OCEAN: an ontology for supporting interoperability service utilities in virtual organisations

Yiannis Verginadis*
Institute of Communications and Computer Systems, National Technical University of Athens, 9, Iroon Polytechniou Str., Athens, Greece
E-mail: jverg@mail.ntua.gr
*Corresponding author

Dimitris Apostolou
Informatics Department, University of Piraeus, 80 Karaoli & Dimitriou Str., Piraeus, Athens, Greece
E-mail: dapost@mail.ntua.gr

Nikos Papageorgiou and Gregoris Mentzas
Institute of Communications and Computer Systems, National Technical University of Athens, 9, Iroon Polytechniou Str., Athens, Greece
E-mail: npapag@mail.ntua.gr
E-mail: gmentzas@mail.ntua.gr

Abstract: Semantic interoperability is a crucial issue in industrial enterprises when they participate in virtual organisations (VOs), i.e., when they dynamically form network-based collaborative alliances of a temporary nature. Addressing semantic heterogeneities aims to ensure that the meaning of information exchanged by VOs is interpreted in the same way by all communicating parties and their systems. In this paper we examine how ontologies can be employed by a system of services for delivering interoperability to enterprises, independent of particular IT deployments. In order to support interoperability service utilities in VOs, this paper presents a top-level ontology for collaborative networked organisations (code named OCEAN). The OCEAN ontology is designed as a lightweight top-level ontology that provides a common terminological reference in terms of VOs. The paper also demonstrates the use of practical tools for achieving consensus of the shared conceptualisation of a virtual organisation (VO), among participants, while it outlines a service-oriented architecture (SOA) for supporting VO knowledge based collaborations using OCEAN. We demonstrate how that usage enables shared understanding in knowledge-intensive collaborations, as well as how it facilitates interoperability of applications that provide collaboration services, presenting concrete examples from the pharmaceutical industry.

Keywords: top level ontology; virtual organisation; VO; networked collaboration; semantic interoperability; interoperability service utility; ISU.

**Biographical notes:** Yiannis Verginadis is an Adjunct Lecturer in the Department of Mechanical and Industrial Engineering of the University of Thessaly and a Senior Researcher in the Institute of Information and Computer and Communication Systems at the National and Technical University of Athens (NTUA) in Greece. He received his Diploma and his PhD in Electrical and Computer Engineering from the National Technical University of Athens, Greece (2001 and 2006). His diploma thesis was in the area of collaboration and workflow management systems over IP, using browser and client server architectures and his PhD was on inter-organisational workflow management systems.

Dimitris Apostolou is a Lecturer in the Informatics Department of the University of Piraeus and a Senior Researcher in the Institute of Information and Computer and Communication Systems at the National and Technical University of Athens (NTUA) in Greece. He received his PhD on Knowledge Management and Decision Support from the School of Electrical and Computer Engineering, NTUA, and his MSc in IT from University College London, UK. He has an extensive experience as a Knowledge Management Consultant.

Nikos Papageorgiou received his Diploma in Electrical and Computer Engineering (1998) and his MSc on Engineering-Economic Systems (2007) from the National Technical University of Athens, Greece. He has an extensive experience as a Software Engineer and IT Consultant. He is currently a Research Engineer at the School of Electrical and Computer Engineering of NTUA.

Gregoris Mentzas is a Professor of Information Management at the School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) in Greece and Director of the Information Management Unit (IMU), a multidisciplinary research unit at the university. His area of expertise is information technology management and his research concerns the integration of knowledge management, semantic web and e-service technologies, collaboration and workflow management, corporate knowledge management in e-government and e-business settings. He received his PhD in Operations Research and Information Systems from NTUA.

---

1 **Introduction**

The need for supporting network enterprise collaborations is mainly attributed to the fact that enterprises are operating in an increasingly dynamic environment where market frequently changes, new technologies continuously emerge and competition is fierce at the global scale (Canavesio and Martinez, 2007). In order to survive in a competitive and complex market environment the single enterprise model is not sufficient and the creation of collaborative networks with other firms is considered to be necessary for the control of more profitable business areas (Brown and Jiangang, 1999). The basic motivation behind the necessity of collaborative networks is the intention of participating enterprises to
exploit new business opportunities, deliver superior quality products and services and at the same time keep the production cost as low as possible (Fox, 1993).

Enterprises wishing to take part in collaborative networks participate in formations often referred to as virtual organisations (VOs) (Davidow and Malone, 1992; Mowshowitz, 1997a, 1997b). In a VO, the participating firm is no longer a physical entity with a stable mission. Instead it is a dynamic entity systematically involved in temporary alliance via networks structures (Mowshowitz, 1997a; Ahuja and Carley, 1999). A VO is a short-term association with a specific goal of acquiring and fulfilling a collaboration opportunity. A key underpinning of VOs is the logical separation of VO members’ requirements (e.g., requests for information, advice, or transactions) from satisfiers (e.g., information services, collaboration services, or transactional services) (Mowshowitz, 1997a). Having such a capability allows management to continually examine service requirements, scan for matching service offerings and switch the assignment of satisfiers to requirements so as to optimise performance on the basis of explicit criteria such as reducing service delivery costs or improving service quality. VOs are practicable because computers make it possible to minimise the cost of switching between alternative requirement satisfiers (Mowshowitz, 1997a). To realise the potential benefits of switching at low cost, VO members’ systems offering and consuming services should be interoperable, e.g., it should be easy to change from, say, UPS to Airborne Express for overnight delivery, without having to make costly changes to the order dispatch system of the VO member that requests the service. Hence, the achievement of interoperability of data, information and systems is considered as a prerequisite for VOs to be efficient. Moreover, since each VO member undertakes particular sub-processes and contributes with critical competencies and knowledge to the joint effort, collaboration services of each partner should be available in an interpretable way. Additionally, VO members need to have a shared understanding of the knowledge needed to perform a knowledge-intensive collaboration. Towards this end, adequate semantic interoperability has to be established by means of a common frame of reference or at least a common terminology (Chituc et al., 2008).

