



**HAL**  
open science

# Combining multi-agent model, SOA and ontologies in a distributed and interoperable architecture to manage multi-site production projects

Bernard Archimède, Muhammad Ali Memon, Karim Ishak

## ► To cite this version:

Bernard Archimède, Muhammad Ali Memon, Karim Ishak. Combining multi-agent model, SOA and ontologies in a distributed and interoperable architecture to manage multi-site production projects. *International Journal of Computer Integrated Manufacturing*, 2016, 30 (8), pp.856-870. 10.1080/0951192X.2016.1224389 . hal-01945210

**HAL Id: hal-01945210**

**<https://hal.science/hal-01945210>**

Submitted on 5 Dec 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <http://oatao.univ-toulouse.fr/20084>

**To cite this version:**

Archimède, Bernard  and Memon, Muhammad Ali  and Ishak, Karim *Combining multi-agent model, SOA and ontologies in a distributed and interoperable architecture to manage multi-site production projects.* (2016) *International Journal of Computer Integrated Manufacturing*, 30 (8). 856-870. ISSN 0951-192X

Any correspondence concerning this service should be sent to the repository administrator: [tech-oatao@listes-diff.inp-toulouse.fr](mailto:tech-oatao@listes-diff.inp-toulouse.fr)

# Combining multi-agent model, SOA and ontologies in a distributed and interoperable architecture to manage multi-site production projects

Bernard Archimède<sup>a\*</sup>, Muhammad Ali Memon<sup>a</sup> and Karim Ishak<sup>b</sup>

<sup>a</sup>*School of Information Technology, INP-ENIT, University of Toulouse, Tarbes Cedex, France;* <sup>b</sup>*Section II (Achrafieh) Department of Business Computing, Faculty of Economic Science and Business Administration, Beirut, Lebanon*

Today, production projects are often achieved by multi-site production and the associated markets become more and more dynamic and service oriented. In this context, the multi-site project management is a difficult task, which has to take into account not only the decision distribution and the information confidentiality, but also the heterogeneity existing between different production management systems used by partners. The generic planning model SCEP (supervisor, customer, environment, producer) based on multi-agent technology offers a well-adapted solution to the decision distribution and information confidentiality in project management. SCEP was integrated with ontological mechanisms and SOA (service-oriented architecture) components in a new architecture Semantic-SCEPSOA, which makes it possible to achieve a distributed and interoperable management of multi-site production projects. In the case of two sites managed by two different planning software applications, the semantic interoperability effectiveness is shown within the S-SCEPSOA architecture.

**Keywords:** multi-site production projects; service-oriented architecture; distributed production management; interoperability; multi-agent system; ontology

## 1. Introduction

To ensure a good competitiveness, companies turn very often to multi-site production, centred on their core competencies and sharing their low value-added activities with other companies. In this collaborative context, companies create electronic marketplaces (e-marketplaces) based on Internet and new Information and Communication Technologies, thus enhancing their competitiveness and their business relations with partners (Park and Yang 2006). Actually, big companies or original equipment manufacturers (OEMs) creating their own markets for manufacturing their products dominate these electronic production markets. In these markets, OEMs often require strong constraints that penalise a large number of small and medium enterprises (SME) having limited capacities and means (Abidi, Su, and Mohiuddin 2011). A new type of dynamic e-marketplaces appeared under the name of service-oriented market to increase the presence of SMEs in these markets (Li et al. 2006). Service-oriented markets put in contact several customers having production projects to realise with several producers or providers of services allowing the realisation of customer's projects (Ghenniwa and Shen 2004). They reduce the strong requirements of the OEM and puts SME in the foreground by describing their skills and know-how in terms of services. Also, these markets federate the know-how of

SMEs in a shared environment, thus allowing companies to seek for service providers corresponding to their needs.

The dynamic aspect of collaborations in a service-oriented market highlights the necessity of a distributed management of multi-site projects that takes into account the heterogeneity between the various systems and applications used by the partners. The heterogeneity of technologies and functions deployed in management systems concerns technical models, data types, platform, etc., whilst semantic and structural heterogeneities that can exist between concepts are caused by differences between meta-models of exchanged data or more particularly structures, representations and terminologies of information exchanged between the partners.

These arguments are in favour of a generic service-oriented architecture, which allows a distributed and interoperable management of multi-site production projects, by cooperation between various production management systems. Interoperability is the possibility for systems or components to exchange information and to be able to use exchanged information (Geraci 1991). The proposed architecture is based on a generic planning multi-agent system SCEP (supervisor, customer, environment, producer) offering a distributed management and integrating interoperability mechanisms based on SOA (service-oriented architecture) allowing a platform- and technology-independent communication (Erl et al. 2014).

\*Corresponding author. Email: [bernard.archimede@enit.fr](mailto:bernard.archimede@enit.fr)

Present address of Muhammad Ali Memon is Institute of Information and communication technology, University of Sindh, Sindh, Pakistan

Ontology mechanisms are also integrated to ensure the good understanding of the information exchanged between heterogeneous systems (Gruber 1993), thus enhancing their communication and cooperation.

In Section 2, difficulties for managing multi-site projects in a service-oriented market are discussed. Section 3 summarises some existing approaches to manage multi-site production projects, focusing on multi-agent systems and particularly on SCEP model and its limits in dynamic collaborations. The proposed architecture is presented in Section 4 and its functioning detailed in Section 5. In Section 6, an application to two heterogeneous planning systems shows the effectiveness of the semantic interoperability strategy in collaboration during a distributed planning of multi-site projects (Taghipour and Frayret 2013). Finally, Section 7 collects the points of interest of the proposed architecture and presents current and future works.

## 2. Literature review

Today, more than ever enterprises need to strengthen their assets in order to evolve according to new company strategies for coping the raising environmental and economic challenges. Especially for those who have to constantly change their products for sustaining in the market. To overcome this problem, companies and organisations joined forces, reusing each other's resources and capabilities organised in a collaborative Network (Papakostas et al. 2013). Collaborative networks such as DMN (dynamic manufacturing networks) are formed by several companies in order to achieve the goal set for the project for which the network is formed (Papakostas et al. forthcoming).

A project is offered by customer (company), while producers, providing services achieving activities of these projects. A producer involved in a customer project can be considered in its turn, as a customer by other producers, which are concerned by the realisation of its project's activities. The so-formed companies' network is temporary with the same lifetime as this of the project. A producer can join or exit the created network at any time, illustrating the dynamic aspect of collaboration and yield the existence of a virtual supply chain. Collaboration or integration of a supply chain is a well-studied area. Several approaches and frameworks are proposed in the literature. For example, Dudek and Stadtler (2005), Schneeweiss and Zimmer (2004) and Taghipour and Frayret (2012) propose planning of supply chain collaboration, but are limited to two partners. There are frameworks proposed for supply chain integration. CXML (commerce XML), OAGIS (open applications group integration specification) and xCBL (XML common business library) are included because they are pioneers in XML-based supply-chain integration. BPML (business process

modelling language), ebXML (electronic business XML) and XPDL (XML process description language), in turn, provide a new understandings of supply chain integration.

cXML, OAGIS and xCBL are cross-industry frameworks providing vocabularies but are limited to the rough process approach. BPML, ebXML and XPDL are process-centric frameworks. They provide no vocabularies and focus only on business processes generic process approach. There are also content-based languages for business communication like. DAML+OIL is an RDF description logic-based language for expressing ontologies. FIPA-SL is used in conjunction with FIPA-ACL. KIF is a knowledge representation language is used as interchange format between knowledge systems. Prolog is also a knowledge representation language. From them, ebXML seems promising, but as it is not a traditional knowledge representation framework, hence it lacks strong basis in predicate logics (Nurmi and Kotinurmi 2004). ebXML also has not delivered core Components for business documents (Botelho et al. 2002).

In such kind of distributed and dynamic environment, the complexity of multi-site projects management is coming from the presence of various production systems cooperating without any prior knowledge and without any pre-imposed constraints. The management difficulties result mainly from characteristics of these systems relative to physical distribution and to technical and semantic heterogeneities (Izza 2006). For a production project, several decisions such as choosing and using production resources, relaxing constraints, etc. have to be distributed between the involved partners. The semantic heterogeneity results from the difference between structures, representations and terminologies of information exchanged between the partners. An illustration of the structural heterogeneity is the generalisation/specialisation, the absence of some concepts, etc. The representational heterogeneity deals with the difference between the representations of the same information, i.e. due date as 01/12/2015 or 2015-12-01, etc., whilst the terminological heterogeneity is relative to the use of same terminology for representing different concepts or the use of different terms to represent the same concept. Some projects to provide a shared business vocabulary are developed for business like Toronto Virtual Enterprise (TOVE) project provided a generic, reusable data model that provides a shared terminology for the enterprise that each agent can jointly understand and use (Schlenoff, Ivester, and Knutilla 1998). TOVE does not touch the domain of integration and how integration is done is unclear (Roy and Hafner 1997). TOVE divides its domain (enterprise modelling) into a number of different sub ontologies. Even within these smaller ontologies in TOVE, no overall taxonomies exist. Its taxonomies seem to be local, each going very few levels deep (Roy and Hafner 1997). Another limitation of TOVE project is inability to find actual ontologies

and the last update of that site made on 18 February 2002, resulted in utilisation of only its theoretical part (Szymczak et al. 2008).

