implemented in many ways but JESS provides a slightly faster and more insightful method as shown previously in section 4. The advantage of using JADE, JESS and the ontology together is that JESS can be configured to receive, process and transmit the XML-based messages without conversion. The XML-based messages are only converted into objects to take user inputs and to display information to the user.

The multi-agent system is being developed to assist the users during the coordination of a manufacturing network. The ontology models the type of data that are important for the coordination of the network of manufacturing shop floors. JESS is used to model the decision making process taking place during the formation and operation of a manufacturing network.

6 Evaluation against Requirements

The proposed ontology has been compared to a number of existing ontologies in the literature with respect to the requirements defined in section two of the paper, as shown in Table 2. S1 to S7 represent the non-functional requirements of the required ontology.

Ontology 1 has been developed to represent the manufacturing resources of a shop floor producing electronic connectors. Here, the ontology does not provide important information such the schedule of raw material, and finished product delivery. It also does not show the bill of material. Moreover, no history of resource performance is available. Ontology 2 is well designed but the logistics, technical and control information is intermixed. A decoupled ontology is preferred to facilitate its use by heterogeneous agents and, and also for maintenance. Moreover, the ontology has no history for order tardiness, resource breakdown, quality failure, etc. Ontology 3 is clearly decoupled into customer, product, manufacturer, transport. The top level ontology is very reusable, but it does not contain history of performance. Also, the top level ontology is very basic while the domain level is too subjective to be reusable for the domain of job shop production. Ontology 4 is well designed and accurately describes a manufacturing plant. However, the ontology represents the domain of mass production. Also the naming convention used for the slots of the concepts, is not appropriate. See Table 1 for examples of the correct naming convention. Ontology 5 is good but acts as a bridge between those with different syntax. Thus it is not designed to contain relevant data for the coordination of a manufacturing network. MASON is the most comprehensive ontology for manufacturing in literature. It is also freely available online in OWL format. The downside here is that it is too specialized for shop floor applications. The system integrator cannot use this ontology to coordinate a manufacturing network. Moreover, the ontology demands information that the system integrator does not have access to since much of the information is owned by the manufacturers.

Literature review reveals that the availability of reusable ontologies in the field of manufacturing is fairly limited, but this is likely to improve significantly due to increasing availability of development tools. Future work will involve the comprehensive development of the knowledge base using artificial intelligence. This will be achieved through a further case study with GFM Srl. The proposed ontology and the emerging knowledge base will be used in tandem with a multi-agent system. This will facilitate the investigation of the effects that rules, relations, recommendations, directives, strategies and heuristics have on the reliability of manufacturing networks.

Table 2 Evaluation of the proposed ontology and the existing ontologies with respect to requirements

Manufacturing ontologies	S1	S2	S3	S4	S5	S6	S7	Comments
Proposed ontology	+	+	+	+	+	+	+	Reducible to essentially order, product, resource domain
Ontology 1 [23]	-	+	+	+	-	+	+	Ontology is limited to resource domain
Ontology 2 [24]	-	+	+	+	-	+	+	Ontology shows no control aspect of resource domain
Ontology 3 [25]	+	+	+	+	-	-	+	Ontology is functionally sound but concepts used are inaccurate
Ontology 4 [26]	+	-	-	+	-	+	-	Naming convention, for relation between concepts, is complex
Ontology 5 [27]	+	+	+	+	-	-	+	Ontology acting as a mediator between dissimilar ontologies
MASON [28]	+	+	+	-	-	-	+	Ontology is very specialized for in-house production

+ Satisfies the requirements **- Does not satisfy the requirements**

7 Conclusion

In this paper, a type of knowledge that a unified ontology should capture has been developed. The proposed ontology was designed based on the principles of product resource order staff architecture (PROSA). Then, a knowledge base was presented with examples of rules for the selection of resources for manufacturing a product. The proposed ontology was evaluated against a set of non-functional requirements and compared with existing manufacturing ontologies. The proposed ontology has met all the requirements that are relevant to the scope of its future use. Its strong foundation from PROSA allows scalability without compromising compatibility, whilst the system integrator can use it to request information from its manufacturers and vice versa. Furthermore, the ontology is uniquely designed for network coordination and its reliability evaluation during the makespan of products.

Acknowledgments. The authors wish to thank GFM Srl (Italy) for their kind contribution in providing the case study for this research.