Advances in semantic web (Berners-Lee, 2007) technologies, which enable machines to process and reason about resources in support of businesses interactions, have paved the way for ontology-based platforms enabling semantic interoperability between heterogeneous information systems. In this paper we examine how ontologies can be employed by a system of services for delivering interoperability to enterprises, independent of particular IT deployments. The main contributions of the paper are the following. First, the paper proposes an ontology representing VO objects, processes, roles and relationships as a formal framework for enabling resolution of semantic heterogeneities. Second, the paper demonstrates the use of practical tools and techniques for customising or extending the ontology for the particular needs of VOs and for achieving consensus of the shared conceptualisation of a VO among participants. Third, the paper outlines a pattern-based, event-driven architecture for supporting VO collaboration and for enabling the resolution of semantic heterogeneities. Concrete examples from the pharmaceutical industry are used to demonstrate the applicability and benefits of the proposed ontology and architecture in supporting VO collaboration.

The remaining of the paper is organised as follows: In Section 2 we discuss the emerging need for supporting semantic interoperability in VOs, while in Section 3 the tools and techniques that were used for the development of the OCEAN ontology are presented. In Section 4 we present the main concepts of this ontology and in Section 5 we
outline the proposed architecture for service-based VO collaboration. Finally, we conclude this paper with the application of our work in the pharmaceutical industry in Section 6 and with a discussion on the research implications and conclusions in Section 7.

2 Semantic interoperability, ontologies and their application in virtual organisations

2.1 Introduction

Interoperability refers to the ability for two systems to understand one another and to use functionality of one another (Chen et al., 2008). In the context of networked enterprises (i.e., enterprises that participate in a VO), interoperability refers to the ability of interactions (exchange of information and services) between enterprise systems. The enterprise interoperability research roadmap – EIRR (2009) argues that interoperability of enterprises in future business ecosystems will be a utility-like capability that enterprises can invoke on the fly in support of their business activities. This model of interoperability services already exists in specific industries such as tourism. The utility metaphor is to indicate that enterprises should be able to expect and afford basic, interoperable IT as a critical infrastructure for operation, just as water, electricity, and indeed the Internet and the web. The European Commission uses the term interoperability service utility (ISU) to denote the enabling system of services for delivering basic interoperability to enterprises, independent of particular IT deployment. The ISU is conceived to be a basic ‘infrastructure’ that supports information exchange between diverse knowledge sources, software applications, and web services.

Current interoperability solutions are often oriented toward integration of data required for executing a common business goal, often specified in terms of a contract. Technologies such as model-driven architecture (MDA) and service-oriented architecture (SOA, EIRR, 2009; OMG, 2009; Jardim-Goncalves et al., 2006) and architectural frameworks (Chen et al., 2008) aim to facilitate technical interoperability between heterogeneous software systems. Protocols and standards (domain-specific and domain-neutral) such as ebXML (2009), electronic data interchange (EDI, 2009), and RosettaNet (2009) have been enablers for the progress made in the ability to integrate heterogeneous information and data. Languages such as unified enterprise modelling language (UEML, Vernadat, 2002; Roque et al., 2008) have been proposed supporting the transformation of an enterprise model from one language to another. Finally, alignment algorithms (Jung, 2008) have been developed for discovering mappings between different but similar enterprise taxonomies.

2.2 Semantic interoperability in VOs

Semantic interoperability aims to achieve a more ambitious goal, that is to assure that the meaning of the information exchanged (e.g., business documents, messages) is interpreted in the same way by the communicating systems (Chituc et al., 2008). Each of the interoperability approaches outlined in Section 2.1 tackles semantic interoperability differently, e.g., semantic interoperability in ebXML relies on core competences, while RosettaNet relies on dictionaries or vocabularies or on a common set of business
documents. Moreover, development efforts have been pursued to extend their semantic capabilities [e.g., (Hofreiter and Huemer, 2002) for ebXML; (Katinurmi and Vitvar, 2006) for RosettaNet]. However, the ability to deal with all types of semantic heterogeneity in the context of VOs is still a challenge.

Semantic heterogeneity, by generating semantic conflicts between systems, represents a major challenge in VOs because it causes misunderstanding between the content and intended meaning of heterogeneous communicating systems (Pollock, 2001). Conflicts can be confounding (e.g., information appears to have the same meaning, but does not), scaling (e.g., different reference systems measure the same value), and naming (e.g., synonyms and homonyms).

For addressing semantic heterogeneity it is essential that the semantic definitions of the knowledge objects, processes, roles and relationships within VOs are defined based on a mathematically rigorous ontological foundation (Lin and Harding, 2007). Moreover, as VO members might come from different fields or have different professional backgrounds, it is necessary to introduce a mechanism to share common understanding of knowledge, and to agree on a controlled vocabulary. An ontology provides a representation of knowledge, which can be used in order to facilitate the comprehension of concepts and relationships in a given domain, the communication between VO members by making the domain assumptions explicit and the resolution of semantic heterogeneities between VO systems. Ontologies have proven to be an unambiguous and compact way of knowledge representation for mutual understanding as they provide a basis for sharing information (Plisson et al., 2007). The EIRR recognises ontologies as necessary to the operation of an ISU almost by definition – Gruber (1994) defines ontologies as “formal, explicit specifications of shared conceptualisations”.