Similar to TOVE, the enterprise ontology project's goal is to provide 'a collection of terms and definitions relevant to business enterprises to enable coping with a fast changing environment through improved business planning, greater flexibility, more effective communication and integration' (Uschold et al. 1998).

There are some commercial solution for heterogeneous multi-site productions like ERP Enterprise Resource Planning, APS Advanced Planning and Scheduling, etc. take into account the finite capacity of the various sites during the planning, to simulate several planning scenarios and to manage several production sites (Telle 2003). Solutions given by these systems centralise the decision, as well as they are deterministic and not robust because the dynamic aspect of markets and the variation of the physical process (time range, variability of the suppliers, etc.) are not taken into account (Genin 2005).

Multi-agent systems (MAS) (Ferber 1999) overcome these drawbacks, by offering a simple framework to model the various components of a production system as well as a natural distribution of the decision. A project management based on the local behaviour of the agents facilitates the ability to react, and guarantees the autonomy of each entity and the flexibility of the enterprises network involved in a multi-site project (Jiao, You, and Kumar 2006).

In the literature, several models based on MAS were proposed in various production contexts, i.e. planning (Nishioka 2004; Lima, Sousa, and Martins 2006; Wang, Cheng, and Lin 2013), management and coordination of virtual enterprise and supply chain context (Monteiro, Roy, and Anciaux 2007; Kovalchuk 2009), etc.

Therefore, it can be deduced from above literature review that a distributed model is required to manage multi-site production projects for overcoming these difficulties and to take into account the dynamic aspect of the service-oriented markets, thus dynamic partners discovery mechanisms are of great interest to put in relation customers and producers, as well as to overcome the heterogeneity dimension of the partners' applications to ensure good collaboration between them.

The generic planning multi-agent model SCEP Archimede and Coudert (2001) proposes a distributed process that allows cooperation via a shared environment (blackboard) between customer agents representing projects and producer agents representing resources, under the control of a supervisor agent. The solution is derived by a negotiation process between agents based on a competition between customer agents. Customer agents may reject the propositions made by producer agents if they consider that the proposition can be improved. The fact that the tasks of all projects are taken into account at the

same time is another interesting property of this model, and differentiates it from models based on the Contract-Net protocol (Smith 1980; Boukreda, Maamri, and Aknine 2012). These characteristics give the SCEP model a better forecasting horizon (so-called visibility) to satisfy the customer objectives and those of the production system.

For a better management of multi-site activities, the SCEP model was enriched by two kinds of ambassador agents – supervisor ambassador agent and producer ambassador agent – as illustrated in Figure 1. These agents achieve connections between several SCEP models. The supervisor ambassador agent (resp. producer ambassador agent) exchanges information with other SCEP models, and more precisely with producer ambassador agents (resp. supervisor ambassador agent) of these models. For example, the management framework of a multi-site project involving two remote production sites consists of four SCEP models. One of them represents the customer offering the project. Two other SCEP models represent the two remote production sites. The fourth SCEP model is dedicated to the inter-site transport management. In Figure 1, regular agents represent local resources.

In a service-oriented market, production systems and applications of the partners involved in a multi-site project must be able to communicate and cooperate without modifying significantly the local process of every partner. Therefore, in this context, a better management of multi-site projects supposes dynamic identification of the partners and interoperability mechanisms to deal with the dynamic aspect of collaboration and technical and semantic heterogeneity (Corella, Rosalen, and Simarro 2013). Although the use of multiple SCEP models offers a distributed way to manage multi-site projects, it presents some limits for its deployment in a service-oriented market because its weak ability to communicate and to cooperate with heterogeneous models and management systems, as well as its difficulty to localise new partners' applications.

### 3. S-SCEPSOA meta-model

Before presenting the proposed architecture of Semantic-SCEPSOA(S-SCEPSOA), first the S-SCEPSOA meta-model is presented, which constitutes the integration of key concepts of SCEP (in bold italic) and SOA (in italic) as well as their relations.

*Function*: A 'function' represents a management function assured by an 'application'. It can be for example planning, scheduling, management of purchases and sales, etc.

*Activity*: An 'activity' represents some know-how in one of the big stages of the product life cycle. For

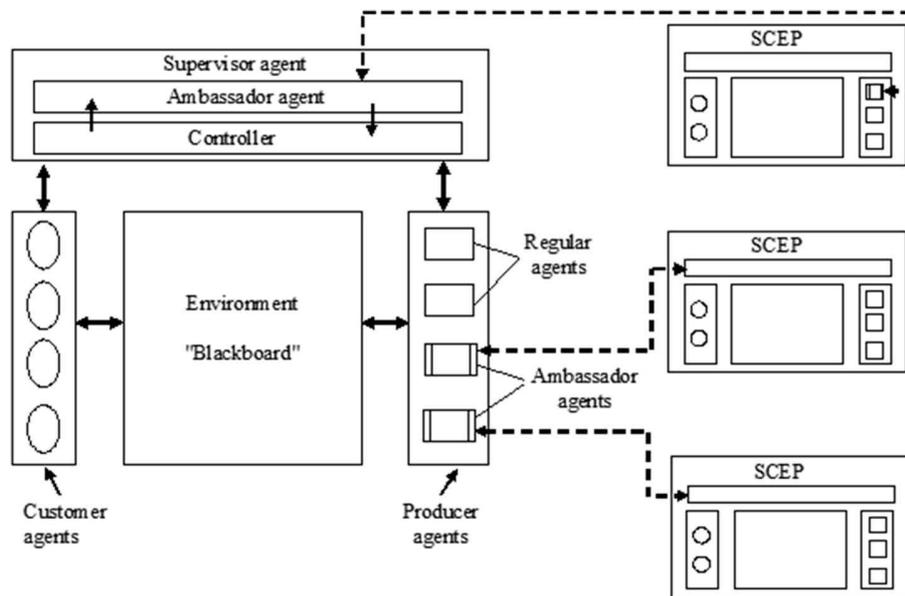


Figure 1. Conceptual framework of a networked SCEP models.

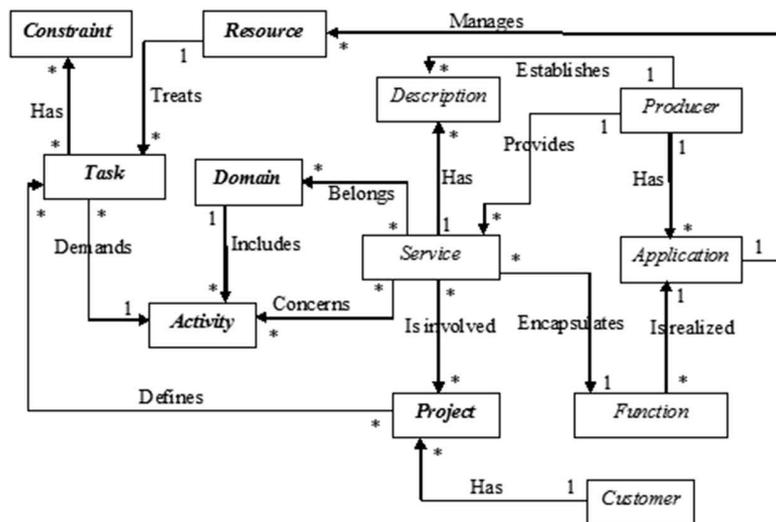


Figure 2. Conceptual meta-model of S-SCEPSOA.

example, in the stage of manufacturing, an activity can be drilling, milling, etc.

**Resource:** A ‘resource’ represents the physical tool managed by an ‘application’ and which allows the realisation of one or several ‘tasks’.

**Domain:** A ‘domain’ collects all the activities associated with a defined stage.

**Task:** A ‘task’ concerns the realisation of an ‘activity’ on a ‘resource’ for definite duration. It possesses a set of constraints, i.e. temporal constraints as a start date, end date, etc. Note that several tasks can be defined for the same activity.

