University of Niš
Faculty of Mechanical Engineering
Chair for manufacturing, information technologies and management

Milan Zdravković, MSc Mech.Eng.

Formal framework for semantic interoperability in supply chain networks

PhD Dissertation

Submitted to Faculty of Mechanical Engineering in Niš, University of Niš for the degree of

PhD in Mechanical Engineering

Niš, June 2012.

Milan M Zdravković: Formal framework for semantic interoperability in Supply Chain Networks, University of Niš, Faculty of Mechanical Engineering, 2012

Keywords: Systems interoperability, Enterprise Information Systems, Semantic Interoperability, Ontology, Semantic Web

Резиме: Main objective of the thesis is the development and verification of relevant formal framework for representation and reasoning of knowledge in supply networks. The framework is based on the neutral specification of existing reference SCOR model for definition of supply chain processes. It is characterized by the modular ontologies, developed on the different levels of abstraction (from the literal translation of SCOR model to a common formalism, to implicit conceptualization of the enterprise information systems and explicit domain ontologies) and different contexts (inter-organizational processes). It is demonstrated that this framework and associated methodology can be used as a basis for establishment of the semantic interoperability of systems in supply networks.

"Two things fill the mind with ever-increasing wonder and awe…
the starry heavens above and the moral law within"

Immanuel Kant

Foreword

It was just last year that Nina asked me what I actually do in my office. Since children's innocence and purity of perception deserve honesty and clarity, I stayed mostly speechless, angry after many failed attempts to put together the possible answers in my head. After the (more or less successful) search in the theater, music, marketing, programming, management and policy expertise, is my research career just a manifestation of the (early) middle-aged crisis or is it, in fact aligned and focused reflection of the many lived joys of acknowledgement and creation?

It's special, because it is the activity whose basic tool is doubt, where ignorance is a blessing, because it drives your hunger for knowledge. It's difficult, because under the increased weight of vanity that everyone carries within himself, it inevitably pulls you into the abysses of mediocrity. It's responsible, because it creates less and inspires more, because it dissolves despondency and makes you and all under your glass bell better than you think you are. Science is not something you do, it is something you preach. This is what I learned from the best, and so different.

Traja is and will remain an advisor, collaborator and friend to self-destruction. In a way antipode to researcher stereotype (which itself is often reflection of the prejudices of the ignorant), with his energy and commitment he showed (to those with wide open eyes) that science work is a sweet and passionate need, more than some tangible or intangible outcome.

From cocky Frenchmen, one could learn a lot about unconditional love, whether it is about wine, women, nation or profession. And unconditionality means honesty. Hervé is one of those rare and valuable people, whose lines between ethics and bigotry are sometimes almost unrecognizable. His pride is nourished by doubt, the most effective scientific method, unknown to vain ones.

And Nina, Matija and Jelena? They, more than anyone else deserve truthful and complete answers to their questions. As long as this work does not consume but creates energy, all of it I dedicate to them.

Niš, Serbia, 12.6.2012

Chapter 1: Overview of the PhD Thesis

Development of the Formal framework for semantic interoperability in supply chain networks

Abstract. Main objective of the thesis is defined as the development and verification of a formal framework for representation and reasoning of knowledge in supply networks. This framework is based on a neutral specification of existing reference SCOR model for definition of supply chain processes. It is characterized by a modular ontological environment, developed on the different levels of abstraction (from implicit formalizations of the enterprise information systems, to explicit domain ontologies) and different contexts (inter-organizational processes, enterprise goals, etc.). It is demonstrated that this framework and associated architecture for its implementation can be used as a basis for establishment of semantic interoperability of systems in supply networks.

1 Description of problem

Supplier and client collaboration is the main factor of enterprise competitiveness in a modern economy. This is particularly important for small and medium enterprises which success is based on the number of supply chains in which they concurrently participate.

To some extent, the level of enterprises' collaboration can be measured by the level of integration of their Enterprise Information Systems (EIS). Integration enables exchange of messages, automation of the business transactions, integrated view to supply chain operations, etc. By integrating EISs of the different enterprises, the boundaries of the conventional enterprises are erased, while focal partner of the supply chain is described by the notion of Extended Enterprise (EE).

However, EIS integration has certain negative effects on the enterprise flexibility. Integration assumes fixed agreements on the message formats, interfaces and other types of technological commitments which implementation is costly and time consuming. Hence, these agreements are made and valorised only in a small number of partnerships of one enterprise.

Today, EIS research community is showing increased interest in the system interoperability. In contrast to system integration, which basically deals with formats, protocols and processes of information exchange, the objective of interoperability is to have two systems invoking each others functions or exchanging information with the consideration that they are not aware of each others internal workings. Furthermore, interoperability aims at correct and complete reasoning on the meaning of the information which is exchanged between two systems. Hence, it is sometimes called "semantic interoperability". Main tools for implementation of the semantic interoperability are ontologies, languages for ontologies' representation, inference tools (engines) and semantic applications.

Increasingly important role in the implementation of the interoperable systems is given to a domain ontology – explicit representation of the specific domain knowledge (e.g. about Supply Chain Management), namely its concepts and logical relations between those. Domain ontology ensures the correctness of the inference on the meaning of the information which is being exchanged. Thus, it has to be: a) expressive (to contain all concepts from one domain and all their relations); b) explicit (to uniquely define all concepts and their relations); c) neutral (to define all concepts objectively, independently from the specific context); and d) relevant (in the sense that there is a consensus in the domain community about used conceptualizations).

Given that domain ontology is a main interoperability facilitator of arbitrary EISs' interoperability, it is obvious that its relevance is the most important feature. Lack of relevance is a weakness of all existing efforts in definition of the supply chain ontology, such as TOVE, The Enterprise Ontology, IDEON, etc. All these ontologies are created in isolation, by applying an inspirational approach, from the scratch, while their verification is performed only in small number of cases.

2 Objectives and content of research

By considering the above definition of the problem, main objective of the research work is set. The main objective is:

the development and verification of the relevant formal framework for representation and reasoning of knowledge in supply chain networks.

An original approach is set to address the identified problems, with general objective to fulfil the conditions for interoperability of systems in inter-organizational environments.

In order to achieve the relevance, ontological model is developed as neutral specification of the existing, widely accepted reference model for definition of processes in supply chains (SCOR – Supply Chain Operations Reference). This model is verified, widely used industrial standard and it describes the processes, activities, good practices, systems and metrics in supply chains. However, it does that in implicit way – it describes the concepts of the supply chain and their relations by using natural language. In order to preserve the integrity of the reference model, and hence, the compatibility of the resulting formal framework with existing systems based on SCOR, first, this implicit model is described by using Description Logic, namely OWL (The Web Ontology Language) language. In the process of the analysis and synthesis of the implicit model, the explicit model of the supply chain operations is developed – a micro-theory which consists of the common, general enterprise terms and their relationships. The concepts of the explicit and implicit model are related by logical correspondences – SWRL (The Semantic Web Rules Language) rules. Finally, in the process of semantic reconciliation between explicit model and corresponding concepts from the existing domain ontologies, an expressivity of the ontology framework is increased.

In this way, an integrated and modular ontological framework is developed. Its modules are characterized by different levels of abstraction. These levels range from implicit knowledge of the supply chain operations to intermediary micro-theory which is result of the implicit knowledge's semantic analysis and synthesis and domain ontologies, in which the concepts of this micro-theory are defined in different contexts. This framework is the basis of the semantic layer, which can be unconditionally exploited by all enterprises from the supply chain network. In this layer, each of the enterprise is represented by its operations, implicitly described in corresponding EISs. The ability of the individual enterprise to interoperate is directly related to completeness and correctness of the logical relations between those representations and the ontological framework, described above. In order to facilitate establishment of these relations, implicit descriptions of the individual enterprises' operations need to be formalized by so-called local ontologies.

Besides ontologies, semantic layer consists also of semantic applications, which are shared resource of all enterprises in the supply chain network. Their role is to support the collaborative activities and functions of the network, such as the management of inter-organizational processes, partner selection, management of use of shared resources, etc. For fulfilment of these roles, each of the semantic applications exploits the individual application (or problem) ontology – formal representation of the individual problem. In the scope of this research, the semantic applications for supply chain process configuration and execution of the semantic queries on the integrated ontological framework are developed.

The last research topic which is addressed in the scope of this work is related to the aspects of the functionalities and technical implementations of the semantic layer, foundational element of the architecture of semantically interoperable EISs in supply chain networks.

2.1 Research questions

The main groups of research questions which are set for the purpose of the work described in this thesis are:

- Which scientific fields and topics are relevant for achievement of the set objectives and what is the state-of-the-art of these fields, with focus on specific, identified topics? Are achieved results in the identified scientific fields and topics arguable? Are there any gaps identified in each of the scientific fields and topics in the context of the set objectives?
- Given the answers to the questions above, the following questions related to the formalization process are asked: What are the main principles for the development of a formal model which may facilitate a semantic interoperability in a supply chain environment? What are the most suitable method and/or approach to its development? How will this model fit into the formal description of the semantic interoperability of systems?
- Which software services, applications, components and associated assets must be developed in order to become possible to exploit the formal framework for seman-

tic interoperability of the systems in supply chain? How they will be configured? What is the level of human involvement in the process of making two systems semantically interoperable?
- Can the described approach be used to deliver some realistic practical benefits for the collaborative enterprise? How?

3 Methodology

Main result of the research work presented in this thesis is the formal model of the supply chain networks, namely, ontological representation of the knowledge about supply chain networks.

In development of this model, a bottom-up approach is applied. Approach includes: 1) analysis of the implicitly defined knowledge of SCOR reference model, namely induction of the relevant enterprise notions; 2) synthesis of the aggregates of the induced notions; and 3) verification of completeness and integrity of the knowledge models, which is performed by identifying and analyzing logical relations between the concepts of the resulting ontological model and existing domain ontologies and enterprise models (e.g. TOVE, The Enterprise Ontology, CIMOSA, etc.). In development of this model (especially in the step of the synthesis), existing efforts in developing so-called foundational ontologies are taken into account.

Verification of the formal model quality is performed by mapping its concepts with local ontology – implicit model of the enterprises' resources. Ontological representation of the implicit model of the enterprise resources is generated by the developed method for semantic analysis of the database schema of selected ERP (Enterprise Resource Planning) system. This method enables mapping of the ER (Entity-Relationship) syntax and structural patterns to elements of the description logic, by exploiting the expressivity of the relevant languages (OWL).

Finally, experiences from the research described above and analysis of the current state-of-the-art in relevant scientific fields will contribute to specification of the architecture of semantically interoperable EISs.

4 Overview of the research results and the thesis' content

The problems described at the beginning of this Chapter are discussed in this thesis from three perspectives: existing relevant work, formalization and implementation.

First, it is shown that existing research results in the identified fields do not provide enough evidence that semantic interoperability of systems (especially in collaborative environments, such as supply chain) can be achieved. While most of the work is focused to achievement of the interoperability of systems (actually, in most cases, certain levels of systems' interoperability), the semantic interoperability must be considered as a new, under-developed scientific topic. In Section 4 of Chapter 2, the attempt to formalize the notion of semantic interoperability is made. This attempt clearly distinct between the notions of semantic and "traditional" interoperability, by taking into account the unified view to the interoperability of systems, presented in Section 2 of

Chapter 2 and available formalisms for conceptualization of the systems' semantics, described in Section 3 of Chapter 2. Formal definition of semantic interoperability of systems provide the basis for this approach and has significant influence on the choices made in the process of development of methodology for this work.

In order to implement and evaluate semantic interoperability, enterprises' realities have to be represented by relevant formal models. In Section 5 of Chapter 2, the existing work on developing different formalisms for enterprise modelling (enterprise architectures, frameworks and ontologies, database schemas) is presented.

Realization of the interoperability value proposition has great impact to the development of new forms of the enterprise collaboration. These forms and associated notions are defined in the Section 6 of Chapter 2, in the context of the issues related to the conventional enterprises' networking, namely Supply Chain Management. There are already some existing formal models of the collaborative enterprises' environment. However, these models are not considered as candidate ontologies for formal framework for semantic interoperability in supply chain networks, mostly because of lack of relevance. Thus, still there is a need for expressive, explicit, neutral and relevant formal model which will enable the partnering enterprises, namely their EISs to exchange the information and services in the supply chain.

The approach to the development of this model is described in Section 1 of Chapter 3. The approach is based on the premises that: 1) expressivity of one model can be achieved by selecting adopted and affirmed industrial reference model for a semantic analysis; 2) explicitness of one model can be achieved by mapping induced enterprise concepts to the formally defined concepts of domain or upper ontologies; 3) neutrality can be achieved by semantic enrichment, namely, synthesis of the recognized concepts; and 4) relevance can be achieved by maintaining the mappings between formal definitions of the enterprise concepts and implicit notions of the reference models, used or exploited by the relevant communities or EISs. The Supply Chain Operations Reference (SCOR) model is selected as a referent model and is described in Section 2 of Chapter 3. In the following Section 4, a formal representation of the implicit SCOR model (SCOR-KOS OWL), its semantic enrichment (SCOR-Full) and the process of mapping its concepts (namely, their explicitation) to the common enterprise notions (represented by the concepts of OWL representation of the selected domain ontology) are described. Finally, in Section 5 it is shown how this modular ontological framework can be exploited for the purpose of achieving the semantic interoperability of systems in supply chain environment.

A formal perspective to the semantic interoperability of systems in collaborative enterprise environments is complemented with the developed implementation approach, namely, architecture of the semantic layer. The approach builds upon the current trends of defining the Interoperability Service Utilities (ISU), presented in Section 1 of Chapter 4. However, it reconsiders some of the ISU conceptual directions in the context of differences between the "simple" and semantic interoperability. The approach is based on the methodology used for definition of the formal ontological framework, in the sense that it identifies Semantic Interoperability Service Utilities (S-ISU) components, certain functional and conceptual levels of the components; it relates these components to specific implicit or explicit formal models and it formally describes these components, associated assets and their inter-relations by corresponding meta-model, namely S-ISU Ontology. This meta-model is presented in Section 2 of Chapter 4. The core services of S-ISU architecture, namely Transformation and Semantic Querying services are realized and described in more detail. They are based on the approach which assumes the formalization of the implicit sources of enterprise knowledge, such as database schema, into so-called local ontologies.

Finally, in the Chapter 5, the evidence on feasibility of presented approach is provided. In two case studies, it is shown that formal framework for semantic interoperability in supply chain networks can be used for reasoning on the configuration of inter-enterprise processes (Section 1) and for retrieving relevant information from heterogeneous information sources (Section 2).

Final conclusions are presented in Chapter 6. This chapter also lists explicit answers to the research questions, set to facilitate the work on the research, presented in this thesis. It also uses gained experiences to define some research directions and topics which may have an impact to bringing currently only assumed benefits of semantic interoperability to reality.

Chapter 2: Theoretical Background

for development of the Formal framework for semantic interoperability in supply chain networks

Abstract. In this Chapter, a theoretical background for the work carried out in the scope of this thesis is given. It includes presentation of state-of-the-art in the selected relevant scientific fields, namely enterprise interoperability, conceptualization and ontologies, semantic interoperability, formalisms for Enterprise Modelling and inter-organizational networks and Collaborative Networked Organizations. Focus of the literature review is made at the key conceptual findings of the relevant works, while technical perspectives are only shortly addressed. These findings are discussed in the context of development of the formal framework for semantic interoperability in supply chain networks.

1 Introduction

General objective of the work presented in this thesis is to contribute to the achievement of semantic interoperability of systems in inter-organizational environments. In order to achieve this, first, it was necessary to determine in which scientific fields, sub-fields and topics, relevant research results are reported.

These results are discussed in this Chapter, the gaps are identified in the context of the set research objectives and discussion is provided. These are considered as a theoretical background for achievement of the above mentioned objective.

The following research topics are selected and considered as relevant for the work, performed in the scope of this thesis:

- Interoperability. Although the work focuses on semantic interoperability of the EISs, here it is considered in a whole. Unified and integrated view to an interoperability as a problem provides the opportunities to position the presented work and determine its contribution and impact on the development of this scientific topic.
- Conceptualization and ontologies. Interoperability cannot be achieved without previous agreements about the conceptual models which unhide the tacit and implicit knowledge of the enterprises. These agreements are made by conceptualizing the relevant domains of discourse, where ontologies are used to specify these conceptualizations.
- Semantic interoperability. Semantic interoperability is a novel concept. It needs to be clearly differentiated from "simple" interoperability. Hence, an attempt is made to make the existing definitions of the semantic interoperability formal.
- Formalisms for enterprise modelling. The practical benefits of the semantic interoperability can be achieved only if industry adopted models of the enterprises are used as formalisms. Are there good candidates for formal enterprise models which can be semantically interoperable? What's missing? These are the research ques-

tions which need serious attention in order to determine if it's possible to bring the value propositions of the semantic interoperability to a reality.

- Inter-organizational networks. Once the above mentioned propositions are made realistic, the one discipline which will receive probably most benefits is the Supply Chain Management. Although the paradigm of supply chain already evolved towards the notions of Collaborative Networked Organizations, Virtual Enterprises, and others, there are many societal, organizational and technical challenges at different levels, which are not yet resolved. Thus, collaborative enterprises are still suffering from the decreased flexibility, namely, capability to simultaneously manage their performances in more than one supply chain (or Virtual Enterprise) in which they are the partners. Since interoperability has great impact on the reduction of the costs and efforts made in the relationship management in inter-organizational networks, the need for its implementation is evident.

1.1 Overview of the literature

In synthesis of the relevant researches, following sources of information are used: scientific papers, position papers, journal articles, technical reports of the different working groups, organizations, associations and projects' deliverables and different web sites. Based on citation analysis, different authors' work is followed for the selected scientific disciplines.

For example, most of the background on enterprise interoperability and architectures is provided by David Chen and work of Interop Network of Excellence, followed by the findings of the EU projects, such as IDEAS, ATHENA and COIN. In the fields of conceptualization and ontologies, the most influential work is considered from the authors such as Nicola Guarino, Martin Hepp and Michael Grüninger. Finally, in the field of collaborative networked organizations, the works of European ECOLEAD project, Luis M. Camarinha-Matos and Bernhard Katzy are heavily referenced.

The total of 176 citations is made to the sources of the scientific knowledge about the relevant topics. 77 citations are made to the papers published in 30 respectable international journals, such as Computers in Industry, International Journal of Advanced Manufacturing Technology, Journal of Intelligent Manufacturing, Lecture Notes in Computer Science, Information Systems, Enterprise Information Systems and others. 32 citations are made to different reports, reference model specifications and white papers. 41 citations are made to the proceedings of the international conferences, symposiums and workshops. 23 books' or books' chapters' citations are made. In addition, the work of 40 EU funded projects is referenced.

