International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: http://www.tandfonline.com/loi/tcim20

An ontology-based multi-agent virtual enterprise system (OMAVE): part 1: domain modelling and rule management

Bahram Lotfi Sadigh, Hakki Ozgur Unver, Shahrzad Nikghadam, Erdogan Dogdu, A. Murat Ozbayoglu & S. Engin Kilic

To cite this article: Bahram Lotfi Sadigh, Hakki Ozgur Unver, Shahrzad Nikghadam, Erdogan Dogdu, A. Murat Ozbayoglu & S. Engin Kilic (2016): An ontology-based multi-agent virtual enterprise system (OMAVE): part 1: domain modelling and rule management, International Journal of Computer Integrated Manufacturing

To link to this article: http://dx.doi.org/10.1080/0951192X.2016.1145811

Published online: 19 Feb 2016.

International Journal of Computer Integrated Manufacturing, 2016
http://dx.doi.org/10.1080/0951192X.2016.1145811

An ontology-based multi-agent virtual enterprise system (OMAVE): part 1: domain modelling and rule management

Bahram Lotfi Sadigh[a], Hakki Ozgur Unver[b]*, Shahrzad Nikghadam[a], Erdogan Dogdu[c], A. Murat Ozbayoglu[c] and S. Engin Kilic[d]

[a]*Mechanical Engineering Department, Middle East Technical University, Ankara, Turkey;* [b]*Advanced Manufacturing Laboratory, Mechanical Engineering Department, TOBB University of Economics and Technology, Ankara, Turkey;* [c]*Computer Engineering Department, TOBB University of Economics and Technology, Ankara, Turkey;* [d]*Manufacturing Engineering Department, Atilim University, Ankara, Turkey*

(*Received 8 May 2015; accepted 17 January 2016*)

New advancements in computers and information technologies have yielded novel ideas to create more effective virtual collaboration platforms for multiple enterprises. Virtual enterprise (VE) is a collaboration model between multiple independent business partners in a value chain and is particularly suited to small and medium-sized enterprises (SMEs). The most challenging problem in implementing VE systems is ineffcient and inflexible data storage and management techniques for VE systems. In this research, an ontology-based multi-agent virtual enterprise (OMAVE) system is proposed to help SMEs shift from the classical trend of manufacturing part pieces to producing high-value-added, high-tech, innovative products. OMAVE targets improvement in the flexibility of VE business processes in order to enhance integration with available enterprise resource planning (ERP) systems. The architecture of OMAVE supports the requisite flexibility and enhances the reusability of the data and knowledge created in a VE system. In this article, a detailed description of system features along with the rule-based reasoning and decision support capabilities of OMAVE system are presented. To test and verify the functionality and operation of this system, a sample product was manufactured using OMAVE applications and tools with the contribution of three SMEs.

Keywords: virtual enterprise; domain ontology; multi-agent systems; OWL; semantic reasoning

Introduction

Drastic advancements in science and technology in recent years have led to a similar rise in consumer expectations. This means that market needs more competitive, innovative and inexpensive products and services that are superior in quality to their predecessors. To remain competitive, and increase market share and profit, corporations tend to introduce a wide range of new product- or service-related choices to satisfy consumer expectations to the extent that they can (Unver and Sadigh 2013; Sadigh, Unver, and Kilic 2012). For this purpose, enterprises need to have highly agile, flexible and re-configurable infrastructures, which are expensive even for large multi-national companies (Camarinha-Matos and Afsarmanesh 2001, 2007; Petersen, Divitini, and Matskin 2001). Under these circumstances, small and medium-sized enterprises (SMEs), as the economic pillar of industrial economies, face a number of daunting of obstacles to survival in highly competitive global market conditions. Collaboration seems to be a reasonable solution for SMEs to deal with this problem. Virtual enterprise (VE) is a temporary collaboration framework among multiple business partners in a value chain designed to attain certain business goals by sharing fundamental capabilities using information and communication technologies (ICT) (Unver and Sadigh 2013; Sari, Sen, and Kilic 2008). By cooperating within a VE framework, SMEs can combine their diverse competencies to develop new, higher quality products and reduce the side-effects of market turbulence (Unver and Sadigh 2013; Petersen, Divitini, and Matskin 2001; Browne and Zhang 1999). However, collaboration among multiple manufacturing SMEs is inadequate for the development of an innovative, high-value-added product. To address the lack of innovation at the product development stage, the addition to research and development (R&D) start-ups or research institutes within the collaboration platform is equally important. Therefore, a new platform to enhance collaboration among multiple design, research and manufacturing SMEs has been proposed (Sadigh, Unver, and Kilic 2012; Sari, Sen, and Kilic 2007) in an ontology-based multi-agent virtual enterprise (OMAVE) system.

A number of attempts to model VE systems have been made to handle various business processes in VE systems. For instance, an information management system to support the formal part of inter-enterprise communications in VE systems was developed by the Bremer Institut für Produktion und Logistik GmbH (BIBA) institute (Hirsch

*Corresponding author. Email: hounver@etu.edu.tr

et al. 1995). Another example of business process modelling in VE is the goal-oriented trust model for partner selection in the VE formation phase developed by Mun et al. (2009).

VE platforms need to deal with heterogeneous data from heterogeneous sources. In order to make appropriate decisions in different VE phases (e.g. partner selection, virtual breeding environment (VBE) benchmarking, risk management, etc.), a unified view of concepts and functions must be captured. To attain these goals, all concepts, relationships and constraints must be defined precisely in a comprehensive domain model and VE-specific data should be stored in a logical and reliable manner (Ye and Li 2005; Huang, Wong, and Wang 2004). Ontologies and the corresponding knowledge bases provide the best tools for modelling such complex and dynamic domain knowledge. Ontology domain modelling is mostly used to enhance the semantics of agent-based systems for VE Systems or the integration of VE system components (Camarinha-Matos et al. 1998; Zhang and Yin 2008).

This study introduces a novel approach to the development of a flexible VE system that is capable of responding to different types of orders and products from varying industrial sectors with enhanced system reusability and reliability. Here we present the results of the ontology modelling component of this research.

Perhaps the most important phase in the VE life cycle is VE formation. The success of a VE consortium is highly dependent on the chosen partners. Therefore, choosing the most appropriate VE partners is a crucial step in the VE system. While many approaches have been proposed to address the VE partner selection problem in the literature, this study introduces a new approach to it. According to it, a hybrid multi-step agent-based partner selection process is developed and implemented in the system. The first step of the partner selection process described here begins with the elimination of unsuitable enterprises using previously defined ontology rules in ontology model. In order to test and verify the performance of the OMAVE system, a sample product was designed, processed and manufactured by applying the tools of the system.

The rest of the article is organised as follows: a brief survey of modelling approaches to VE systems is provided in the section entitled 'Related works', which also contains a discussion of ontology-based modelling approaches. Next section is devoted to a description of challenging issues in VE systems as well as the proposed solutions to these in the literature. The architecture and ontology model of the proposed OMAVE system are detailed in the methodology section, and the definition and execution of the partner selection rule are provided in semantic rules in the section 'OMAVE partner selection process'. The case study and its results are discussed in the section 'Case study', and the conclusions are given in the last section of this article.

Related works

The concept of virtual enterprise was first introduced by Byrne in 1993 as the 'temporary nature of interactions between independent enterprises using information and communication technologies (ICT)' (Byrne, Brandt, and Port 1993). Since then, several studies have been conducted to create a robust and reliable information management system for collaboration platforms, and numerous information management systems and data models have been proposed to this end. The resulting information management systems and related models have focused on different phases of VE in varying industrial sectors. The main target of most platforms and systems that have been developed has been to enhance system performance by increasing flexibility, reconfigurability and intelligence. A brief history of related work in this field follows.

An information system that can only support the formal part of inter-enterprise communication was developed by the BIBA institute (Hirsch et al. 1995). A specific information management system for process planning and control activities in the X-CITTIC semiconductor manufacturing project was also introduced by Zhou and Nagi (2002). One of the earliest studies on the development of an Internet-based information system for VE involved the development of an innovative network-centric information system for VE, and was proposed by Park and Favrel (1999). Soares, Azevedo, and De Sousa (2000) proposed an ontology-based decision support system for VE systems in semiconductor manufacturing.

Collaborative design and engineering is an essential part of the VE life cycle. To enhance collaboration among multiple entities in VE systems, an information infrastructure was developed by Ye (2002). This infrastructure helps integrate different types of information and activities to support a collaborative engineering platform.

Supported by the European Union, the PRODNET project was one of the fundamental implementations of VE systems. In this project, in order to manage the complex requirements of VE information systems, a federated information access mechanism was proposed and developed. A distributed information management system (DIMS) for different user levels was developed, and access rules were defined and used in the project. The DIMS was developed by using a relational database management system, and the federated schema management modules were written in C++ (Frenkel et al. 2001; Klen et al. 1998; Rabelo, Camarinha-Matos, and Afsarmanesh 1998; Candido and Barata 2007).

Persistent distributed data store (preDiS), which is based on distributed shared memory for VE software infrastructure, was developed by Sandakly et al. (1999).

They proposed a persistent software infrastructure with concurrent access as well as a coherent and secure distributed data store based on the distributed shared-memory paradigm. Another study related to the architecture of distributed information systems, implemented using the CORBA and STEP standards for workflow management and information exchange, was developed by Zhou and Nagi (2002). They attempted to develop a transparent communication channel and a uniform data model format to deal with heterogeneous data and knowledge in workflow management for virtual enterprises.

Vanderhaeghen and Loos developed a distributed model management platform to describe the complex business processes of virtual enterprises, either globally or locally, and adequately support virtual enterprise network design and implementation (Vanderhaeghen and Loos 2006). Vifrebras is a VBE environment that concentrates and contains enterprises from the die-and-mould sector. A VE framework called AmbianCE was developed by Vallejos et al. by relying on Vifrebras VBE to enhance the competitiveness of enterprises in VE (Rabelo, Camarinha-Matos, and Vallejos 2000).

Ontologies for VE initiators and potential partners were developed by Wang et al. to support a Service-oriented Architecture (SOA)-based interoperable and distributed platforms for negotiation among automated agents in a multi-agent-based VE negotiation process ((Wang, Wong, and Wang 2010). A goal-oriented trust model approach containing project constraints and strategies for partner selection in virtual enterprises was developed by Mun et al. (Mun et al. 2009). To facilitate and develop a context-aware work flow management system based on a multi-agent architecture, a design methodology was introduced by Hsieh and Lin. VE Petri-net models were used to describe workflows and resource activities for agents' schema coordination (Hsieh and Lin 2014). The initial Framework for the Inter-sensing Enterprise (FISEA) is an inter-enterprise architecture that supports behaviour-sensing technology introduced by Vargas et al. to show collaboration components and real business relations in virtual enterprises (Vargas, Boza, and Cuenca 2011).

Ontologies not only help model and capture complex domain knowledge, but also improve the sharing and reusability of data, and provide a suitable environment for communication between agents and humans (Gruber 1995) (Chandrasekaran, Josephson, and Benjamins 1998). Ontologies represent concepts, including knowledge formally agreed upon, and their relationships to an application domain. Therefore, unlike task-specific and implementation-oriented data schema, ontologies should be as generic and task independent as possible (Spyns, Meersman, and Jarrar 2002). Ontologies play an important role in defining the terminology used by agents in the exchange of knowledge-level messages; therefore, the choice of an ontology representation language is a significant issue when designing a multi-agent system (Cranefield, Haustein, and Purvis 2001). Considering these properties of ontological knowledge representation, ontology-based domain models have been developed to increase flexibility and meaningful-message communication between VE agents and entities, and to enhance the integration of various business process management systems (e.g. MRP or ERP) (Camarinha-Matos et al. 1998; Zhang and Yin 2008). The main reasons for developing an ontology-based domain model for agent-based VEs are (1) to create a common vocabulary for software agents to understand one another, and (2) to enable agents to reuse and analyse domain knowledge (Gruber 1995; Noy and McGuinness 2001).

To improve the integration of business processes among member enterprises, increase interoperability among system components and develop a common model for VE business activities, an ontology-based process model for the business architecture of VE was developed by Kim et al. (Kim et al. 2013).

VE applications are designed according to a modular software architecture, and serve VE partners based on the Software as a Service (SaaS) concept (SIIA 2006). These applications require continual updating and flexible, dynamic and reliable database structures to handle heterogeneous data from diverse sources in different formats. In order to establish such a database, an ontology-based data store has been developed. This structure empowers the VE system to maintain and manage dynamic and heterogeneous data and knowledge from heterogeneous resources (SIIA 2006).