**Application:** An ‘application’ represents a production management tool which can have one or several ‘functions’. An application can be for example an ERP, an APS, etc.

**Service:** A ‘service’ encapsulates a precise ‘function’ assured by an ‘application’. It includes one or several ‘activities’ and belongs to one or several ‘domains’. It can be involved in one or several projects. A service possesses one or several ‘descriptions’. It is supplied by a single ‘producer’. However, services supplied by different producers can have the same ‘function’ and concern the same activities.

*Producer:* A ‘producer’ has an ‘application’, which provides one or several ‘services’ encapsulating each one a ‘function’ assured by the ‘application’.

*Project:* A ‘project’ defines a set of ‘tasks’ for one or several ‘activities’ supplied by one or several ‘services’, the realisation of which allows to have a finished product. A project is considered feasible if all the required activities are supplied by available services. A service ‘description’ is made by the ‘producer’ and contains functional (what service does) and non-functional (how service is supplied) properties. Functional properties (resp. non-functional properties) can be for example functions, type of parameters, etc. (resp. quality of service QoS, etc.).

*Ontology:* Ontology is an explicit specification of a conceptualisation (Gruber 1993), and makes it possible to share the arrangement of the structure of information between software agents and people (Corcho, Fernandez-Lopez, and Gomez-Perez 2006). It facilitates the reuse of knowledge for systems operating in a similar domain.

#### 4. S-SCEPSOA architecture

The proposed architecture Semantic-SCEPSOA (S-SCEPSOA) encapsulates the SCEP model of Archimede and Coudert (2001) within the SOA framework (Erl et al. 2014). This architecture combines three kinds of technologies. First, SOA in order to facilitate the identification and forming a relation of actors on the internet. Second, ontology technology is used in order to facilitate the understandings for the exchange of information. Third, SCEP multi-agent model is used to elaborate the planning of project. SOA allows technical and operational interoperability by gathering functionalities of enterprise applications in a loosely coupled and technology independent manner (technical model, platform, etc.), offering mechanisms of dynamic identification of partners. S-SCEPSOA also ensures distributed and confidential framework for the management of multi-site projects inherited from SCEP model, as well as the technological interoperability and mechanisms of dynamic collaboration for the cooperating of the various production applications used by the partners inherited from SOA model. The S-SCEPSOA meta-model depicts in Figure 2 the integration of key concepts of SCEP and SOA as well as their relations.

The semantic interoperability strategy, necessary to ensure the good understanding and interpretation of the exchanged information between heterogeneous systems is based on ontologies (Guarino 1997), (Maedche 2012).

More precisely, the semantic interoperability strategy adopted in S-SCEPSOA is based on the ontological hybrid approach (Noy 2009), which is well adapted to the dynamic context, because it takes advantage from mono-ontology and multi-ontology approaches by combining the use of a common ontology as background ontology (Aleksovski et al. 2006) to establish mappings (De

Bruijn et al. 2006) between ontologies. In this strategy, each partner keeps its own local ontology and establishes bidirectional mappings with a global and common ontology serving as Interlingua between the various partners production systems. Reasoning mechanisms are needed by customers and producers to transform data described according to the global ontology in data expressed according to the local ontology and vice versa.

S-SCEPSOA is organised around three kinds of actors: the SOAregister containing information about the provided services, the SOAproducer proposing the service and the SOAcustomer consuming the discovered services by invoking them at the corresponding SOAproducers. Interaction between the different actors is achieved via messages. The structure of these actors is given in Figure 3, which shows technical, and semantic, components set up at each actor.

##### 4.1. SOAregister

The SOAregister contains an interface of communication and acts as a service broker by discovering the services leading to the realisation of SOAcustomer projects, and declaring the know-how of SOAproducers in terms of SOA services. The discovery service looks for services as well as their SOAproducers. The publication service publishes the description of SOAproducers services. These two services communicate with a Register DataBase (RDB) containing information on the supplied services and their SOAproducers. In this database, services are stored by domains. Domains are defined and implemented by the administrator of the SOAregister. OntoBase represents the global and common ontology used as reference ontology in the exchange of information between production systems and applications. This ontology is set up at the SOAregister and is imported in the mapping module of SOAcustomer and SOAproducers during the service publication and discovery phases.

##### 4.2. SOAproducer

The SOAproducer has its own production application to manage the activity achievement on the production resources. The exposition of the activities to the public is offered by services, which implement the production functions assured by the production application. These services are described in the Producer DataBase (PDB) storing the description of the services, the information concerning the description of the SOAcustomer’s projects according to the supplied activities, and the results of management for these projects established by the production application, i.e. a scheduling plan in the case of planning services. A description and publication agent interacting with the SOAregister for the publication task realises the description of services and its publication.

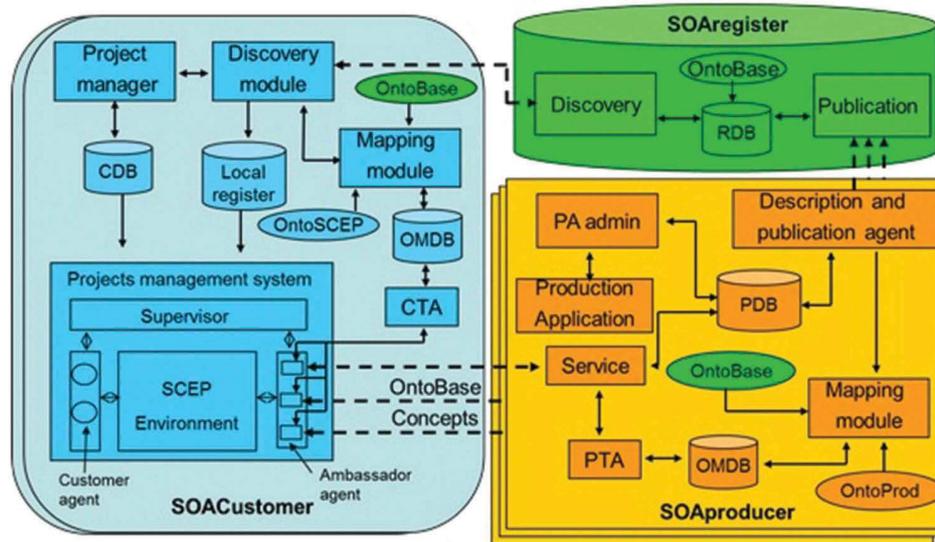


Figure 3. S-SCEPSOA architectural framework.

The administrator of the Production Application (PA Admin) interacts with the producer database PDB to get the projects description received from the SOAcustomer in order to process them. It stores also in the producer database PDB the results established by the production application for these projects. Each SOAproducer has its own ontology noted OntoProd. Mappings between this local ontology and the global one OntoBase are established by the mapping module and stored in the Ontology Mappings DataBase OMDB. These mappings are used during the translation phase by the Producer Translation Agent PTA to translate information from OntoProd to OntoBase and *vice versa*.

#### 4.3. SOAcustomer

The SOAcustomer has a project manager which is responsible for its projects realisation, a Customer DataBase (CDB) containing the description of the projects defined by the project manager, a discovery module to discover the requested services as well as their SOAproducers, a local register and a project management system. The local register is a limited copy of the SOAregister. Based on a collaboration coefficient, which represents the number of times that the producer is selected after a successful collaboration over the number of times that the producer is contacted for collaboration. It stores information about the most frequently invoked services and their SOAproducers with which the SOAcustomer had good collaborations. Also, the local register accelerates the service discovery and is a better guide for the SOAcustomer in the choice of its partners by privileging the most requested SOAproducers. That enhances the partnership relations between the SOAcustomer and the SOAproducers with

which it has already collaborated. The coherence between the information contained in the local register and those contained in the SOAregister must be regularly verified to take into account the modifications made in this last one, i.e. deletion of certain SOAproducers, modification of the description of certain services, etc.

The project management system allows managing in a distributed way the customer's projects by collaboration with the selected SOAproducers. Because the SCEP model is well adapted to this objective, the agents of this model are integrated into the project management system to manage in a distributed and autonomous way the realisation of the projects. The management of the projects is made on one hand by cooperation between the SCEP customer agents representing the projects, and the SCEP ambassador agents representing the SOAproducers, via the shared SCEP environment, and on the other hand by invoking services at the concerned SOAproducers by the corresponding ambassador agents. In the rest of this paper, an ambassador agent represents a producer ambassador agent and not a supervisor one.

The SOAcustomer has its own ontology denoted OntoSCEP. Mappings between this local ontology and the global one OntoBase are established by the mapping module and stored in the ontology mappings database OMDB. These mappings are used during the translation phase by the translation agent CTA to translate information from OntoSCEP to OntoBase and *vice versa*.