1.2 EU Framework programme perspectives on research of the enterprise interoperability and collaborative enterprising

In this section, a short overview of the EU funded research about enterprise interoperability and collaborative enterprising and related opportunities is presented. Its purpose is to give an overview of the relevant scientific topics from the perspective of

EU needs, as well as to provide information to interested reader about possible cooperation opportunities in this topic. This overview is focused to EU research funded by the Framework Programme (FP6-FP7) for Research and Technological Development of the European Commission.

Five major building blocks of FP7 are the Specific Programmes: Cooperation, Ideas, People, Capacities and Nuclear Research. The core of FP7, representing two thirds of the overall budget, is the Cooperation specific programme. It fosters collaborative research across Europe and other partner countries through projects by transnational consortia of industry and academia. Research is carried out in ten key thematic areas: Health; Food, agriculture and fisheries, and biotechnology; Information and communication technologies; Nanosciences, nanotechnologies, materials and new production technologies; Energy; Environment (including climate change); Transport (including aeronautics); Socio-economic sciences and the humanities; Space; Security. The content of the funded research is typically defined by the high level objectives of the specific programmes, which are mapped to the research priorities and challenges. The latter are defined by the Work Programmes, published by the European Commission, for one or two year period. Based on the challenges defined by the work programmes and specific interlinked objectives of each of the challenges, a content of the calls for proposals for this period is planned. For example, **Fig. 1** illustrates the objectives of the ICT Challenge 1: Pervasive and Trusted Network and Service Infrastructures.

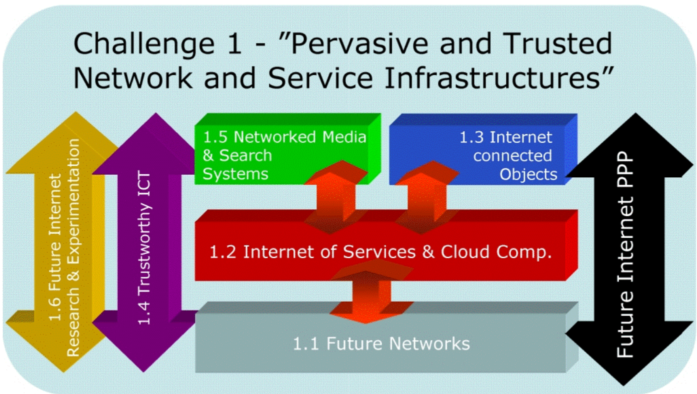

Fig. 1. The specific objectives of ICT Challenge 1

The content of the work programmes shows that enterprise interoperability is currently researched at the implementation level, where mostly technical paradigms are developed on the top of the current Internet infrastructure, to enable the adaptation and implementation of the conceptual frameworks, developed in the past. These paradigms are described in Section 2.2. The vast majority of the currently funded relevant projects fall into the objectives of the ICT Specific Programme Challenge 1. “Pervasive and Trusted Network and Service Infrastructures”, and “Factories of the Future” cross-thematic Coordination between ICT and NMP (Nanosciences, Nanotechnologies, Materials and new Production Technologies) Specific Programmes.

In this thesis, the results of the selected FP7 funded projects are referenced and used in defining the research state of the art. The following projects are considered as highly relevant:

- An interoperability service utility for collaborative supply chain planning across multiple domains supported by RFID devices (ISURF) (ICT-2007.1.3 ICT in support of the networked enterprise)
- Envisioning, Supporting and Promoting Future Internet Enterprise Systems Research through Scientific Collaboration (ENSEMBLE)
- Collaboration and interoperability for networked enterprises (COIN) (ICT-2007.1.3 ICT in support of the networked enterprise)
- Supporting highly adaptive Network enterprise collaboration through semantically enabled knowledge services (SYNERGY) (ICT-2007.1.3 ICT in support of the networked enterprise)

In addition, some FP5 and FP6 projects are also referenced.

Some future directions of the research of enterprise interoperability may be defined by recently launched relevant projects, such as:

- Enabling business-based Internet of Things and Services - An Interoperability platform for a real-world populated Internet of Things domain (EBBITS) (ICT-2009.1.3 Internet of Things and enterprise environments)
- Internet of Things Architecture (IOT-A) (ICT-2009.1.3 Internet of Things and enterprise environments)
- Innovative networks of SMEs for complex products manufacturing (NET-CHALLENGE) (NMP-2008-3.3-1 Supply chain integration and real-time decision making in non-hierarchical manufacturing networks)
- Virtual Enterprises by Networked Interoperability Services (VENIS) (FoF-ICT-2011.7.3 Virtual Factories and enterprises)
- Innovative End-to-end Management of Dynamic Manufacturing Networks (IMAGINE) (FoF-ICT-2011.7.3 Virtual Factories and enterprises)

While enterprise interoperability is still getting a big attention of the European Commission (at least from the technological perspective), the research of collaborative organizational forms was mostly funded at the beginning of this century. Most of the relevant results in this area were developed in the scope of the projects from the VOSTER (Virtual Organizations Cluster) cluster.

Scientific and technological objectives of the VOSTER initiative were: to consolidate relevant concepts and relationships, types, features and indicators of the virtual organizations; to identify and recommend the approaches for modelling the virtual organizations; to identify relevant technologies and standards and to evaluate their potential use in virtual organizations; and to define the functional perspective of the virtual organizations' infrastructure. VOSTER cluster encompassed following FP6 projects:

- ALIVE (Working group on Advanced Legal Issues in Virtual Enterprise), 2001-2002,
- BAP (Business Integrator Dynamic Support Agents for Virtual Enterprise), 2000-2002,
- COVE (COoperation infrastructure for Virtual Enterprises and electronic business),
- E-Colleg (Advanced Infrastructure for Pan-European Collaborative Engineering), 2000-2003,
- eLegal (Specifying Legal Terms of Contract in ICT Environment),
- EXTERNAL (Extended Enterprise Resources, Network Architectures and Learning), 2000-2002,
- GENESIS (Global Enterprise Network Support for the Innovation Process), 2000-2002,
- GLOBEMEN (Global Engineering and Manufacturing in Enterprise Networks),
- ISTFORCE (Intelligent Services and Tools for Concurrent Engineering), 2000-2002,
- NIMCUBE (New-use and Innovation Management and Measurement Methodology for R&D), 2000-2002,
- OSMOS (Open System for Inter-enterprise Information Management in Dynamic Virtual Environments),
- PRODNET II (Production Planning and Management in an Extended Enterprise),
- SYMPHONY (A dynamic management methodology with modular and integrated methods and tools for knowledge-based, adaptive SMEs), 2001-2004,

- PRODCHAIN (Development of a decision support methodology to improve logistics performance of globally acting production networks), 2002-2004,

Besides the VOSTER cluster, there were other projects, related to the consolidation of knowledge about the virtual organizations, such as: VOmap (Roadmap design for collaborative virtual organizations in dynamic business ecosystems) and ECOLEAD.

2 Interoperability

Despite the continuous developments of standard ICT and organizational infrastructures, enterprises will definitely continue to have mixed environments for the foreseeable future. First, many businesses have very specific requirements which cannot be handled by the "standard" hardware and software systems. Second, the move to the new platforms needs to be gradual and evolutionary, because of the (sometimes, critical) changes this move implies and businesses' needs to leverage existing investments. Themistocleous et al (2001) revealed that 38 percent of companies are not replacing their legacy systems when they implement an ERP system. Following to this, they also found that 58 percent of companies did not succeed to integrate their ERP systems with existing legacy systems. Sprott (2000) attributed this "to differences in semantics and business rules between different applications that were never intended to collaborate".

Despite the decrease in operational costs and complexity, it is unlikely that many organizations will be able to have completely homogenous systems environment. Thus, interoperability becomes very important requirement for the systems architecture. In general, it is considered as the ability for two systems to understand one another and to use functionality of one another (Chen et al, 2008).

2.1 Definitions of interoperability

ISO/IEC 2382 defines interoperability as the "capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units".

In more broad sense, IEEE (IEEE, 1990) defines interoperability as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged". Here, interoperability should not only be considered as a property of ICT systems, but it also considers the business processes and the business context of an enterprise. Therefore, interoperations are meaningful, only when all levels of an enterprise are taken into account. Hence, the diversity, heterogeneity, and autonomy of software components, application solutions, business processes, and the business context of an enterprise must be considered.

From the systems perspective, interoperability refers to the ability of heterogeneous, autonomous EISs to perform interactions (exchange of information and services)

(Chen and Vernadat, 2004). In this case, two systems function jointly and give access to their resources in reciprocal way. Interoperability is related to the federated approach, which implies that systems must accommodate on the fly in order to interoperate – no pre-determined assets are assumed.

Interoperability may be considered and evaluated on multiple levels, such as: information/data, services, processes, systems, enterprise models, enterprises and communities. Each of the levels is characterized by the specific challenges. For example, data formats without semantics are the main issue of information interoperability; static definition is serious restriction for services interoperability; lack of correspondences between standard and implemented models and realities poses the challenge for enterprise models interoperability. Although each of these challenges can be associated to a particular level of interoperability, they cannot be addressed in isolation. Namely, EISs capture implicit knowledge of the enterprise; systems are exposed by their services, which are then used to exchange information through enterprise or cross-enterprise processes. Thus, only holistic approach to enterprise interoperability can produce the knowledge and associated assets for realizing its value proposition.

2.2 Scientific topics relevant for research of enterprise interoperability

Research of Enterprise Interoperability involves a mix of relevant scientific topics, each of which has its own state-of-the-art. For example, state-of-the-art for Interoperability architecture approaches of InterOP Network of Excellence, addresses (Berre et al, 2004):

- Interoperability architectures;
- Model Driven Development (as a bridge to the areas of Enterprise Modelling and Ontologies, but also as a foundation for explicit system models, and Model Driven Architectures as an approach for achieving interoperability);
- Service-Oriented Computing (as extension of the area of Web Services);
- Component-oriented and message-based computing (as an implementation foundation for areas such as Service-Oriented Computing);
- Agent-oriented Computing;
- Business Process Management and Workflow; and
- Non-functional aspects of systems, with respect to interoperability, such as security, trust, quality of service, etc.

State-of-the-art in enterprise modelling techniques and technologies to support enterprise interoperability of ATHENA Project addresses (Dietz, 2004): enterprise frameworks and languages; industry initiatives and, standardization efforts; and enterprise modelling languages.

Recently, some novel paradigms related to so-called Future Internet emerged. The Future Internet is a summarizing term for worldwide research activities dedicated to further development of the original Internet and is endorsed by NSF[25] and EC[26], as

[25] http://www.geni.net/
[26] http://cordis.europa.eu/fp7/ict/fire

such. Regarding the current status of Future Internet research, it seems too early to identify any technical consensus or even standardization steps. Therefore, this term should be used with caution only, especially not as a specific technology but instead as an abstract reference to the visible, worldwide activities in this direction[27].

While Internet of Services and Internet of Things are often considered as some of the key features of the Future Internet, the Future Internet Enterprise Systems (FInES) emerged as a field of activity that aims at enabling enterprises to exploit the full potential of the Future Internet. Interoperability is considered as one of the main facilitators of those paradigms.

The Internet of Services is a part of the vision of the future internet where everything that is needed to use software applications is available as a service on the Internet, namely, the software itself, the tools to develop the software, and the platform (servers, storage and communication) to run the software. Cloud computing is a relatively new model of Internet-based computing, whereby servers, storage, networking, software, and information are provided on demand. Advantages of the "Internet of Services" include the little upfront investments to develop an application and the possibility to reuse or build upon other users' efforts. The risk involved in pursuing new business ideas is decreased, and might lead to more innovative ideas being tried out in practice.

Internet of Things (Ashton, 2009) is defined (Vermesan et al, 2009) as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols. In Internet of Things, physical and virtual 'things' have identities, physical attributes, and virtual personalities and use intelligent interfaces. In the Internet of Things, 'things' are expected to become active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging information 'sensed' about the environment, while reacting autonomously to the 'real/physical world' events and influencing it by running processes that trigger actions and create services with or without direct human intervention. Interfaces in the form of services facilitate interactions with these 'things' over the Internet, query and change their state and any information associated with them, by also taking into account security and privacy issues.

2.3 Interoperability frameworks

The main purpose of interoperability frameworks is to provide an organizing mechanism so that concepts, problems and knowledge on enterprise interoperability can be represented in more structured way (Chen et al, 2008). Typically, as seen in many works, this mechanism provides different perspectives to the problem of interoperability, such as conceptual, organizational and technical. Then, these perspectives are used to analyze the interoperability of the different business entities, such as enterprise, process, system, function, data, etc.

[27] http://en.wikipedia.org/wiki/Future_Internet

In this thesis, following frameworks are referenced: LISI, IDEAS, ATHENA and INTEROP NoE. They are shortly described in the subsequent sections.

Levels of Information Systems Interoperability (LISI)

LISI (C4ISR, 1998) (Levels of Information Systems Interoperability) is a maturity model and process, developed for US Department of Defense, for determining the joint interoperability needs, assessment of the systems to meet those needs and selection and implementation of solutions to achieve higher levels of capability to interoperate.

LISI (C4ISR, 1998) Reference model (see **Fig. 2**) is used to review interoperability maturity levels (Process, Applications, Infrastructure, Data) by assessing the capability to interoperate in context of enabling attributes of interoperability, namely, procedures, applications, infrastructure (hardware, communications, security and system services) and data.

Nature of Operational Information Interaction	Corresponding Interoperability Level		Implications			
			P	A	I	D
Cross-Domain Interactive Manipulation	Enterprise	4	Enterprise Level	Interactive	Multiple Topologies	Enterprise Model
Shared Applications & Databases	Domain	3	Domain Level	Groupware	World Wide Networks	Domain Model
Complex Media Exchange	Functional	2	Program Level	Desktop Automation	Local Networks	Program Model
Simple Electronic Exchange	Connected	1	Local/Site Level	Standard System Drivers	Simple Connection	Local
Manual Gateway	Isolated	0	Access Control	N/A	Independent	Private

Fig. 2. LISI Reference model

IDEAS Interoperability Framework

IDEAS Interoperability Framework (IDEAS, 2002) is developed in scope of IDEAS project, the first interoperability initiative in Europe, carried out under FP5. It defines the capabilities to interoperate on different levels, structured into layers of enterprise model (organizational issues, including business and knowledge level) and system architecture (ICT issues, including application, data and communication level). The holistic view on the interoperability is ensured by using semantic models to make correspondences between different models of different levels.

IDEAS Framework considers interoperability at three levels of detail. First, all interoperability concerns are classified into organizational (enterprise model) and architectural (ICT) (See **Fig. 3**).

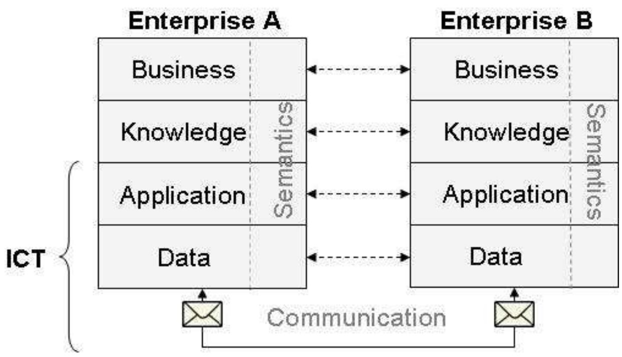

Fig. 3. IDEAS Framework Interoperability levels

Organizational concerns are discussed at the levels of business and knowledge. The interoperability at the business level is considered as organizational and operational ability of an enterprise to cooperate with other enterprises. This level includes the decisional, business and process model. Decisional model defines what/ how decisions are taken and the degree of responsibility of each operating unit, role and position. The business model is the description of the relationships between an enterprise and the way it offers products or services to market. Business processes model defines the set of activities that deliver value to the customers. Interoperability at the knowledge level should be seen as the compatibility of the skills, competencies and knowledge assets of the enterprise with those of other enterprises. Knowledge level includes the models for defining roles, skills/competences and enterprise knowledge assets (procedures, norms, rules and references).

Interoperability at ICT systems level should be seen as the ability of an enterprise's ICT systems to cooperate with systems of other external organizations. In the context of the system architecture, interoperability is discussed on the levels of application, data and communication. Application level includes models for solutions management (tools and procedures required to administer an enterprise system), workplace interaction (interaction of the user and the system), application logic (computation carried out by a system to achieve some result) and process logic (order in which the application is carried out). Data level describes which data is required and produced by the system, by using the models of product data, process data, knowledge data and commerce data.

IDEAS roadmap considers interoperability as significant only if the interactions take place at least on three different levels: data, services and processes, with a semantics defined in a given context (IDEAS, 2007).

ATHENA Interoperability Framework

While IDEAS focuses on structuring interoperability issues, ATHENA Interoperability Framework (Berre et al, 2007) (AIF) aims at providing solutions for those. A common feature of the ATHENA solutions is the fact that they are all model-driven.

The solutions focus on modelling the interactions and information exchanges that occur both on a business level and a technical level. AIF is structured into parts of:

- conceptual integration, which provides a modelling foundation for various aspects of interoperability,
- applicative integration, which provides guidelines and principles for resolving the interoperability issues, and
- technical integration, which provides ICT tools and platforms.

The ATHENA Interoperability Framework adopts a holistic perspective to interoperability by inter-relating three research areas supporting the interoperability of EISs. The three areas are: 1) enterprise modelling (which defines interoperability requirements), 2) architectures and platforms (which provide implementation frameworks), and 3) ontology to identify interoperability semantics in the enterprise. ATHENA identifies the levels where interoperations can take place: enterprise/business, process, service and information/data (see **Fig. 4**). Then, for each of these levels a model-driven interoperability approach is prescribed, where meta-models are used to formalize and exchange the provided and required artefacts that must be agreed upon.