The content of an ontology depends both on the amount of information, their specificity and on the degrees of formality that is used to express it. In terms of formality, ontologies can be lightweight and heavyweight (Mowshowitz, 1997a). A lightweight ontology can take the form of a graph or taxonomy, where concepts are arranged in a hierarchy with taxonomic (is-a) relationships between them. A heavyweight ontology adds more meaning to this structure by providing axioms and constraints, which tend to reduce the ambiguity of concepts by restricting and constraining the usage of information.

2.3 Existing approaches

Among the wide spectrum of approaches which differ in the amount of information and specificity, four categories of approaches can be distinguished for developing ontologies i.e., top level, domain, task and application level ontologies (Huang and Diao, 2008; Rajpathak et al., 2006; Andersson et al., 2006). Top level ontologies are used to represent the building blocks for a particular domain and basically constitute the first step toward knowledge representation for a domain (SUO WG, 2009). Basically, this kind of ontology is limited to concepts that are meta, generic, abstract and philosophical, and therefore are general enough to address (at a high level) a broad range of domain areas. In the last decade, many projects aimed at creating top level ontologies for different purposes: word net (Fellbaum, 1998), SUMO (Niles and Pease, 2001), DOLCE (Gangemi et al., 2002), AIAI enterprise ontology (Uschold et al., 1998), PROTON (Kiryakov, 2006), ECOLEAD (2009; Plisson et al., 2007) and the business management

Among these most relevant to our work is the ECOLEAD ontology which proposes an ontology for virtual breeding environments, which are long-term associations of enterprises that have the potential and the will to form a VO. The OCEAN ontology builds upon and extends the ECOLEAD ontology to cover the creation, operation and termination phases of VOs. OCEAN is designed as a lightweight top-level ontology providing a common terminological reference for humans participating in VOs and enabling semantic interoperability of VO and ISU systems. In particular, we focus on knowledge-oriented collaboration within VOs and subsequently OCEAN aims to enable interoperability of systems providing services for enabling knowledge-based collaboration.

3 The methodology for the development of the OCEAN ontology

Among the various ontology development methods that have been proposed (Cristani and Cuel, 2005), for the development of the OCEAN top level ontology we opted for a collaborative method because it addresses the objective of achieving a shared representation of a domain knowledge. Following the ontology development framework proposed in Holsapple and Joshi, (2002), we aimed to support domain experts to reach consensus through iterative evaluations and improvements. According to Holsapple and Joshi (2002), the phases of collaborative ontology design are:

1. the preparatory phase that defines the design criteria for the ontology for guiding and assessing its success
2. the anchoring phase that includes the development of the initial ontology that will seed the next phase
3. the iterative improvement phase that revises the ontology until consensus is reached through a consensus building technique
4. the application phase that applies the final ontology structure to the case in order to demonstrate its uses.

The first three phases that we followed for the collaborative design of the OCEAN ontology are discussed below, while Phase 4 is presented in Section 5 with the application of the OCEAN ontology in a VO from the pharmaceutical field.

3.1 Phase 1: definition of the design criteria

The design criteria will guide us throughout the whole process and will evaluate objectively the design of the ontology. As in Uschold and Gruninger, (1996) these criteria are clarity, coherence – consistency, extensibility, minimal ontological commitment and encoding bias. In our case the encoding bias criterion is not applicable because we will not use directly any formal representation language but via the PROTÉGÉ editing tool that automatically encodes our effort into OWL (2009).
3.2 Phase 2: design of the initial ontology

To develop the initial version of the ontology we opted for the main ontology building steps described in (Noy and McGuiness, 2001). We added one additional step to cater for the utilisation of ontology learning tools that can be used for semi-automatic ontology development. Before starting designing the initial ontology, we did an extensive literature review and discussed with domain experts about the scope of the top-level ontology. This briefing was aiming to prepare the participants for the whole venture, to leverage their active involvement as well as to familiarise them with the possible applications and benefits of the ontology. Domain experts were carefully selected in order to complement each other and represent diverse viewpoints resulting to a group of five academics and five practitioners with extensive experience in VOs. The steps of this methodology follow:

Step 1 Determine the domain and scope of the ontology.

Step 2a Re-using existing ontologies. We have used the ECOLEAD top-level ontology as a starting point for our work regarding the development of OCEAN top-level ontology.

Step 2b Utilisation of ontology learning tools. We analysed a corpus of 79 scientific journal papers and conference presentations from the related literature with the aim to identify important terms and relationships between terms. This process has been leveraged with Text2Onto (Cimiano and Völker, 2005), an ontology learning tool. In particular, we used Text2Onto’s ability to extract prominent terms using text mining algorithms and to mine relations between terms (taxonomical, mereological and general relations) using natural language processing algorithms.

Step 3 Enumerate the terms of the ontology.

Step 4 Class definition and class hierarchy. We followed the top-down approach took into account suggestions for class hierarchies provided by Text2Onto.

Step 5 Determine the properties of classes. Suggestions for object properties from Text2Onto were again taken into account. We must note however that we did not systematically model data type properties of classes as these are often not accounted for in top-level ontologies (see e.g., Plisson et al., 2007 for a similar approach).

Step 6 Determine the restrictions of the data type and the object properties.