### 5. S-SCEPSOA functioning

The management process is achieved by three steps for which sequence diagrams describe associated treatments. The first one is the identification of partners, which

concerns services publication, and discovery phases in the SOA context. The second step deals with instantiation of SCEP components at the SOAcustomer side and connection to SOAproducers selected at the previous step. The SCEP supervisor agent gets from the customer database CDB the projects' description and creates the shared environment as well as the customer and ambassador agents. One customer agent (respectively ambassador agent) is created for each project, similarly one ambassador agent is created for each SOAproducer. At all levels of interaction between actors and agents is achieved always via messages. The third/last step concerns interactions and cooperation between SOAcustomer and its SOAproducers through the SCEP instantiation to manage multi-site production projects.

The rest of this paper focuses on the third step. The objective is to show interoperability in collaboration between partners' production systems during the multi-site projects management. These interactions are achieved in a finite number of management cycles. Each cycle consists of three phases: initialisation and transmission of projects' description to the SOAproducers, treatment of projects by the SOAproducers, and validation of results by the SOAcustomer (Archimede and Coudert 2001).

### 5.1. Initialisation and transmission of projects' description to the SOAproducers

This phase is detailed in Figure 4 by the sequence diagram.

At the beginning of the first management cycle, the supervisor agent invites customer agents to deposit their projects' realisation constraints in the SCEP environment respecting exactly the wishes of the SOAcustomer.

After the initialisation of projects, for an invitation of the supervisor agent, every ambassador agent gets from the environment, constraints for activities

concerned by the service of the SOAproducer, which it represents.

After the collect of information, the ambassador agent asks for the customer translation agent CTA, which uses the ontology mappings stored in the OMDB to translate it from OntoSCEP to OntoBase in order to send it to the corresponding SOAproducer. After that, the ambassador agent generates an invocation request containing the project description structured as OntoBase's concepts and invokes the service at the corresponding SOAproducer. The quantity of information transmitted in each request depends on the service granularity. When the transmitted information is relative to a single project's task, the granularity is called fine. In this case, the complexity of the management's algorithm increases because the customer has to make a significant number of service invocations to manage its projects. That slows down the algorithm convergence speed, which represents the number of cycles before obtaining a validated management result. For this reason, it is recommended in practice a coarse granularity corresponding to the choice of all tasks for every customer's project; the convergence speed of the management algorithm is faster in this case.

### 5.2. Treatment of projects by the SOAproducers

At the SOAproducer side, a management cycle corresponds to a 'timeout' at the end of which the production application is launched. The duration of this 'timeout' is defined in the service description in order to be taken into account in the management cycle at the SOAcustomer side. The duration of the management cycle at the SOAcustomer corresponds to the longest 'timeout' of the concerned SOAproducers. In fact, the interest of this strategy is to avoid the SOAcustomers waiting for the invocation result of the requested service for an unknown duration. Because of the use of statefull services (the

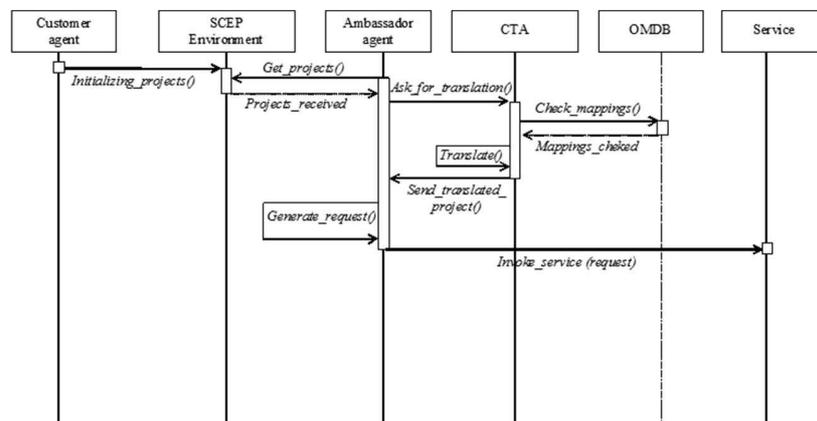


Figure 4. Initialisation and transmission of constraints to SOAproducer.

state of the previous service invocations is taken into account), the SOAproducer has to keep in its PDB the results established by the production application for certain duration while waiting for the confirmation on behalf of the SOAcustomer having invoked the service. In fact, stateless services (which do not care about previous service invocations) are not convenient in this case because they slow down the management process. That is due to the fact that for a validation of a service invocation result, the SOAcustomer has to call this service at the corresponding SOAproducer, permanently.

The treatment processing of received projects at the SOAproducer is described in Figure 5 by the sequence diagram. Once the service invocation request is received by the service, this latter asks the translation agent PTA to translate it from OntoBase to OntoProd in order to store it in the PDB, identifying the various projects to be managed.

Normally, the duration of the storage of a result in the PDB database is of the order of a management cycle. Indeed, for an established result by the SOAproducer, this latter cannot wait, for a long time, the answer of a SOAcustomer about the validation or not of this result, because in that case, it has to keep the production resources reserved for several cycles as long as the SOAcustomer has not answered. This strategy penalises the SOAproducer having resources saturated for a long time. That is not in the interest of the SOAproducer because after this long wait, the SOAcustomer can refuse the proposed results.

After the timeout for launching of the production application, the administrator of this latter gets from the PDB the non-treated projects and loads them in the production application. After the treatment, the administrator of the production application registers in the PDB the results of this management cycle with the state not validated and changes the state with treated for the treated projects. The result established by the production application and described according to OntoProd, is translated by the translation agent PTA to OntoBase for being sent by the service as an invocation response to the corresponding ambassador agent of the concerned SOAcustomer.

The interactions with a SOAcustomer finish when there are no more projects (consequently results) for this customer in the PDB of the SOAproducer, or if every project and result for this SOAcustomer existing in the PDB database is stated validated.

### 5.3. Validation of results by the SOAcustomer

Every ambassador agent receives from the corresponding SOAproducer the result of the invoked service. In a SOAcustomer project, several ambassador agents can be involved. For each management cycle, at the SOAcustomer, results must be received for all defined projects. For that purpose, at the end of each management cycle, it is considered that there is at least one result for each project. The sequence diagram concerning the treatment and the validation of a result for the SOAcustomer is illustrated in Figure 6.

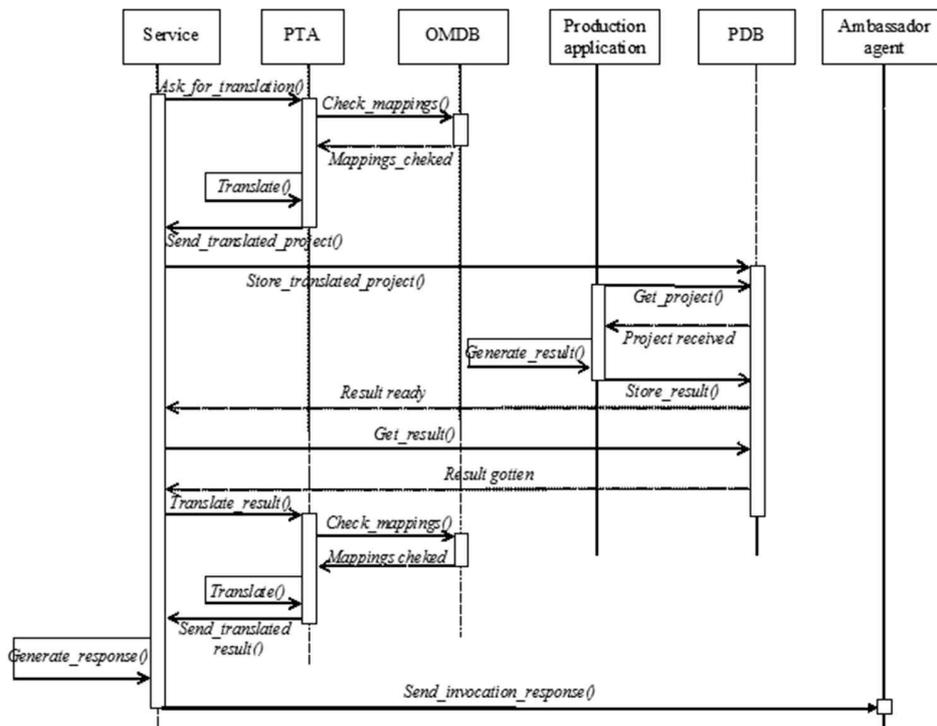


Figure 5. Sequence diagram of the projects processing of the SOAproducer.