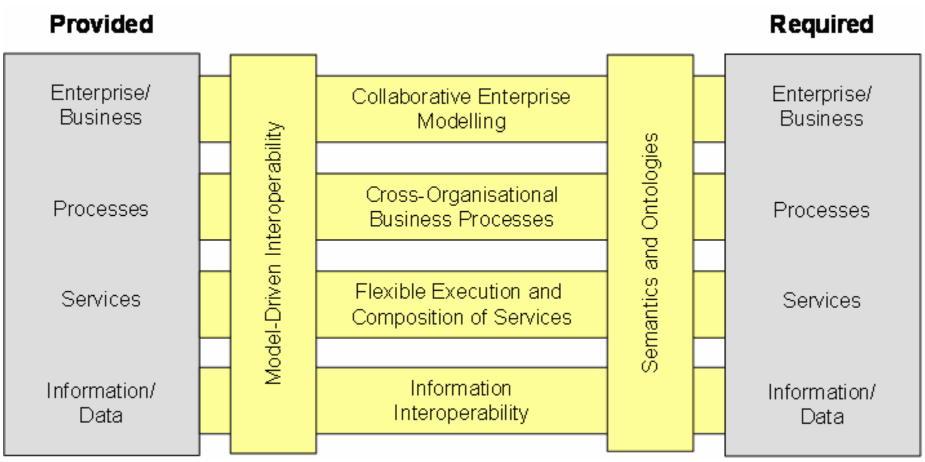

Fig. 4. AIF Conceptual Framework

The applicative integration of ATHENA is based on Enterprise Unified Process[28] (EUP), for modelling the software lifecycle. In this perspective, ATHENA Interoperability Methodology (AIM) for managing the lifecycle of the interoperability project is described, including activities of business collaboration modelling, interoperability maturity analysis, solution mapping and design, implementation, testing, deployment and assessment and project management. The technical framework of the AIF describes an integrated architecture supporting collaborative enterprises. The architecture focuses on a set of tools and infrastructure services to support collabora-

[28] EUP, Enterprise Unified Process, http://www.enterpriseunifiedprocess.com/

tive product design and development, cross-organizational business process, service composition and execution, and information interoperability.

INTEROP NoE Enterprise Interoperability Framework

Enterprise Interoperability Framework (Chen and Daclin, 2006) (EIF) developed within a frame of INTEROP network of excellence follows barriers-driven approach to define the domain of enterprise interoperability and identify and structure knowledge (solutions) of the domain using the framework.

Identified barriers are classified into syntactic and semantic differences of exchanged information (conceptual), incompatibility of the information technologies (technical) and incompatibility of organization structures (organizational barriers). Then, the barriers may be discussed in the context of each of the interoperability levels, defined by ATHENA: business, process, service and data, so solutions can be identified. These two dimensions define the enterprise interoperability domain (see **Fig. 5**).

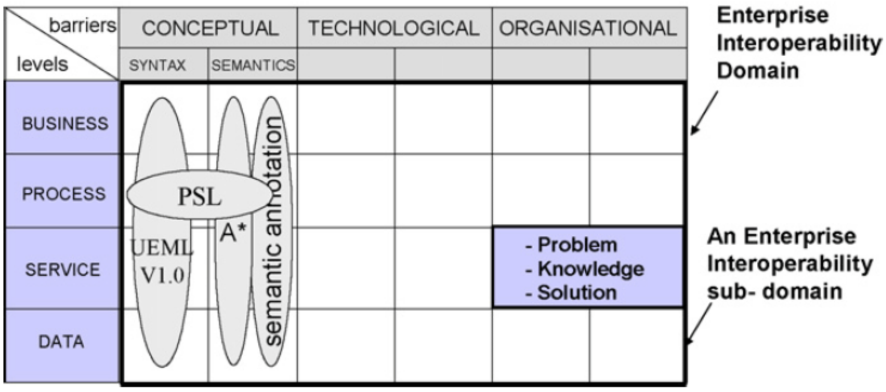

Fig. 5. Use of the framework to define the domain and to structure knowledge

The third dimension (interoperability approaches) is added to the two-dimensional framework. This third dimension allows categorizing knowledge and solutions relating to enterprise interoperability according to the ways of removing interoperability barriers.

Other frameworks and initiatives

Besides the enterprise and production areas, interoperability initiatives are carried out in other fields, such as e-business, e-health, e-government and others. Some of them are E-Health Interoperability Framework (NEHTA, 2005), The European Interoperability Framework (COMPTIA, 2004) and others.

2.4 Technical issues for enterprise interoperability

Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. The perceived value of SOA is that it provides a framework for matching needs and capabilities and for combining capabilities to address those needs. While both needs and capabilities exist independently of SOA, in SOA, services are the mechanism by which needs and capabilities are brought together (MacKenzie et al, 2006). The key technical background for SOA is provided by the technology of Web Services.

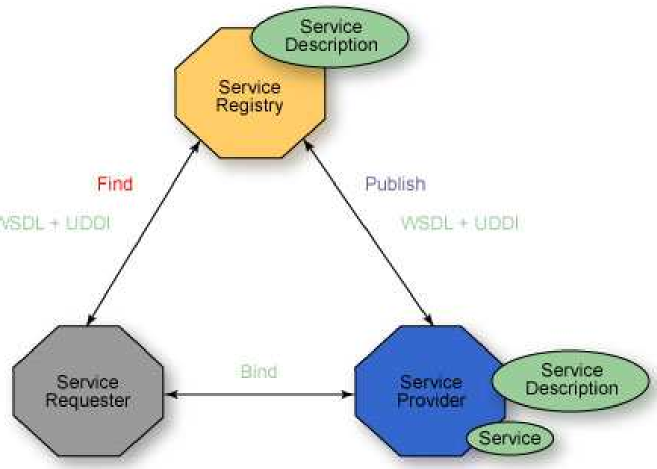

Fig. 6. Service-Oriented Architecture

SOA is the form of organization of integrated enterprise application environment, characterized by supply, demand and usage of its distributed functions, implemented by services. It enables a concept of uniform tools for exposition, discovery, interaction and usage of individual business functions in context of fulfilment of defined objectives.

Growth of internet, electronic business (B2B), as well as supporting protocols and standards, and in particular - XML (eXtensible Markup Language), motivated the development of technical solutions for exposition of business functions in wider context, even publicly. Today, primary tool for enterprise collaboration, as well as integration of its internal business functions, are web services – basis of SOA infrastructure. Basic standards for realization of web services are WSDL (Web Services Description Language) (Christensen et al, 2001), used for definition of structure of service - its „contract“; and UDDI (Universal Description Discovery and Integration), which prescribes the methods, principles and guidelines for management of service registry. BPEL (Andrews et al, 2003) (Business Process Execution Language) language for process modelling is a tool for orchestration of web services. It facilitates

Business Process Management (BPM) – area of ICT application in definition, simulation, execution, optimization, evaluation and control of business processes.

While SOA is a candidate approach for an exposure of the enterprise systems' functionality, ATHENA project and INTEROP NoE suggest that Model-Driven Architecture (MDA) is a candidate for the design and development of the interoperable systems architecture.

MDA is an approach for the software development, developed by Object Management Group (OMG), in 2001, which uses different models of software requirements to support software engineering (Model-Driven Engineering, MDE). It defines system requirements by using Platform-Independent Model (PIM), expressed in Domain-Specific Language (DSL). Then, given a Platform-Definition Model (PDM), the PIM is translated to one or more Platform-Specific Models (PSM), which can be executed by computers (See **Fig. 7**).

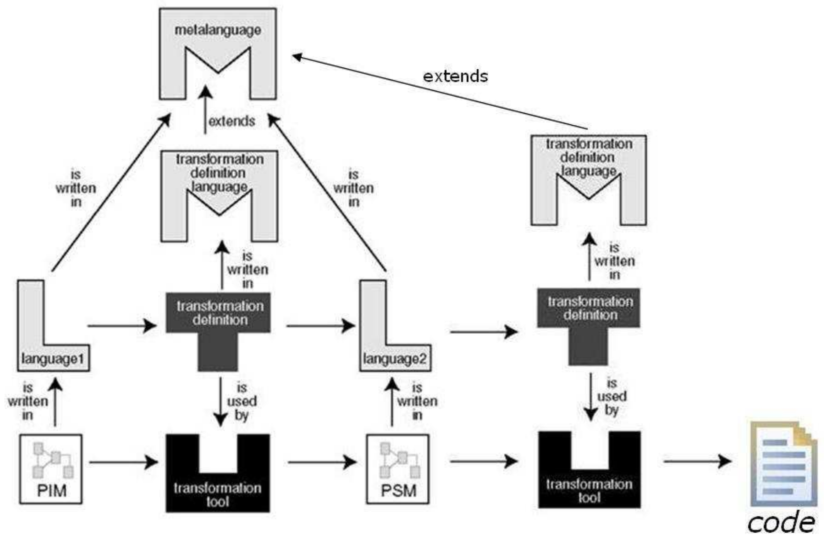

Fig. 7. MDA Framework

It is expected that MDA will evolve to cover the level of enterprise modelling and interoperability requirements. Some efforts in this direction are already reported (Chen et al, 2008) by ATHENA project and INTEROP NoE.

3 Ontologies

There is an agreement in the research community that ontologies need to be used for reconciliation of the interoperating systems. Even so, there are opinions that the main conditions for achievement of interoperability of the loosely coupled systems are: 1) to maximize the amount of semantics which can be utilized and 2) to make it increasingly explicit (Obrst, 2003).

The ontologies are considered as logical theories for formal, explicit, partial specification of conceptualization (Guarino and Giaretta, 1995). The notion of ontology comes from the domain of philosophy. Angeles (1981) defines the ontology as “that branch of philosophy which deals with the order and structure of reality in the broadest sense possible”. Bateman (1995) argues that “the general programme of ontology relies on it being possible to uncover properties that could not fail to be as they are for the world to exist”. Guarino (1995) is more specific and defines philosophical ontology as “the study of organization and the nature of the world independently of the form of our knowledge about it”. This definition separates ontology from epistemology and hence implies independence among those two.

In computer science, ontology is considered (Guarino and Giaretta, 1995) as a (partial) specification of the semantic structure which is defined by the conceptualization process. It is a logical theory that explicitly expresses the conceptualization in some language. In this context, ontology is a specification used for making ontological commitments. Practically, an ontological commitment is an agreement to use a vocabulary (i.e., ask queries and make assertions) in a way that is consistent (but not complete) with respect to the theory specified by an ontology. While conceptualization is language independent, ontology depends on the used language. In this sense, ontology is important for the purpose of enabling knowledge sharing and reuse. The most widely used syntax for representing ontologies today is defined by OWL (W3C OWL, 2009) (OWL 2 Web Ontology Language). The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. The languages are characterized by formal semantics and RDF/XML-based serializations for the Semantic Web. OWL is endorsed by the World Wide Web Consortium (W3C) and has attracted academic, medical and commercial interest.

3.1 Definitions of conceptualization

Conceptualization is a decision process (Guarino, 1998), a view in which knowledge of the studied part of reality, typically available in an implicit and complex form, is reorganized and generalized in different aggregates, for some purpose. In stronger manner, a conceptualization can be defined as an intensional semantic structure that encodes implicit knowledge constraining the structure of a piece of a domain (Obitko, 2007). In the latter definition, intensional semantic structure refers to the sufficient and necessary conditions for classification of the aggregates’ individuals.

Conceptual models range in type from the more precise, such as the mental image of a familiar physical object, to the abstractness of mathematical models which cannot be visualized in mind. They can be developed in different levels of abstraction of a single domain (Zdravkovic et al, 2011). Conceptual models also range in terms of the scope of the subject matter that they are taken to represent. The variety and scope of conceptual models is due to the variety of purposes that people had while using them. The same applies for conceptualization approaches, which are numerous and have been developed in different knowledge domains (LaOngsri, 2009).

According to a definition of Engelbart (1962), developing conceptual models means specifying the essential objects or components of the system to be studied, the relationships of the objects that are recognized, the types of changes in the objects or their relationships which affect the functioning of the system and the types of impact these changes have on the system. Similarly, Genesereth and Nilson (1987) define conceptualization as “the objects, concepts and other entities that are assumed to exist in some area of interest and their inter-relationships”. Both definitions assume extensional character of the conceptualization process, in the sense that they imply that the elements of the mental image of the specific domain are simply enumerated or listed. Some researchers (Guarino, 2007) argue that this contradicts to an intentional character of a human thinking, where the meaning of elements is constituted by their necessary and sufficient conditions.

3.2 Approaches to ontology engineering (conceptualization)

One of the major challenges in the efficient use of computer systems is interoperability between multiple representations of reality (data, processes, etc.) stored inside the systems, or actual representations and reality itself – systems’ users and their perception of reality (Hepp, 2007).

Where latter can be formalized by the domain ontologies, as shared specifications of the domains’ conceptualizations, former relies upon the local ontologies – wrappers for heterogeneous sources of information, business logic and presentation rules.

The top level of abstraction in the conceptualization process is typically described by so-called upper ontologies. An upper ontology (or foundation ontology) is a model of the common objects that are generally applicable across a wide range of domain ontologies. It employs a core glossary that contains, the terms, and associated object descriptions, as they are used in various, relevant domain sets.

Interoperability of information systems depends on the quality and mutual consistency of the underlying ontologies (Smith, 2003). Differences in conceptualizations (or paradigmatic stands) to which ontologies are committed can cause semantic mismatch, and hence, have a negative effect on interoperability. Namely, in ontology development, knowledge workers or domain experts can choose descriptive or prescriptive approach, temporal or static representations, objectivist or subjectivist paradigm, etc. Negative effects of the inconsistent conceptualizations can be reduced by employing additional efforts in mapping, alignment, translation, transformation or merging the corresponding ontologies (Noy, 2004). The above listed methods for making two ontologies interoperable are described in detail in Section 3.3 of this Chapter.

In addition, ontologies may differ by the level of granularity applied in the conceptualization process. Using different levels of granularity is a common approach to engineering of ontological framework. It is applied in building upper ontologies, which often combine continuant, enduring perspectives of reality and concepts extended through time (Grenon and Smith, 2004). Four-dimensional perspective on reality within a single framework can be granularized further to strategic, operational and tactical sub-perspectives. Thus, it also contributes to the development of modular ontological framework.

A variety of granularity levels in an ontological framework extends the scope of inference. Use of modular ontologies also addresses performance issues of the semantic environments because it enables distributed reasoning. Also, it contributes to achievement of the semantic interoperability of systems.

3.3 Ontology interoperability

Many works rely on the assumption that a single ontology is shared by all the participants of the system. However, in the heterogeneous environments, such as inter-organizational networks, this assumption is not realistic anymore. On the contrary, one has to consider that the partners of the networks create their ontologies independently of each other. Thus, most often the ontologies differ. Still, the distinctness of ontologies does not prejudice logical inconsistency of their terms, especially if they focus on different contexts of the same concepts. Namely, ontology is not a tool for checking correctness of reality, but for its explicit representation.

In this section, the problem of ontology interoperability is discussed in context of methods or techniques and processes for making two ontologies interoperable.

Methods for ontology interoperability

To tackle the above problems, research on ontology interoperability proposes several techniques: ontology mapping/matching, alignment, translation, transformation, merging/integrating, checking, evolution/versioning, and mappings management.

Mapping of two ontologies assume that, for each of the entities (concept, relation, attribute, etc.) of one ontology, the corresponding entity in another ontology is found, with the same meaning. Typically, correspondences are 1-1 functions. They can be expressed by logical equivalences, subsumption or sameness relations, assertions of constraints, based on the object properties or identification of rules, with the form of logical implication between the antecedent and consequent statements.

One of the most cited approaches (Kalfoglou and Schorlemmer, 2003) to defining ontology mapping is based on the algebraic definition of ontology. Ontology is considered as a pair O=(S,A), where S is the (ontological) signature, describing the vocabulary, and A is a set of (ontological) axioms which specify the intended interpretation of the vocabulary in some domain of discourse. Typically, an ontological signature is modelled by partially ordered set (Poset). Poset formalizes and generalizes the intuitive concept of an ordering, sequencing, or arrangement of the elements of a set. It consists of a set together with a binary relation that indicates that, for certain pairs of elements in the set, one of the elements precede another. Such a relation is called a partial order to reflect the fact that not every pair of elements need be related: for some pairs, it may be that neither element precedes the other in the Poset.

Ontology mapping is the task of relating the vocabulary of two ontologies that share the same domain of discourse in such a way that the mathematical structure of ontological signatures and their intended interpretations, as specified by the ontological axioms, is respected. Structure-preserving mappings between mathematical structures are called morphisms. For example, a function f of two Posets that preserves the partial order: ($a \leq b$ implies $f(a) \leq f(b)$) is a morphism of Posets. Hence they characterize ontology mappings as morphisms of ontological signatures as follows:

A total ontology mapping from $O_1 = (S_1,A_1)$ to $O_2 = (S_2,A_2)$ is a morphism $f:S_1 \rightarrow S_2$ of ontological signatures, such that, $A_2 \vdash f(A_1)$, i.e., all interpretations that satisfy O_2 axioms also satisfy O_1 translated axioms. In order to accommodate a weaker notion of ontology mapping they also provide a definition for partial ontology mapping form $O_1 = (S_1,A_1)$ to $O_2 = (S_2,A_2)$ if there exists a sub-ontology $O'_1=(S'_1,A'_1)$ ($S'_1 \leq S_1$ and $A'_1 \leq A_1$), such that there is a total mapping from O'_1 to O_2.

In literature, ontology alignment is often used as a synonym for ontology mapping. In some works (Klein et al, 2002), ontology alignment is considered as a process of making two ontologies consistent and coherent, where it is possible that some of their elements will be transformed. Other authors consider an ontology mapping as a morphism which typically consists of the set of functions which assign the symbols used in one vocabulary to the symbols of the other.

When binary relations are used, instead of the functions, then this process is called ontology alignment. Since a binary relation can itself be decomposed into a pair of total functions from a common intermediate source, the alignment of two ontologies O_1 and O_2 can be described by a pair of ontology mappings from intermediate source ontology O_0 (depicted in the figure below). The intermediate ontology O_0, together with its mappings is called the articulation of two ontologies.

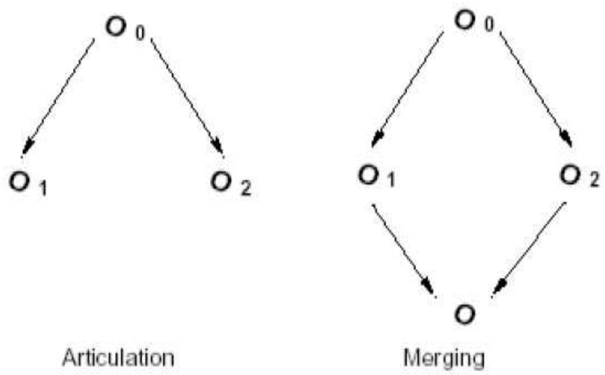

Fig. 8. Ontology articulation and merging

Finally, articulation allows for defining a way in which the fusion or merging of ontologies need to be carried out. The intuitive idea is to construct the minimal union of vocabularies S_1 and S_2 and axioms A_1 and A_2 that respects the articulation. Again, this strong notion of merging can be relaxed by taking the articulation of two sub-ontologies of O_1 and O_2 respectively, and defining the merged ontology O according to their articulation.

Ontology translation assumes that new ontology is created from the existing one, by using different formal languages for expression of the same meaning. The ontology translation is performed when it is necessary to use the ontology in changed circumstances, such as another information system, inference engine, etc.