3.3 Phase 3: consensus building

In the previous phase we created an initial version of the ontology, according to information collected from the literature and domain experts. In this phase we worked with experts to evolve the initial version of the ontology by asking them to evaluate it and finally reach consensus and agree upon the final version. To reach consensus between experts that were not co-located and did not collaborate synchronously, we followed an adaptation of the Delphi method (Fitch et al., 2001), a technique which involves multiple iterative rounds of anonymous responses to a questionnaire until either the opinions converge or until no further substantial change in the opinions can be elicited. During
rounds, the facilitator can omit marginal opinions of experts in order to drive consensus. The Delphi method has been selected because it can efficiently drive consensus due to the power given to the facilitator to omit marginal opinions, it can be combined with established statistical measures as shown below and it can be easily followed by participants that are non-IT experts as it does not require the use of any specialised tools.

In phase 3, participants evaluated and agreed upon (Porzel and Malaka, 2004):

1. the scope of vocabulary-concepts
2. the relation between classes and subclasses (taxonomy)
3. the adequacy of non-taxonomic relations (semantic relations between classes).

In particular, participants were asked to rank using a five-point Likert scale each concept, and each taxonomic and non-taxonomic relation of concepts for relevancy to the project and for ambiguity. Moreover, for each concept synonyms were collected in order to broaden the vocabulary of the domain. Finally, participants could enter new concepts and relations in each round which were then fed again into the evaluation process.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Calculation of medians and IQRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scores</td>
<td>5, 5, 5, 2, 1, 4, 5</td>
</tr>
<tr>
<td>Ordered scores</td>
<td>1, 2, 4, 5, 5, 5, 5</td>
</tr>
<tr>
<td>Median</td>
<td>5</td>
</tr>
<tr>
<td>Upper quartile</td>
<td>5</td>
</tr>
<tr>
<td>Lower quartile</td>
<td>2</td>
</tr>
<tr>
<td>IQR</td>
<td>(5 – 2) = 3</td>
</tr>
</tbody>
</table>

The results of the rankings in the above columns were gathered and analysed. Analysis was based on the application of statistical measures such as medians and interquartile ranges (IQR, Linstone and Turoff, 1975). For items to be considered ‘acceptable’ their median should be more than or equal to the second best score in the scale (e.g., 4 or 5 for a five-point Likert scale) and their IQR range score should be less than or equal to 1.5. An IQR of equal or less than 1.5 means that a majority of the scores does not greatly differ from one another. A median of less than or equal to the second worse score (e.g., 1 or 2 for a five-point Likert scale) means that the item was deemed ‘not acceptable’ or that participants had ‘strong disagreement’ with the item being included in the OCEAN top level ontology; this however did not preclude the item being included in lower level domain ontologies. The example presented in Table 1 illustrates the calculation of medians and IQRs. In practical terms, scores have been interpreted as follows:

- **Agree** (median between 4 and 5, IQR less than 1.5): the item (concept or relation) remains in the ontology because the majority of the participants approve it.
- **Disagree** (median between 1 and 2, IQR less than 1.5): the item (concept or relation) does not pass and have to be deleted from the ontology.
- **Neither agree – nor disagree** (all other cases): participants do not have consensus on the item and further deliberation is needed. The facilitator explains in detail this particular item to participants and they rate it again in order to decide explicitly whether they agree or not about this item.
4 The OCEAN ontology

The OCEAN ontology aims to represent a conceptual schema of the domain of VOs typically referred to as terminology box or TBox. The domain of VOs includes concepts such as collaborative network organisation, virtual breeding environment and business opportunity that model the external environment in which VOs are being bred; such concepts are modelled in the ECOLEAD ontology. OCEAN mainly focuses on knowledge-oriented collaborations apposite for VOs. Nevertheless, to fully cover the domain of VOs, we have used the part of the ECOLEAD ontology which covers extensively the VO breeding environment and built upon it towards a unified model that captures the general aspects of collaborative network organisations and at the same time present details about knowledge-oriented collaborations that are important during the creation, operation and termination phases of VOs.

For developing the OCEAN ontology we have used Protégé (2009) and for validating it we have used the OWL-DL reasoner Pellet (2009). Pellet provides reasoning services and performs consistency checking and computation of inferred hierarchies, equivalent classes and inferred individual types (Sirin et al., 2007). Due to spatial restrictions we cannot depict the whole (53 terms and 77 relationships were identified and modelled) of OCEAN; instead we depict the critical concepts, only. We have categorised the critical OCEAN concepts as follows:

- breeding environment related OCEAN concepts
- service and collaboration related OCEAN concepts (refer to knowledge-enabled collaboration services)

4.1 Breeding environment related OCEAN concepts

We have organised the presentation of OCEAN by putting first concepts and relationships that describe the VO’s breeding environment, as a necessary artefact to describe the full picture of the domain (Figure 1). The highlighted concepts of this part of the top level ontology were taken from the ECOLEAD ontology, while the remaining concepts and relationships appear as extensions. Some of the breeding environment related OCEAN concepts and relationships are:

- A VO is a short-term association (of organisations) with a specific goal of acquiring and fulfilling a collaboration opportunity. A VO member represents an entity collaborating with other entities in the VO (Plisson et al., 2007). In simpler words VO members are the organisations which participate in a VO. A VO is bread in a VBE, an association of organisations and their related supporting institutions, which have both the potential and the will to cooperate with each other through the establishment of a base long-term cooperation agreement and interoperable infrastructure (Camarinha-Matos and Afsarmanesh, 2005). VO’s aim is to deliver products (anything an organisation may produce: goods or services), has a CommonGoal, undertakes a Project, uses CollaborativeMethodsAndTools and exploits a CollaborationOpportunity. With the term CollaborativeMethodsAndTools we define all the synchronous or asynchronous tools and methods that are going to be developed in terms of a system in order to support and enhance collaboration within a VO.
Figure 1  Breeding environment related OCEAN concepts (see online version for colours)
Every VO member has (or should have) CollaborationCapability which declares the capability that is relevant to the participation of an enterprise in collaboration with partner enterprises. It includes both HR capabilities of personnel involved in management and operation of collaborative activities, and interoperability of software systems. The concept of CollaborationCapability concerns mainly the pre-creation phase of a VO (i.e., identification phase for Plisson et al., 2007) as it focuses on the knowledge about the capability of future VO partners to collaborate. A critical factor, that is often disregarded in efforts that describe and support VOs, is the fact that two potential partners may be unable to collaborate, although they appear to have all the necessary assets for participating in a specific VO (e.g., two partners that had unsuccessful collaborations in previous VOs, partners that have been engaged in lawsuits against each other etc.). Within the system that will use the OCEAN top level ontology, a VO may use an ISUService (described in the next section).

The information about the structure of a VO is described with the term topology which stands for the arrangement of the participants inside the VO (e.g., ‘Star alliance: a grouping of independent organisations, with a core organisation taking the lead management role’). By declaring that a VO is a kind of CNO we express that a VO is a collaborative network of organisations.

4.2 Service and collaboration related OCEAN concepts

In this section we present the top-level ontology concepts that refer to collaborations and services (Figure 2) that are to be provided by the ISU. The ISU is the enabling system of services for delivering basic interoperability to enterprises, independent of particular IT deployment. It may also denote and an enterprise providing such services. A service is a provider-client interaction that creates and captures value (IBM, 2009). An ISU service is technical, commoditised functionality, delivered as services provided by an ISU to support the collaboration between enterprises (EIRR, 2009). A non-exhaustive list of ISU services is presented below. Lower-level domain ontologies further specify each one of the ISU services.

- DecisionMaking, ConsensusBuilding, ConflictResolution services and other Group Support Services. For example, reach decision on production plans, budget expenditure, etc.
- KnowledgeManagementServices helping a company that wants to enter the VO, to efficiently build up and manage a knowledge base of collaboration-oriented internal knowledge, together with knowledge sharing and exchange services which guarantee adequate treatment of confidentiality concerns.
- Specific IntelligentServices such as OpportunityDetection (e.g., detection of opportunity to develop a new product) and RiskAssessment (e.g., risk of failure of the new product, risk of conflict between partners).
- CollaborationPatternServices as a means to use and reuse proven, useful, experience-based ways of doing and organising communication and collaboration activities in specific knowledge-oriented collaborative tasks. A Collaboration Pattern (Papageorgiou et al., 2009) has Pre-Conditions, Post-Conditions, category (CPatCategory), Application Area, and Triggers that are comprised of complex events.
Figure 2  Service and collaboration related OCEAN concepts (see online version for colours)
5 Using the OCEAN ontology in an event-driven architecture for supporting VO collaborations

Effective VO collaboration requires knowledge about generic collaboration practices, technological capabilities of past, current and potential network enterprise partners, as well as experience and knowledge of best practice in the formation and operation of networked enterprises. Each VO develops, through its lifetime, project specific knowledge. This is in part knowledge specific to the network’s product or service, and to the processes and technologies involved. For the most part, such knowledge needs to be maintained as available only to the network and its partners as it is of no use, and possibly even very confusing, outside that context. Nevertheless, such knowledge can be analysed in order to derive collaboration patterns (CPats) as means for capturing collaboration best practices, thus forming a basis for organisational learning. CPats aim to provide proven, useful, experience-based ways of organising collaboration activities and support their execution with collaboration services (Briggs, 2003; de Moor, 2006). The maintenance of a library of appropriate CPats by the ISU, available as process and service templates to be specialised as necessary and applied to network enterprises either as they form or as they subsequently evolve, is central to the support of collaborations within VOs.

In order to demonstrate the applicability of OCEAN in a system for supporting collaboration within VOs, we propose an architecture that leverages CPats in order to enable collaboration within the different phases of the VO life cycle. Moreover, we propose an event-driven architecture in order to enable adaptation of services based on the dynamics of VO collaborations. Specifically, our approach exploits events generated continually by collaboration services and uses them as triggering mechanisms for relevant CPats. The proposed architecture comprises the following software components (Figure 3):

- An event service bus which acts as a ubiquitous integration backbone through which events flow through intelligent routing mechanisms. It acts as a bridge between the several event sensors that perceive events from an event cloud and the complex event processing engine (CEP Engine).
- CEP Engine which has the capability to process simple events, combine them and produce complex constructs of events that may trigger CPats.
- A workflow/service execution engine which is responsible for implementing or solutions proposed by the triggered CPat by executing services or by invoking services provided by other systems, and generates events during the execution of the workflow/service.
- A collaboration pattern processing layer (CPP Layer) which validates CPats according to a set of predefined restrictions, and instantiates CPats for the specific collaboration context. It also manages execution and access rights related to CPats.
- A collaboration pattern editor (CPat Editor) which defines, edits, searches and simulates CPats. It produces abstract CPats which can be instantiated and adapted during run-time.
A collaboration pattern assistant (CPat Assistant) which, upon notification from CEP for complex events, uses facts from Collaboration Knowledge Base to evaluate the preconditions specified in rules and derives CPat recommendations (Verginadis et al., 2009). If the CPat Assistant user accepts the recommendation for a CPat, the CPat Assistant instructs through the CPP layer the execution of the CPat solution in the workflow/service engine.