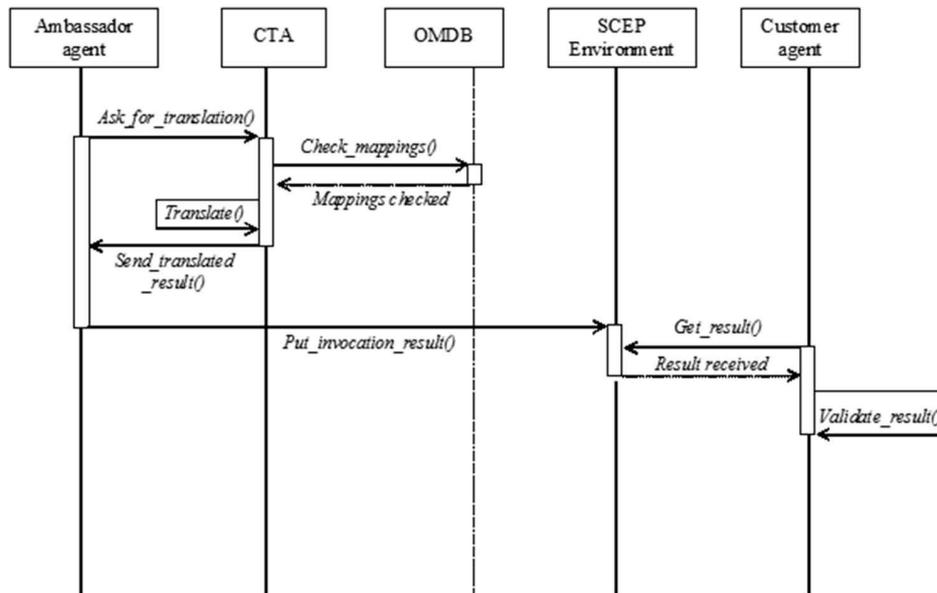


Figure 6. Treatment and validation of a result for the SOAcustomer.

The response containing the invocation result received by an ambassador agent contains the identifiers of the concerned projects, and the constraints proposed by the SOAproducer for the requested activities. The information contained in the response is extracted by the ambassador agent, which asks the translation agent CTA to translate it from OntoBase to OntoSCEP in order to deposit it in the SCEP environment for being discovered by the corresponding customer agents.

In the validation phase, four cases can take place. (1) The customer agent validates the received result for the totality of the project activities. (2) It completely refuses the result for any reason, i.e. the validation of a result for another SOAproducer. (3) It refuses the received result and proposes new realisation constraints for the concerned activities. (4) It cancels or confirms some activities of the project relative to the received result.

A re-invocation of the service is necessary in cases 1, 3 and 4 and that before the end of the current management cycle at the corresponding SOAproducer. For the second case, re-invocation of service is not necessary because the result was not confirmed by the SOAcustomer at the corresponding SOAproducer. That leads to the deletion of this result from the BDP at the corresponding SOAproducer at the beginning of the next management cycle.

In the third case, new constraints are defined for the concerned activities. If all activities are concerned by the cancellation, the procedure is the same as in the second case. If the SOAcustomer wants to confirm certain activities of a result and not the totality, it re-invokes the concerned service by setting new constraints for the not validated activities.

Interactions with a SOAproducer finish the following two cases: when the SOAcustomer cancels all the projects

asking for services supplied by this SOAproducer; when it receives a confirmation response about all the validated activities from this SOAproducer.

After the validation phase, the decision is deposited by customer agents in the SCEP environment for being treated by the concerned ambassador agents in the next management cycle.

## 6. Application to two heterogeneous planning systems

The objective herein is to show the application of S-SCEPSOA to two heterogeneous planning systems. For this reason, the planning is not the main interest of this section, but rather the exchange of information and interoperability between planning systems during the planning of a manufacturing project.

The case study concerns a service-oriented marketplace, which proposes planning services associated with manufacturing activities offered by two enterprises A and B. It is considered that these enterprises are close enough in order to neglect inter-site transport activities. An enterprise C wishes to plan its manufacturing project requiring activities offered by enterprises A and B. Enterprise C is using a SCEP-based planning system called R@mSES (Re@ctive Multi-site SystEm for Scheduling) (Archimede, Charbonnaud, and Firmin 2003) for managing its project planning. The same system is also used by enterprise A. However, enterprise B has a different planning system called TAPAS (The Almost Perfect Approach to Scheduling) (Moutarlier, Geneste, and Grabot 2000) to manage the planning of the provided activities.

The meta-model of R@MSES (Figure 7) (respectively TAPAS Figure 8) is represented by the ontology R@m sesontology (respectively TAPASontology), which captures the semantic of the used planning concepts. The semantic heterogeneity between these two ontologies is illustrated by structural and terminological heterogeneities

as shown in Figure 9 (Ishak, Archimede, and Charbonnaud 2009).

The difference between the structures of the Operation concept in these two ontologies illustrates the structural heterogeneity. In fact, Operation concept in R@m sesontology is equivalent more precisely in

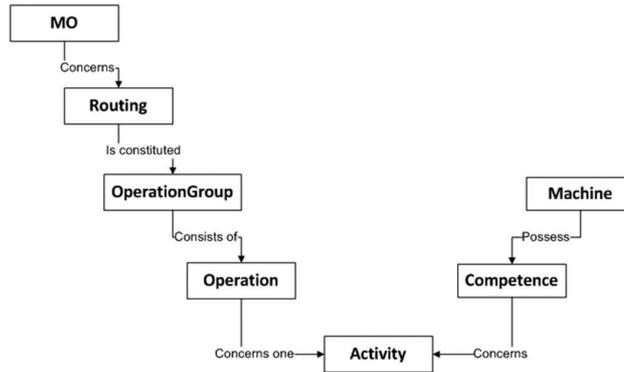


Figure 7. Concepts of R@m ses and their relations.

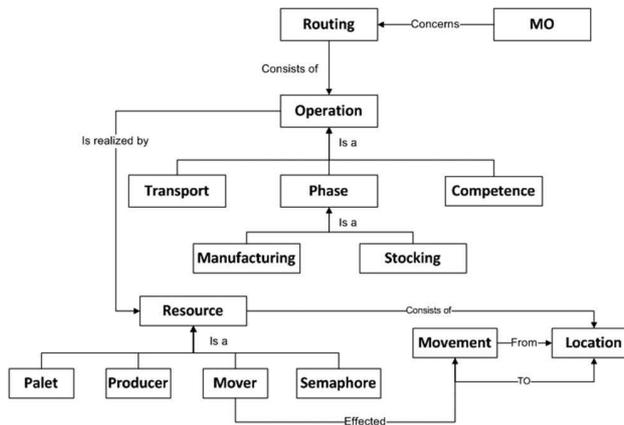


Figure 8. Concepts of TAPAS ontology.

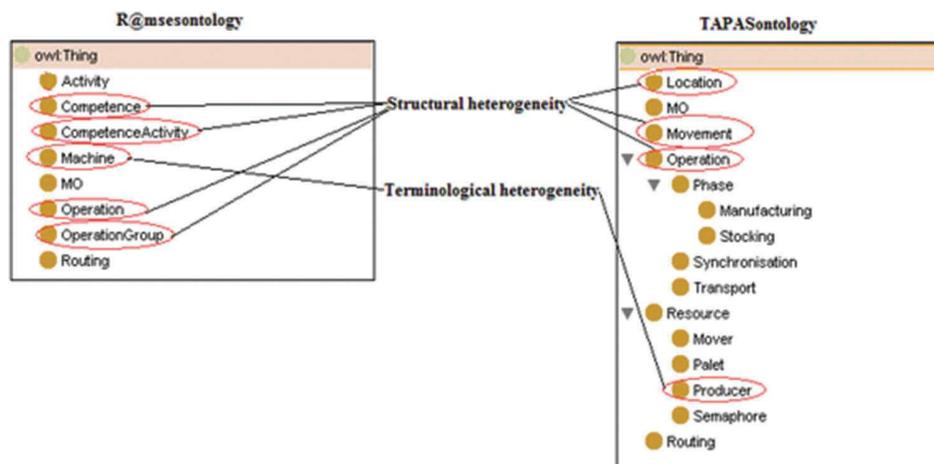


Figure 9. Semantic heterogeneity between R@m sesontology and TAPASontology.