It is evident from VE modelling-related literature that few studies have used ontology modelling tools to model a virtual enterprise system. Researchers have mostly used ontology models in VE systems to enhance semantics in agent communication. One of the rare studies to have implemented an ontology for VE systems is by Wang Wang, Wong, and Wang (2010). In this study, an ontology to support the negotiation process among agents in a virtual enterprise was developed. An ontology model was implemented for agents to understand one another during the bidding procedure (Wang, Wong, and Wang 2010). Another important work in implementing an ontology-based business model in VE systems was proposed by Cheolhan et al. In this study, an ontology-based process model for VE business architecture was designed. Process models were separated into activity-oriented and communication-oriented process models. The researchers here developed a common semantic model to standardise VE activities and communication protocols for data and information transmission between VE entities.

A summary of the literature surveyed here is given in Table 1. In this study, we develop a flexible and reconfigurable VE system that can be used in different industrial sectors to satisfy various manufacturing, production and

Table 1. History of the development of information and data management systems for VE platforms.

Authors	Title	Focus area in VE	Novelty	Sector/cluster focus
Byrne, Brandt, and Port (1993)	VE platform	VE System	For the first time the term VE introduced in literature	General
Hirsch et al. (1995)	Information system concept for the management of distributed production	Formal part of the inter enterprise communication	A VE information system introduced	General
Camarinha-Matos et al. (1998)	Towards an architecture for virtual enterprises	VE Architecture and Framework	VE life cycle and system architecture have been revealed	General
Zhou, Souben, and Besant (1998)	An information management system for production planning and control in virtual enterprises	Production Planning and Control	An information system for production planning in VE was developed	Semiconductors Manufacturing
Park and Favrel (1999)	Virtual enterprise- information system and networking solution	VE Information Systems-Networking	Networking solutions for VE were presented	General
Soares, Azevedo, and De Sousa (2000)	Distributed planning and control systems for the virtual enterprise: organisational requirements and development life cycle.	VE Information Systems-Distributed Planning and control	A new innovative planning and control systems were developed for VE	General
Garita, Afsarmanesh, and Hertzberger (2001)	The PRODNET cooperative information management for industrial virtual enterprises	VE information Systems	Cooperative information management system for VE has been developed	PRODNET
Klen (2001)	Managing distributed business processes in the virtual enterprises	VE Business Process	Distributed business processes management system was revealed	General
Osório and Barata (2001)	Reliable and secure communications infrastructure for virtual enterprises	VE Information System and Data Transaction	VE platform was equipped with a communication infrastructure	General
Sandakly et al. (2001)	Distributed shared memory infrastructure for virtual enterprise in building and construction	Data Share and Information System	Shared memory infrastructure was applied in VE	Building and Construction
Zhou and Nagi (2002)	Design of distributed information systems for agile manufacturing virtual enterprises using CORBA and STEP standards	Distributed Information System	CORBA and STEP standards were applied in VE	General
Ye (2002)	Information infrastructure of engineering collaboration in a distributed virtual enterprise	Collaborative Design and Engineering	Information Infrastructure for Engineering Activities	General
Vanderhaeghen and (2007)	Distributed model management platform for cross enterprise business process management in virtual enterprise networks	Modeling and Management in VE	Distributed model and business process management application in VE were introduced	General
Vallejos, Lima, and Varvakis (2007)	Towards the development of a framework to create a virtual organisation breeding environment in the mould and die sector	VBE	VBE framework introduced	Dies and Mould
Wang, Wong, and Wang (2010)	Service-oriented architecture for ontologies supporting multi-agent system negotiations in virtual enterprises	VE architecture and Negotiation Process	SOA, ontology and MAS technologies were implemented in VE	General
Mun, Shin, and Jung (2011)	A goal-oriented trust model for virtual organisation creation	VE modelling	Goal oriented trust model implemented in VE	General
Kim et al. (2013)	Ontology-based process model for business architecture of a virtual enterprise	VE Process Modeling	Implementation of ontology based modelling	General

(*continued*)

Table 1. (*Continued*).

Authors	Title	Focus area in VE	Novelty	Sector/cluster focus
Moisescu and Sacala (2014)	Towards the development of inter-operable sensing systems for the future enterprise	VE Communication Infrastructure	Sensing systems implemented and tested in VE	General
Hsieh and Lin (2014)	Context-aware work ow management for virtual enterprises based on coordination of agents	Work ow Management	Implementation of Petri net models in Agent coordination scheme in MAS based VE work ow management systems	General
Vargas et al. (2014)	Towards the development of the framework for inter sensing enterprise architecture	VE Framework	Inter- sensing enterprise architecture was developed	General

standards requirements. The other objective of this study is to find the most appropriate partners for imminent VE projects by matching ordered product and design requirements with the capabilities of the available partners in VBE using a rule-based reasoning model.

Problem description

A VE consortium is formed based on the requirements of the ordered product, and partners of the VE consortium are selected accordingly. The capabilities of the enterprise partners are key to finding and selecting the most suitable partners. A company's capabilities depend on its resources and their properties. Resources can consist of machine tools, available software, human resources, etc. A VE system must be able to demonstrate the types of resources, their location and the available free capacity that can be used in the consortium. Based on these constraints and project requirements, the most suitable companies for joining the project consortium should be chosen.

The other important issue is the variety of orders and customer requirements. In other words, since a VE system is designed to cope with different types of orders by customers, it is expected that heterogeneous data will be entered into the system. The VE system should hence be able to handle heterogeneous data types in different projects. Thus, a novel VE infrastructure must have the following capabilities:

(1) Ability to easily establish variable components, and their dependencies and relationships;
(2) Flexibility sufficient to manage different types of data and knowledge;
(3) A reconfigurable data model.

It seems that these characteristics are essential for a VE system to deal with dynamic, heterogeneous data as well as knowledge from heterogeneous data sources. In order to fill the gap mentioned in the VE literature and VE requirements, an ontology-based domain model for VE systems, its associated triple-store data storage and an information management system called OMAVE have been developed here. The results of the development and implementation of an ontology-based VE model, with the attendant unstructured triple store as a flexible data warehouse for OMAVE system, are also presented.

Requirements of ontology-based multi-agent virtual enterprise system

Efforts to establish a collaboration framework for SMEs to increase their competitiveness and the quality of products began about two decades ago. Several attempts have since been made to resolve this issue, some of which have been partly successful in creating at least some tools to enhance VE creation and formation procedures. However, several problems persist at different stages of VE systems. For instance, in the VE formation phase, several studies have been undertaken to solve the partner selection problem and establish the best possible consortium from a pool of enterprises. In the VE operation phase, the focus of research has been on operation management and risk analysis of the system. Most studies in this regard have been conducted on project management, the establishment of collaborations among consortium partners, and the prognostication of possible bottlenecks or problems during the operation phase. The most common investigation in the VE dissolution phase concerns the performance evaluation of partners in the VE consortium and relating these results to future partner selection processes. Several studies have proposed methodologies to assess partners' performances and score enterprises based on their past performance.

Recent advancements in ICT facilitate the development of appropriate infrastructures or tools to address the above problems. In order to obtain a better understanding of VE systems and their requirements, IDEF0[1] models of VE systems were developed. In order to study system components and their interactions with, Unified Modeling Language (UML[2]) diagrams (Use case, activity and sequence and entity relationship diagrams) of OMAVE have been developed as well. The main node decomposition of the IDEF0 diagram of the OMAVE system is shown in Figure 1, and a UML activity diagram for the project submission process in OMAVE is shown in Figure 2.

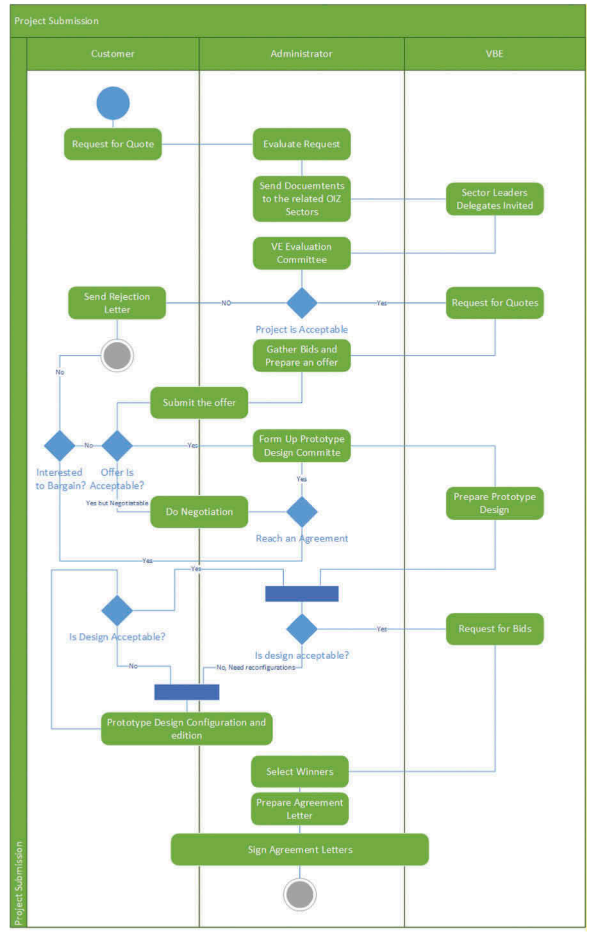

Figure 2. OMAVE System project submission process activity diagram (UML Activity Diagram).

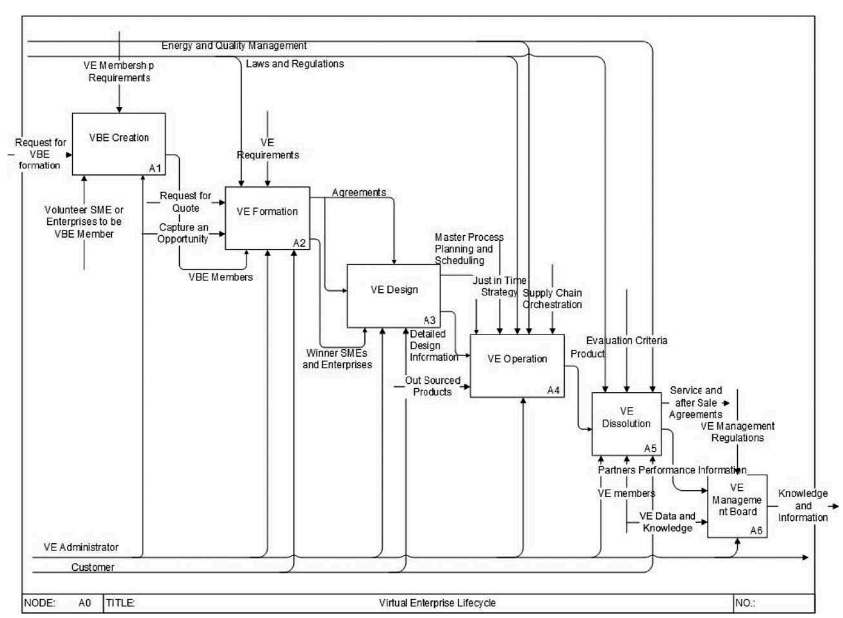

Figure 1. OMAVE IDEF0 diagram: main node decomposition.

To build a dynamic, flexible and reusable VE system, it is clear that several tools and applications should be developed for different stages of the VE life cycle. However, the most important issue here is to establish a new infrastructure for handling and managing dynamic and heterogeneous VE data to support all tools and applications in a VE system. By analysing the IDEF and UML models, various requirements for different parts of the VE system can be captured. Three recognised emerging necessities for VE systems are as follows:

- **Machine- and human-readable, flexible data infrastructure**: Virtual enterprise platforms need to be structured and designed to handle different types of orders, products and projects. Heterogeneous data from heterogeneous data sources should be stored, managed and analysed properly. Data analysis and evaluation is critical for various decision-making procedures in different VE phases. This is the most challenging aspect of virtual enterprise systems. It is extremely important to create a collaboration platform with a common language understandable by all entities in the system in order to establish a collaboration network for various industrial sectors with different structures and cultures.
- **Unpredictable, neutral and robust partner selection platform**: Perhaps the most important stage in VE processes is the partner selection stage. By selecting the most proper partners for a forthcoming VE consortium, the chance of the project's success increases significantly. Making the correct decision is mostly based on the following items:
- ***Comprehensive enterprise information***: The comprehensiveness of the information regarding registered enterprises in the Virtual Breeding Environment (VBE) is critical for the partner selection process.
- ***Selection criteria***: Considering the most appropriate criteria to evaluate enterprises and their performance is also critical to the success of a consortium. Enterprises in the OMAVE system are evaluated through two categories of criteria: dynamic and static. Static criteria are stored in the system database, and embed background enterprise information and history. Dynamic data contains the proposal data of the enterprises.
- ***Selection objectivity***: The evaluation and analysis procedures for enterprises should be objective. An integrated multi-agent-based, semantic reasoning approach is used to analyse and assess the procedures of enterprises. Product–process–machine tool–enterprise relations are created automatically using semantic rules, and enterprises unrelated to the ordered product are first eliminated. Following this, different types of agents interact and compete with one another to provide the best offer on behalf of their assigned enterprises to win a semi-English, reverse auctioning process. This auctioning process is integrated with a Fuzzy Analytical Hierarchy Process (AHP) for the Order of Preference by Similarity to Ideal Solution (TOPSIS) multi-criterial decision-making algorithm.
- **Collaborative design and data management platform**: The other important objective in the development of OMAVE is a collaborative design and information management system. OMAVE targets the enhancement of the quality of manufacturing of products produced by SMEs in the OSTIM-organised industrial zone by improving collaboration between manufacturing SMEs and R&D start-ups, research institutions and universities. Implementing a collaborative, concurrent, central design and product life cycle management (PLM) system is indispensable to such a platform.