Ontology transformation is the process in which the structure of the input ontology is changed, while the meanings of its elements remain the same (transformation without semantic losses) or changed (transformation with semantic losses), with objective to use ontology for the purpose which is different from the original one.

Detection of correspondences between two ontologies is performed by calculating the semantic similarities between any of their elements, by using different methods. Each of the methods is suitable for particular circumstances in which ontology mapping is done. Two of the most used methods are analysis of the structural description of the entities and analysis of the terminological description of the entities (based on the lexical or linguistic similarities). Both methods are addressing what is considered as a problem of semantic mismatch.

Semantic mismatch

Semantic mismatch is a difference in the representations of the single entity in two or more different conceptualizations. The semantic mismatch is analyzed on two levels: language and ontology (model) (Klein, 2001).

A language mismatch may be the consequence of using different formalisms in defining the same entity (for instance, OWL and LOOM). This type of mismatch is resolved by ontology transformation. However, sometimes, it is not possible to avoid the semantic losses, because different languages are characterized by the different levels of expressivity. For example, while OIL language can represent the cyclic relation of inheritance, this is not possible with RDFS.

Ontological (model) mismatches occur when two or more ontologies which need to be integrated, describe (partly) overlapping domains. The sources of this type of mismatch are differences in a way one domain (or some of its parts) are conceptualized and explicated (Visser et al, 1997). While conceptual mismatches are differences in a way one domain is interpreted and conceptualized, explication mismatch is considered as a difference in a way the conceptualization is specified.

The conceptual mismatches occur due to different considerations of the ontology scope and model coverage (or granularity). Scope mismatch occurs when two classes, which seem to represent the same concept, do not have same instances, although they intersect. Model coverage and granularity mismatch occurs when there is a mismatch in parts of the domain which is represented by the ontologies or the level of detail to which different ontologies are committed in the representation of the same concepts. In this case, the problem of mismatch is not typically resolved, but two or more ontologies (actually, their overlapping parts) are aligned.

Explication mismatches can be classified into style of modelling and terminological mismatches. The first category of mismatches is harder to resolve and usually involves human work. It occurs when different paradigms (usually for explication of the abstract notions, such as time, action, plan, causality, etc.) or modelling conventions (for example, the choice between extensional or intensional conceptualization) are used in the explication of the conceptualized domain. Paradigm mismatches occur also when different upper ontologies are used for modelling the same domain.

Meanings from ontologies, developed in isolation (assumingly, by using different paradigms), can be reconstructed or re-created by using contextualization or logical theories, such as ontology of descriptions and situations (DnS) (Gangemi et al, 2002). DnS enable the first-order manipulation of micro-theories and models, independently from an upper ontology.

Terminological mismatches occur when 1) individual concept is described by two different names (synonym terms), or 2) the meaning of a term is different in different contexts (homonym terms).

Semantic integration process

The process of semantic integration (CROSI, 2005) is characterized by the set of activities which enable the semantic interoperability of two software systems, based on different local ontologies. The process of semantic integration is illustrated on **Fig. 9**.

The process consists of the activities of preparation for integration (normalization, lifting), similarities discovery, similarities representation and similarities execution.

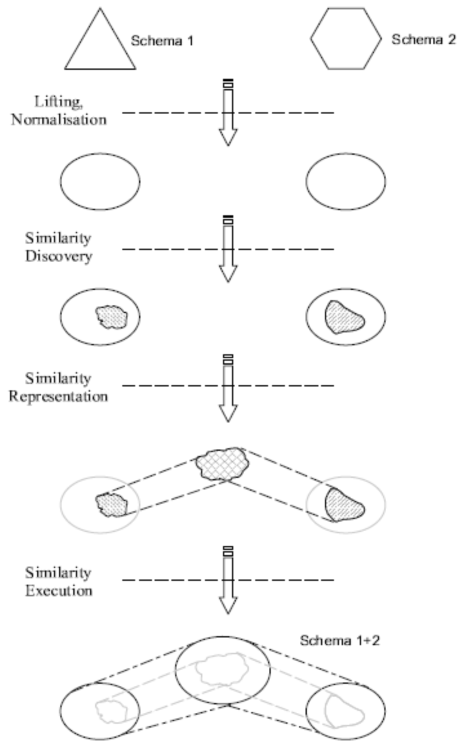

Fig. 9. Semantic integration process

In the preparation phase, ontologies are normalized and uniformly represented, so conflicts due to syntax heterogeneity are avoided. Then, in the phase of similarities discovery, the correspondences between their entities are identified, ranked, evaluated and confirmed. These correspondences need to be represented in a formal way, by using languages, such as RDF, RDFS, OWL or XML, so they can be affirmed. The outcome of the correspondences affirmation can be ontology which merges source ontologies, the set of articulation rules or query rewriting template.

Typically, the process of semantic integration is two-dimensional. Besides the sequential set of activities, it also involves supportive actions and assets (Maedche et al, 2002), related to: 1) evolution, namely, managing the representations of correspondences; 2) building the cooperative consensus on the correspondences; and 3) acquisition and use of domain knowledge in the similarity analysis.

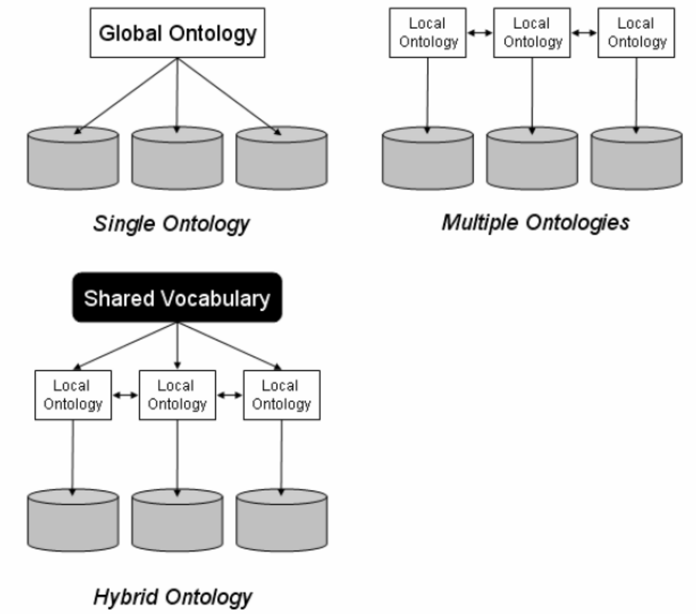

Fig. 10. Architectures for ontology interoperability

Different architectures (Wache et al, 2001) can be used in semantic integration process (see **Fig. 10**). Single ontology approach assumes that a single ontology is used to formalize the semantics of all concepts from the source schemas. Multiple ontologies approach is employed when implicit semantics of each of the source schemas is made explicit in corresponding local ontology. Then, each of the local ontologies is asserted with logical axioms which are used to formalize the correspondences between its and other ontologies' concepts. Finally, hybrid approach is used when shared vocabulary or ontology is used to relate concepts from the local ontologies.

4 Semantic Interoperability

In many interoperability frameworks, the semantic tools, namely, ontologies are intended to be used as facilitators for the interoperability. However, it's very important to distinguish semantically supported interoperability from the semantic interoperability as the latter goes beyond mere data exchange and deals with its interpretation.

Semantic interoperability of systems means that the precise meaning of exchanged information is uniquely interpreted by any system not initially developed for the purpose of interoperation. Thus, it is sometimes called “General Semantic Interoperability”. It enables systems to combine and consequently process received information with other information resources and thus, to improve the expressivity of the underlying ontologies and consequently – to increase the relevance of the data models which are formalized by those ontologies.

Semantic Interoperability is also considered as a synonym for “Computable Semantic Interoperability”. In this sense, it is the ability of computer systems to communicate information and have that information properly interpreted by the receiving system with the same meaning as intended by the transmitting system[29].

Semantic Interoperability, in more general sense, refers to ability of receiving system to correctly interpret transmitted sufficient and necessary information, from sender, but also it is related to awareness and agreement of both actors about their behaviours for given interaction.

Syntactic Interoperability is a prerequisite to semantic interoperability. It assumes that common data formats, languages and structures of the messages are defined, so receiving system may read, interpret and reason about the further processing of the message, based on its structure. In this sense, formats correspond to the protocols used for exchange; languages are related to formalisms used to describe the meanings of the messages; structures are related to conceptualization approach, used to describe the meaning of the concepts from these messages. The specification of this conceptualization, namely, ontology allows all interoperating systems to interpret meanings of terms with precision, by exploiting the message’s terms used in specific contexts, to the ontology elements that describe the meanings of those terms in logical format.

Some researches suggest that upper ontology must be involved in reconciliation of the systems’ semantics. This need is argued by the statement that no single ontology can describe all possible terms related to all possible uses of the different information systems. However, limited set of basic (primitive) concepts may be combined to create the logical descriptions of the meanings of terms used in local or domain ontologies.

Thus, if following assumptions hold true:

(1) the meanings and usage of the primitive ontology elements in the foundation ontology are agreed on, and

(2) the ontology elements in the domain ontologies are constructed as logical combinations of the elements in the foundation ontology,

Then:

The intended meanings of the domain ontology elements can be computed automatically using a reasoner, by any system that accepts the meanings of the elements in the foundation ontology, and has both the foundation ontology and the logical specifications of the elements in the domain ontology.

Therefore:

Any system wishing to interoperate accurately with another system need to transmit only the data to be communicated, plus any logical descriptions of terms used in

[29] http://en.wikipedia.org/wiki/Semantic_interoperability

that data that were created locally and are not already in the common foundation ontology.

4.1 Basics of human communication

In order to better understand the architecture of the semantically interoperable systems, the human communication process is described and some analogies are identified in this subsection.

Besides the intelligent interpretation, one of the key differences in establishing the semantic interoperability for people and systems is the amount of precision that is needed. When two people are communicating, a lot of redundant information is available for confirming assumptions on the meaning of transmitted information and refining of understanding. This redundant information includes facial expressions, tone of voice, repetition using different words and gestures. Also, many terms can be (and are) approximately translated, partially understood in the moment of transmission. This understanding may be improved when context is provided.

Unfortunately, computers typically do not transmit redundant information. They do require precise correspondence of terms, and have an extremely limited set of communication protocols to deploy when communication does not produce expected results. Thus, creating semantic interoperability among computer systems requires significantly more attention to detail than creating it among people.

In a simplified view, the human communication is considered as interplay of 4 physiological and psychological groups of processes: sensation (physiological), perception, cognition and articulation (psychological). **Fig. 11** illustrates this simplified view.

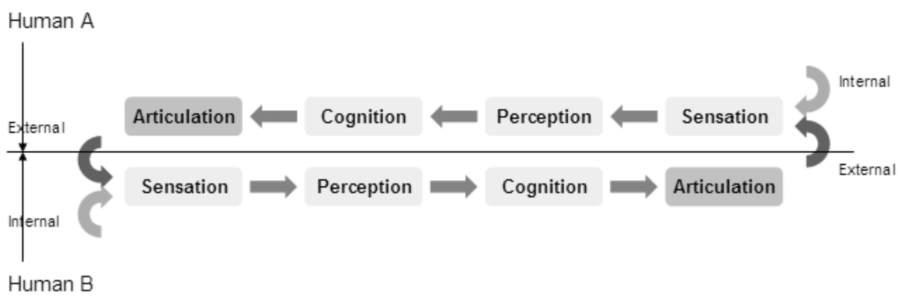

Fig. 11. Illustration of the human communication processes

The human communication typically starts when some stimulus sensory energy is recorded by a human. This stimulus energy is then transformed to electrochemical signals to a human's brain. Thus, sensation is basically physiological process, although it also involves selection of sensations and making decisions on which information is worth percepting.

Then, perception is carried out. It is the psychological process of organizing, analyzing and providing meaning to various sensations. It is reflection of our consciousness and it is carried out in the contexts of expectations from the communication process, experience, culture, etc.

In the next phase, knowledge and comprehension is gained on the basis of provided meanings. This corresponds to the mental processes of cognition. The processes of cognition include reasoning, problem solving, imagining and maybe the most important mental process – conceptualizing. In the process of conceptualizing, attributes or features (can be general, defining or characteristic) of provided meanings are combined into concepts. In psychology, a concept is a thought or idea that represents a set of related ideas, or attributes. In the process of conceptualization, a set of concepts is combined into larger units. Concepts are combined into propositions; when multiple related propositions are clustered, a mental model is built; finally, mental models are combined into schemas. Schemas are basic units of understanding that represent the world.

4.2 Architecture of the semantically interoperable systems

Analogies of the human communication processes with the semantic interoperability of EISs can be used to classify the technologies, tools, models and other artefacts, necessary for outlining the basic architecture for semantically interoperable systems. This basic architecture outline is illustrated at **Fig. 12**.

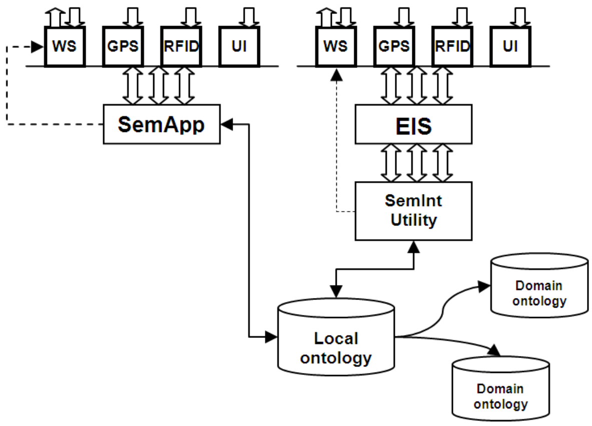

Fig. 12. Basic architecture of the semantically interoperable system

Sensation processes of the human communication correspond to receiving raw information which needs to be processed in the consequent phases, so it can be percepted and understood by the systems. Thus, sensation technologies and tools may be the facilities for sensing the analogue signals: conventional telecommunication facilities, such as phone or fax; cameras and microphones; different types of sensory equipment; RFID interrogators (RFID) and GPS devices (GPS). Examples of the sensation tools that can receive digital signals are: user interfaces (UI), software agents, web services (WS), even database triggers. As it is shown on **Fig. 12**, these signals can be received by EISs or dedicated Semantic Application (SemApp).

Once the information is acquired or received, it is being percepted. This typically means that the attempt to attach semantics to acquired digital information is made.

Thus, some operations, such as translation, data mapping or ontology matching with this information are performed. These operations are using existing knowledge to make the acquired information explicit. This knowledge is formalized by enterprise architectures and models, goal models, dictionaries and taxonomies, other reference models, etc. In case where the signal is received by EIS, this process is carried out by Semantic Interoperability Utility, which maintains the correspondences between implicit semantics of EIS (or its database) and local ontology. Otherwise, it is carried out by the Semantic Application.

In the subsequent process of cognition, an attempt to analyze and understood acquired information (now, with the attached semantics) is made. This step may involve activities such as trial-and-error, root-cause or impact-difficulty analysis. It also involves storage of the percepted and analyzed information. The analysis outcomes are typically related to answering the questions such as: What is the impact of the received information? What kind of response is needed? Which messages should be articulated and sent to which actors? This analysis should be carried out by SemInt Utility or SemApp, on the basis of the business rules, which are formalized in the local ontology.

Finally, a decision is made about the appropriate feedbacks to the received, processed and understood information. Each of those feedbacks is articulated to a digital message which is sent to the recipient. Three types of decisions are made in this case: the decision on the content of the message (for example, manufacturing or procurement order), its structure and its format (such as XML message, SQL statement, etc.). Again, feedback is articulated by Utility or SemApp, which directly deliver the appropriate response, through the web service interface.

4.3 Definition of semantic interoperability

In semantically interoperable systems, there is no need for any kind of data structures or meta-information which is typically used to assign values so the receiving system can understand the meaning of those values. Instead, exchanged information is considered as a logical statement or a set of the logical statements which describe the semantics of the message from one system to another. When OWL language is used, these statements correspond to subject-predicate-object triples.

The differences between the "traditional" interoperability and the semantic interoperability research also arise because of lack of abstract, formal descriptions of semantic interoperation, independently of implementation details (Lee et al, 1996). In research reported in this thesis, the formal definition (Sowa, 2000) of semantic interoperability by John Sowa[30] is adopted. Also, it is represented in a formal way so it can be used to evaluate semantic interoperability of two enterprise systems:

"A sender's system S is semantically operable with a receiver's system R if and only if the following condition holds for any data p that is transmitted from S to R: For every statement q that is implied by p on the system S, there is a statement q' on the system R that: (1) is implied by p on the system R, and (2) is logically equivalent to q. The receiver must at least be able to derive a logically equivalent implication for every implication of the sender's system.".

[30] The IEEE Standard Upper Ontology web site. http://suo.ieee.org

This definition is represented in controlled natural language, as asymmetric logical function semantically-interoperable(S,R):

```
data(p) ∧ system(S) ∧ system(R) ∧ semantically-
interoperable(S,R) ⇒
∀p (
(transmitted-from(p,S) ∧ transmitted-to(p,R)) ∧
∀q(statement-of(q,S) ∧ p⇒q) ∃q’(statement-of(q’,R) ∧
p⇒q’ ∧ q’⇔q)
)
```

Here, systems S and R are represented by the so-called local ontologies.

In this work, the following assumption has been made: when two local ontologies of two corresponding systems are mapped to the same domain ontology, these systems will become semantically interoperable (see **Fig. 13**). In other words, if there exist two isolated EISs S and R and corresponding local ontologies O_S and O_R and if there are mappings M_{SD1} and M_{RD1}, established between the concepts of O_S, O_R and domain ontology O_{D1}, respectively, then there exist mappings M_{SR} which can be inferred as logical functions of M_{SD1} and M_{RD1}.

Obviously, the assumption of semantic interoperability depends on the accuracy and completeness of the mappings. In the Semantic Web environment, these mappings can evolve in interest-driven activities, thus, increasing the information fluidity over the World Wide Web (Jiang et al, 2006).