**Figure 3** An EDA for supporting VOs (see online version for colours)

The persistence layer comprises three repositories. The Collaboration Knowledge Base contains domain models as well as facts about the collaboration that may change as the collaboration proceeds. It provides information about the state of collaborators and the state of collaborative work, while it aggregates events and their consequences for the ongoing collaborations. Moreover, a collaboration pattern base and an event pattern base are needed for storing CPats and event patterns respectively.

To illustrate the intended use of the proposed architecture, consider a scenario of a meeting scheduling service provided by the ISU. This service allows two or more VO members to schedule a physical meeting on a specific date, time and location. This service needs to be automatically engaged when the ‘schedule a meeting’ CPat is selected (Table 2).

CPats recommendations are based on event-condition-action (ECA) rules (Behrends et al., 2006). This type of rules is understood as follows: whenever a (complex) event occurs and conditions hold, the actions of all matching rules are executed. Simple events arrive at the CEP, for example simple event 1 (SE1) is generated when more than three months passed from the last progress meeting; SE2 is generated when the VO coordinator requests a meeting. CEP maintains partially detected complex events in its pattern base, for example complex event 1 (CE1): SE1 and SE2. Event patterns, remain
in the store until they are evaluated as true (occurred) or false (impossible). As soon as an event pattern is evaluated as true, CEP notifies CPat Assistant about the corresponding complex event occurrence. CPat Assistant checks for the preconditions of the CPat by searching the collaboration knowledge base. Upon preconditions validation, CPat Assistant derives recommendations by matching the current collaboration state to one or more CPats stored in the CPat base. The actual engagement of the workflow/service execution engine takes place upon the acceptance of the CPA recommendation by a qualified user. The ECA rule that triggers the execution of the CPat in this scenario is the following:

\[
\text{WHEN CE1 (\textit{?VO, ?VOCoordinator})} \\
\quad \text{IF (Number of available collaborators} \geq 4(\textit{?VOmember})) \text{ AND} \\
\quad (\text{available(\textit{?VOCoordinator}) AND (available(\textit{?VOBudget})}) \\
\quad \text{THEN Schedule a Meeting CPat1} \\
\quad (\textit{?VO, ?VOCoordinator, ?VOmember, ?VOmember list, VOmeetingLocation})
\]

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Collaboration pattern’s description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>&lt;Schedule a meeting&gt;</td>
</tr>
<tr>
<td><strong>Problem:</strong></td>
<td>Check the work progress for a specific deliverable</td>
</tr>
<tr>
<td><strong>Application area:</strong></td>
<td>All industries</td>
</tr>
</tbody>
</table>
| **Pre-conditions:** | Number of available collaborators > (Minimum number of collaborators) AND 
VO coordinator available AND 
budget available |
| **Triggering:** | The last progress meeting was held three months ago AND 
VO coordinator requested meeting |
| **Input resources:** | Project description of work, number of collaborators, 
collaborators’ contact details |
| **Solution:** | VO coordinator notifies collaborators about the need of 
organising a meeting 
VO coordinator proposes the use of a specific web application 
for agreeing on the date of the meeting |
| **Output resources:** | Meeting minutes document |
| **Exception:** | <Postpone scheduled meeting CPat> |
| **Post-conditions:** | Meeting took place AND 
agreed minutes produced |

The proposed architecture is being currently implemented in a prototype system for which we use open source for implementing the aforementioned components. The OCEAN top-level ontology comprises the underlying interconnecting framework
between the architecture components and domain ontologies, and is used to resolve semantic heterogeneities as described in Section 6.

6 Application of the OCEAN ontology and architecture in VO collaboration

In this section, we present the application of the OCEAN ontology and architecture for network enterprise collaboration in the pharmaceutical industry. The pharmaceutical industry is considered a typical example of knowledge-intensive sector where the problem of dealing with heterogeneous and vast number of information appears to be insurmountable. An important effort for the development of a semantic nomenclature for the pharmaceutical sector based on networked ontologies has been undertaken in the Neon (2009) project (Cárcel and Lobo Pariente, 2008). Instead of using a single ontology to model the domain, Neon developed a networked ontology model for binding and managing relations between existing, heterogeneous ontologies. The OCEAN top-level ontology and architecture relies on the existence of domain ontologies such as Neon’s in order to provide a shared understanding of the domain by pharmaceutical VO participants and to enable semantic interoperability of systems and applications providing collaboration and knowledge provision services to the VOs.

According to investigational new drug application process –IND (2009), the process of developing a new dermatological drug involves several different stages starting from pre-clinic studies (testing the drug in the lab, use it on guinea pigs etc.) and continuing with the four phases imposed by Foods and Drug Administration – FDA (2009) and the European Medicines Agency – EMEA (2009). During these phases a formal proposal is introduced to the FDA or EMEA with all the details of the new drug. Upon approval, phase one starts with the testing on a group of healthy people in order to decide on the drug toxicity, liver and spleen reaction, the best dose amount, the best way to administer the new drug (oral, patch, intravenous, intradermal). The next two phases involve the testing on a group of sick people in order to decide on the new dermatological drug effectiveness. Phase two involves 100–300 sick people while phase three involves the testing on an extensive group with ethnographic differences that takes place in different hospitals. Since only a 5% of new drugs are approved to be circulated in the public, not many efforts continue with Phase four where the approved drugs continue to be tested for side effects for many years after their first circulation.