The focus in this article is on the development of a flexible and reconfigurable VE data store and information management system. The remaining research objectives regarding the OMAVE partner selection system are presented in the second part of this article.

Methodology

OMAVE framework

The focus in the proposed framework is on enhancing data and knowledge reusability and improving system capability to respond to a diversity of requirements from a variety of industrial sectors. This framework aims to provide a collaboration platform among manufacturing SMEs, design and research SMEs, and research institutions. In order to attain innovative research results, such as redesigning or optimising existing products operationally and/or environmentally, another step prior to the operation phase is required. This phase is called the 'collaborative design and engineering phase'. In some VE hierarchies, this phase is assumed to be part of the operation phase. However, it does not yield the expected results (Alzaga and Martin 1999; Siqueira and Bremer 2001). When collaborative design and engineering is part of the operation phase, only minor changes by the enterprise, which is responsible for manufacturing the product, are applicable. Such changes cannot be considered as part of the collaborative design because they feature very few or no contributions from other consortium partners or the customer. The VE architecture developed in this article separates the design phase from the operation phase, and proposes a completely distinct phase within the VE life cycle (Figure 3).

Figure 3. OMAVE system life cycle (Sadigh et al. 2012).

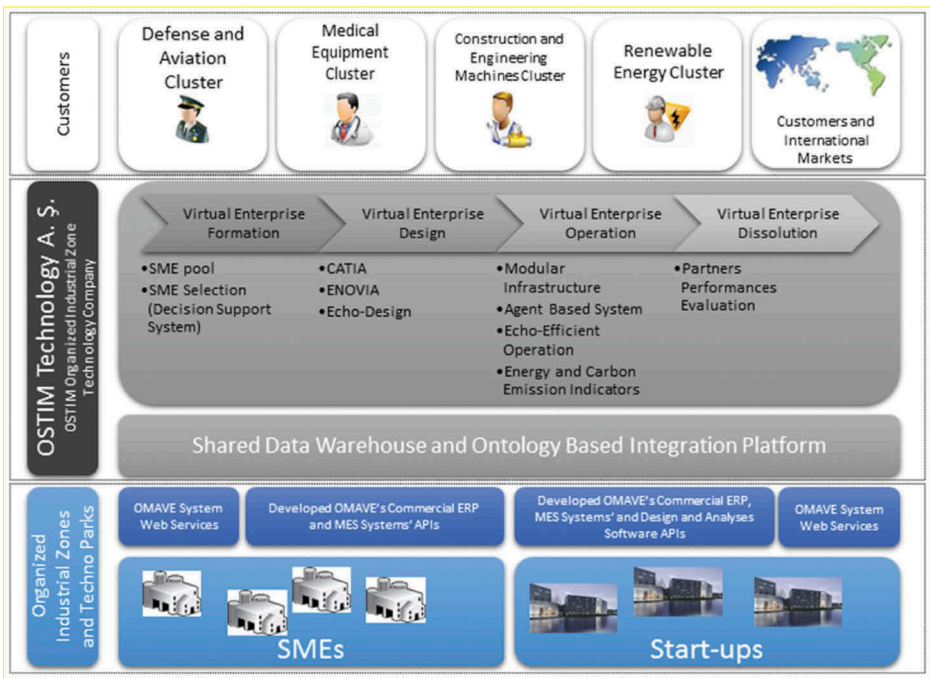

Figure 4. OMAVE framework (Sadigh et al. 2014).

As shown in Figure 4 a, a three-layered Information and Communication Technologies (ICT) framework is developed for the OMAVE system, consisting of

(1) The Enterprise Layer;
(2) The Shared Database and Applications Layer;
(3) System Interfaces.

The bottom level of this framework involves data and information pertaining to system members and partners, such as SMEs, original equipment manufacturers (OEMs), research institutes and R&D start-ups in techno parks. Information concerning members is gathered manually or automatically by agents or Web services. Different types of agents are designed to capture various types of data from member enterprises. In case a Virtual Breeding Environment (VBE) member enterprise is equipped with an automated enterprise management system, such as enterprise resource planning (ERP), manufacturing execution systems (MES) or similar tools, it is required to provide appropriate interfaces to gather the required information from member organisations. Here, suitable data collection agents or Web services directly connect to the databases of commercial systems and immediately collect the required information. It is clear that for different commercial software, different application programming interfaces (APIs) need to be developed. However, if the enterprise does not have the required level of automation, information is collected manually by sending questionnaires to the relevant authorities within the member enterprises. A sample of these agents can consist of enterprise agents in the partner selection process responsible for gathering enterprise information regarding the negotiation process. In this scenario, it is assumed that the enterprise does not have any pre-installed commercial management software (which is the case in most OSTIM-organised industrial zones), and provides a simple graphical user interface (GUI) for enterprise authorities to obtain the required information for the OMAVE system. Figure 5 shows a simple enterprise agent GUI to obtain an enterprise bid for a product.

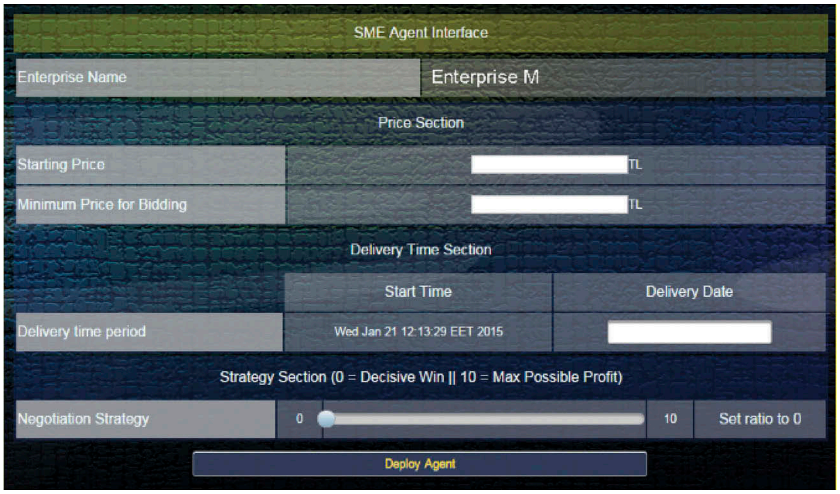

Figure 5. Enterprise agent GUI to obtain enterprise bid.

The middle layer of the framework embeds the shared system data warehouse, the system ontology model and all integrated applications. This layer is divided into three sub-layers. The lowest sub-layer is the OMAVE ontology model. The shared system data warehouse (triple store), which is based on the OMAVE ontology model, is placed in the middle sub-layer. Despite other approaches in literature, instead of using an object-oriented or relational database, a triple store is utilised in this system. This is based on the OMAVE ontology model. The OMAVE system database is developed based on the OMAVE ontology model without using any specific database management system; therefore, the OMAVE system can be considered platform independent and portable, and capable of integration into other commercial software and systems.

The middle-upper layer of the system encloses all applications and tools of different phases of the VE life cycle. For instance, VBE tools for collecting information from pool enterprises and agent-based decision support tools can be listed as OMAVE formation phase tools. To manage product design and development activities as well as production processes and operation management, the recently introduced Web-based Dassault Systems CATIA/ENOVIA V6[3] tools were utilised. In this approach, all design and development activities are carried out on a Web-based platform, and all system users connect to the system's main frame, which runs on a central Web server. Clients connect to the system's main frame, and contribute to the design and engineering process based on the defined rules and credentials assigned by the project manager. ENOVIA V6 is the backbone of this software package. Data and information generated during the design and development stages are supported by ENOVIA, the Product Lifecycle Management (PLM) tool that supervises and handles data and information flow during all stages of the product life cycle.

Various enterprises with different core competencies contribute remotely during the product development stage by benefiting from these Web-based tools to produce high-value-added, high-tech products in cooperation with manufacturing SMEs and R&D start-ups, and universities and other institutes. These applications in the upper-administrative level of the OMAVE framework serve members based on Software as a Service (SaaS) concept (Rush 2010).

In the dissolution phase of OMAVE, a partner performance evaluation tool is needed. This tool evaluates the performance of contributing partner enterprises after project completion. In order to assess the companies' performance and use this for forthcoming projects, it is crucial to evaluate their past performances in such projects (Sari 2006). When finalising each project, the activities and operations of partner enterprises are evaluated and their current information, past performance, quality and service evaluations are updated in the system accordingly.

The system has two types of interfaces. GUIs are in the upper layer of the OMAVE architecture to interact with system clients (customers or enterprise delegates). The other interfaces are special agents or Web services designed to exchange data and information with other interconnected systems in client enterprise facilities.

The OMAVE system is an open system where different types of alliances can be formed in different domains and applications (e.g. casting, moulding, etc., besides machining and metal forming, which are exemplified in the case study). The main target of this platform is to instantly match customers' order requirements (new projects) with suitable service providers (such as SMEs). Since it is possible to have too many production processes that are too complex to be completely covered, from a manufacturing engineering point of view, implementing all manufacturing processes and their requirements for all industrial sectors as a 'one-stop solution' is almost impossible. Therefore, a VE collaboration platform catering to these needs should be easily extensible and reconfigurable according to the requirements of novel domains. The OMAVE system has been designed with these requirements in mind. This is accomplished by the non-fixed data model infrastructure in OMAVE; the data model is based on ontologies and is extensible, such that new concepts, classes and relationships can be added, or existing ones can be removed or updated, based on the requirements of a new application or domain. Therefore, the system is dynamic and adaptable to emerging requirements.

Ontology design

The most important requirement of the infrastructure of OMAVE systems is flexibility. The VE environment is dynamic and unpredictable. Thus, the domain and the system should be sufficiently flexible to be easily reconfigured and reset according to new conditions and situations. New circumstances are the result of changes in the prerequisites of the product or project, type and customer conditions. Such flexibility needs to be supported by an extremely adjustable data and knowledge infrastructure. OMAVE tools automatically require decisions on incoming data and parameters from VBE members and OMAVE partners. These tools, machines and applications in the enterprises should be able to exchange data and interact with one another smoothly. Therefore, a machine-readable and understandable common language among system entities should be developed and implemented.

In order to make appropriate decisions in VE systems (e.g. partner selection, VBE bench marking, risk management, etc.), all concepts, relationships and constraints must be defined precisely in the domain model, and VE-specific data should be stored in a logical and reliable manner (Ye and Li 2005; Huang, Wong, and Wang 2004). Ontologies and the corresponding knowledge bases provide the required tools for modelling such complex domain knowledge. Ontologies not only help model and capture complex domain knowledge, but also improve the sharing and reusability of data and knowledge (Gruber 1995; Chandrasekaran, Josephson, and Benjamins 1998). Domain knowledge is represented formally according to the defined concepts and their relationships for a specific domain. Therefore, unlike task-specific and implementation-oriented data schema, ontologies should be as generic and task independent as possible (Spyns, Meersman, and Jarrar 2002). Ontologies play an important role in defining the terminology used by system entities in the exchange of knowledge-level messages. The choice of an ontology representation language is a significant issue in designing a complex and intelligent system (Cranefield, Haustein, and Purvis 2001). Considering these properties of knowledge representation, ontology-based domain models are developed to establish meaningful communication among VE entities and enhance the integration of various business management processes (e.g. MRP or MES) used by VE partners (Camarinha-Matos et al. 1998; Zhang and Yin 2008). The main reasons for developing an ontology-based domain model for agent-based VEs are to create a common vocabulary for system entities to understand one another and enable them to reuse and analyse domain knowledge (Gruber 1995; Noy and McGuinness 2001). All types of concepts used in the domain are defined comprehensively and clearly by the ontology domain model. The resulting ontology models can be categorised into one of three main types:

- Informal: These types of ontologies are not readable and, consequently, not understandable by machines. In these ontologies, only a set of concepts is defined.
- Semi-formal: These ontologies are machine readable, but the defined concepts or relations are not equipped with the required metadata and descriptions. Therefore, machines cannot understand the meaning of the defined concepts.
- Formal: Formal ontologies are completely machine-readable models that can be processed by software.