Within the single enterprise, different systems may implement different functions of the enterprise. Thus, their conceptual models describe the enterprise in the specific contexts (C_1-C_n)

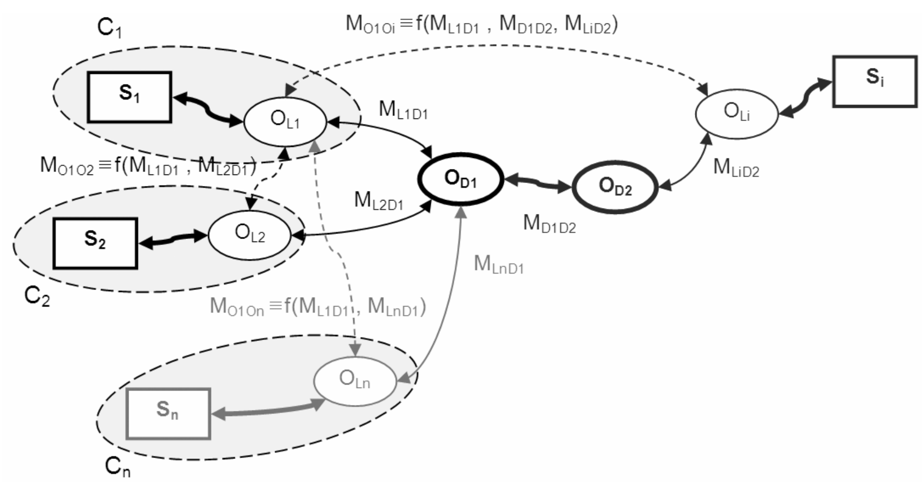

Fig. 13. Semantic interoperability of systems.

Local ontologies are considered as the models of implicit enterprise knowledge. This knowledge is made explicit and hence, machine-processable, when implicit terms of the local ontologies are logically related to appropriate enterprise conceptualizations (e.g. standard models), represented by domain ontologies. Furthermore, each of the local ontologies may represent one of the contexts of the enterprise. Hence, the isolated systems become not only interoperable, but also more expressive, as they become capable to exploit enterprise knowledge, represented by the different local ontologies. Expressivity can be improved further when focal domain ontology is related to another domain ontology in the same manner. This approach may be exploited for the benefit of assertion of the enterprise knowledge by using different conceptualizations, encoded in the different domain ontologies.

4.4 Local ontologies

While realities of the particular domain can be formalized by the domain ontologies, their representations by the corresponding EIS relies upon the local ontologies – wrappers for heterogeneous sources of individual enterprises' information, business logic and rules. The local ontologies formalize the implicit data from the heterogeneous sources in order to facilitate the semantic interoperability of the systems which store this data.

In order to cope with the implicitness of semantics of the enterprises' realities, the following assumptions are made (Zdravkovic et al, 2011) in this thesis for the purpose of defining the source of this semantics, and consequently, building the local ontologies:

- enterprises' realities are represented by the corresponding EISs, and
- enterprises' message models are based on EISs' data models, represented implicitly in their databases.

Hence, the database-to-ontology method is employed in order to transform implicit Entity-Relationship (ER) models to explicit OWL representations, namely, local ontologies.

Then, these local ontologies can be mapped to a common, shared knowledge of the enterprise collaboration environment, namely, different domain ontologies, developed for different contexts. Each of the contexts corresponds to a domain ontology, whose concepts are logically related to the concepts of the local ontologies. Thus, domain ontology becomes a dictionary – a common knowledge of particular enterprise perspective one can use to query the hidden, implicit knowledge stored in EISs, so single, integrated access to the multiple contexts of the particular enterprise concept becomes possible.

The above assumptions about correlations between local ontologies and ER models are made for the purpose of making the process of local ontology creation – automatic. Otherwise, the precondition for this process would be a detailed analysis of the involved EISs. Example of the work which follows this approach can be found in (Castano and De Antonellis, 1998). The authors "analyzed the process descriptions for the aspects related to information and operation similarity, to evaluate semantic correspondences between processes and identify activity replication and overlapping, as well as for the aspects related to interaction/cooperation, to evaluate the degree of coupling between processes and identify the type and the nature of exchanged information flows".

In the work carried out for this thesis, the range of semantic interoperability is clearly set to EISs. The semantic interoperability of the enterprises is considered as more complex problem and is not addressed explicitly, but, to some extent can be derived to the semantic interoperability of systems.

The conceptualization of the enterprises' information systems is made also on basis of the business logic, which is hidden in the actual code, in most cases, and data model, represented by the corresponding relational database structure. Obviously, business logic which is encapsulated in the EIS' will remain hidden – only underlying data model is exposed by ontology. The exceptions are database's triggers, which can be considered as business rules, if they are not implemented only to enforce referential integrity of the database.

4.5 Semantics in Entity-Relationship schemas

Current research and practices of database interoperability are based on the earlier efforts in schema integration. Schema integration typically occurs (Batini et al, 1986) in the context of view integration (during database design) or in database integration (in distributed database management). The process of schema integration implied the development of a single integrated schema – a federal schema (Sheth and Larson, 1990), expressed by using a common data model, for the purpose of integrating the schemas of existing or proposed databases into global, unified one.

The mismatch between the schemas is caused by the fact that a single concept in the universe of discourse is sometimes represented in different ways, while there are also cases where the single representation is associated to the meaning of different concepts. Typically, schema integration assumes that these conflicts are resolved in the process of schema transformation. This process is formalized by McBrien and Poulovassilis (1998). Its outcomes are equivalent schemas, which may then participate in the database federation. While the information capacity of the schemas was considered as the basis for measuring their equivalence, Miller et al (1994) have shown that the problem of inferring the information capacity equivalence and dominance of schemas that occur in practice is undecideable and they have proposed more restrictive notions of equivalence.

It is important to note that most of the approaches to schema integration did not make an attempt to interpret or formalize the implicit semantics of the schemas. Instead, they used a notion of common data model (which does not necessarily reflect an ontological commitment) to enable the federation of databases and thus, to make those interoperable. With the development of the formalisms for semantics representation, the new approaches to database interoperability are increasingly focused to transformation of the implicit semantics of the database schemas to explicit conceptual models. Many researchers have worked on schemas mapping (Rahm and Bernstein, 2001) (Doan and Halevy, 2005) or data integration in ontology (Wache et al, 2001). William et al (1996) considered different groups of semantic relations between schema objects in order to find the corresponding similarities. Zhao and Ram (2007) took into account the instance information in the process of integrating heterogeneous data sources.

In general, the existing approaches suffer of their applicability on existing large data sets. Moreover, the most of these approaches cannot be implemented in real cases because of the large amount of manual intervention. Some of the examples of the existing but practical work in database to ontology mapping are presented below.

Existing database to ontology mapping approaches and tools

Review of the relevant literature reveals several approaches which address database to ontology mapping. In this section, the main features of four distinctive frameworks, made with different objectives, is presented. Also, some gaps are identified, in terms of the selected criteria.

In particular, the focus of this gap analysis is made on how the existing frameworks resolve three specific groups of problems related to database-to-ontology process: 1) semantic interpretation of ER patterns, namely a level of database schema conceptualization; 2) data population, namely ontology concepts instantiation; and 3) use of the framework, namely translation of semantic to database queries. As the latter two are mostly related to the technical challenges, the level of database schema conceptualization is considered as the most important. In this section, also some remarks on the existing approaches, regarding these groups of problems, are provided.

Work on DB2OWL mapping facility is a part of development of a general interoperability architecture (Ghawi and Cullot, 2007) that uses ontologies for explicit description of the semantics of information sources, and web services to facilitate the communication between the different components of the architecture. DB2OWL (Cullot et al, 2007) looks for some particular cases of database tables to determine which ontology component has to be created from which database component. According to these cases, conversion process is performed (table -> class, column-> property, constraint -> relation) where the set of correspondences between database and ontology components is conserved, thus enabling the translation of ontological to SQL queries and retrieval of corresponding entities. However, it remains unclear how this translation will be implemented.

More important, the semantics of existential constraints of the columns and cardinality of relations is not taken into account. The major feature of this approach, as claimed by the authors, is that it aims at separating data mapping from schema mapping. Any data manipulation with a database will not affect the ontology. However, the consistence of two corresponding data and individuals' sets will be maintained by the queries which will populate the ontology with instances at the moment of the semantic query execution. This method is referred to as a query-driven population, in contrast to a massive dump, which maintains the full correspondences between ontology individuals and database table data. The latter approach is taken by the Relational.OWL model.

Relational.OWL (De Laborda and Conrad, 2005) is a candidate for data and schema representation format, relevant for database to ontology mapping, developed with a primary motivation to facilitate data and schema exchange in Peer-To-Peer (P2P) database environment. It provides a meta-model, which describes the components of the relational database. In contrast to DB2OWL, it does not attempt to interpret the semantics of the ER patterns. In does not conceptualize the ER model but only provides its replica. However, it can be used as an intermediary in the process of database to ontology mapping, instead of a document with correspondences, used by DB2OWL. In that sense, it can be considered as a complementary work. Unfortunately, same like DB2OWL, it does not model multiplicity of the foreign keys. Thus, it is not possible to use it to assign source and destination cardinality to OWL properties. Moreover, source multiplicity determines important aspect of the semantics of the underlying concept or database table. Namely, where source multiplicity of the foreign key equals 1, the corresponding OWL relation shall be necessary condition for instantiation of the concept in its domain. This is important semantic feature, because it enables intensional conceptualization of the entity.

Where DB2OWL and Relational.OWL are used to create new ontologies from existing schemas, there are tools that takes different approach by facilitating automatic, semi-automatic or manual mapping between existing ontologies and schemas. In this thesis, the work of Konstantinou et al., and Xu et al is reported.

Vis-A-Vis tool (Konstantinou et al, 2006) uses the Protégé libraries for graphically representing ontology and a database model (MySQL or PostgreSQL) and it facilitates manual establishment of the mappings between those. In this sense, its not relevant to discuss on the level of ER schema conceptualization as it mainly depends on the outcomes of the manual work. The Protégé plug-in allows queries to be asked to the ontology and returns results from the database. Hence, it takes a query-driven approach to instance population. The key motivation of this approach is to keep the instances stored in a database while maintaining a link to the dataset, so ontologies become smaller.

In contrast to Vis-A-Vis which only facilitates manual mapping, D2OMapper (Xu et al, 2006) is a tool for automatic or semi-automatic creation of the mappings between database schema and existing ontology. This work is based on the authors' experience in developing ER2WO (Xu et al, 2004) tool for translating ER schema into OWL ontology. The key motivation of the authors was to develop a framework which will facilitate the generation of ontological annotations for dynamic Web pages, extracted from the database. D2OMapper outputs express the conceptual, in specific element (naming matching) and structural (predefined heuristic rules) correspondences between the schema and ontology. Although it is not explicitly mentioned in the reported work, the purpose of the approach implies that query-driven approach to data population is taken.

5 Formalisms for enterprise modelling

One of the key challenges of the semantic interoperability problem is how to discover and make explicit – the implicit information about the enterprise, or its information system. The research and provided assumptions on the relationships between the actual knowledge about the enterprise and the structure and content of its systems' databases, presented in Sections 4.4 and 4.5 of this Chapter, define the directions for tackling this challenge.

However, for two local ontologies to become interoperable, it is needed to establish the logical mappings between their typically implicit concepts and common domain knowledge. The sources of this domain knowledge are standard enterprise architectures and models. In this sense, both are considered as formalisms, the common sets of concepts and relationships which can be used to model an enterprise.

In this context, standard enterprise architectures provide at least two benefits for the semantic interoperability of systems. First, they, in a lesser formal way, describe an enterprise and thus, provide the dictionaries for developing their more formal descriptions - the domain ontologies, which can be used for interpretation of the concepts of the local ontologies. Second, they align organizational and ICT perspectives of the enterprise, and hence, they may even determine the outline of the enterprise systems' architecture. The latter argument implies that, in some cases, conceptual models used to develop ER schemas of the EISs may, to some extent, correspond to the enterprise architectures and hence, make matching of the local and domain ontologies easier.

5.1 Definitions of the Enterprise Architecture

ISO 15704[31] defines architecture as a description of the basic arrangement and connectivity of the parts of a system (either a physical or a conceptual object or entity). The architecture may be used to guide the implementation of the system, its design and evolution over time. It may also be used to communicate about the system among all of its stakeholders.

Enterprise Architecture (EA) should be organized in a way that supports reasoning about the structure, properties and behaviour of the system (Chen et al, 2008). It defines the components that make up the overall system and provides a blueprint from which the system can be developed. It provides a vision of the future system. EA is seen as a complementary architecture to software architecture, to document system-wide organizational and business context in which software operates.

EA should not be mixed with Enterprise Modelling (EM). EM describes the EA from various viewpoints in detail to allow specification and implementation of the systems. In other words, while enterprise architecture describes the significant characteristics or features of a system, the enterprise models specify in detail the system itself.

According to ISO 15704, there are two types of architectures. While system architectures (Type 1) deal with the high level design of a system, Type 2 architectures are actually frameworks which are used to structure concepts and activities necessary to design and build that system.

[31] ISO 15704, Industrial Automation Systems - Requirements for Enterprise-reference Architectures and Methodologies, 2000

5.2 Enterprise architectures and frameworks

Enterprise architectures emerged in 1980's. Among these earlier efforts, the most known are Computer Integrated Manufacturing Open System Architecture (CIMOSA) (AMICE, 1993), that established the notion of enterprise architecture; ARIS (Scheer, 1994) and Zachman Framework (Zachman, 1996). All these frameworks are of Type 2.

Both CIMOSA and ARIS are process oriented approaches which aim at integrating enterprise and system functions by modelling and monitoring the flow of activities.

Zachman framework structures various enterprise modelling and engineering concepts according to the perspectives of various stakeholders involved in the enterprise engineering. The multiple perspectives are introduced because different stakeholders use different levels of abstraction to describe an enterprise and consider different deliverables. Among the most significant work on enterprise architecture in US, the most known are TOGAF[32], developed by Open Group and DoDAF[33] (Department of Defense Architecture Framework).

Although these early architectures are considered as complementary, they are developed in independent efforts and in most cases, for different purposes. Hence, some amount of redundancies existed and consequently, a need for harmonization became evident. In response to this, IFAC/IFIP Task Force on enterprise integration developed Generalized Enterprise-Reference Architecture and Methodology (GERAM, 1999). GERAM defines basic concepts to be used in enterprise engineering and integration. It harmonizes contributions from CIMOSA, GRAI Integrated Methodology (Chen and Doumeingts, 1996) (GRAI/GIM) and Purdue Enterprise-Reference Architecture (Williams, 1994) (PERA).

Also, Bernus et al (2003) analyzed other frameworks, such as Zachman and DoDAF in context of GERAM, to facilitate better understanding of the similarities and differences of those and others.

Enterprise architectures developed in the past are contextual, in the sense that they reflect the background and purpose of their developers: CIMOSA for computer integrated manufacturing, GRAI for production management, PERA for system engineering, Zachman for information systems and DoDAF for military operations management. GERAM is considered as the best candidate as a reference architecture to which the concepts of these architectures can be mapped, analyzed and compared (Chen et al, 2008).

The main results of the work of standardization bodies, relevant for enterprise architectures and modelling are ISO 15704[34] – Requirements for Enterprise Reference Architecture and Methodologies, and EN/ISO 19439[35] – Enterprise Integration – Framework for Enterprise Modelling, where the latter is considered as implementation of requirements, defined in former.

---

[32] Open Group, http://www.opengroup.org/togaf/

[33] The DoDAF Architecture Framework Version 2.02, http://dodcio.defense.gov/dodaf20.aspx

[34] ISO 15704, Industrial Automation Systems - Requirements for Enterprise-reference Architectures and Methodologies, 2000

[35] EN/ISO I9439, Enterprise Integration—Framework for Enterprise Modelling, 2003

IEEE 1471[36] standard is concerned with “Recommended Practice for Architectural Description of Software-Intensive Systems-Description”. It addresses the activities of creation, analysis and evolution of architectures of software-intensive systems, and description of such architectures. Although the approach is developed for software engineering, its concepts are also relevant for enterprise architecture. For the purpose of architectural descriptions, a conceptual framework is established (displayed at **Fig. 14**).

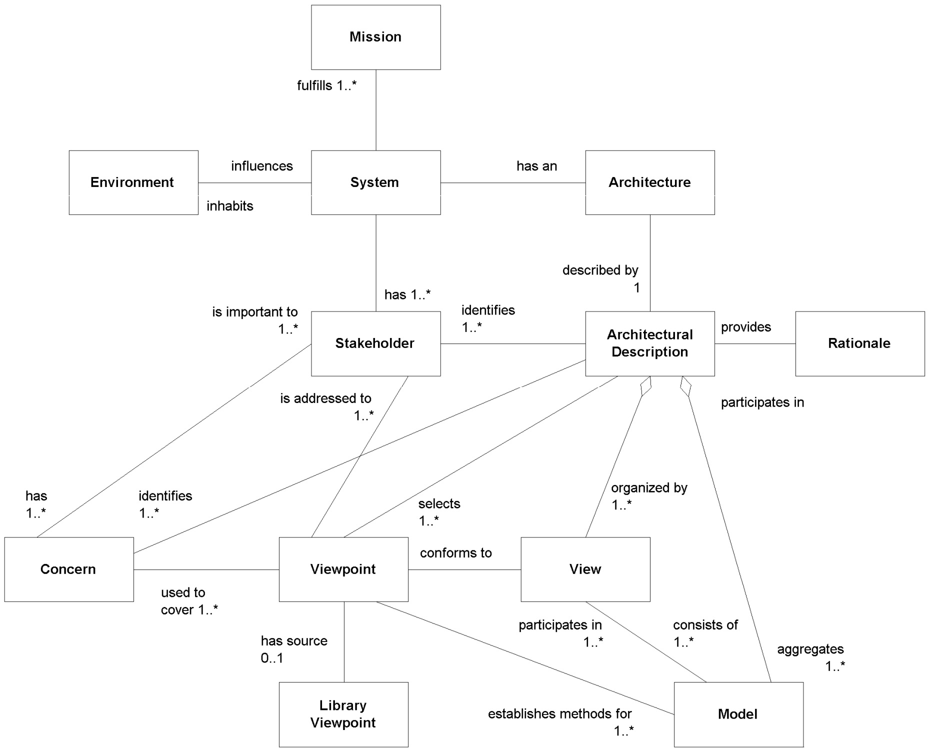

Fig. 14. IEEE 1471 Conceptual framework

Some remarks on the individual concepts of this framework are provided below.

- System. The System could be an application, a subsystem, a service, a product line, system of systems or an enterprise. The system may be man-made or natural. The premise of the standard is that it provides guidance for documenting the system's architecture, independently of the specific definition of system.
- Mission. Most systems exist to fulfil one or more missions, or functions or objectives. The architecture should help the system meet its missions.