In our case, we consider that the new dermatological drug has reached the critical phase three where the testing must proceed in different hospitals. According to the international conference on harmonisation of technical requirements for registration of pharmaceuticals for human use (ICH, 2009) that was held in Helsinki four decades ago, there was an agreement upon a set of good clinical practices. Of course these best practices may be altered by the ethics committees of each country involved that may decide on the details of the drug testing (e.g., people with age less than fourteen should not be tested) or by the release of a new regulation from the FDA or EMEA. Such a change on the clinical practices can be considered as a new opportunity in terms of a VO.
Figure 4  Breeding environment related OCEAN concepts – instantiated (see online version for colours)
As shown in Figure 4, the OCEAN ontology has been instantiated in order to describe our domain. The VO follows a certain topology: Star Alliance. This specific topology for structuring a VO involves the grouping of independent organisations, with a core organisation taking the lead management role (Lethbridge, 2001). The VO comprises two pharmaceutical companies with expertise in dermatological drug development and two hospitals with their own assets (testing knowledge, doctors supervising and volunteers). The common goal for this VO has been agreed to be the development of dermatological drugs according to the regulations and ethics taking into account the profit maximisation. The VO has been bred by a drug development virtual breeding environment (VBE) that combines pharmaceutical companies that are capable of developing any new drug and hospitals for the testing processes.

6.1 Enabling shared understanding of the knowledge needed to perform a knowledge-intensive collaboration

The ability of OCEAN to provide a common terminological reference and a shared understanding for human participating in VOs, is demonstrated by the following set of questions for which we were able to get answers from our instantiated ontology. We have used the SPARQL language for assessing the expressiveness capability of OCEAN. SPARQL (2009) is a query language for the semantic web that can be used to query an RDF schema or OWL model in order to filter out individuals with specific characteristics.

One such question could be: Which are the assets of each VO member? In Figure 5, it is shown how we can make such a question using SPARQL. Regarding our application, we get as an answer the group of assets per VO members (Hospital 1, Hospital 2, Pharma Company 1 and 2).

Figure 5 Retrieval of VO members’ assets (see online version for colours)
In Table 3 the reader can find more questions that can be answered using SPARQL queries through the instantiated OCEAN top level ontology.

**Table 3** SPARQL queries

<table>
<thead>
<tr>
<th>Query</th>
<th>SPARQL query</th>
</tr>
</thead>
</table>
| In which VOs have a specific pharmaceutical company participated in the past? | ```
SELECT ?VO
WHERE {
  :PharmaCompany1 :participatesInVO ?VO.
}
``` |
| Which are the projects that the DermaDrugDevelopmentVO has undertaken so far? | ```
SELECT ?Proj
WHERE {
}
``` |
| What are the possible moderator services depending on common goals for the DermaDrugDevelopmentVO? | ```
SELECT DISTINCT ?ModSrv ?CGoal
WHERE {
  :DermaDrugDevelopmentVO :hasCommonGoal ?CGoal.
  :DermaDrugDevelopmentVO :usesModeratorService ?ModSrv
}
``` |

Unlike databases, ontologies built in OWL such as OCEAN has a so-called open-world semantics in which missing information is treated as unknown rather than as false and OWL axioms behave like inference rules rather than as database constraints. For example, if we have asserted that BiotechOne is a VO member and that it participates in (which is the inverse property of hasParticipant) BioAlliance, then, because only VOs have VO members as participants, this leads to the implication that BioAlliance is a VO. If we were to query the ontology for instances of VO, then BioAlliance would be part of the answer. We can also ask if any collaborative network organisation that has VO members as participants is necessarily of VO. Query answering in OWL is analogous to theorem proving; therefore the OCEAN top level ontology plays itself an important role and is actively considered at query time. Considering both the schema and the data represented in OCEAN can be very powerful, making it possible to answer conceptual and extensional, queries as well as to deal with incomplete information.

### 6.2 Enabling semantic interoperability of systems and applications providing VO collaboration services

OCEAN is intended to support semantic interoperability of systems and applications providing collaboration services. OCEAN, combined with lower level domain ontologies, functions as a universal vocabulary so that interoperating services have the same interpretation of shared concepts. Typically, the ontology is directly mapped to the heterogeneous data sources giving applications direct access to the data through the ontology. In this section, we illustrate a new way of supporting semantic interoperability through the use of CPats. OCEAN, combined with lower level domain ontologies,
OCEAN: an ontology for supporting interoperability service utilities

semantically enriches the terms contained in the CPats. This semantic description can then be automatically processed by software entities so that service discovery, composition, selection, execution and monitoring can be done without human intervention.

Consider the scenario presented in Section 5 regarding the ‘schedule a meeting’ collaboration pattern. To make use of this service automatically, the calendar systems of the VO members should technically (i.e., via connecting to appropriate APIs) connect to the event scheduling service (Figure 6). Moreover, to schedule a specific event, several parameters have to be specified, such as alternative dates, time slots, event location, event duration, etc. These parameters are included in the specification of the event scheduling service. However, the calendar systems, in their own specifications, may not comply with this terminology. For example, one of them makes use of the terms ‘intervals’ instead of ‘timeslot’ and ‘event place’ instead of ‘event location’. Thus, the service can only be automatically invoked upon making the mappings between the different terms that refer to the same concepts.

Figure 6 The role of OCEAN top-level and domain ontologies in addressing semantic interoperability (see online version for colours)

Ad hoc mapping of terms by translator or middleware software is possible but this would only solve this specific mapping problem. We would have to do the same again when we encountered the next calendar system specifications. A better way to handle the semantic heterogeneity in this example is by providing a domain ontology about event scheduling. Apart from concepts, attributes and taxonomical relationships, the domain ontology includes semantic (Synset) relations between concepts by integrating information from
the Wordnet lexical database. As shown in the snapshot of the ontology in Figure 7, through the OWL equivalentProperty axiom, a matching between ‘interval’ and ‘timeslot’, and ‘place’ instead of ‘location’ is accomplished.

**Figure 7** Snapshot of event scheduling ontology