TAPASontology to Manufacturing concept, which is a sub-concept of Phase, which is itself a sub-concept of Operation concept of TAPASontology. This heterogeneity is also illustrated by the existence of some concepts in one ontology, which are absent in the other, i.e. Competence, CompetenceActivity, OperationGroup in R@msonontology and Movement, Location in TAPASontology. The use of different concepts to describe the same information, i.e. Machine in R@msonontology and Producer in TAPASontology illustrates the terminological conflicts between these two ontologies. Heterogeneities between these two ontologies do not concern only ontology classes but concern also properties and instances of these classes.

The proposed semantic strategy requires a development of a global and common ontology OntoBase structuring planning concepts to be exchanged between partners. For this purpose, the OZONE planning ontology (Jiang and Hou 2012) was used and improved with new planning concepts to take into account more planning aspects as illustrated in Figure 10 (Ishak, Archimede, and Charbonnaud 2010). Only concepts linked to the added ones are represented.

Concepts written in bold represent the added concepts to OZONE. This latter is based on the five following basic concepts: Demand, Activity, Resource, Product and Constraint. A Demand is an input request for one or

more Products, designating the goods or services required. Satisfaction of Demands is centred on the execution of Activities. An Activity is a process that uses Resources to produce goods or provide services. The use of Resources and the execution of Activities are restricted by a set of Constraints. In this context, scheduling project is defined as a process of feasibly synchronising the use of Resources by Activities to satisfy Demands over time. To take into account the concept of cost for activities and demands, a new sub-concept was added as a constraint for activities and demands (Cost\_Constraint). A Planned\_Demand concerns a manufacturing order already established by a customer, which is represented in OntoBase as Demand. A Planned\_Demand has a Cost\_Constraint and temporal constraints, i.e. Due\_Date, Release\_Date and concerns one or several Planned\_Activity. A Planned\_Activity concerns an activity already existing in the concerned Demand. It has a Cost\_Constraint and can be scheduled according to one or several Activity\_Scheduling\_Plans. An Activity\_Scheduling\_Plan is modelled by temporal constraints concerning the beginning of the Planned\_Activity and its duration.

Before sending invocation request to the enterprise B, the enterprise C project's description is translated from R@msonontology to OntoBase. An example of a R@msonontology project description and its translation

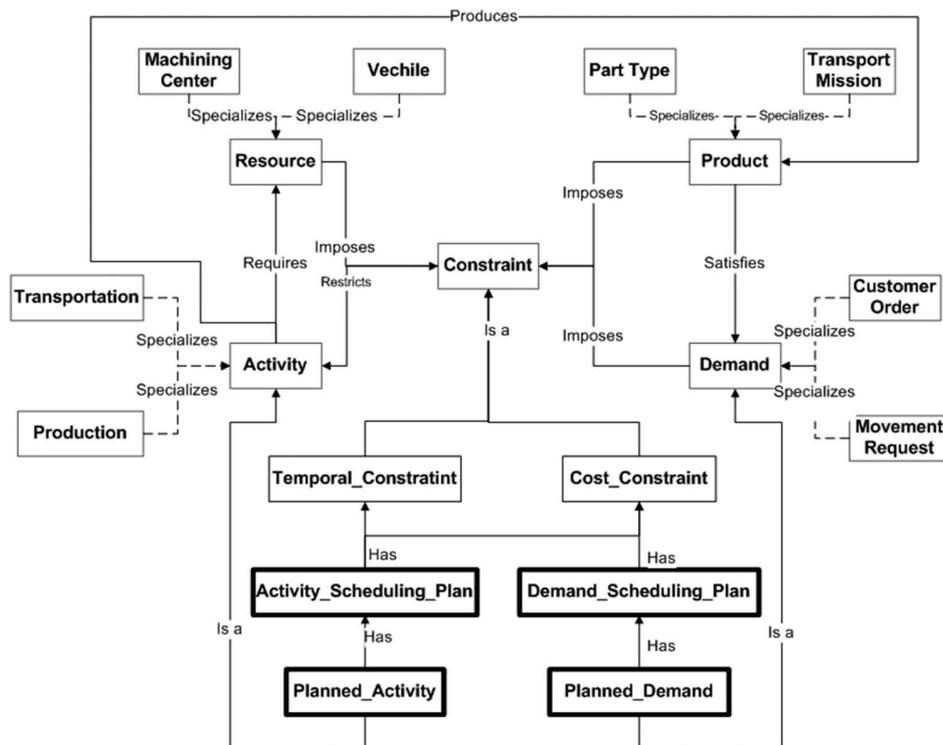


Figure 10. Concepts of OntoBase and their relations.

to OntoBase is presented respectively in Figures 11 and 12. Once the invocation request is received by the planning service of enterprise B, the project description is extracted and is translated according to TAPASontology in order to be loaded and treated by the TAPAS planning system of this partner. Note that the TAPAS system does not take into account the activity cost constraint.

Alignment here is achieved using semantic-based similarity algorithm proposed in Karray, Chebel-Morello, and Zerhouni (2010). This similarity algorithm applies three techniques for mapping, terminological, internal structure and extensional. After establishing the planning result with the TAPAS system of enterprise B (denoted by

Producteur2 in Figures 11 and 12), the cost constraint is added for every activity having the same value requested by the customer. That is for not penalising this partner (enterprise B) if all other constraints requested by enterprise C are respected. An example of a result described according to the TAPASontology and its translation to OntoBase is given in Figure 13. When enterprise C receives the OntoBase planning result, it is translated to R@msesontology for being treated and validated. The validation concerns the confirmation of the planned activities in the proposed result. This validation is global for all planned activities or partial for some planned activities.

```

-<Project_Description ID="Project1">
  <Producer>Producteur2</Producer>
  <MO ID="OF1">
    <MO_Due_Date>01/01/2015 00:00:00</MO_Due_Date>
    <MO_Release_Date>01/01/2015 00:50:00</MO_Release_Date>
    <MO_Cost>20$</MO_Cost>
    <MO_Quantity>1</MO_Quantity>
    <Routing>Gamme_pieds</Routing>
    -<MO_OperationGroups>1</OperationGroups>
      -<MO_OperationGroup ID="grop1">
        <Type>Sequential</Type>
        -<Operations>
          <Operation ID="Sciage_pieds">
            <Activity>SciageBois</Activity>
            <Start_Time>01/01/2015 00:00:00</Start_Time>
            <End_Time>01/01/2015 00:20:00</End_Time>
            <Operating_Time>00:20:00</Operating_Time>
            <Setup_Time>00:00:00</Setup_Time>
            <Operation_Cost>10$</Operation_Cost>
            <Customer_Confirmation>NO</Customer_Confirmation>
            <PreviousOperation />
            <NextOperation>Percege_pieds</NextOperation>
          </Operation>
          <Operation ID="Percege_pieds">
            <Activity>PercegeBois</Activity>
            <Start_Time>01/01/2015 00:40:00</Start_Time>
            <End_Time>01/01/2015 00:50:00</End_Time>
            <Operating_Time>00:10:00</Operating_Time>
            <Setup_Time>00:00:00</Setup_Time>
            <Operation_Cost>10$</Operation_Cost>
            <Customer_Confirmation>NO</Customer_Confirmation>
            <PreviousOperation>Sciagee_pieds</PreviousOperation>
            <NextOperation />
          </Operation>
        </Operations>
      </MO_OperationGroup>
    </MO_OperationGroups>
  </MO>

```

Figure 11. XML content for a project description according to R@msesontology.

```

-<Project_Description ID="Project1">
  <Producer>Producteur2</Producer>
  -<Demand ID="OF1">
    <Due_Date>01/01/2015 00:00:00</Due_Date>
    <Release_Date>01/01/2015 00:50:00</Release_Date>
    <Cost>20$</Cost>
    <Currency>$</Currency>
    <Quantity>1</Quantity>
    <Product>Gamme_pieds</Product>
    <Activity ID="Sciage_pieds">
      <Activity_Name>SciageBois</Activity_Name>
      <Start_Time>01/01/2015 00:00:00</Start_Time>
      <End_Time>01/01/2015 00:20:00</End_Time>
      <Duration>00:20:00</Duration>
      <Setup_Time>00:00:00</Setup_Time>
      <Cost>10$</Cost>
      <Currency>$</Currency>
      <Customer_Confirmation>NO</Customer_Confirmat
    -<Activity_Temporal_Relation>
      -<Synchronization ID="Start_Before">
        -<Related_Activities>
          <Activities>Percege_pieds</Activities>
        </Related_Activities>
      </Synchronization>
      -<Synchronization ID="End_Before">
        -<Related_Activities>
          <Activities>Percege_pieds</Activities>
        </Related_Activities>
      </Synchronization>
    </Activity_Temporal_Relation>
  </Activity>
  <Activity ID="Percege_pieds">
    <Activity_Name>PercegeBois</Activity_Name>
    <Start_Time>01/01/2015 00:40:00</Start_Time>
    <End_Time>01/01/2015 00:50:00</End_Time>
    <Duration>00:10:00</Duration>
    <Setup_Time>00:00:00</Setup_Time>
    <Cost>10$</Cost>
    <Currency>$</Currency>
    <Customer_Confirmation>NO</Customer_Confirmation>
  -<Activity_Temporal_Relation>
    -<Synchronization ID="Start_After">
      -<Related_Activities>
        <Activities>Sciage_pieds</Activities>
      </Related_Activities>
    </Synchronization>
    -<Synchronization ID="End_After">
      -<Related_Activities>
        <Activities>Sciage_pieds</Activities>
      </Related_Activities>
    </Synchronization>
  </Activity_Temporal_Relation>
  </Activity>
</Demand>

```

Figure 12. XML content of the same project description translated to OntoBase.