In this research, an OMAVE domain ontology model is developed to encapsulate different aspects of enterprises, products, orders, etc. There are formal languages for encoding ontology models. The most commonly used languages for creating ontology models are resource description framework (RDF), RDF schema (RDFS) and Web ontology language (OWL). OWL knowledge representation can be encoded using XML. OWL is in fact an extension of the vocabulary of RDF and RDFS to capture complex concepts and relationships. Some shortcomings of RDF/RDFS languages have been eliminated by the development of the OWL standard. Some of these extra features added to OWL are property characteristics, disjoint classes, restriction classes, etc. OWL itself has three sub-languages – OWL Lite, OWL DL and OWL Full (Heflin 2004) – based on computational complexity.

Since an agent-based infrastructure is proposed and developed for the OMAVE system, a common communication language is developed by creating a separate agent communication ontology in order to satisfy semantic requirements of agent interaction. The OMAVE system needs to respond intelligently at different stages of the decision-making and operational phases. These decision-making, operations management or partner selection procedures require sophisticated computational or conjectural calculations, which can be supported by the OWL DL standard. The other requirement for using the OWL DL standard is the availability of augmentation in standard concepts. The OMAVE system requires flexible infrastructure for serving different industrial sectors. The OWL DL allows the enhancement of concept definitions by taking advantage of the OWL standard according to the requirements of the domain model. Therefore, in order to accomplish these goals, and by considering the semantic requirements of the targeted system domain, computational requirements and the augmentation in OWL concepts, the OWL DL language is selected to model the system domain.

Declaring concepts, rules and axioms of the ontology model

Axiomatisation

The ontology model contains statements based on triples of subject–predicate–object. Subjects and objects are defined concepts in the model, and predicates cover the properties of subjects. Concepts in ontology models are represented as classes, individuals and properties. Classes (owl:Class) are the main building blocks for representing concepts. They are defined to classify a set of entities or objects that share one or more similarities. Individuals are members of classes that represent unique instances in the system. Properties bind subjects to objects. There are two types of properties; these are created as instances of the *owl:ObjectProperty* and *owl:DatatypeProperty* classes. The first property type (object property) declares relationships between concept instances or resources. The second type (data-type property) is used to describe detailed information regarding concepts and link individuals to data values. For instance, in the OMAVE model, a machine tool, e.g. 'Machine Tool A', is an individual of the classes 'Resources/Physical Resources/Manufacturing/Manufacturing Resources/Machining Equipment/Machining Centers'. An SME, e.g. 'SME A', is an individual of the classes 'Organization/Company/SME'. An ownership relationship (property or predicate) between these two individuals can be declared in the model as follows:

$$\text{Machine tool A} \xrightarrow{veonto{:}AnAssetOf} \text{SME A}$$

$$\text{SME A} \xrightarrow{veonto{:}Owns} \text{Machine Tool A}$$

veonto:isAnAssetOf and *veonto:owns* are inverse properties of each other. 'Machine Tool A' is an asset of 'SME A' under the relation (object property) *veonto:isAnAssetOf*, and the inverse object property indicates that 'SME A' owns 'Machine Tool A' (object property *veonto:owns*).

Formalisation

Defining concepts and relations alone is not enough for a formal ontology model. In order to bring formalism and add rule-based reasoning capabilities to the model, SWRL (semantic Web reasoning language)[4] rules are developed and added to the ontology model. The definitions of OWL without rule axioms are those of an abstract model containing a set of concepts and facts, such as subclasses or individual axioms. Rules bring greater reasoning capabilities to the model. Rules consist of an antecedent and a consequent part in the following format:

$$antecedent \Rightarrow consequent$$

Each of these parts consists of a set of atoms. In a rule, if the antecedent holds the value 'true', the consequent of the rule must also hold. This logic is used in the reasoning process to generate new knowledge; whenever matching data is found for the antecedent, the corresponding consequent data is generated by the rule engine. Empty antecedent values are presumed true; however, empty consequent values imply falsity. The rules are specified as follows:

$$\begin{aligned} rule &::= {}'Implies('[URIreference]\{annotation\} \\ &\qquad antecedenconsequent')' \\ antecedent &::= {}'Antecedent'\{atom\}' \\ consequent &::= {}'Consequent'\{atom\}' \end{aligned}$$

Two types of notations are used to define rules: the SWRL notation and the Jena[5] rule notation. Jena is a Java framework for building semantic applications developed by researchers at the HP labs. Based on the W3C recommendations, Jena provides Java libraries to enable developers to handle RDF, RDFS, OWL and SPARQL rules and queries. Jena also contains an inference engine to execute rules of reasoning on defined ontologies (Horrocks et al. 2004). An example of Jena notations for different rules in the VE system is given in Table 2. For instance, in order to show the first mentioned rule in the table (machine tool selection rule), the SWRL notation will be as follows:

hasType (? w,? TurningProcess) ∧ enables (? MultiTaskingMachine,? TurningProcess)
⇒ enables (? MultiTaskingMachine,? w)
hasType (? y,? MillingProcess) ∧ enables (? MultiTaskingMachine,? MillingProcess)
⇒ enables (? MultiTaskingMachine,? y)
hasType (? z,? DrillingProcess) ∧ enables (? MultiTaskingMachine,? DrillingProcess)
⇒ enables (? MultiTaskingMachine,? z)

As an example, let us consider a partner selection procedure in VE. Assume that producing product 'Part A' requires a manufacturing process 'Manufacturing Process A'. Each manufacturing process has its own enablers. These enablers can be machine tools, software, skilled staff, etc. Assuming that 'Manufacturing Process A' can be performed using 'Machine Tool A', this relation can be represented by the *veonto:isEnabledby* property. Therefore, if an enterprise owns (*veonto:Owns*) 'Machine Tool A', the enterprise is capable of performing

Table 2. List of assigned rules for reasoning in OMAVE ontology model.

Rule name	Rule description	Sample Jena notation of rule
Machine tool selection	Searches the system based on process definition and list enabler machine tools for this process and assign listed machines to the named process. This rule is activated after every process or machine tool submission automatically. This rule also assigns reverse relation between submitted entities, which is called 'is enabled by' relation.	(veonto:MultiTaskingMachine veonto:enables veonto: TurningProcess) (veonto:MultiTaskingMachine veonto:enables veonto: MillingProcess) (veonto:MultiTaskingMachine veonto:enables veonto: DrillingProcess) (?mtm rdf:type veonto:MultiTaskingMachine) (?w rdf:type veonto:TurningProcess) (?y rdf:type veonto:MillingProcess) (?z rdf:type veonto:DrillingProcess) (?mtm veonto:enables ?y) (?mtm veonto:enables ?z) (?mtm veonto:enables ?w)
Dimensional verification	Based on selected part's physical size and volume, suitable machine tools are filtered out.	(?part veonto:hasDiameter ?a) (?part veonto:hasSizeX ?b) (?machineTool veonto:maxMachiningDiameter ?x) (?machineTool veonto:maxMachiningLength ?y) greaterThan(?x ?a) greaterThan(?y ?b) (?part veonto:canBeFittedTo> ?machineTool) (?machineTool veonto:hasAdequateSpaceFor ?part)
Power verification	Minimum required power for the selected process is compared to the listed machine tool power and machine tools with less than needed power are eliminated.	(?process veonto:hasMinPowerRequirement ?a) (?machineTool veonto:motorOutputPower ?x) greaterThan (?x ?a) (?process veonto:canBeDoneIn ?machineTool) (?machineTool veonto:canMachine ?process)
Max spindle speed verification	Checks for process' spindle speed requirements in machine tools and available machines' list is populated	(?process veonto:hasMinSpindleSpeedRequirement ?a) (?machineTool veonto:spindleSpeed ?x) greaterThan(?x ?a) (?process veonto:speedIsAvailableIn ?machineTool) (?machineTool veonto: meetsProcessSpeedRequirement ?process)
Surface roughness verification	Process' minimum necessity for surface finish is considered and machine tools capable of machining with the required design specifications are selected from potential machine tools list.	(?process veonto:hasReqSurfaceFinish ?a) (?machineTool veonto:hasMaxSurfaceFinish ?x) greaterThan(?x ?a) (?process veonto:surfaceFinishIsObtainableFrom ? machine Tool) (?machineTool veonto: hasSu_cientSurfaceFinishCapabilities ?process)
Controller resolution verification	Process' minimum necessity for machine tool resolution is considered and machine tools capable of machining with the required design specifications are selected from potential machine tools list.	(?process veonto:hasReqResolution ?a) (?machineTool veonto:hasMaxResolution ?x) greaterThan(?x ?a) (?process veonto:resolutionObtainableFrom ? machineTool) (?machineTool veonto:hasResolutionCapabilities ? process)
Part weight verification	Checks part weight requirements of the machine tools. According to the machine tool structure and capabilities, work pieces weight is restricted. Therefore over-weighted parts will be prevented from assigning to the unsuitable machine tools.	(?part veonto:hasWeight ?a) (?machineTool veonto:hasMaxPartWeightLimit ?x) greaterThan(?x ?a) (?part veonto:lightEnoughtobeMachinedin ? machineTool) (?machineTool veonto:rigidEnoughtoMachine ? process)
Accuracy verification	Process' minimum necessity for accuracy is considered and machine tools capability to satisfy required design specifications are checked and appropriate machine tools are selected.	(?part veonto:hasAccuracy ?a) (?machineTool veonto:hasMaxAccuracy ?x) greaterThan(?x ?a) (?part veonto:accuratelyMachinable ?machineTool) (?machineTool veonto:isAccurateEnoughtoMachine ?process)

'Manufacturing Process A'. This can be represented as a rule, and the above reasoning can be automatically carried out by an inference engine based on predefined rules in the OMAVE model. The following triples represent this inference:

$$\text{Part A} \xrightarrow{\textit{veonto:requires Process}} \text{Manufacturing Process A}$$

$$\text{Manufacturing Process A} \xrightarrow{\textit{veonto:is enabled by}} \text{Machine Tool A}$$

$$\text{Enterprise A} \xrightarrow{\textit{veonto:owns}} \text{Machine Tool A}$$

$$\text{Enterprise A} \xrightarrow{\textit{veonto:isCapableToProduce}} \text{Part A}$$

The rule language is used to specify this inference procedure as a general rule as follows: If

$$\text{?part} \xrightarrow{\textit{veonto:requires Process}} \text{?process}$$

$$\text{?process} \xrightarrow{\textit{veonto:is enabled by}} \text{?tool}$$

$$\text{?enterprise} \xrightarrow{\textit{veonto:owns}} \text{?tool}$$

then,

$$\text{?enterprise} \xrightarrow{\textit{veonto:isCapableToProduce}} \text{?part}$$

Here, the placeholders starting with a question mark (?) such as '?part', '?process', etc., are variables that can be replaced with the subjects and objects in the ontology model. If such subjects and objects can be found, the reasoning rule is applied. This is a very simplified example of how the production of a part can be associated with a special enterprise, following which the enterprise can be nominated as a potential partner for producing the particular product. In the real world, of course, complex products with complex requirements need to be defined in a system, which should evaluate different aspects of products to assign possible enablers and potential partners. In Table 2, the samples of the Jena rules for our OMAVE ontology model are presented. The following prefixes (URL shortcuts) are used there:

PREFIX rdf: http://www.w3.org./1999/02/22-rdf-syntax-ns#
PREFIX owl: http://www.w3.org./2002/07/owl#
PREFIX xsd: http://www.w3.org/2001/XMLSchema#
PREFIX rdfs: http://www.w3.org./2000/01/rdf-schema#
PREFIX veonto: http://www.ostim.org.tr/veonto#

Order submission scenario

The system assigns the most appropriate resources to a production process based on product design, process properties and required capabilities. Therefore, in order to provide the information needed by the system to evaluate a product and assign suitable resources, product engineering design and process planning are necessary. The customer can submit the product order with or without product design. If the customer does not provide product design for the ordered product, a consortium for designing new products should be formed. For this reason, the customer's order is divided into two separate orders, one for product design and the other for product manufacturing processes. Enterprises with design capability, or research and development start-ups with the required capabilities, may enter the design consortium. In this approach, design partners collaborate on a new product design for the customer through the Web-based CATIA v6 design tool. Following the final design approval from the customer, the verified design is the input data for the next manufacturing consortium. Based on this design and the corresponding process planning, the related task groups are formed, and the OMAVE administrator sends out a call for potential partners to submit their bids: the OMAVE formation phase begins.

In the second scenario, which is most common in organisations like the OSTIM-organised Industrial Zone, the customer's order is submitted along with the product design and the process plan. In this case, according to the provided design and process plans, task groups are formed and potential partners are asked to submit their bids. These two scenarios are shown in Figure 6.

Information management system

One of the main contributions of this research is the development of a flexible and reconfigurable information monitoring and management system. This system is developed based on the established ontology model. A triple store database is developed based on the ontology model. To edit and manage data, three main user roles are defined: administrator, customer and enterprise user. Enterprise and customer users can monitor data and information related to their own enterprises and configure their information. System administrators have more authority to edit and configure information, manage system activities, and create, edit or remove events, components, properties or rules. System administrators can also make more substantial changes to the system architecture. Based on changing system requirements, system administrators may accordingly adjust system structure.