```
<owl:Class rdf:ID="event"/>
<owl:DatatypeProperty rdf:ID="date">
  <rdfs:domain rdf:resource="#event"/>
  <rdfs:range rdf:resource="&xsd:date"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="timeslot">
  <rdfs:domain rdf:resource="#event"/>
  <rdfs:range rdf:resource="&xsd:time"/>
  <owl:equivalentProperty rdf:resource="interval"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="location">
  <rdfs:domain rdf:resource="#event"/>
  <rdfs:range rdf:resource="&xsd:string"/>
  <owl:equivalentProperty rdf:resource="place"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="duration">
  <rdfs:domain rdf:resource="#event"/>
  <rdfs:range rdf:resource="&xsd:float"/>
</owl:DatatypeProperty>
```

Furthermore, the OCEAN top-level ontology can be used to specify conditional parameters in CPats. For example, a rule can be defined to derive the ‘Minimum number of collaborators’ required in order to schedule a meeting, which is a parameter that depends on the topology of the VO. The following indicative SWRL (2009) rule states that, in the case of a VO whose topology is ‘network’, the minimum number of available collaborators should be half the number of collaborators.

\[
\text{Ocean : VirtualOrganisation}(x) \land \text{Ocean : hasTopo}(x, \text{"Network"}) \Rightarrow \\
\text{EventScheduling : MinimumNumberOfCollaborators}(y) \land \\
\text{EventScheduling : NumberOfCollaborators}(z) \land \text{swrlb : divide}(y, z, 2)
\]
7 Conclusions

In this paper we presented OCEAN, a top-level ontology for collaborative networked organisations. The OCEAN ontology covers the creation, operation and termination phases of VOs and is designed as a lightweight top-level ontology that provides a common terminological reference for VO concepts and relations. For the development of OCEAN we have used specific tools and techniques for successfully completing the phases of the collaborative ontology design. Among these are the use of a semi-automatic ontology development tool for the identification and refinement of high level concepts of the domain and a technique, inspired by Delphi, to facilitate consensus building among experts. Regarding the application of OCEAN, we have instantiated it based on a VO example that focuses on the development and testing of new dermatological drugs. We validated the OCEAN ontology as an expressive tool for describing such VOs using SPARQL queries.

We believe that the explicit representation of the semantics of all relevant data through the OCEAN top level ontology enables collaboration within network enterprises and can serve as the basis to provide a qualitatively new level of functionalities. While several ontology roles may all be subsumed under the term ‘knowledge sharing’ (Chungoora and Young, 2008), different purposes of (re)use of ontologies also put different requirements on the ontologies (Valente and Breuker, 1996). In the following, we describe briefly the several aspects of the role of the OCEAN top-level ontology.

The OCEAN ontology formalises and enables network enterprise collaboration as it models formally the main factors that affect/enable the network enterprise collaboration orchestrated by an entire system. It targets specifically the relationships between ‘high level pieces’ of domain knowledge, explaining how they contribute altogether to the network enterprise collaboration. This top level ontology also enables better communication by defining a common-agreed vocabulary that: ensures shared meaning and understanding regarding project goals; facilitates knowledge acquisition in situations where teams have to work together because the ontology becomes a common, agreed-upon understanding of the terms, which can be understood by team members with different background knowledge (Valente and Breuker, 1996). Ultimately, the OCEAN ontology supports semantic interoperability between software components by formalising the used vocabulary explicitly in a machine-readable form.

Since one of the VOs’ main characteristics is the continuous change, their effectiveness is determined by their ability to adapt collaborations to changing circumstances and by the speed and accuracy with which information can be shared among the business partners (Sandakly et al., 2001). The proposed event-driven architecture exploits events generated continually by collaborations within the VO and managed by the ISU, and uses them as triggering mechanisms for identifying and recommending CPats, experience-based ways of organising VO collaboration activities. A system providing knowledge-oriented collaboration services for VOs may produce many events that might be relevant for other services. Such a dynamic behaviour of services cannot be defined in advance explicitly; nevertheless events can be categorised in advance and used as triggering mechanisms for CPats. Specifically, the architecture introduces reactive invocation of services based on signals sensed/derived within a VO and combines CPats with ontologies to address semantic heterogeneities in the application domain.
Our work has implications for researchers because it provides a blueprint for the development of ISU services for the support of networked enterprises. Based on the proposed architecture, ISU services enabling collaboration, supporting decision making, identifying opportunities, mitigating risks and providing knowledge for facilitating the VO lifecycle can be developed. Moreover, our work has implications for practitioners because it provides a ready to use top-level VO ontology that covers the creation, operation and termination phases of VOs as well as tools and techniques for extending or customising it to fit specific needs. We plan to implement an open source system based on the proposed architecture and to further validate the application of OCEAN as a framework for successfully interconnecting domain ontologies and for using them to resolve VO semantic heterogeneities.

Acknowledgements

This work has been partially funded by the European Commission under project SYNERGY ICT No 63631. The authors would like to thank the project team for comments and suggestions.

References