---

36 IEEE 1471, Recommended Practice for Architectural Description of Software-Intensive Systems, 2000

- Environment. A system exists within its environment. The system acts upon that environment and vice versa. A system's environment determines the range of influences made by the system and on the system.
- Architecture. Every system has architecture. In fact, a system could have many architectures. In IEEE 1471, architecture is considered as a conception of a system.
- Architectural Description. An architectural description (AD) is a collection of artifacts or work products used to describe architecture. Those descriptions are the primary subject of the standard. Any architecture may be described by one or more description. In the standard, an AD describes exactly one architecture for a system of interest. An AD, per the standard, is made up of various contents: identification of stakeholders, architectural concerns, architectural viewpoints, architectural views and architectural models.
- Stakeholder. A stakeholder is any person, organization or group with an interest in the system. Examples of stakeholders are: architect, designer, client, user, maintainer, auditor, and certification authority. Within the standard, a stakeholder has one or more (architectural) concerns pertaining to the system of interest.
- (Architectural) Concern. A concern is any interest in the system. A concern may be held by one or more stakeholders. Just as an architectural description is a specific representation of architecture, the identification of a system's stakeholders and concerns is a specific representation of its environment in terms of its influences.
- (Architectural) Viewpoint. A viewpoint is a set of conventions for constructing, interpreting and analyzing a view in terms of viewpoint languages and notations, modelling methods and analytic techniques to be used to address a set of concerns held by stakeholders. A viewpoint covers one or more concerns and stakeholders.
- (Architectural) View. A view is a representation of the whole system from the perspective of a related set of concerns. A view conforms to exactly one viewpoint.
- (Architectural) Model. A view is comprised of one or more models. Each model is constructed in accordance with conventions established by the viewpoint. A model may be a part of one or more views. Models are provided for sharing details between views and for the use of multiple viewpoint languages within a view.
- Library Viewpoint. A library viewpoint is one that is predefined (reusable) and does not need to be spelled out within an AD in which it is used.
- (Architectural) Rationale. Rationale captures the reasons why certain architectural choices have been made (such as viewpoints selected for use and architectural decisions).

There was no collaboration between ISO and IEEE during development of the above standards, so it’s necessary to establish mappings between them so to achieve the interoperability between models and systems which are using those standards.

5.3 Enterprise ontologies

Although there are many enterprise modelling frameworks used in an industry, this is not the case with enterprise ontologies. In this thesis, three existing enterprise ontologies are presented. They are developed on different levels of expressivity.

TOVE (TOronto Virtual Enterprise) ontology (Fox et al, 1996) applied a formal approach to enterprise modelling, representing activities, states and time (top-level ontology), organization, resources, products, costs and activity-based cost management. The primary focus of the TOVE enterprise model has been in linking the structure and behaviour through the concept of empowerment – the right of an organization agent to perform status changing actions. TOVE aims at providing sophisticated support to decision making by enabling the inference, not only on basis of what is explicitly stated in the model, but also on the basis of what is implied by the model. It introduces the notion of an ontology competence and corresponding competency questions – the ontology benchmarks, in the sense that the ontology is necessary and sufficient to represent the tasks specified by the competency questions and their solution. Thus, TOVE organizational ontology defines three sets of competency questions: 1) Structure; 2) Behaviour; and 3) Authority, empowerment and commitment competency.

The Enterprise Ontology (Uschold et al, 1998) is a result of the work in development of a method and a computer tool set for enterprise modelling, based on a formal models. It aims at ensuring that all parties, involved in the enterprise have a shared understanding of the relevant aspects. Its role is to act as a communication medium (primarily, but not exclusively – between humans). Secondly, it is intended to assist acquisition, representation and manipulation of enterprise knowledge. Thirdly, it is intended to enable the interoperability, by using the ontology as an interchange format for terms related to business enterprise. The main criteria for selection of the terms were common usage (consensus on the meaning) and avoid of ambiguity. The building blocks on the Enterprise Ontology are notions of an Entity, a Relationship, a State of Affairs and a Role. These are the primitives that are used for expressing the definitions in ontology and they are referred to as concepts of "meta-ontology". They are specialized to the concepts of 4 sections: 1) Activity, Plan, Capability and Resource; 2) Organization; 3) Strategy; and 4) Marketing.

IDEON™ ontology (Madni et al, 2001) is one of the candidate ontologies for modelling collaborative distributed enterprises. It employs four complementary perspectives to capture the key concepts and relationships of the enterprise. First, the enterprise context view represents the interaction between an enterprise and its external environment (partners, customers, competitors, etc.). It introduced the concept of "sensors", employed for observing the environment, enabling the enterprise to act upon the assessment of the observation, with a goal to achieve a specific effect to its environment. Second, the enterprise organizational view captures its inner structure, assigns goals, and selects strategies for their achievement and acts upon them, by employing relevant processes. Third, the process view represents planning-execution-control cycle. Fourth, the resource view elaborates on the various types of resources that might be needed to execute a process.

Table 1 show the comparative overview of the above enterprise conceptualizations, which illustrates used development approaches (purpose, modularization decisions) and implementation features (expressivity, notations, applications).

	The Enterprise Ontology	**TOVE**	**IDEON^{TM}**
Purpose (motivation)	Enterprise modelling	Enterprise modelling	Collaborative enterprise modelling
Domains (modularization approach)	1) Activity, plan, resource, 2) Organization, 3) Strategy, 4) Marketing	1) Structure, 2) Behaviour, 3) Authority, empowerment and commitment	1) Interaction, 2) Organization, 3) Process, 4) Resource
Expressiveness	Low	High	Low
Existing (known) notations	Ontolingua	OWL (Web Ontology Language)	Unified Modeling language (UML), Knowledge Interchange Format (KIF)
Known applications	The Enterprise Toolset (Business Process models, Agent-based architecture for integration)	Integrated Supply Chain Management, Process integration in Enterprise Engineering	Process-centric crisis action planning and execution, Integrated Product-Process Development (IPPD)

Table 1. Overview of the enterprise ontologies

Although these ontologies had some impact to enterprise modelling scientific community and some briefly reported applications, there are no strong evidences of their industry acceptance with their intended purpose. While obvious lack of practical implications can be justified by the technology-related implementation difficulties, it is a fact that many of the existing efforts in development of the common enterprise model are based on an inspirational approach to enterprise modelling, implemented by the groups of experts, not by community (Grubic and Fan, 2010). Moreover, enterprise ontologies are usually created from scratch. As a result, the development and, especially validation processes take a very long time to complete (Yan, 2007), due to a typically large amount of work needed for analysis and synthesis of the domain knowledge, as well as achievement of the consensus on developed conceptualizations within the relevant community.

6 Inter-organizational networks and Collaborative Networked Organizations

Today's globalization of the marketplace and technological innovations are driving increased trend of diversification of market demand. The market dynamics is putting a lot of pressure at long-term planning activities and introduces demand for flexible production environments, capable to quickly and competitively respond to the opportunities.

Recently, it became obvious that single enterprise cannot meet these challenges independently. For gaining the competitive market advantages, Supply Chain Management (SCM) approach to business is widely recognized. Significant number of new production and collaboration concepts has been derived from SCM paradigm. Joint coordination of previously isolated individual functions of an enterprise has been one of the major challenges in exploration of opportunities for improvement of supply chain productivity, particularly since current view of supply chains shifted from linear chains of supplier's supplier to customer's customer into complex networks which entail groups of companies with varying degrees of integration. In order to address this challenge, a concept of Extended Enterprise have been proposed, defined as function of closer coordination in the design, development, costing (Childe, 1998) and the coordination of the respective manufacturing schedules of cooperating independent manufacturing enterprises and related suppliers (Jagdev and Thoben, 2001).

It is expected that Supply Chain Management practices would need to evolve in order to enable efficient collaboration of loosely-coupled, diverse businesses, networking their core competences towards fulfilment of shorter term objectives.

Miles and Snow (1984) introduced concept of external groups, which they called "dynamic networks" – combinations of independent business processes with each contributing what it does best to the network. This concept gained attention of the practitioners and academia and led to a discussion in clarification of a new term - Virtual Enterprise. The main challenge of a Virtual Enterprise formation is to establish optimal balance of dynamic, competent and compatible set of temporary relationships, rather than simply enable collaboration of physical or legal entities. Virtual enterprises are derived from the underlying inter-organizational network – kind of relatively long-term cooperation, in contrast to temporal forms of collaboration it sets up. Network is responsible for preparation, setup and lifecycle management of the Virtual Enterprise.

6.1 Supply Chain Management

Supply chain is a complex, dynamic networked environment which consists of a number of different actors, assets, goals, competencies, functions and roles. The interest in creating a new discipline of Supply Chain Management was developed in the early '60s with the initial motivation to investigate the increase in demand fluctuation (known as "bullwhip effect") which occurred in deeper levels of the manufacturing supply tree (Forrester, 1961). With the development of processing power in the '90s, it became possible to quantify and manage this effect.

However, despite the technology development, it appears that SCM paradigm is adopted unexpectedly slow. Some of the main reasons are: lack of feasible technology support; inconsistency of supply chain and individual enterprises' business strategy; and difficulties in change management, from internal and external perspective. These issues are related to the three pillars of SCM: objectives, IT systems and business functions.

Any inter-organizational collaborative form is characterized by a singular objective, expressing the common interest of involved parties to collaborate. Where supply chain has a singular objective, its actors are individually characterized by different objectives, not necessarily compatible with the cooperative ones. This misalignment may have a negative impact on the capability of an enterprise to act upon its business strategy, when the enterprise is involved in more than one supply chain.

Advances in ICT have great impact on social, economic and technical aspects of doing business. However, rapid progress also resulted in increasing complexity and heterogeneity of systems, having a negative effect on realization of one of the fundamental requirements for ICT applications - enterprise integration capability and interoperability (Panetto and Molina, 2008).

Besides different integration challenges imposed by inter-organizational collaboration requirements, the lack of internal, horizontal integration still remains the issue in many enterprises. Weaknesses of isolated business functions become critical when enterprise-wide information systems, such as ERP system, are implemented. This is evident from the proportion of change management uptake in ERP implementation, in overall, sometimes as high as 70% (Motwani et al, 2005). Using standard processes included in an ERP is considered as valuable implementation tool. These processes are often seen as “best practices” - collective, organized and empirically validated knowledge, enabling increase in company performance, and providing a powerful tool for change management (Grabot, 2008).

Many researches are trying to show that the effective solution for all three classes of SCM problems is related to the use of knowledge-based technologies. Cross-functional, horizontal enterprise integration often relies on the existing body-of-knowledge, commonly represented by standards and reference models. Mainstream research of interoperability of applications focuses on federation, where mapping is done at the semantic level, with the use of interfaces, reference models or ontologies. Finally, the coherence between local and global objectives is enabled by ensuring the consistency of system-wide decision making, a concept of enterprise integration in the frame of enterprise modelling (Vernadat, 2002). For the reasons above, SCM researchers today are shifting towards the exploration of semantic web technologies, based on the use of ontologies.

Industry practice shows that manufacturing supply chains are still primarily focused on a cost reduction as a key aspect of collaboration. The fact that supplier relationship management contributes largely to the overall costs of the supply chains’ final products has great impact to their configuration-related decisions. For example, manufacturers tend to reduce the number of suppliers. Moreover, relationships are dyadic – rarely expanded to include vendors’ vendors and customers’ customers. Also, high level of integration is required in order to reduce costs – manufacturers tend to view their suppliers as extensions of themselves.

Traditional approaches to supply chains’ configuration may have negative impact to their performance. First, high-speed, low-cost supply chains are often unable to respond efficiently to unexpected structural changes in (customized) demand or supply. Second, high level of integration reduces flexibility of small and medium enterprises, main constituents of the lower levels of supply chains, because it assumes fixed agreements on the message formats, interfaces and other kinds of technological commitments which implementation is costly and time consuming. Third, investments in technical framework for enterprise integration, which could maximize the efficiency and productivity, cannot be returned in a short term. Moreover, it is evident that starting collaboration in such traditional settings is reactive and not proactive decision. Namely, relationship establishment or development is motivated by the internal, rather than external factors: complexity and volume of supply relationships, potential for cost reduction (Lamber and Knemeyer, 2004), high frequency of transactions between parties (Jespersen and Larse, 2006), degree of asset specificity (Williamson, 1985), etc.

6.2 Approaches to inter-organizational networking

Traditional ways of organizing enterprises in stable supply chains, based on long-term partnerships will no longer be sufficient in today's global environment (Hamel, 1999). Today, physical boundaries of collaboration are expanded and more open than ever, due to improved visibility of market and accessibility of information relevant for establishment of cooperation. This situation is driving the market towards the vision of global business networking, where enterprise networks would take over the dominant role on the market from individual corporations. Despite the consolidation trend of business acquisitions on the market, it is argued that network of collaborating companies is much more agile than single integrated company (Katzy and Dissel, 2001). Main expectation from the virtual organizations is to behave in agile manner towards market opportunities (Goldman et al, 1995). This also poses necessity for flexible and agile behaviour of involved partners. Thus, SME sector is identified as the most promising for networking. It is expected that future enterprise networks would provide umbrella framework for SME's and perform in competition with other networks, according to development roadmaps and strategic guidance, provided by few remaining global corporations – mainly distribution networks (Katzy et al, 2004).

Overall capability of the network depends on objective and realistic performance, as well as potential to generate a new value in a collaboration process. For this reason, capability of each partner in a network must be described explicitly in a measurable manner, usually by quantification of their capacity potential. Traditional view to enterprise capability, based on its resource-based representation is too implicit for this purpose. Namely, sole availability of particular machine or a tool is not sufficient to clearly determine potential value it could bring in a collaborative effort. In this case, partner selection depends on great deal of other factors, like unit and caution cost, completion probability and past performance (Sari et al, 2006), as well as the others.

Key driving forces for establishment of inter-organizational networks are advances in overall development of information and communication technologies, and growing trends of specialization and outsourcing.

Complexity of the scope of inter-organizational networks as well as collaboration issues cannot be managed without application of advanced ICT technologies. Their primary goal is to provide infrastructure and overall environment for development of virtual cooperative platform for coordination of relevant actors within the network (Felix and Chan, 2005). Main objectives of virtual cooperative platform are to:

- Enable semi-automated or automated selection of competences relevant to meet customer requests, based on transparent, realistic, actual and measurable image of individual capabilities;
- Expose and distribute individual partners' business services throughout collaboration space;
- Coordinate collaborative performance of individual partners' business services within inter-organizational processes.

Another driver for the inter-organizational networking is increased relevance of specialization, both vertical and horizontal. It brings the expertise in better coverage of particular market segment, and enables enterprise to excel in this segment. Also, trend of specialization influenced market differentiation and appearance of new business segments, especially in horizontal direction. One of the examples is growing practice of specializing managerial capability, where management is becoming a service, instead of a position (Katzy et al, 2004).

Relationship of trends of specialization and inter-organizational networking is two-way. Where performance of networks benefits from highly specialized partners who provide the top expertise in handling market opportunity, individual partners' involvement in networking can serve as transitional stage to help them to become leaner, more innovative and responsive.

Direct source of specialization trend is another phenomenon, appearing in the eighties at global market – outsourcing, defined as delegation of non-core operation from internal production to an external entity, specializing in that operation. Zeffane (1995) argues that outsourcing is fundamental argument for inter-organizational networking. In manufacturing, the most frequently outsourced function today is logistics, involving transport, purchasing, inventory control, production planning, warehousing, forwarding and customs brokerage (Berry, 1994).

Success of inter-organizational networks greatly depends on imposition of equal opportunities for all of its partners, whether they are participating on system or human collaboration level. With growth of general interest in knowledge management technologies and appearance of semantic web paradigm, more efforts in exploring human-oriented collaboration services are involved (Lee and Kim, 2007). This is important for networks where diverse levels of members IT maturity is present. For the demonstration of equal opportunities and resolution of possible priority conflicts, transparency of partner's competences information is crucial. Its scope must be managed on a voluntary basis - companies involved in a network must be enabled to gain full control over their sensitive data manipulation and distribution through precisely defined, secured and controlled channels.

Interaction maturity levels

In attempt to clarify various concepts involved in what is considered as inter-organizational networking, Camarinha-Matos and Afsarmanesh (2006) proposed the working definitions of the concepts of networking, coordinated networking, cooperation and collaboration.

Networking involves basic communication and information exchange for mutual benefit. A simple example of networking is the case in which a group of entities share information about their experience on the use of a specific tool. They can all benefit from the information made available/shared, but there is not necessarily any common goal and hence, there is no value generation. In addition to communication and information exchange, coordinated networking involves aligning/altering activities so that mutual benefits are achieved more efficiently.

Cooperation involves not only communication, information exchange, and adjustments of activities, but also resources sharing for achieving compatible goals. Cooperation is achieved by division of some labour among participants. In this case the aggregated value is the result of the addition of individual ‘‘components” of value generated by the various participants. A traditional supply chain based on client-supplier relationships and pre-defined roles in the value chain, is an example of a cooperative process. Each participant performs its part of the job, although it coordinates with others. There exists however, a common plan, which in most cases is not defined jointly but rather designed by a single entity, and that requires some low-level of co-working, at least at the points in time when one partner's results are delivered to the next partner.

Collaboration is a more demanding process in which entities share information, resources and responsibilities to jointly plan, implement, and evaluate a program of activities to achieve a common goal and therefore jointly generating value. It implies sharing risks, resources, responsibilities, losses and rewards. Here, the individual contributions to the value creation are much more difficult to determine. The example of the collaboration process is concurrent engineering, when a team of experts jointly develop a new product.

The concepts of networking, cooperation and collaboration are used to evaluate the maturity of interaction of two enterprises and corresponding levels of their integration (Camarinha-Matos and Afsarmanesh, 2008) (see **Fig. 15**).

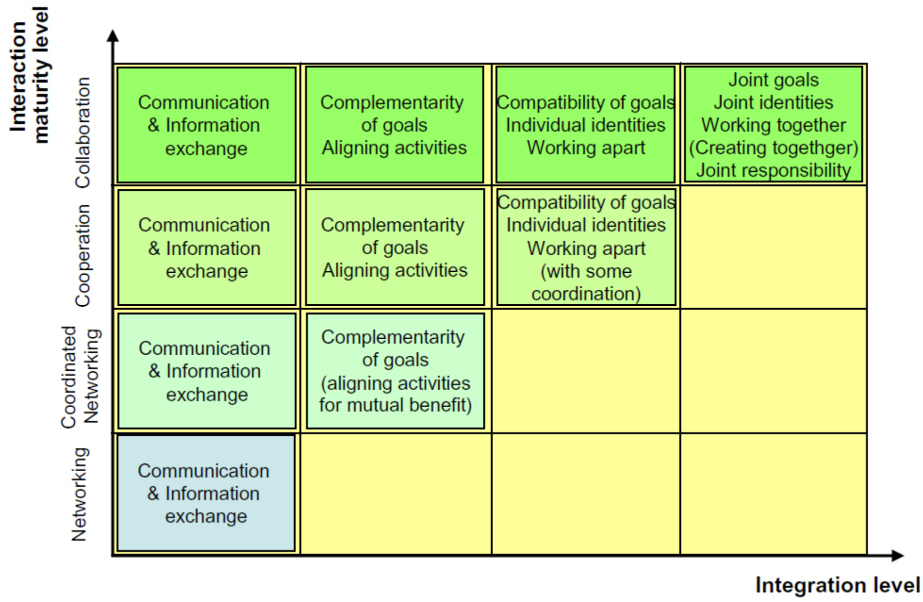

Fig. 15. Interaction maturity levels

6.3 Virtual Organizations and Virtual Breeding Environments

In a response to the issues of static and integrated architecture of the supply chain, and as a result of the research of new approaches to inter-organizational networking, a notion of Virtual Enterprise has been introduced and widely discussed in academic community.