As mentioned above, domain ontology-based modelling for VE systems is mostly used to support agent interaction in a multi-agent platform. In this study, in addition to this approach, a new paradigm in the design and implementation of the VE management system is proposed. In this case, the OMAVE management system is developed based on the system ontology model. All

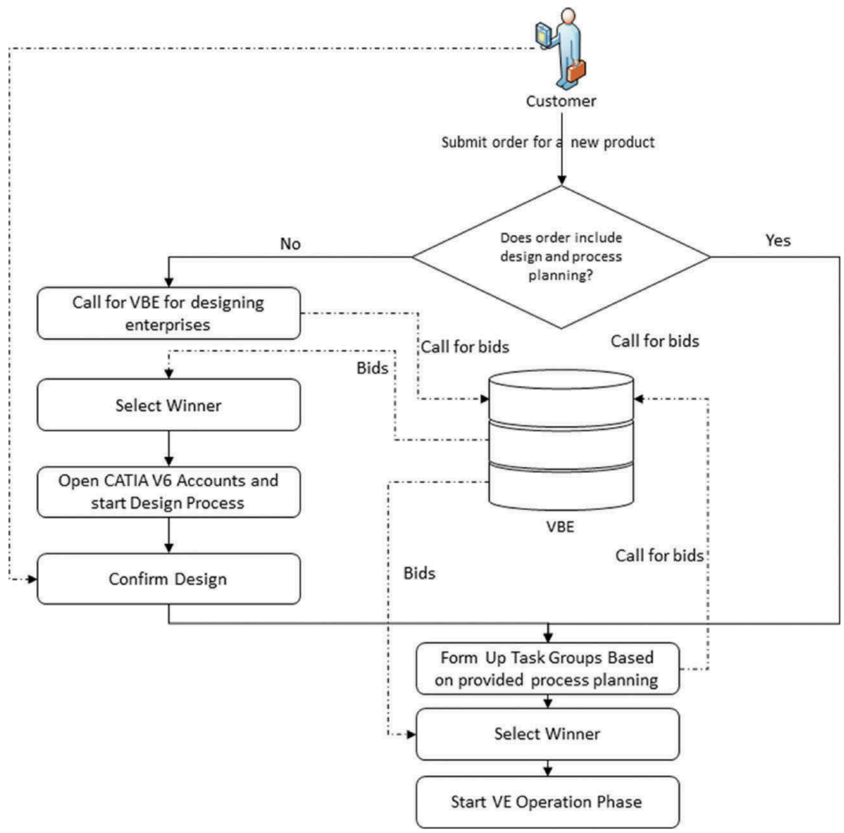

Figure 6. Order submission process with and without product design.

data and information in the system are maintained as triples in a Triple Data Base (TDB). TDB is a Jena platform component that deals with storing and querying RDF data structures. Using Jena API, the TDB store is accessible and manageable. As TDB data sets are stored as triples (or statements), in order to query these types of data sets, a semantic query language called SPARQL (SPARQL Protocol and RDF Query Language)[6] is used.

Data management tools

In the OMAVE system it is possible to change the system architecture by changing the system components, like classes, instances, properties, relations and rules, to solve problems. Data and knowledge on system triple stores are created directly using the ontology model (Dogdu, Hakimov, and Yumusak 2014). By editing information in the system, the results are directly applied to the system ontology model. Therefore, the system structure is also open to configuration. This means that the system's user may restructure the OMAVE system to fulfil new requirements and change the data and information hierarchies. Figure 7 illustrates the relation between system interface and the system ontology model. The OMAVE system management section is composed of several sections designed to change and configure system components, including classes, instances, properties, relations and rules.

To facilitate data and information searches on the OMAVE data store and enhance searching performance, specific search capabilities are developed. A search module enables users to look up all system databases or refine searches on special classes, like resources, capabilities, organisations, etc. For instance, a user may find a specific machine tool with the requested properties, like power, controller resolution, working space, etc., to fit with his/her requirements. The designed search interfaces are shown in Figure 8.

As shown in Figure 9, the system interfaces were designed using the Java Server Faces (JSF)[7] technology in XHTML format. Based on a user's requested task, an associated Web service is called. A reconfiguration procedure activates the related Java Bean, and the requested change is realised using SPARQL on the triple data store. If a user requests data monitoring (for example, running a search module), the SPARQL query runs over the system data store and returns the results to the user.

This system shows a higher flexibility, which is highly desirable in VE systems. The system administrator can change the system hierarchy or the way data items are

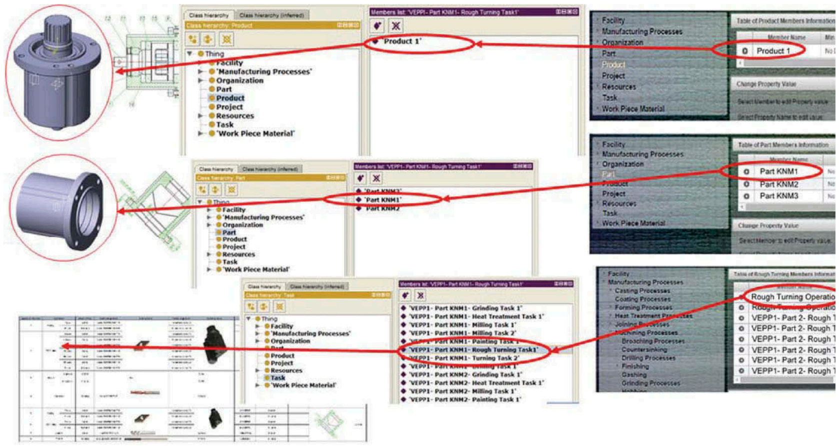

Figure 7. Relation between OMAVE interfaces and ontology model.

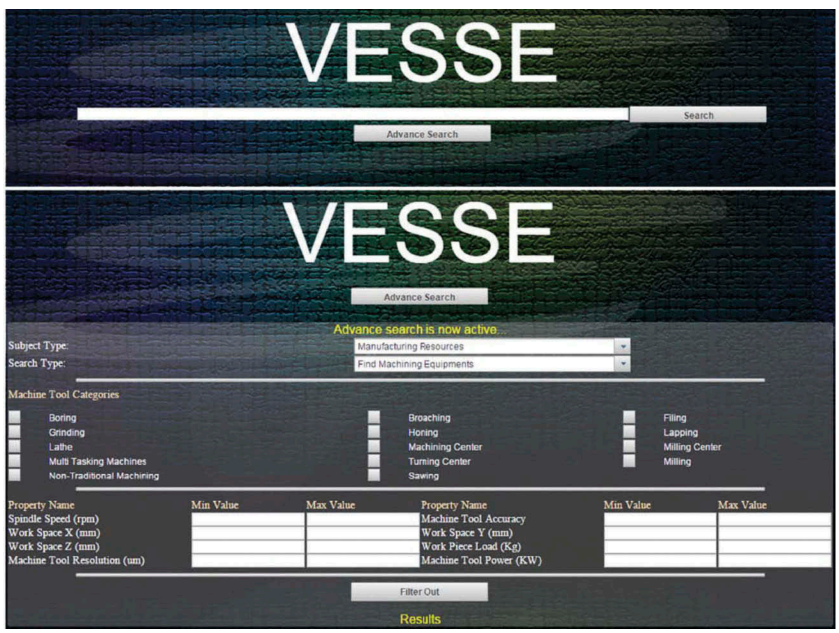

Figure 8. Simple and advanced machine tool search engines.

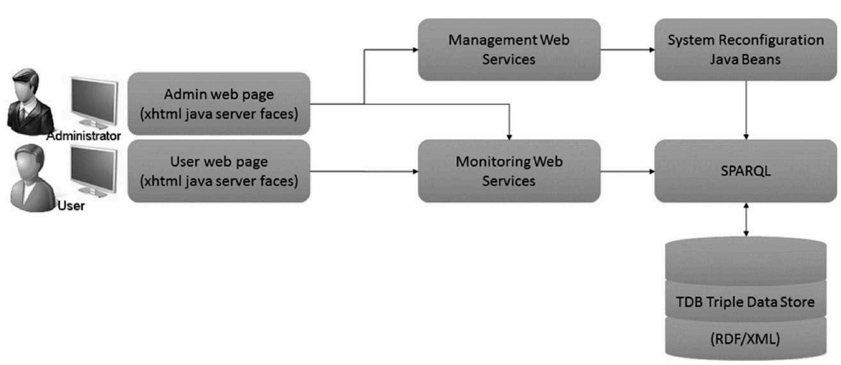

Figure 9. Data flow structure in OMAVE information management system.

stored or interrelated to one another according to new requirements and conditions. This was one of the main priorities in designing and developing OMAVE systems.

Semantic rules in OMAVE partner selection process

In order to form a consortium to manufacture a product in the OMAVE system, the first step consists of the analysis of data and the location of the best matches for the product. To find the best matches for a product, different ontology rules are applied to the system's data store. Outcomes of the kind shown in Figure 10 are sent to the OMAVE system's hybrid partner selection algorithm, the results are evaluated and the final ranks are presented to the system administrator.

In order to select the most appropriate partners in the OMAVE system, a hybrid multi-agent-based approach is proposed. As shown in Figure 11, three main decision-making steps should be used to choose the winning enterprises:

SWRL rule elimination: Irrelevant instances are removed or filtered from the potential list of partners according to predefined rules specified in the SWRL language in the ontology model. An inference or reasoning engine is used to execute the rules. The remaining partners (as instances in the ontology) are candidates for the final VE project consortium.

DEA elimination: The outcome of the first step consists of all candidate partners for the final VE project consortium. However, in order to enhance the success of the VE consortium, the best possible partners must be selected from among the remaining ones. The data envelopment analyses (DEA) method is applied in this stage to separate efficient and inefficient candidate enterprises. Inefficient enterprises are disqualified in this step. The DEA is not used to rank the enterprises, but simply removes inefficient companies from the list of available candidate partners. The remaining enterprises are invited to join the auctioning process in the last step.

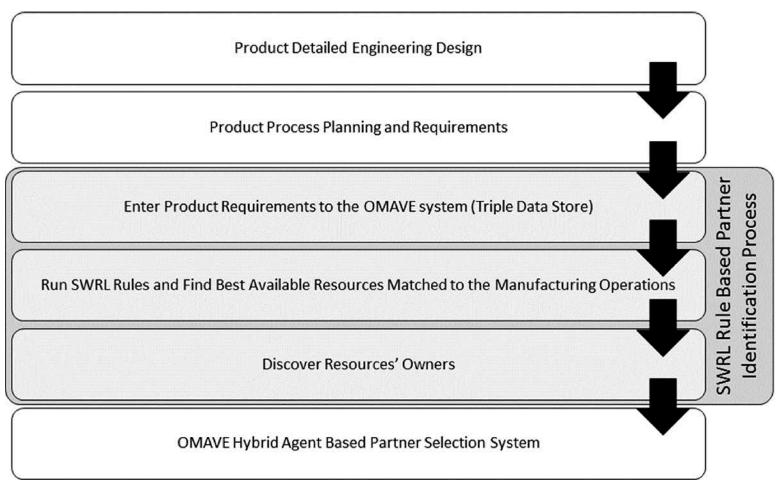

Figure 10. The steps of partner selection, and the function of SWRL rules.

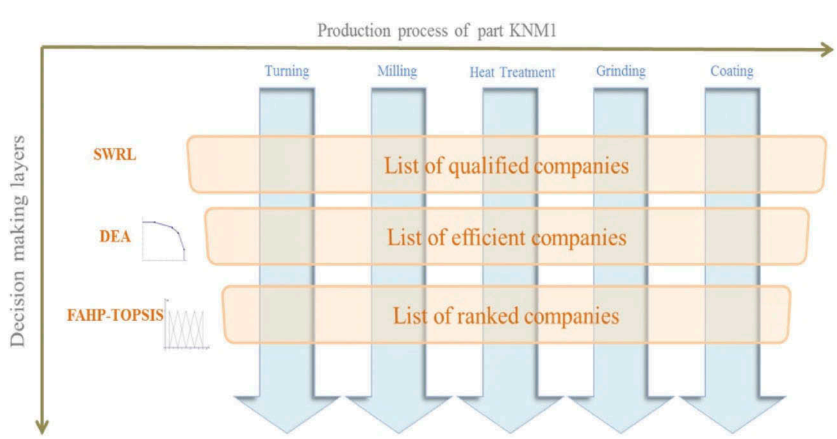

Figure 11. Step-wise decision-making scheme.