<pre> -&lt;Project_Planning ID="Project1"&gt;   &lt;Customer&gt;cust1&lt;/Customer&gt;   -&lt;MO ID="OF1_project1_cust1"&gt;     &lt;Start&gt;01/01/2015 00:00:00&lt;/Start&gt;     &lt;End&gt;01/01/2015 00:50:00&lt;/End&gt;     -&lt;Activity ID="sciage_pieds"&gt;       &lt;Resource&gt;SciageProducer&lt;/Resource&gt;       &lt;Begins&gt;01/01/2015 00:00:00&lt;/Begins&gt;       &lt;Duration&gt;00:20:00&lt;/Duration&gt;     &lt;/Activity&gt;     -&lt;Activity ID="perage_pieds"&gt;       &lt;Resource&gt;PerageProducer&lt;/Resource&gt;       &lt;Begins&gt;01/01/2015 00:40:00&lt;/Begins&gt;       &lt;Duration&gt;00:10:00&lt;/Duration&gt;     &lt;/Activity&gt;   &lt;/MO&gt;   -&lt;MO ID="OF3_project1_cust1"&gt;     &lt;Start&gt;01/01/2015 01:40:00&lt;/Start&gt;     &lt;End&gt;01/01/2015 02:00:00&lt;/End&gt;     -&lt;Activity ID="perage_assise"&gt;       &lt;Resource&gt;PerageProducer&lt;/Resource&gt;       &lt;Begins&gt;01/01/2015 01:40:00&lt;/Begins&gt;       &lt;Duration&gt;00:20:00&lt;/Duration&gt;     &lt;/Activity&gt;   &lt;/MO&gt; </pre>	<pre> -&lt;Project_Planning ID="Project1"&gt;   &lt;Customer&gt;cust1&lt;/Customer&gt;   -&lt;Planned_Demand ID="OF1"&gt;     -&lt;Planning_Demand_Scheduling_Plan&gt;       &lt;Start_Date&gt;01/01/2010 00:00:00&lt;/Due_Date&gt;       &lt;End_Date&gt;01/01/2010 00:50:00&lt;/Release_Date&gt;       &lt;Demand_Cost&gt;20&lt;/Demand_Cost&gt;       &lt;Demand_Currency&gt;\$&lt;/Demand_Currency&gt;     &lt;/Planning_Demand_Scheduling_Plan&gt;     -&lt;Planned_Activity ID="sciage_pieds"&gt;       &lt;Planned_Activity_Scheduling_Plan&gt;         &lt;Activity_Name&gt;SciageBois&lt;/Activity_Name&gt;         &lt;Start_Time&gt;01/01/2015 00:00:00&lt;/Start_Time&gt;         &lt;End_Time&gt;01/01/2015 00:20:00&lt;/End_Time&gt;         &lt;Duration&gt;00:20:00&lt;/Duration&gt;         &lt;Activity_Cost&gt;10&lt;/Activity_Cost&gt;         &lt;Activity_Currency&gt;\$&lt;/Activity_Currency&gt;         &lt;Producer_Confirmation&gt;NO&lt;/Producer_Confirmation&gt;       &lt;/Planned_Activity_Scheduling_Plan&gt;     &lt;/Planned_Activity&gt;     -&lt;Planned_Activity ID="perage_pieds"&gt;       &lt;Planned_Activity_Scheduling_Plan&gt;         &lt;Activity_Name&gt;PerageBois&lt;/Activity_Name&gt;         &lt;Start_Time&gt;01/01/2015 00:40:00&lt;/Start_Time&gt;         &lt;End_Time&gt;01/01/2015 00:50:00&lt;/End_Time&gt;         &lt;Duration&gt;00:10:00&lt;/Duration&gt;         &lt;Activity_Cost&gt;10&lt;/Activity_Cost&gt;         &lt;Activity_Currency&gt;\$&lt;/Activity_Currency&gt;         &lt;Producer_Confirmation&gt;NO&lt;/Producer_Confirmation&gt;       &lt;/Planned_Activity_Scheduling_Plan&gt;     &lt;/Planned_Activity&gt;   &lt;/Planned_Demand&gt; </pre>
---	--

Figure 13. XML description of result according to TAPASontology and OntoBase.

A new service invocation is necessary for communicating the validation response to the partner. In the invocation request sent to enterprise B, enterprise C sets to 'Yes' the confirmation parameter noted customer\_confirmation for validated planned activity and to 'No' for those not validated. In the invocation response, enterprise B sets to 'Yes' the confirmation parameter producer\_confirmation for validated activities for which production resources are reserved at this partner. Otherwise, the producer\_confirmation parameter is set to 'No' indicating to enterprise C that the validated activities are not taken into account. It is due to the deletion of the corresponding result at enterprise B maintaining a planning result in its database for only one planning cycle.

The project is planned where all its activities are planned and validated by all partners. This studied case illustrates a single planning cycle for one project. For the planning of many projects, the same process is repeated for all other cycles until obtaining a validated plan for every project.

Proposed developments in the context of the SCEP-SOA architecture will find their full potential with the major revolution expected for by the industry 4.0 or factory of the future, which is characterised by a fusion of Internet and factories. The extension of mass customisation, the generalisation of the concepts of service and product-service, the need to respond in the best time to customer needs, the desire of manufacturers to deport the inter-site transport constraints on customers will result management by the customer of his or her projects and therefore very important potentialities of applicability in terms of planning

## 7. Conclusion

In service-oriented markets, managing multi-site projects is a difficult task due to the distributed aspect of the management, highlighting the heterogeneity existing between partners' applications. It was proposed S-SCEPSOA, a distributed and interoperable architecture based on a multi-agent system for the distributed management, integrating concepts of the SOA model and ontologies to ensure interoperability between heterogeneous applications and offering dynamic collaborations between partners. This architecture masks the complexity of the networks of applications to be set up to realise the projects. It facilitates cooperation between the various production applications in a platform independent manner, offering a loose coupling between the management models used by these applications.

The case study shows the effectiveness of the semantic interoperability according to the strategy integrated in S-SCEPSOA. This strategy is based on the use of a common and global ontology to structure the exchanged information between partners. Every partner has to establish bi-directional mappings between its local ontology and the global one. In future work, authors consider to test the proposed framework with more robust case study or use the existing benchmarks of other approaches if found. Authors also consider to enhance architecture's application from multi-site production to multi-site maintenance planning as well as multiple transportation systems planning. Authors also consider to work on the collaborative planning of all production, transportation and maintenance planning together.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## References