Virtual enterprise (organization) is a temporary network of independent enterprises (organizations), who join together quickly to exploit fast-changing opportunities and then dissolve (Browne and Zhang, 1999). It can be formed as an autonomous market entity with owned product or even within the existing supplier network, as so-called instant virtual enterprise (Grefen et al, 2009). It is characterized by a short-living appearance of a supply chain, capable to produce low volume of high variety of products, by drawing from the loosely-coupled, heterogeneous environment of available competences, capabilities and resources. This environment is sometimes referred to as Virtual Breeding Environment (Sánchez et al, 2005) or Organization (Panetto and Molina, 2008), defined as a pool of organizations and related supporting institutions that 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.

Paradigms of Virtual Enterprises and their breeding environment are based on the capability of an enterprise to configure or reconfigure quickly, according to the circumstances of the market, often not known in advance or even in the moment of configuration. Hence, efficiency and effectiveness of this joint endeavour depends on the interoperability of enterprises, rather than their integration, because the latter implies the number of technical and organizational preconditions, which are not possible or feasible to achieve in a short term.

The core concept of Virtual Enterprise has been addressed by numerous authors, providing different approaches. Virtual enterprises are customer-oriented, focused primarily to single opportunity, in contrast to supply chains which are being built on basis of a market share. They can be formed to perform one-of-a-kind production or service task (Sari et al, 2006) or even to deliver after sales services for a product line (Hamel, 1999). Although Virtual Enterprise is designed to create a value of a business opportunity, it is argued (Katzy and Dissel, 2001) that the value of a Virtual Enterprise is also created within itself, as internal processes and services are adapted to the requirements of short-term business opportunity. Hence, one of the impacts of enterprises' competences restructuring is also stimulation of organizational flexibility, resulting with improved performance in future occasions (Katzy and Dissel, 2001).

Approach to a legal form of organization of the network, as well as derived Virtual Enterprises, is currently not unified, and it is directly related to a level and the scope of coordination. In some circumstances, only small headquarters staffs is required, to deal with administrative details. It is also argued (Katzy et al, 2004) that the role of business brokers or business architects, in charge of order acquisition, network marketing and internal assembly, must be foreseen. Management of the networks will become the responsibility of independent business services brokering companies, fully committed and dedicated to improvement of network's performance.

Classification of the collaborative organizations

Given the large diversity of types of collaborative networks in different application domains, often using different terminologies, it is important to provide the definitions or descriptions of the used terms. Camarihna-Matos et al (2009) provided taxonomy of different collaborative organizational forms and definitions (see **Fig. 16**).

The definitions of following main categories are given:

- Category 1: A collaborative network (CN) is a network consisting of a variety of entities (e.g. organizations and people) that are largely autonomous, geographically distributed, and heterogeneous in terms of their operating environment, culture, social capital and goals, but that collaborate to better achieve common or compatible goals, thus jointly generating value, and whose interactions are supported by computer network.
- Category 2: Supply chain is a stable long-term network of enterprises each having clear roles in the manufacturing value chain, covering all steps from initial product design and the procurement of raw materials, through production, shipping, distribution, and warehousing until a finished product is delivered to a customer. The level of stability of these organizations is being challenged, leading to dynamic supply chains where, for instance, the participants can change more often.

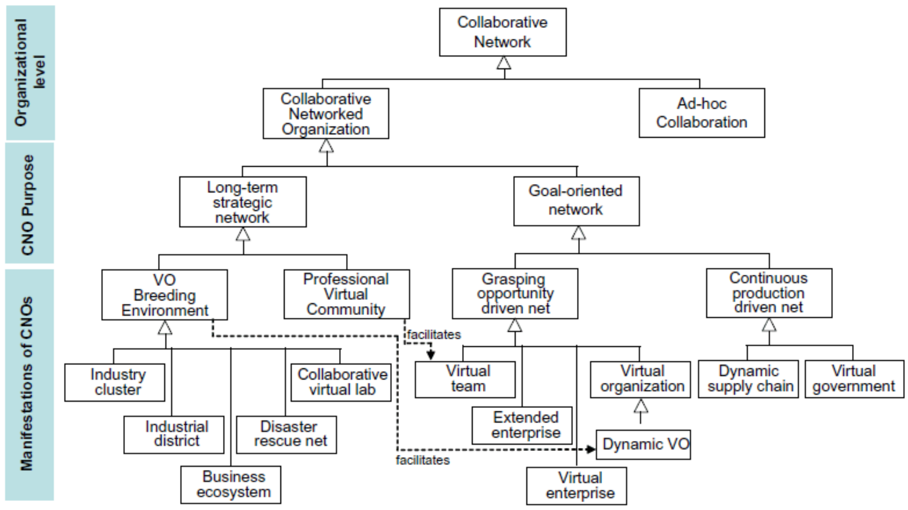

Fig. 16. Taxonomy of collaborative organizational forms

- Category 4: Virtual enterprise (VE) represents a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks.
- Category 5: Virtual Organization (VO) represents a concept similar to a Virtual Enterprise, comprising a set of (legally) independent organizations that share resources and skills to achieve its mission/goal, but that is not limited to an alliance of for profit enterprises. A Virtual Enterprise is therefore, a particular case of virtual organization.
- Category 5.1: Dynamic Virtual Organization typically refers to a VO that is established in a short time to respond to a competitive market opportunity, and has a short life cycle, dissolving when the short-term purpose of the VO is accomplished.
- Category 6: Extended Enterprise (EE) represents a concept typically applied to an organization in which a dominant enterprise "extends" its boundaries to all or some of its suppliers. An extended enterprise can be seen as a particular case of a Virtual Enterprise.
- Category 7: Virtual team (VT) is similar to a VE but formed by humans, not organizations. A virtual team is a temporary group of professionals that work together towards a common goal such as realizing a consultancy job, a joint project, etc., and that use computer networks as their main interaction environment.
- Category 8: VO Breeding environment (VBE) represents an association of organizations and their related supporting institutions, adhering to a base long-term cooperation agreement, and adoption of common operating principles and infrastructures, with the main goal of increasing their preparedness towards rapid configuration of temporary alliances for collaboration in potential virtual organizations.

– Category 11: Professional virtual community is a long-term alliance of professional individuals that provides an environment to facilitate the agile and fluid formation of virtual teams (VTs), similar to what VBE aims to provide for the VOs.

Virtual organizations architectures and frameworks

For enterprise network design and implementation, business process modelling is considered as a fundamental starting point (Vanderhaeghen and Loos, 2007), with business processes as ideal design items. Cooperative process design and management has been dominated subject of research in area of Virtual Enterprise networking for some time, with different approaches.

Virtual Enterprise Chain Collaboration Framework – VECCF (Choi et al, 2006) is developed on the premise that seamless integration of network partners' business processes is indispensable for implementation of a Virtual Enterprise. Therefore, it proposes solution for incompatibility related problems, based on combining elements from existing frameworks and models. VECCF aims at development of value chains within the inter-organizational network by using reference models, provision of independent operational domain for each of the chains, and finally – means for communication between them. It provides new enterprise architecture framework combined from two existing reference models (DoDAF, FEAF) and business process methodology, where SCOR (Supply Chain Operations Reference) model is used, restructured with concept of components. Basic idea of VECCF is to solve the inconsistency of enterprise-owned individual, context-dependent business processes by mapping them semantically to predefined, context-independent, reusable process templates. They encompass and coordinate self-contained business processes or services with predetermined functionality, exposed through particular interface and implemented by specific technology or a standard.

In their work, related to development of Synchronization Point Model (SPM), Perrin et al (2003) explore specific features of cooperative processes, where partners from different enterprises realize atomic and composite activities. They argue that dynamic business process definition and change are crucial for Virtual Enterprises. Main argument for this is uncertainty of the business process structure, due to impossibility to predict and anticipate human collaboration activities and intermediate results exchange necessity, imposed by strong interdependency of partners' parallel work. SPM foresees exposition of internal business processes to networked environment through public abstract definitions of the outcomes enterprise is able to deliver. Usage of so-called process services clearly separates the enterprise public capability from its implementation in order to respect the privacy needs and protect intellectual property. Cooperative process is, then, realized through orchestration of process services, where interactions between each of two or more cooperative process services are coordinated by the Synchronization Point (SP). SP is generic inter-organizational activity which provides facilities for coordination of two process services. It implements project management functions like managing information flows, verification of outcomes, re-planning (dynamic process changes), re-allocation of resources, etc.

While VECCF and SPM focus on the business processes, VEM – Virtual Enterprise Methodology (Sari et al, 2006) takes a different approach in providing a set of guidelines outlining the activities enterprise should consider in relation to managing lifecycle of the Virtual Enterprise. VEM foresees relatively straightforward collaboration scenarios, but focuses on partner selection by examining application of Analytic Hierarchy Process (AHP) and Program Evaluation Review Technique (PERT). AHP is method, developed in early 1970's, for structuring complex, multi-attribute, multi-person and multi-period problem hierarchically. PERT is review technique mainly used to schedule the projects and to cover uncertainty of activity times estimates. Also, VEM adopts multilayer neural network approach for performance assessment of partners for particular tasks.

While above approaches aim at resolution of some specific technical problems (process integration and partner selection), work related to development of Value System Designer framework (Katzy and Dissel, 2001) emphasizes significance of sociological aspects of networking for overall inter-organizational network performance. It identifies a cooperative culture of partners as critical success factor for effective access to individual partners' competences and in general, formation and performance of Virtual Enterprises. The framework is a direct result of converting the sociological research findings into method and supportive tools specification. It encompasses three technical components: integrated tool for business network and inter-organizational processes modelling, performance assessment tool and infrastructure for setup of project-specific IT support.

ARCON (A Reference model for Collaborative Networks)

A large body of theoretical and empiric knowledge related to inter-organizational networking is already available. There is an urgent need to consolidate this knowledge and build the foundations for a more sustainable development of this area. The objective of the ECOLEAD FP6 project was to establish a proposal of the reference model for Collaborative Networked Organizations (CNO) (Camarinha-Matos and Afsarmanes, 2008) – a common basis for understanding and explaining the different manifestations of this new paradigm. This reference model is called ARCON (A Reference model for COllaborative Networks).

ARCON (see **Fig. 17**) takes a holistic approach by combining and aligning the technology and business perspectives, but also by including the other aspects, such as culture, trust and values. It does so from internal (In-CNO) and external (About-CNO) perspectives of the enterprise.

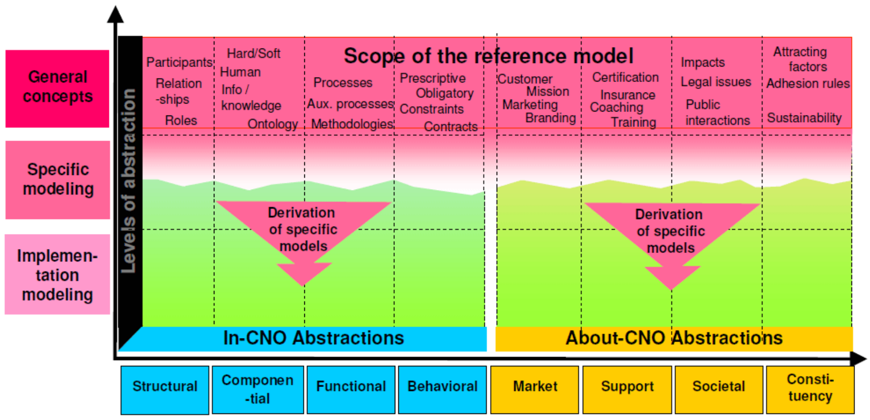

Fig. 17. A modelling framework for CNOs

In-CNO perspective is characterized by four dimensions, as follows:

- Structural dimension addresses the structure of the CNO in terms of its constituting elements (participants and their relationships) as well as the roles performed by those elements and other characteristics of the network nodes, such as the location, time, etc.
- Componential dimension focuses on the individual tangible/intangible elements in the CNO's network, e.g. the resource composition such as human elements, software and hardware resources, information and knowledge. Elemental dimension also consists of ontology and the description of the information/knowledge.
- Functional dimension addresses the "base operations" available at the network and the execution of time-sequenced flows of operations (processes and procedures) related to the "operational phase" of the CNO life cycle.
- Behavioural dimension addresses the principles, policies, and governance rules that drive or constrain the behaviour of the CNO and its members over time. Included here are elements such as principles of collaboration and rules of conduct, contracts, conflict resolution policies, etc.

About-CNO perspective is defined by the characteristic properties that CNO reveals in its interaction with its surrounding environment. The following modelling dimensions are proposed for the external or About-CNO perspective:

- Market dimension covers both the issues related to the interactions with "customers" (or potential beneficiaries) and "competitors". The customers' facet involves elements such as the transactions and established commitments (contracts), marketing and branding, etc. On the competitors' side issues such as market positioning, market strategy, policies, etc. are considered. The purpose / mission of the CNO, its value proposition, joint identity, etc. are also part of this dimension.

- Support dimension considers the issues related to support services provided by third party institutions. Examples include certification services, insurance services, training, external coaching, etc.
- Societal dimension captures the issues related to the interactions between the CNO and the society in general. Although this perspective can have a very broad scope, the idea is to model the impacts the CNO has or potentially can have on the society (e.g. impact on employment, economic sustainability of a given region, potential for attraction of new investments) as well as the constraints and facilitating elements (e.g. legal issues, public body decisions, education level) the society provides to the CNO development.
- Constituency dimension focuses on the interaction with the universe of potential new members of the CNO, i.e. the interactions with those organizations that are not part of the CNO but that the CNO might be interested in attracting. Therefore, general issues like sustainability of the network, attraction factors, what builds / provides a sense of community, or specific aspects such as rules of adhesion and specific "marketing" policies for members, are considered here.

In addition to these perspectives, a CNO model can be defined at multiple levels of abstraction. Currently, three levels are considered in ARCON:

- General concepts level – that includes the most general concepts and related relationships, common to all CNOs independently of the application domain.
- Specific modelling level – an intermediate level that includes more detailed models focused on different classes of CNOs.
- Implementation modelling level – that represents models of concrete CNOs.

Cases of collaborative organizations

Camarinha-Matos et al studied several cases of applied CNO concepts in the manufacturing industry. In table below, some main features of the cases are given. More details about the cases can be found in the referenced paper (Camarinha-Matos et al, 2009).

Table 2. Summary of CNO cases analysis

CNO, Region, Size	Main entities	Industry sector	Business processes	Governance structure	ICT tools
IECOS, Mexico, 30 companies	Virtual enterprise broker, virtual industry clusters, Virtual Enterprises	Manufacturing: metal-mechanic, medical products	- Search and select business opportunities - Project planning - Project execution - Customer follow-up	General director that manages two main groups: engineering Group and brokerage Group	Web site/portal, Automatic diagnosis methodology (for evaluating members), Internal management system (Peñaranda et al, 2006), Administrative system (based on excel sheets)
Virtuelle Fabrik, Switzerland and Germany, 90 companies	Broker, Breeding environment, Virtual enterprise	Manufacturing: design and engineering, metal-mechanic, plastics	- Network development - Order processing - Marketing and sales - Training and further training - Finance and controlling	Five working parties (formed by representatives of each company) executive committee (formed by five members and headed by a chairman).	Web site/portal, Webcorp (Katzy and Ma, 2002) (internal order management system), International portal VF2VF (Huber and Pluss, 2003) (for posting customer demands)
Virfebras, Brazil, 12 companies	VE breeding environment, VE, VE coordinator (defined for each VE)	Manufacturing: mold and dies	- Training and education - Technology set-up - Market strategy - Benchmarking - Identification of shareable resources - Organizational structure - Operation	Directory board composed by a president, a vice-president and a financial responsible.Statute and ethical rules are also defined.	Web site/portal, Virfebras information system (VIS), with two modules (Lima et al, 2004): Marketing information (public online information and order tracking), VEs operation information (only for members)

CNO, Region, Size	Main entities	Industry sector	Business processes	Governance structure	ICT tools
VEN, UK, 250 companies	Associate members, accredited members, professional community members, lead integrators, broker	Manufacturing, Digital industries, Healthcare and bioscience, Food and drink, Chemicals, Construction	- Member-related processes (business health check, workshops, forums, risk management) - Information processes (quotation, exports, getting a product to market, partnering) - Broker processes (broker registration and approval, opportunities registration and assessment) - VENabled^{TM} processes (ICT that support the virtual factory operation, and the marketplace)VENpro^{TM} processes (VEN bid supporting processes and systems)	Advisory board as the final accountable body, with the power to hire and fire service providers to the VEN and sanction or dismiss VEN members.	Web site/portal, VEN main tools: Opportunity management, Virtual factory building process: VENabled^{TM} and VENpro^{TM}, Advanced competency profiling, CRM modules, e-marketplace, ERP and MRP interface, Knowledge management, Networking forums, Access to legal/financial resources
Supply Network Shannon, Ireland, 25 members	Business network (VBE), formed by: companies, development agencies, universities	Engineering and electronics sub-supply companies	Two core activities: training and promotion (marketing and quotation).Three main areas of activities: - Supply chain management (SCM) - Technical issues relevant to engineering and electrical manufacturers - ICT usage	Steering committee (nine members and two development agencies) four subcommittees: marketing, environment, training and projects	Currently SNS have no common ICT infrastructure in place. Simple web site

CNO, Region, Size	Main entities	Industry sector	Business processes	Governance structure	ICT tools
Torino Wireless, Italy, 47 members	VBE network formed by: national and local authorities, social partners, enterprises, universities and financial institutions	ICT (wireless, software, multimedia, technologies, microelectric and optical devices, wire-line technologies)	- R&D - Enterprise acceleration (creation of new entrepreneurship and development of SMEs) - Financial support (stimulating private and public investment) - IPR valorization and technology transfer - Communication and media relations	Torino wireless foundation (Administrative Committee, President, Vice-President, Reviewers College, Ordinary Assembly)	Internet portal with functionalities for: Searching companies members and viewing their profiles, Seeing company news (news can be posted by members), Promoting networking events, Members' area (publication of profiles, news and products launches, press review, access to specialized information)
CeBeNetwork, Germany, 30 members	Company members, broker	IT market, engineering sciences and software development; Main customer: aeronautical industry	- Cooperation management and brokerage services (which includes coordination and management of VOs) - Project and quality management - On-site support	CeBeNetwork group formed by companies: CeBeNetwork engineering and IT, CeBeNetwork services, CeBeNetwork France, CeBeNetwork UK, Werucon automation GmbH	This network doesn't use a specific ICT tool. Standard office tools are used. Web site
Swiss MicroTech, Switzerland and China, 7 companies	Company members, education and research centers, technological parks and specialized consulting centers	Watch-making and other micro-technology applications	- Support for set-up a business in Switzerland - Marketing and promotion (workshops, forums, exhibitions) - Job search - Research and training - News posting - Technological development	An association with: Steering committee, President, General assembly	Web site, Search engine (for partners search and technical information), Job search, News and events posting

6.4 Formal models of collaborative networked organizations

The collaborative networked organizations are modelled by using "weak" formalisms of the enterprise architectures or modelling frameworks, such as ARCON, described in Section 6.3. The architectures and frameworks are considered as weak formalisms because they do not provide fully explicit and expressive descriptions of the involved concepts and relationships between those.