FAHP-TOPSIS: The final evaluation and ranking step is a hybrid multi-agent structure integrated with a Fuzzy Analytic Hierarchy Process (FAHP) and Technique for Order of Preference by Similarity to the Ideal Solution (TOPSIS). In this step, the candidate enterprises are asked to join the VE consortium, and submit their bids using their associated enterprise agents. A negotiation process among the enterprise agents, the task manager agent and the customer agent is then performed. Enterprises are evaluated based on the information regarding them in the system and their submitted bids. This evaluation process is accomplished by executing the FAHP-TOPSIS method, and the ranked list of bidder enterprises is presented to the system administrator at the end of the process.

This article focuses on the first step of the above process – the SWRL rule elimination step. The final results of the second and the third steps are summarised at the closing stages of this section. As mentioned above, SWRL rules are defined to relate concepts and instances in the OMAVE ontology model to enable the reasoning engine to establish the inferred model. In case any new instances are added, such as concepts, relations or data properties, the system instantly checks the data store and adds new inferred statements to the model. These new statements are in fact new ripples resulting from the latest changes to the system. SWRL rules are applied at different stages of the OMAVE system. One of these applications is the partner selection process.

Case study

According to the type and requirements of ordered products, several types of VE consortiums can be formed. This depends on the system administrator, the ordered product and the requirements and conditions when deciding how to form the project consortium. If the ordered product is a simple product, a single consortium can be suffcient to produce it. But if the assembly of the product is complex, multiple consortiums can be formed to produce different parts of it, and then assembling it. In this second scenario, it is assumed that a separate VE consortium can be formed for the production of each part of the assembly.

The case study in this article covers 'turning' and 'milling' processes involved in 'sheet metal forming'. Different types of alliances from several manufacturing sectors could be formed in this system. The authors do not claim that this system covers all requirements of all manufacturing processes at this stage. However, during the development process, this system can respond to the heterogeneous requirements of different types of products and orders. However, to do so, the system administrators should define an appropriate model and extend existing ones to cope with new requirements.

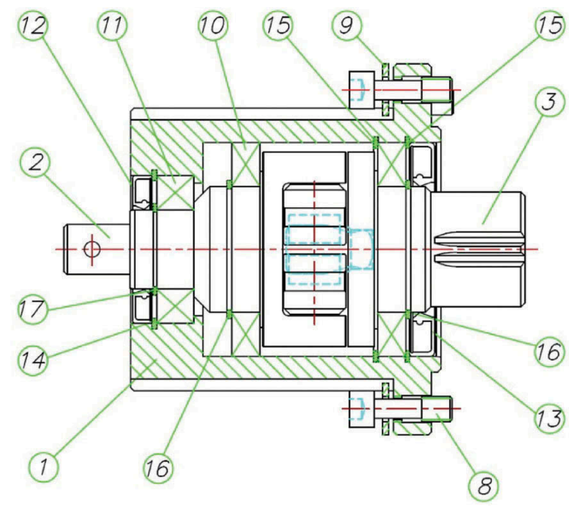

Figure 12. A sketch of the ordered product.

The case study for producing a sample product is shown in Figures 12 and 13. In this scenario, different tools and applications were developed in the OMAVE system, and were tested and verified using the sample production process. The design enterprises collaboratively designed the product via the CATIA V6 collaborative Web-based design tool on the OMAVE platform.

This sample product consisted of several sub-components. The assembly shown above had eight components: Part KNM1 to Part KNM8. In this section, the process of establishing a virtual enterprise to produce Part KNM1 is

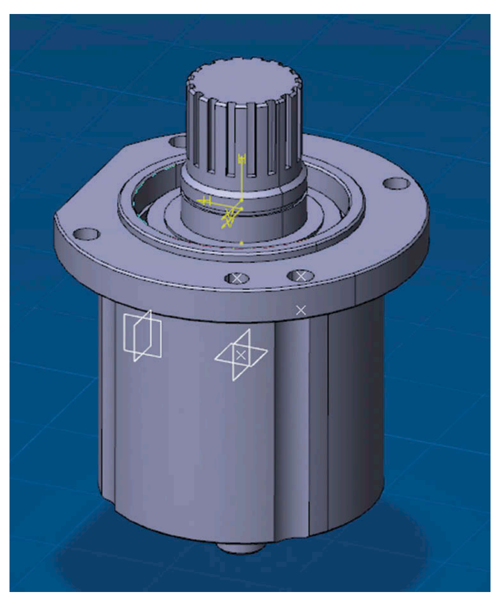
Figure 13. Assembly part designed on CATIA V6.

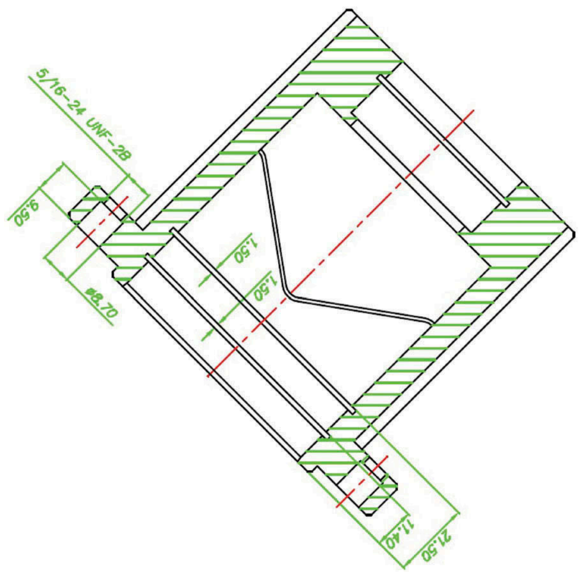

Figure 14. Drawing of a KNM1 part.

Figure 15. Task management screen.

described. A technical drawing of Part KNM1 is shown in Figure 14. Based on the design specifications and the manufacturing requirements, the production process of each part was decomposed into manufacturing processes, and product process plans were created. The manufacturing operation information of all parts was entered manually into the OMAVE system. The OMAVE reasoning engine checked manufacturing operations and created task groups by automatically considering semantic rules and raw data in the triple store based on the ontology model. Similar manufacturing processes (automatically distinguished by the system) were selected one after the other, and together created a single manufacturing task. For instance, two or more successive turning processes were placed in one turning task. However, if any other operation occurred between two turning tasks, the turning operations were divided into two turning tasks.

Rule-based elimination

The major factor for selecting the candidate partner enterprises was their resources and capabilities. In other words, the aim was to relate a new job or task to an enterprise through task enablers or enterprise resources. Each task demands certain requirements, like specific machine tools with given properties, a software, etc. On the contrary, enterprises own different types of physical, software or human resources. The plan was to match task requirements to enterprises resources. This matching was accomplished automatically by the reasoning engines by inferring knowledge according to predefined semantic rules. Part KNM1, as shown in Figure 15, required six manufacturing processes: two turning tasks, two milling tasks, painting and grinding. According to the manufacturing process of each part, the planning and the manufacturing necessitated that the inference engine group similar consecutive processes as a single task. These processes needed to be accomplished consecutively as a single task with no interfering operation. This can be seen from Figure 15, where two distinct turning and milling tasks are created by the model.

Based on the characteristics of the designed part and its operational requirements, specific manufacturing operations with the relevant technical parameters were considered. It is important to find the proper machine tools, equipment and devices capable of executing the required manufacturing process. Several critical parameters need to be identified and specified for the best possible matching of a machine tool to a specific manufacturing operation. Turning task 1 contained two turning operations to shape outer diameters of $\phi 118.2\ mm$ and $\phi 90\ mm$. To find the most appropriate machine tool to accomplish this job, several rules shown in Table 2 were applied. The first was machine tool selection based on operation type. In other words, this rule aimed to find all machine tools capable of performing the turning operation. In this case, all resources capable of accomplishing turning operations were selected.

Operation type checking rule aims to find the machine tools capable of performing the target operation-type in the task group. Here, the turning operation could be accomplished by two types of machine tools in the system.

Table 3. Applied semantic rule for selecting suitable machine tool type from available machine tools.

Available machine tools	Applied semantic rule	Machine tool type	Selected
CCMCO Trunnion Type	(veonto:MultiTaskingMachine veonto:enables	Horizontal Casting	FALSE
Deckel HM Series	veonto:TurningProcess)	Milling	FALSE
DMG Mori CTX beta 1250	(veonto:MultiTaskingMachine veonto:enables	Lathe	TRUE
DMG Mori-DMU 70 Series	veonto: MillingProcess)	Milling	FALSE
DMG Mori-NEF400	(veonto:MultiTaskingMachine veonto:enables	Lathe	TRUE
DMG Mori-NTX 2000/1500s	veonto: DrillingProcess)	Multi-tasking machine	TRUE
EMCO Concept Turn 60	(?mtm rdf:type veonto:MultiTaskingMachine)	Lathe	TRUE
EXIN EXE-04	(?w rdf:type veonto:TurningProcess)	Automatic Sintering	FALSE
GSD-WD300 C	(?y rdf:type veonto:MillingProcess)	Dual Wave Soldering	FALSE
Hall 6 H Tilt-Pour	(?z rdf:type veonto:DrillingProcess)	Tilting Machines	FALSE
JINKANG RH60	→	Commutator press	FALSE
MAIKENI MKNF	(?mtm veonto:enables ?y)		FALSE
Mazak 3D Space Gear U44	(?mtm veonto:enables ?z)	3D LASER	FALSE
Mazak Integrex i-200 ST	(?mtm veonto:enables ?w)	Multi-tasking machine	TRUE
Mazak Integrex J-200		Multi-tasking machine	TRUE
Mazak Variaxis II		Machining Center	FALSE
MMA160 IGBT mma		Welding	FALSE
Sandry EPC-A		Casting	FALSE
SOLTERRA Automatic Screw Fastening		Fastening	FALSE
TECHMIRE 44NT Multi Side		Die Casting	FALSE
Zhenhuan J-45		Low Pressure Casting	FALSE

Table 3 shows that two types of machines were selected: lathes and multi-tasking machines (or turn mill centres). These two machine types fitted the target turning operation requirement. Six machine tools of 21 were capable of the turning operation. Both operations are very similar to each other, and the part sizes were the same. Thus, it was expected that for both operations, the same list of machine tools would be selected. The chosen six turning machine tools were then checked for part size and availability matching.

All rules shown in Table 2 were applied over the six machine tools. As is clear from Table 4, some machine tools did not meet the operation requirements. Therefore, the rules eliminated these non-matching machine tools during the reasoning procedure. The final list of selected machine tools should satisfy all requirements of the target operation. Inside a task can exist multiple operations, and each operation has its own requirements. Based on these requirements, different lists of enablers can be selected. The final list of enablers for a task contains common machine tools for all operations. Therefore, if a machine tool is inappropriate for even one operation in the task, that machine tool is removed in the final list of task enablers. In this case, the list of enablers selected by OMAVE semantic rules is depicted in Figure 16. The final four machine tools chosen for the KNM1-turning task 1 satisfied all manufacturing requirements for all operations packed inside turning task 1.

The owners of the selected machine tools were nominated as potential partners for the forthcoming VE consortium. As an instance, it could be seen that Enterprise I, as owner of machine tool Mazak Integrex i-200ST, was capable of machining part KNM1:

$$\text{Part KNM1} \xrightarrow{\textit{veonto:hasOperation}} \text{Turning Task 1}$$

Table 4. Properties of machine tools.

Name	Power (kW)	Max workpiece diameter (mm)	Controller resolution (μm)	Max spindle speed (rpm)	Max. part weight (kg)	Accuracy (in.)
DMGMoriCTXbeta1250	32	340	0.1	5000	10	**<0.0001**
DMGMori-NEF400	11.5	350	0.1	4500	3	**0.0001**
DMGMori-2000/1500s NTX	22	610	0.01	5000	5	**<0.003**
EMCOConceptTurn60	1.1	60	1	4200	2	**<0.01**
MazakIntegrexi-200ST	22	658	0.2	5000	5	**<0.001**
MazakIntegrexJ-200	11	500	0.2	5000	5	**<0.001**
TurningOperation1Req.	6	120	0.5	4000	4	**<0.005**
TurningOperation2Req.	5	100	0.5	4000	3.8	**<0.005**

Figure 16. Detailed task information screen.

Turning Task 1 $\xrightarrow{veonto:isEnabledBy}$ Rough Turning Operation 1

Rough Turning Operation 1 $\xrightarrow{veonto:isOwnedBy}$ Mazak Integrex i-200ST

Mazak Integrex i-200ST $\xrightarrow{veonto:isOwnedBy}$ Enterprise I

In order to define these enterprises as nominated partners for the task, new statements must be added to the model. Following the reasoning process, an inference model was created by the reasoning engine. In this case, new symmetrical relations needed to be established between operations, tasks, enablers (or resources) and owner enterprises. Upon adding a new instance, like a machine tool, new product or part, enterprise, etc., the system automatically starts the reasoning procedure for the added item(s) and immediately establishes new relations between newly added source(s) and available resources in the system. The reasoning procedure starts after any changes in the system, and restructures the model and data. Figure 17 shows newly created relations between system resources following a reasoning procedure.