- Abidi, O., Z. Su, and M. Mohiuddin. 2011. "Strategic and Organizational Evolutions of High-Tech SME on Global Market." *Chinese Business Review* 10 (5): 327–339.
- Aleksovski, Z., M. Klein, W. Ten Kate, and F. Van Harmelen. 2006. "Matching Unstructured Vocabularies using a Background Ontology." In *International Conference on Knowledge Engineering and Knowledge Management*, edited by S. Staab and V. Svatek, 182–197. Berlin: Springer.
- Archimede, B., P. Charbonnaud, and C. Firmin. 2003. "A Supervised Multi-Site Reactive Production Activity Control Method for Extended Enterprise." *Journal of Decision Systems* 12 (3–4): 309–328. doi:10.3166/jds.12.309-328.
- Archimede, B., and T. Coudert. 2001. "Reactive Scheduling Using a Multi-Agent Model: The SCEP Framework." *Engineering Applications of Artificial Intelligence* 14 (5): 667–683. doi:10.1016/S0952-1976(01)00025-2.
- Botelho, L., S. Willmott, T. Zhang, and J. Dale. 2002. *Review of Content Languages Suitable for Agent-Agent Communication*. No. EPFL-REPORT-52454.
- Boukreda, D., R. Maamri, and S. Aknine. 2012. "A Timed Colored Petri-Net-based Modeling for Contract Net Protocol with Temporal Aspects." Proceedings of the Seventh International Multi-Conference on Computing in the Global Information Technology (ICCGI 2012), Venice, June 24–29, 40–45.
- Corcho, O., M. Fernandez-Lopez, and A. Gomez-Perez. 2006. "Ontological Engineering: Principles, Methods, Tools and Languages." In *Ontologies for Software Engineering and Software Technology*, 1–48. Berlin: Springer.
- Corella, V. P., R. C. Rosalen, and D. M. Simarro. 2013. "SCIF-IRIS Framework: A Framework to Facilitate Interoperability in Supply Chains." *International Journal of Computer Integrated Manufacturing* 26 (1–2): 67–86. doi:10.1080/0951192X.2012.681912.
- De Bruijn, J., M. Ehrig, C. Feier, F. Martín-Recuerda, F. Scharffe, and M. Weiten. 2006. "Ontology Mediation, Merging and Aligning." In *Semantic Web Technologies: Trends and Research in Ontology-based Systems*, edited by J. Davies, R. Studer, and P. Warren, 95–113. West Sussex: Wiley.
- Dudek, G., and H. Stadtler. 2005. "Negotiation-Based Collaborative Planning between Supply Chains Partners." *European Journal of Operational Research* 163 (3): 668–687. doi:10.1016/j.ejor.2004.01.014.
- Erl, T., C. Gee, P. Chelliah, J. Kress, H. Normann, B. Maier, L. Shuster, et al. 2014. *Next Generation SOA: A Concise Introduction to Service Technology & Service-orientation*. Upper Saddle River, NJ: Prentice Hall Press.
- Ferber, J. 1999. *Multi-agent Systems: An Introduction to Distributed Artificial Intelligence*. Vol. 1. Reading: Addison-Wesley.
- Genin, P. 2005. "La planification industrielle et ses limites." Techniques de l'ingénieur. L'Entreprise industrielle AG5115.
- Geraci, A. 1991. *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*. Piscataway, NJ: IEEE Press.
- Ghenniwa, H., and W. Shen. 2004. "Service-Oriented E-Marketplace: An Agent-Based Model." *Production Planning & Control* 15 (7): 696–709. doi:10.1080/09537280412331298193.
- Gruber, T. R. 1993. "A Translation Approach to Portable Ontology Specifications." *Knowledge Acquisition* 5 (2): 199–220. doi:10.1006/knac.1993.1008.
- Guarino, N. 1997. "Understanding, Building and using Ontologies." *International Journal of Human-Computer Studies* 46 (2): 293–310.
- Ishak, K., B. Archimede, and P. Charbonnaud. 2009. "Integration of Semantic Interoperability in a Distributed Architecture of Multi-site Planning." *IFAC Proceedings Volumes* 42 (4): 181–186.
- Izza, S. 2006. "Intégration des systèmes d'information industriels: une approche flexible basée sur les services sémantiques." PhD diss., Ecole Nationale Supérieure des Mines de Saint-Etienne.
- Jiang, Y.-Q., and L. Hou. 2012. "Representation of Ontology-based Modeling of Production Scheduling." *Coal Mine Machinery* 8: 132.
- Jiao, J. R., X. You, and A. Kumar. 2006. "An Agent-Based Framework for Collaborative Negotiation in the Global Manufacturing Supply Chain Network." *Robotics and Computer-Integrated Manufacturing* 22 (3): 239–255. doi:10.1016/j.rcim.2005.04.003.
- Karray, M.-H., B. Chebel-Morello, and N. Zerhouni. 2010. "A Contextual Semantic Mediator for a Distributed Cooperative Maintenance Platform." 2010 8th IEEE International Conference on Industrial Informatics, July 13–16, Osaka, 181–188. IEEE.
- Kovalchuk, Y. 2009. "A Multi-agent Decision Support System for Supply Chain Management." PhD diss., The University of Essex.
- Li, Y., X. Lu, K.-M. Chao, Y. Huang, and M. Younas. 2006. "The Realization of Service-Oriented E-Marketplaces." *Information Systems Frontiers* 8 (4): 307–319. doi:10.1007/s10796-006-9006-3.
- Lima, R. M., R. M. Sousa, and P. J. Martins. 2006. "Distributed Production Planning and Control Agent-Based System." *International Journal of Production Research* 44 (18–19): 3693–3709. doi:10.1080/00207540600788992.
- Maedche, A. 2012. *Ontology Learning for the Semantic Web*. Vol. 665. New York: Springer Science & Business Media.
- Monteiro, T., D. Roy, and D. Anciaux. 2007. "Multi-Site Coordination Using a Multi-Agent System." *Computers in Industry* 58 (4): 367–377. doi:10.1016/j.compind.2006.07.005
- Moutarlier, P., L. Geneste, and B. Grabot. 2000. "Tapas: A Modular Framework to Support Reuse in Scheduling Software Development." *Production Planning & Control* 11 (7): 648–659. doi:10.1080/095372800432115.
- Nishioka, Y. 2004. "Collaborative Agents for Production Planning and Scheduling (CAPPS): A Challenge to Develop A New Software System Architecture for Manufacturing Management in Japan." *International Journal of Production Research* 42 (17): 3355–3368. doi:10.1080/00207540410001695989.
- Noy, N. F. 2009. "Ontology Mapping." In *Handbook on ontologies*, edited by S. Staab and R. Studer, 573–590. Berlin: Springer.
- Nurmilaakso, J.-M., and P. Kotinurmi. 2004. "A Review of XML-Based Supply-Chain Integration." *Production Planning & Control* 15 (6): 608–621. doi:10.1080/09537280412331283937.

- Papakostas, N., K. Efthymiou, K. Georgoulas, and G. Chryssolouris. 2013. "On the Configuration and Planning of Dynamic Manufacturing Networks." In *Robust Manufacturing Control*, s.l., 247–258. Berlin: Springer.
- Papakostas, N., K. Georgoulas, S. Koukas, and G. Chryssolouris. forthcoming. "Organisation and Operation of Dynamic Manufacturing Networks." *International Journal of Computer Integrated Manufacturing* 1–9.
- Park, J., and J. Yang. 2006. "An International SME E-Marketplace Networking Model." In *The Economics of Online Markets and ICT Networks*, 245–257. Heidelberg: Physica-Verlag HD.
- Roy, N., and C. Hafner. 1997. "The State of the Art in Ontology Design." *AI Magazine* 18 (3): 53–74.
- Schlenoff, C., R. Ivester, and A. Knutilla. 1998. "A Robust Process Ontology for Manufacturing Systems Integration." Proceedings of 2nd International Conference on Engineering Design and Automation, Maui, 7–14.
- Schneeweiss, C., and K. Zimmer. 2004. "Hierarchical Coordination Mechanisms within the Supply Chain." *European Journal of Operational Research* 153 (3): 687–703. doi:10.1016/S0377-2217(02)00801-9.
- Smith, R. 1980. "Communication and Control in Problem Solver." *IEEE Transactions on Computers* 29: 12.
- Szymczak, M., G. Frackowiak, M. Gawinecki, M., Ganzha, M., Paprzycki, M. W. Park, Y. S. Han, and Y. T. Sohn. 2008. "Adaptive Information Provisioning in an Agent-based Virtual Organization—Ontologies in the System." In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, 271–280. Berlin: Springer.
- Taghipour, A., and J.-M. Frayret. 2012. "Mutual Adjustment Search with Incentive for Supply Chain Planning Coordination." *International Journal of Computer Integrated Manufacturing* 25 (10): 946–962. doi:10.1080/0951192X.2011.646307.
- Taghipour, A., and J.-M. Frayret. 2013. "Coordination of Operations Planning in Supply Chains: A Review." *International Journal of Business Performance and Supply Chain Modelling* 5 (3): 272–307. doi:10.1504/IJBPSM.2013.055729.
- Telle, O. 2003. "Gestion de chaînes logistiques dans le domaine aéronautique: aide à la coopération au sein d'une relation donneur d'ordres-fournisseur." PhD diss.
- Uschold, M., K. Martin, M. Stuart, and Z. Yannis. 1998. "The Enterprise Ontology." *The Knowledge Engineering Review* 13 (1): 31–89. doi:10.1017/S0269888998001088
- Wang, L.-C., C.-Y. Cheng, and S.-K. Lin. 2013. "Distributed Feedback Control Algorithm in an Auction-Based Manufacturing Planning and Control System." *International Journal of Production Research* 51 (9): 2667–2679. doi:10.1080/00207543.2012.738944.