References

1. Mezgár, I., Kovács, G.L., Paganelli, P.: Co-operative production planning for small- and medium-sized enterprises. International Journal of Production Economics 64(1-3), 37–48 (2000)
2. Durkin, J.: Expert Systems - Design and Development. Prentice Hall International Inc., NJ (1994)

3. Galbraith, C.S., Rodriguez, C.L., DeNoble, A.F.: SME Competitive Strategy and Location Behavior: An Exploratory Study of High-Technology Manufacturing. Journal of Small Business Management 46(2), 183–202 (2008)
4. Noori, H., Lee, W.B.: Dispersed network manufacturing: adapting SMEs to compete on the global scale. Journal of Manufacturing Technology Management 17(8), 1022–1041 (2006)
5. Sherer, S.A.: Critical success factors for manufacturing networks as perceived by network coordinators. Journal of Small Business Management 41(4), 325–345 (2003)
6. Sharyn Smith, S.H.: Role of trust in SME business network relationships. In: 1997 USASBE/ICSB World Conference, San Francisco, California (1997)
7. Halevi, G., Wang, K.: Knowledge based manufacturing system (KBMS). Journal of Intelligent Manufacturing 18(4), 467–474 (2007)
8. Cardin, O., Castagna, P.: Using online simulation in Holonic manufacturing systems. Engineering Applications of Artificial Intelligence 22(7), 1025–1033 (2009)
9. Das, K., Lashkari, R.S., Sengupta, S.: Reliability consideration in the design and analysis of cellular manufacturing systems. International Journal of Production Economics 105(1), 243–262 (2007)
10. Seifoddini, H., Djassemi, M.: The effect of reliability consideration on the application of quality index. Computers & Industrial Engineering 40(1-2), 65–77 (2001)
11. Kim, K.S., Knotts, T.L., Jones, S.C.: Characterizing viability of small manufacturing enterprises (SME) in the market. Expert Systems with Applications 34(1), 128–134 (2008)
12. Abdul-Aziz, Z., Chan, J.F.L., Metcalfe, A.V.: Quality practices in the manufacturing industry in the UK and Malaysia. Total Quality Management 11(8), 1053–1064 (2000)
13. Koestler, A.: The act of creation. Picador, London (1964)
14. Van Brussel, H., et al.: Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry 37(3), 255–274 (1998)
15. Babiceanu, R., Chen, F.: Development and Applications of Holonic Manufacturing Systems: A Survey. Journal of Intelligent Manufacturing 17(1), 111–131 (2006)
16. Paulo, L.: Agent-based distributed manufacturing control: A state-of-the-art survey. Engineering Applications of Artificial Intelligence 22(7), 979–991 (2009)
17. Uschold, M., et al.: The Enterprise Ontology. Knowl. Eng. Rev. 13(1), 31–89 (1998)
18. Schalkoff, R.J.: Intelligent System: Principles, paradigms and pragmatics. Jones and Bartlett Publishers, Boston (2011)
19. Jules, G., Saadat, M., Owliya, M.: A holonic systems approach to the formation of manufacturing networks. In: 2010 IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS), Reading, UK (2010)
20. Asuncion Gomez-Perez, M.F.-L., Corcho, O.: Ontological Engineering with examples form the areas of Knowledge Management, e-commerce and the Semantic Web. Springer, London (2004)
21. Ian Horrocks, P.F.P.-S., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language combining OWL and RuleML (2004)
22. Friedman-Hill, E.J.: Jess, The java Expert System Shell (1998)
23. Lin, L.F., et al.: Developing manufacturing ontologies for knowledge reuse in distributed manufacturing environment. International Journal of Production Research 49(2), 343–359 (2011)

24. Jiang, Y., Peng, G., Liu, W.: Research on ontology-based integration of product knowledge for collaborative manufacturing. The International Journal of Advanced Manufacturing Technology 49(9), 1209–1221 (2010)
25. Yan, J., et al.: Ontology of collaborative manufacturing: Alignment of service-oriented framework with service-dominant logic. Expert Systems with Applications 37(3), 2222–2231 (2010)
26. Giret, A., Botti, V.: Engineering Holonic Manufacturing Systems. Computers in Industry 60(6), 428–440 (2009)
27. Lin, H.K., Harding, J.A.: A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. Computers in Industry 58(5), 428–437 (2007)
28. Lemaignan, S., et al.: MASON: A Proposal for An Ontology of Manufacturing Domain. In: IEEE Workshop, DIS 2006, pp. 195–200 (2006)