Based on part complexity, resources, enterprises and the materials library, hundreds of new statements can be added to the system after adding a new instance to the system. The OMAVE system activates the reasoning procedure after adding any new instances to it and instantly creates a new inferred model.

The inference models were created based on the OMAVE model by activating the reasoning engine to deduce new statements from existing ones, according to the predefined semantic rules in the model. One of the applications of these inference models is to find new relations between orders and suitable potential partners. After finding appropriate machine tool(s) for the target manufacturing operations, the owner(s) of these resources was selected. The chosen organisations were potential

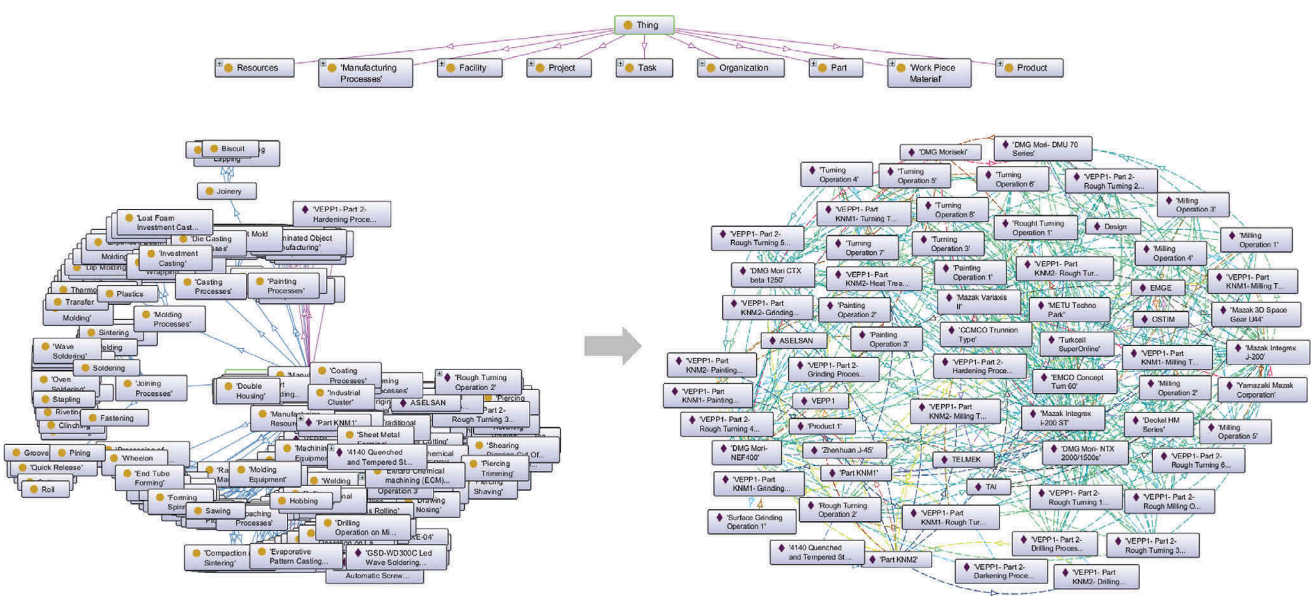

Figure 17. The main classes of the model, and the relations between instances in the OMAVE ontology model before and after reasoning.

candidate partners for the desired tasks, which were necessary for the manufacturing operation. Based on the reasoning procedure, eight enterprises were detected that owned five nominated machine tools suitable for the turning operations of KNM1-turning task 1. The nominated enterprises were displayed on the system administrator's screen (Figure 16).

As mentioned above, the partner selection process in the OMAVE system is a multi-step hybrid process. The first step is rule-based automatic partner selection that filters out partners based on defined rules in the ontology model and customer order requirements. The outcome of this step is a list of potential partners capable of entering the VE consortium to produce the ordered product for the customer. But it is necessary to further refine the selected enterprises and find the most appropriate enterprises. The result of the first step is sent to the OMAVE hybrid agent-based partner selection system for refinement. The refinement process is described in the second part of this article in detail. Here, only the final results of the partner selection procedure are shown in Table 5.

As shown in Table 5, Enterprises I and L had more than one available resource. These enterprises, despite getting more than one resource, could not perform better than second and forth places. This means that having the required resources for the nomination as a potential partner is necessary but not suffcient. To obtain better results, other parameters need to be acquired to win the negotiation process and be selected as the winner enterprise. Based on this ranking Enterprise J was announced as the winner for Turning Task 1, and joined the Part KNM1 VE consortium. This process was repeated for all task groups of Part KNM1. The members of the created VE consortium for producing Part KNM1 were the winners of the negotiation process for all task groups. The six task groups and their winner enterprises are shown in Table 6.

Based on the results shown in Table 6, the Part KNM1 consortium members were Enterprises J, I, L, K, X and Y. More detailed information regarding the negotiation procedure in the formation process is provided in the second part of this article.

Table 6. Part KNM1 task winners.

Task name	Enterprise name
Turning Task 1	Enterprise J
Turning Task 2	Enterprise I
Milling Task 1	Enterprise L
Milling Task 2	Enterprise K
Painting Task 1	Enterprise X
Grinding Task 1	Enterprise Y

Conclusion

The main obstacle to the implementation of VE platforms is the lack of flexibility in their informational structure in contrast to the highly dynamic nature of VE environments and the uncertainties in their processes. For example, different order types with completely different requirements challenge VE system capabilities and may cause it to collapse. Each order has its own characteristics and requirements. In such situations, the VE structure cannot respond to the new requirements of incoming orders. This problem is a result of ineffciency in the information management infrastructure of the VE system. Heterogeneous data and information from various types of enterprises, machine tools, manufacturing processes and products need to be handled.

Improvements in computer science and ICT technologies have yielded new opportunities to solve these problems. The development of triple data stores as non-SQL databases is one such advancement in this area. In traditional relational databases, any sudden radical change, for example, in data type results in a complete change of the database hierarchy and/or scheme. Therefore, the reusability level of these systems, especially in dynamic environments like VE systems, is insuffcient. This is one of the limitations in the enhancement of the flexibility of VE systems. However, instead of keeping data in tables and relating them through different types of relations and, consequently, creating a very complicated database scheme, the idea of keeping data and knowledge in the form of triples or quads is quite promising. In order to

Table 5. Enterprise bidding information for KNM1 rough turning task 1.

Rank	Enterprise name	Owned resource	Obtained points (0–1)
1	Enterprise J	DMG Mori NTX 2000/1500s	0.766
2	Enterprise I	Mazak Integrex J-200 DMG Mori CTX beta 1250 Mazak Integrex i-200ST	0.747
3	Enterprise F	DMG Mori CTX beta 1250	0.706
4	Enterprise L	Mazak Integrex i-200ST DMG Mori NTX 2000/1500s	0.698
5	Enterprise B	Mazak Integrex i-200ST	0.559
6	Enterprise K	Mazak Integrex J-200	0.540
7	Enterprise H	DMG Mori NTX 2000/1500s	0.490
8	Enterprise E	Mazak Integrex J-200	0.144

model a domain or system and create the associated triple data store, all domain objects, relations, properties, rules, axioms, functions, etc., should be defined. Then, with this model, it is possible to change and reconfigure the system without losing information and requiring system reconstruction.

In this study, in order to increase the flexibility and reconfigurability of VE system an ontology domain model of VE systems was created in the standard ontology languages RDF, RDFS and OWL using Protégé ontology editor. Based on the ontology model, an associated triple data store was established. This Triple Data Base (TDB) was constructed by taking advantage of the Apache Jena platform. To enable users to configure the system data and information as well as the system's hierarchy and structure, several Web services were designed to benefit from the Java Server Faces technology. Different types of business processes in various phases of the VE life cycle can be managed and realised by these Web services. These Web services use SPARQL to retrieve and manipulate data and continuously update the VE model.

The system developed and coded based on this model benefited from the Jena and the JADE platforms. Both these platforms are free and open-source Java frameworks. Jena targets building semantic Web and linked data applications, whereas JADE provides middleware compiled by FIPA specifications for ease in the implementation of multi-agent systems. These specifications, in addition to the TDB database structure of the OMAVE system, make this system highly flexible and portable for implementation in different industrial disciplines, and enable it to be immediately reconfigured to respond to customer and system admin requirements.

The OMAVE system was tested and verified by producing a sample product using the OMAVE platform and the tools with the contributions of three enterprises from the OSTIM-organised Industrial Zone in Ankara. In the case study, these three enterprises collaborated on the CATIA V6 collaborative design platform of the OMAVE system to accomplish the detailed engineering design of the product, and developed product process plans. In the partner selection process, these enterprises were included in the auctioning process, and won in different task groups. The subsequent manufacturing process was started. Each enterprise produced its associated components, and ordered product was assembled in one of the companies, and the production process was finalised. The proposed ontology model and information management system were tested and verified successfully in this case study.

Researchers aim to develop a highly flexible information management system for virtual enterprise platforms in the OMAVE project. To realise this feature in OMAVE systems, a large number of interfaces are needed to facilitate the transfer of heterogeneous data and information among different system parts. This, along with the increasing volume of data stored, lead to a substantial decline in system performance. To deal with this problem, a new hybrid approach for a system database hierarchy is needed. This enhancement in the system is considered part of future research. A new algorithm also is being developed for data read/write and analysis. The performance of this novel system is expected to be much better than the OMAVE system proposed here.

Acknowledgements

The authors are grateful for the continuous support of the management of the OSTIM Industrial Park and OSTIM Technology A.S. staff.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This study has been funded by SAN-TEZ project No. 00979.stz.2011-12 of the Turkish Ministry of Science, Technology, and Industry.

Notes

1. The Integration Definition for Function Modeling (IDEF0) method covers the field of operation of most enterprises, and empower analysers and system engineers to develop system models at any level of detail with simple graphical representation tools (IDEF0 1993).
2. The Unified Modeling Language (UML) is a visual modelling language to develop software and systems (Fowler 2003).
3. ENOVIA Web site: http://www.3ds.com/products-services/enovia/.
4. SWRL: http://www.w3.org/Submission/SWRL/.
5. Jena: https://jena.apache.org/.
6. SPARQL: http://www.w3.org/TR/sparql11-query/.
7. JSF: https://javaserverfaces.java.net/.

References

Alzaga, A., and J. Martin. 1999. "A Design Process Model to Support Concurrent Project Development in Networks of SMEs." In *Infrastructures for Virtual Enterprises. Vol. 27 of IFIP The International Federation for Information Processing*, edited by L. Camarinha-Matos and H. Afsarmanesh, 307–318. Boston, MA: Springer.

Browne, J., and J. Zhang. 1999. "Extended and Virtual Enterprises Similarities and Differences." *International Journal of Agile Management Systems* 1 (1): 30–36. doi:10.1108/14654659910266691.

Byrne, J. A., R. Brandt, and O. Port. 1993. "The Virtual Corporation." *Bloomberg Businessweek Archives*. http://www.businessweek.com/stories/1993¬02-07/the-virtual-corporation/.

Camarinha-Matos, L., and H. Afsarmanesh. 2001. "Virtual Enterprise Modeling and Support Infrastructures: Applying Multi-agent System Approaches." In *Multi-Agent Systems and Applications. Vol. 2086 of Lecture Notes in Computer*

Science, edited by M. Luck, V. Mařík, O. Štepankova, and R. Trappl, 335–364. Berlin, Heidelberg: Springer.

Camarinha-Matos, L., H. Afsarmanesh, C. Garita, and C. Lima. 1998. "Towards an Architecture for Virtual Enterprises." *Journal of Intelligent Manufacturing* 9 (2): 189–199. doi:10.1023/A:1008880215595.

Camarinha-Matos, L. M., and H. Afsarmanesh. 2007. "A Framework for Virtual Organization Creation in A Breeding Environment." *Annual Reviews in Control* 31 (1): 119–135. doi:10.1016/j.arcontrol.2007.03.006.

Candido, G., and J. Barata. 2007. "A Multiagent Control System for Shop Floor Assembly." In *Holonic and Multi-Agent Systems for Manufacturing. Vol. 4659 of Lecture Notes in Computer Science*, edited by V. Mařík, V. Vyatkin, and A. W. Colombo, 293–302. Berlin, Heidelberg: Springer.

Chandrasekaran, B., J. R. Josephson, and V. R. Benjamins. 1998. "The Ontology of Tasks and Methods." [Accessed December 16 2013]. http://Web.cse.ohio¬state.edu/chandra/Ontology-of-Tasks-Methods.PDF

Cranefield, S., S. Haustein, and M. Purvis. 2001. "UML-Based Ontology Modeling for Software Agents." The Information Science Discussion Paper Series, 2001 (07).

Dogdu, E., S. Hakimov, and S. Yumusak. 2014. "A Data-Model Driven Web Application Development Framework." Proceedings of the 2014 ACM Southeast Regional Conference (ACM SE '14). Kennesaw, GA, Article 47, 1–4.

Fowler, M. 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Boston, MA: Addison-Wesley Longman.

Frenkel, A., H. Afsarmanesh, C. Garita, L. O. Hetzberger. 2001. "Supporting Information Access Rights and Visibility Levels in Virtual Enterprises." In *E-Business and Virtual Enterprises. Vol. 56 of IFIP — The International Federation for Information Processing*, edited by L. Camarinha-Matos, H. Afsarmanesh, R. Rabelo, et al., 177–192. Boston, MA: Springer.

Garita C., H. Afsarmanesh, and L. O. Hertzberger. 2001. "The PRODNET Cooperative Information Management for Industrial Virtual Enterprises." *Journal of Intelligent Manufacturing* 12 (2): 151–170.

Gruber, T. R. 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing?." *International Journal of Human-Computer Studies* 43 (5–6): 907–928. doi:10.1006/ijhc.1995.1081.

Heflin, J. 2004. "OWL Web Ontology Language." [Accessed February 1 2013]. http://www.w3.org/TR/owl-features/

Hirsch, B. E., T. Kuhlmann, Z. K. Marciniak, and C. Maβow. 1995. "Information System Concept for the Management of Distributed Production." *Computers in Industry* 26 (3): 229–241. Computer Integrated Manufacturing and Industrial Automation. doi:10.1016/0166-3615(95)00009-S.

Horrocks, I., P. F. Patel-Schnider, H. Boley, S. Tabet, B. Grosof, and M. Dean. 2004. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." Accessed June 12, 2014. http://www.w3.org/Submission/SWRL/

Hsieh, F.-S., and J.-B. Lin. 2014. "Context-Aware Workflow Management for Virtual Enterprises Based on Coordination of Agents." *Journal of Intelligent Manufacturing* 25 (3): 393–412. doi:10.1007/s10845-012-0688-8.

Huang, X., Y. Wong, and J. Wang. 2004. "A Two-Stage Manufacturing Partner Selection Framework for Virtual Enterprises." *International Journal of Computer Integrated Manufacturing* 17 (4): 294–304. doi:10.1080/09511920310001654292.

IDEF0, INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0). 1993. *Announcing the Standard for 183*. East College Station, TX: Draft Federal Information Processing Standards Publication, University Drive.

Kim, C., K. Kim, J. Lee, D. Kang, and K. Ryu. 2013. "Ontology-Based Process Model for Business Architecture of a Virtual Enterprise." *International Journal of Computer Integrated Manufacturing* 26 (7): 583–595. doi:10.1080/0951192X.2012.749529.

Klen A. A. P. 2001. "Managing Distributed Business Processes in the Virtual Enterprise." *Journal of Intelligent Manufacturing* 12 (2): 185–197.

Klen, A. A. P., R. J. Rabelo, L.M. Spinosa, and A. C. Ferreira. 1998. "Integrated Logistics in the Virtual Enterprise: The PRODNET-II Approach." Proceedings of 5th IFAC Workshop on Intelligent Manufacturing Systems (IMS'98), Gramado, November 9–11.

Moisescu, M. A., and I. S. Sacala. 2014. "Towards the Development of Interoperable Sensing Systems for the Future Enterprise." *Journal of Intelligent Manufacturing* 27 (1): 33–54.

Mun, J., M. Shin, and M. Jung. 2011. "A Goal-Oriented Trust Model for Virtual Organization Creation." *Journal of Intelligent Manufacturing* 22 (3): 345–354.

Mun, J., M. Shin, K. Lee, and M. Jung. 2009. "Manufacturing Enterprise Collaboration Based on a Goal-Oriented Fuzzy Trust Evaluation Model in a Virtual Enterprise." *Computers & Industrial Engineering* 56 (3): 888–901. Intelligent Manufacturing and Logistics. doi:10.1016/j.cie.2008.09.022.

Noy, N. F., and D. L. McGuinness. 2001. "Ontology Development 101: A Guide to Creating Your First Ontology." Accessed June 21, 2013. http://protege.stanford.edu/publications/ontologydevelopment/ontology101.pdf

Osório, A. L., and M. M. Barata. 2001. "Reliable and Secure Communications Infrastructure for Virtual Enterprises." *Journal of Intelligent Manufacturing* 12 (2): 171–183.

Park, K. H., and J. Favrel. 1999. "Virtual Enterprise-Information System and Networking Solution." *Computers & Industrial Engineering* 37 (1–2): 441–444. Proceedings of the 24th international conference on computers and industrial engineering. doi:10.1016/S0360-8352(99)00113-8.

Petersen, S. A., M. Divitini, and M. Matskin. 2001. "An Agent-Based Approach to Modelling Virtual Enterprises." *Production Planning & Control* 12 (3): 224–233. doi:10.1080/095372801300107707.

Rabelo, R., L. Camarinha-Matos, and H. Afsarmanesh. 1998. "Multiagent Perspectives to Agile Scheduling." In *Intelligent Systems for Manufacturing. Vol. 1 of IFIP | The International Federation for Information Processing*, edited by L. Camarinha-Matos, H. Afsarmanesh, and V. Marik, 51–66. New York: Springer.

Rabelo, R. J., L. M. Camarinha-Matos, and R. V. Vallejos. 2000. "Agent-based Brokerage for Virtual Enterprise Creation in the Moulds Industry." In *Proceedings of the IFIP TC5/WG5.3 Second IFIP Working Conference on Infrastructures for Virtual Organizations: Managing Cooperation in Virtual Organizations and Electronic Business towards Smart Organizations: E-Business and Virtual Enterprises: Managing Business to-Business Cooperation*, edited by Luis M. Camarinha-Matos, H. Afsarmanesh, and R. J. Rabelo, 281–290. Deventer: Kluwer, B.V.

Rush, J. 2010. "IBM I and SaaS Positioning IBM I as a Software as a Service Platform." [Accessed January 1 2013]. https://www-304.ibm.com/partnerworld/wps/servlet/ContentHandler?contentId=XhrjdgScEXMiPCA$cnt&roadMapI

Sadigh, B. L., F. Arikan, A. M. Ozbayoglu, H. O. Unver, and S. E. Kilic. 2014. "A Multi-agent System Model for Partner Selection Process in Virtual Enterprise." *Procedia Computer Science* 36 (0): 367–372. Complex Adaptive Systems Philadelphia, PA, November 3–5, 2014. doi:10.1016/j.procs.2014.09.007.

Sadigh, B. L., H. O. Unver, E. Dogdu, and S. E. Kilic. 2012. "An Ontology Based Model for Virtual Enterprise." In *Proceedings of 15th International Conference on Machine Design and Production, UMTIK 2012, Denizli, Turkey, June 19–22*, 885–898. Ankara: METU-MATIMAREN.

Sadigh, B. L., H. Unver, and S. E. Kilic. 2012. "Design of a Multi Agent Based Virtual Enterprise Framework for Sustainable Production." In *Virtual and Networked Organizations, Emergent Technologies and Tools. Vol. 248 of Communications in Computer and Information Science*, edited by G. Putnik and M. Cruz-Cunha, 186–195. Berlin, Heidelberg: Springer.

Sandakly, F., J. Garcia, P. Ferreira, and P. Poyet. 1999. "PerDiS: An Infrastructure for Cooperative Engineering in Virtual Enterprise." In *Infrastructures for Virtual Enterprises. Vol. 27 of IFIP — The International Federation for Information Processing*, edited by L. Camarinha-Matos and H. Afsarmanesh, et al., 319–332. Boston, MA: Springer.

Sandakly, F., J. Garcia, P. Ferreira, and P. Poyet. 2001. "Distributed Shared Memory Infrastructure for Virtual Enterprise in Building and Construction." *Journal of Intelligent Manufacturing* 12 (2): 199–212.

Sari, B. 2006. "Methodology Development for Small and Medium Sized Enterprise (SME) Based Virtual Enterprises." PhD thesis, Middle East Technical University, METU, Mech. Eng. Dept.

Sari, B., T. Sen, and S. Kilic. 2007. "Formation of Dynamic Virtual Enterprises and Enterprise Networks." *The International Journal of Advanced Manufacturing Technology* 34 (11–12): 1246–1262.

Sari, B., T. Sen, and S. E. Kilic. 2008. "AHP Model for the Selection of Partner Companies in Virtual Enterprises." *The International Journal of Advanced Manufacturing Technology* 38 (3–4): 367–376. doi:10.1007/s00170-007-1097-6.

SIIA. 2006. "Software as a Service; A Comprehensive Look at the Total Cost of Ownership of Software Applications." [Accessed June 21 2013]. http://www.winnou.com/saas.pdf

Siqueira, J., and C. Bremer. 2001. "Action Research: The Formation of a Manufacturing Virtual Industry Cluster." In *E-Business and Virtual Enterprises. Vol. 56 of IFIP — The International Federation for Information Processing*, edited by L. Camarinha-Matos, H. Afsarmanesh, and R. Rabelo, 261–268. Boston, MA: Springer.

Soares, A. L., A. L. Azevedo, and J. P. De Sousa. 2000. "Distributed Planning and Control Systems for the Virtual Enterprise: Organizational Requirements and Development Life-Cycle." *Journal of Intelligent Manufacturing* 11 (3): 253–270. doi:10.1023/A:1008967209167.

Spyns, P., R. Meersman, and M. Jarrar. 2002. "Data Modelling versus Ontology Engineering." *SIGMOD Rec* 31 (4): 12–17. doi:10.1145/637411.

Unver, H. O., and B. L. Sadigh. 2013. "An Agent-Based Operational Virtual Enterprise Framework Enabled by RFID." In *Small and Medium Enterprises: Concepts, Methodologies, Tools, and Applications*, edited by M. M. Cruz-Cunha and F. Moreira, 198–215. Hershey, PA: Business Science Reference.

Vanderhaeghen, D., and P. Loos. 2006. *Distributed Model Management Platform for Virtual Enterprise Networks*. Network-Centric Collaboration and Supporting Frameworks, Vol. 224 of IFIP — The International Federation for Information Processing, 507–514. Boston, MA: Springer.

Vanderhaeghen, D., and P. Loos. 2007. "Distributed Model Management Platform for Cross-Enterprise Business Process Management in Virtual Enterprise Networks." *Journal of Intelligent Manufacturing* 18 (5): 553–559.

Vallejos, R. V., C. P. Lima, G. Varvakis. 2007. "Towards the Development of a Framework to Create a Virtual Organization Breeding Environment in the Mould and Die Sector." *Journal of Intelligent Manufacturing* 18 (5): 587–597.

Vargas, A., A. Boza, and L. Cuenca. 2011. "Towards Interoperability through Inter-enterprise Collaboration Architectures." In *Proceedings of Confederated International Conference on the Move to Meaningful Internet Systems*, OTM'11 Crete, Greece, edited by R. Meersman, T. Dillon, and P. Herrero, 102–111. Berlin: Springer.

Vargas, A., L. Cuenca, A. Boza, I. Sacala, M. Moisescu. 2014. "Towards the Development of the Framework for Inter Sensing Enterprise Architecture." *Journal of Intelligent Manufacturing* 27 (1): 55–72.

Wang, X., T. Wong, and G. Wang. 2010. "Service-Oriented Architecture for Ontologies Supporting Multi-Agent System Negotiations in Virtual Enterprise." *Journal of Intelligent Manufacturing* 23 (4): 1331–1349. doi:10.1007/s10845-010-0469-1.

Ye, F., and Y. Li. 2005. "Multi-Attribute Decision-Making Model for Virtual Enterprise Partners Selection under Uncertain Information Condition." Proceedings of ICSSSM '05. 2005 International Conference on Services Systems and Services Management, Vol. 2, June 13–15, 1162–1165. IEEE. doi:10.1109/ICSSSM.2005.1500180.

Ye, N. 2002. "Information Infrastructure of Engineering Collaboration in a Distributed Virtual Enterprise." *International Journal of Computer Integrated Manufacturing* 15 (3): 265–273. doi:10.1080/09511920110059098.

Zhang, W., and J. Yin. 2008. "Exploring Semantic Web Technologies for Ontology-Based Modeling in Collaborative Engineering Design." *The International Journal of Advanced Manufacturing Technology* 36 (9–10): 833–843. doi:10.1007/s00170-006-0896-5.

Zhou, L., and R. Nagi. 2002. "Design of Distributed Information Systems for Agile Manufacturing Virtual Enterprises Using CORBA and STEP Standards." *Journal of Manufacturing Systems* 21 (1): 14–31. doi:10.1016/S0278-6125(02)90009-9.

Zhou, Q., P. Souben, and C. B. Besant, 1998. "An Information Management System for Production Planning in Virtual Enterprises." *Computers & Industrial Engineering* 35 (1–2): 153–156.