Usage of more "stronger" formalisms, such as the ones based on the Description Logics, enables not only explicit descriptions of the concepts, but also inference of their relationships, by exploiting the standard logical relations. For example, the definitions of the interaction activities, given in Section 6.2, can be expressed by using Controlled Natural Language in order to enable the evaluation of the maturity of interaction between two enterprises.

Networking is defined as a simple information exchange for some benefit. This definition can be made explicit by following rule:

```
network-with(A,B) ⇒
∃p(information(p) ∧ (send(A,p) ∧ receive(B,p)) ∨
(send(B,p) ∧ receive(A,p)))
```

Furthermore, coordinated networking implies aligning activities of two parties:

```
coordinate-with(A,B) ⇒
network-with(A,B) ∧
∃m∃n(task(m) ∧ task(n) ∧ responsible-for(A,m) ∧ re-
sponsible-for(B,n) ∧ has-precondition (n,
status(m,'completed')))
```

Cooperation also involves resource sharing for achievement of the compatible goals. Hence, following rule can be used to infer the cooperation relationship between two enterprises:

```
cooperate-with(A,B) ⇒
coordinate-with(A,B) ∧
∃m∃n(task(m) ∧ task(n) ∧ responsible-for(A,m) ∧ re-
sponsible-for(B,n) ∧
∃r(resource(r) ∧ consumed-by(r,m) ∧ consumed-by(r,n))
∧
∃g∃f(goal(g) ∧ goal(f) ∧ has-goal(A,g) ∧ has-goal(B,f)
∧ is-compatible-with(g,f))
```

Finally, collaboration means that common goal is setup:

```
collaborate-with(A,B) ⇒
cooperate-with(A,B) ∧
```

```
∃m(task(m) ∧ responsible-for(A,m) ∧ responsible-
for(B,m)) ∧
∃g(goal(g) ∧ has-goal(A,g) ∧ has-goal(B,g))
```

Semantics analysis can be useful at different levels of inter-organizational networks. First, the semantic representation of queries and information may improve the relevance of the results and thus, improve the quality of partners' selection process. It can be used instead of or in addition to usual requests representation.

Second, semantics can be used to represent participants, or groups of them, leading participants to better know each other. Such information can be useful for routing the requests to other participants in order to obtain the relevant answers within a short time and with a low traffic load.

Third, this information can also be used to organize the network so as to improve efficiency. This is very important for the open settings of the inter-organizational networks, where the traditional approaches to business process management, which attempt to capture processes as monolithic flows, have proven to be inadequate, resulting to moving research focus from process to interaction modelling (Desai et al, 2006).

The use of domain ontology is already proven as beneficial for Supply Chain Management, in the development of self-integrating SCM systems (Jones et al, 2001), or facilitating collaboration of inter-enterprise design teams (Lin and Harding, 2007), simulation of supply chain Network (Favez et al, 2005), or online negotiations (Pathak et al, 2000), development of approaches to semantic integration of industrial information systems (Izza, 2009), etc. There are also influential efforts to provide the exhaustive ontology-based semantic models for SCM (Ye et al, 2008), organized in a modular way to support the reusability and maintainability of the involved micro-theories. Also, to some extent, enterprise ontologies, elaborated in Section 5.3 formalize the semantics of the supplier-customer relationships, where IDEON^{TM} ontology is actually focused at collaborative distributed enterprises.

Chapter 3: Formalization of the supply chain operations

Abstract. Reference models play an important role in the knowledge management of the various complex collaboration domains (such as supply chain networks). However, they often show a lack of semantic precision and, they are sometimes incomplete. In this Chapter, an approach to overcome semantic inconsistencies and incompleteness of the Supply Chain Operations Reference (SCOR) model is presented. First, a literal OWL specification of SCOR concepts (and related tools) is described. It is developed with the intention to preserve the original approach in the classification of process reference model entities and hence, to enable effectiveness of usage in original contexts. Then, the SCOR-Full ontology and its relations with relevant domain ontology are presented. It is shown how it can be exploited for improvement of SCOR ontological framework competence. Finally, the potential impact of the presented approach, to interoperability of systems in supply chain networks is elaborated.

1 Introduction

Analysis of the relevant enterprise ontologies, presented in Section 5.3 of the Chapter 1, shows that lack of relevance is one of the greatest challenges in building usable domain ontologies. Typical source of this problem is the fact that existing enterprise ontologies are created from scratch. As a result, the development and, especially validation processes take a very long time to complete (Yan, 2007), due to a typically large amount of work needed for analysis and synthesis of the domain knowledge. More important, the consensus on developed conceptualizations within the relevant community is extremely hard to achieve. Finally, those ontologies cannot be considered as interoperable because of the different approaches to the conceptualization of the domain.

While selected enterprise ontologies are developed in the process of conceptualization of the domain by the experts, practice of ontology engineering suggests that process of domain conceptualization should also take into account some upper ontology. An upper ontology (or foundation ontology) is a model of the common objects that are generally applicable across a wide range of domain ontologies. It employs a core glossary that contains the terms, and associated object descriptions. Upper ontologies typically combine continuant, enduring perspectives of reality and concepts extended through time (Grenon and Smith, 2004). Then, four-dimensional perspective on reality within a single framework can be granularized further to strategic, operational and tactical sub-perspectives, so modular ontological framework with different levels of specialization of different ontologies is developed. Hence, domain ontology engineering implies that new specific concepts are created and inherited from general notions residing in upper ontology, so a consistent and expressive ontological framework is built. More important, the conditions of making a new ontology interoperable with other ontologies developed by using selected upper ontology are met. A variety of granularity levels in an ontological framework extends the scope of inference. Use of modular ontologies also addresses performance issues of the semantic environments because it enables distributed reasoning. The approach of developing domain ontology by specialization of the notions of the selected upper ontology is considered as a top-down approach.

However, bottom-up approach also has some advantages. First, it is usually built upon the implicit, but common, widely accepted knowledge, such as reference models, standard specifications, Domain Specific Languages (DSL), etc. Second, the development time is shorter, because the process of ontology engineering is reduced to semantic analysis of the reference models or standards. Finally, the evaluation problem can be only reduced to consistency checking and completeness assessment.

Typical problems of the bottom-up approach is lack of the explicitness of the resulting model which poses many difficulties and reduces scope in the efforts of making this model interoperable with other, relevant models. In this thesis, the bottom-up approach to formalization of the supply chain operations is based on the selected reference model – SCOR (Supply Chain Operations Reference) model, described in Section 2. The potential for interoperability of resulting models is increased by introducing two models at different level of abstraction – implicit SCOR model and its semantic enrichment, which is then, semantically mapped to the OWL representation of the selected domain ontology.

1.1 Description of approach to formalization of the supply chain operations

The approach to formalizing the supply chain operations, presented in this thesis builds upon three of the five general approaches to ontology design: inspiration, induction, deduction, synthesis and collaboration (Holsapple and Joshi, 2002). This design decision is implied by the choice of bottom-up approach.

Induction is used in the phase of semantic enrichment of a reference model, by improving the semantic precision of the categorizations. It is combined with inspirational approach which is characterized by an individual (authors') viewpoint about the used abstractions. Inspiration is also used for formalizing problem solving models – application models, based on the design goals. Finally, synthesis is employed in mapping of a semantically enriched model with relevant ontologies, with aim to enable semantic interoperability and/or to extend an inference scope.

In order to formalize the reference model, it is proposed to employ the semantically aligned layers of a literal specification of a reference model, its semantic enrichments and resulting domain ontology and application models, developed on basis of the different design goals. Approach reflects the practices from AI (Artificial Intelligence) domain of using the different granularities of domain knowledge in solving engineering problems of different abstraction (Falkenhainer and Forbus, 1991). The approach is demonstrated in Chapter 5, on the case study of using the SCOR model for development of the semantic application for supply chain process configuration.

Fig. 18 illustrates the proposed framework for semantic enrichment of reference models. The framework includes source data (reference models and existing domain ontologies), various tools, and resulting models: literal OWL specification, semantically enriched model, unifying model and application ontologies.

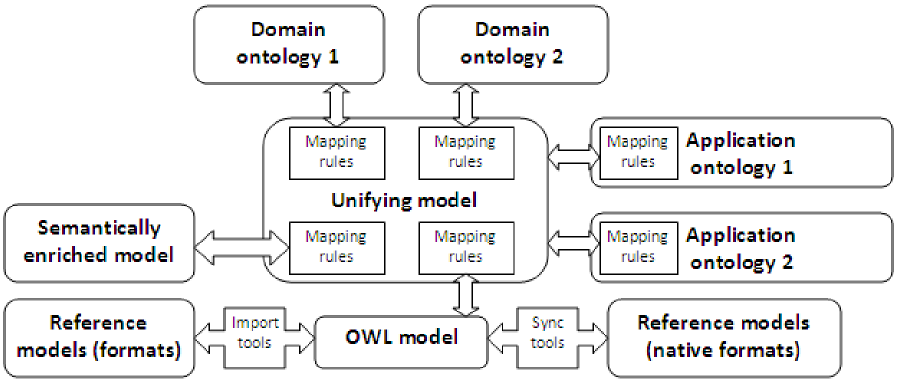

Fig. 18. Framework for semantic enrichment of reference models

As reference models are stored in number of different formats and representations, the use of import facilities in support to initial development and continuous update of the OWL model is recommended. Some of the examples of the import tools are Euler-GUI[38], a lightweight IDE that translates UML XMI format and XML schemas into N3 triples and Anzo for Excel[39], which extracts RDF data from Excel spreadsheets. Update of the models or instantiation of relying concepts can be automated when the import tools and respective API's are used for alignment of OWL models and native data formats of the applications which are using the reference models. In case of SCOR, some of the examples of such applications are ARIS EasySCOR[40] by IDS or e-SCOR[41] by Gensym , used for the benefit of SCOR implementation process.

The (implicit) concepts of the OWL models are then semantically analyzed and semantically enriched model is created. In this process, conceptualization approaches (for example, key properties of the main concepts) to selected domain ontologies or upper ontologies may be taken into account. A unifying model which imports the semantic enrichment model and OWL model stores the rules for establishment of the correspondences between the explicit and implicit concepts from these two models and/or explicit concepts of the (different) domain ontologies.

Hence, the unifying model becomes a single point of access to the enterprise knowledge (in context of its supply chain), as it unifies one implicit (reference model) and two explicit (one general, common and another, as a bridge to the reference model) views. Thus, it may be exploited by the semantic applications, which are using

[38] http://eulergui.sourceforge.net
[39] http://www.cambridgesemantics.com/semanticexchange
[40] http://www.ids-scheer.com/ru/ARIS/ARIS_Reference_Models/ARIS_EasySCOR/115741.html
[41] http://www.gensym.com

it to launch semantic queries against the integrated knowledge environment for the benefit of resolving some specific application problem.

Layering of application and domain representation models reflects the paradigm of separation of domain and task-solving knowledge (Gangemi, 2005) and assume their mutual independence (Guarino, 1997). Thus, arbitrary design goals can be defined, formalized to set of competency questions and used for development of a task-solving, application ontology. Although problem domain is restricted to a supply chain context, level of its variety can be extended in process of synthesis, namely, mapping of semantically enriched model with other relevant ontologies, such as enterprise ontology, TOVE ontologies, and others.

In the remainder of this Chapter, the core methodology presented above is implemented in the case of development of the formal framework for supply chain operations. Section 2 presents the Supply Chain Operations Reference (SCOR) model, which is used as a starting point for its development. In Section 3, OWL model and semantic enrichment of SCOR are presented. Also, the terms of the semantic enrichment are additionally explicated in the process of establishment of correspondences between its concepts and the concepts of relevant domain ontology – TOVE ontology. Finally, in Section 4, the potential impact of the proposed methodology is elaborated in the context of semantic interoperability of systems in a supply chain environment.

2 Supply Chain Operations Reference (SCOR) model

In order to gain real benefits from Supply Chain Management, relevant systems must span full horizontal organization of enterprises and beyond – its customers and suppliers. For dealing with the complexity of such an environment, reference models play an important role. Supply Chain Operations Reference (SCOR) (Stewart, 1997) is a standard approach for analysis, design and implementation of five core processes in supply chains: plan, source, make, deliver and return. SCOR defines a framework, which aims at integrating business processes, metrics, best practices and technologies with the objective to improve collaboration between partners.

In this Section, SCOR model is described. Some considerations of its possible use in the context of the semantic systems are provided. Hence, existing attempts to ontologize or semantically enrich the SCOR model are described.

2.1 Reference models

The development of reference models in different domains is a community response to interoperability problems. They aim at the standardization of domain collaboration by providing categorization schemes or taxonomies – knowledge structures, interpreted in organized way – to be used as guidelines in the collaboration of humans and systems.

Industrial reference models are not formal models. They are descriptive languages. They were created with an objective to aggregate entities for some purpose, rather than to describe the nature of the entities. Hence, they are very hard to maintain and to evolve in a consistent way. Dynamics and volatility of concepts are much easier to manage if they are represented by a set of meaningful statements or expressions, rather than by narrative descriptions. Also, higher levels of expressivity and axiomatization extend the opportunities for automated support. However, industry acceptance of the reference models shows that practical benefits are more likely to be achieved when they are focused on highly contextualized approaches where formalizing domain knowledge is involved. Domain knowledge evolves at highest rate at lower levels of abstraction in domain community interaction, where consensus is more likely to be reached.

Defining a reference model of processes is a pre-condition for their description, implementation, performance measurement, management, control and revision. One process reference model typically consists of standard descriptions (templates) of processes, specifications of the relations which can be established between those processes, standard metrics for the measurement of their performances, the specification of management practices which can be employed for gaining the top performances and descriptions of relationships between processes and functions. The process reference model may be neutral, such as SCOR, or developed for a specific industry sector, such as ENUM (telecommunications), POSC (petrochemical industry), BASEL II (banking and finance) (Phelps, 2006). Finally, process reference models may exploit some data reference models, such as product classifications (UNSPSC, eClass).

2.2 Overview of SCOR

The SCOR model is developed by supply chain Council (SCC), non-profit organization established in 1996, by AMR Research and PRTM. Initially, SCC's mission was to evaluate the market of ERP systems. While organization had 69 members at the moment of establishment, now it grew up to the membership of 1000 different organizations, all over the world.

The SCOR model is implemented from the perspective of the single enterprise and it resembles all interactions two levels ahead from the enterprise, towards its supply and customer directions (from the suppliers of the enterprise suppliers to the customers of the enterprise customers).

Core of the model is illustrated at **Fig. 19**.

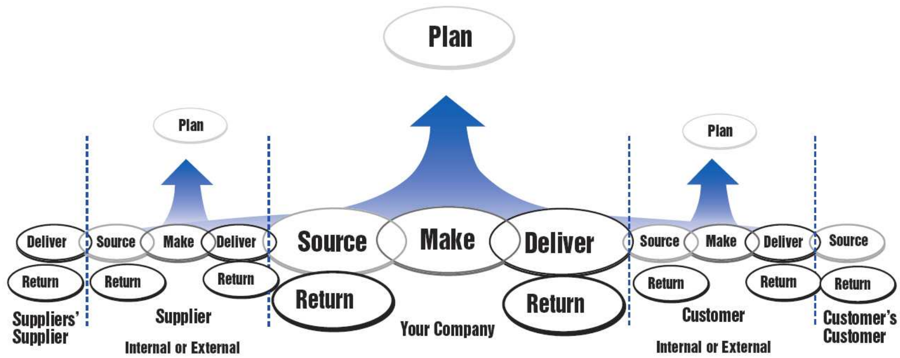

Fig. 19. Core of SCOR model

In contrast to traditional decomposition methods, process reference models are developed by identifying and analyzing the processes at different levels of detail. SCOR model does that at three levels: top level, configuration and process element level. At the top level, SCOR model defines key processes: Plan, Source, Make, Deliver and Return. They are described in more detail below.

In the first level of detail, the relationships between SCOR processes and process types are determined. At this level, a strategic character of the supply chain is determined by choosing the process categories. At the second level of detail, the chosen processes are decomposed – process elements, information inputs and outputs, metrics attributes and best practices are defined.

Key processes of the SCOR reference model and activities they include are (Bolstorff and Rosenbaum, 2003):

- Planning (P). This key process include: gathering customer requirements, collecting information on available resources, and balancing requirements and resources to determine planned capabilities and resource gaps.
- Sourcing products and material (S). The key process includes: issuing purchase orders, scheduling deliveries, receiving, shipment validation and storage, and accepting supplier invoices.
- Make (M). The Make processes describe the activities associated with the conversion of materials or creation of the content for services. It focuses on conversion of materials rather than production or manufacturing because Make represents all types of material conversions: assembly, chemical processing, maintenance, repair, overhaul, recycling, refurbishment, remanufacturing, and other material conversion processes.
- Delivery (D). This key process includes the receipt, validation, and creation of customer orders; scheduling order delivery; pick, pack, and shipment; and invoicing the customer.
- Return (R). The Return processes describe the activities associated with the reverse flow of goods back from the customer. The Return process includes the identifica-

tion of the need for a return, the disposition decision making, the scheduling of the return, and the shipment and receipt of the returned goods.

In the second level of detail, the processes above are classified into process categories (see **Fig. 20**). Planning processes are classified into P1-P5, based on which key process is planned. Source, Make and Deliver processes are considered as Execution processes and are classified on the basis of which strategy is used for manufactured, sourced and/or delivered product: make-to-stock, make-to-order or engineer-to-order. Finally, Enable processes are classified on the basis of which key process is facilitated by the Enabler process (EP, ES, EM, ED, ER).

		SCOR Process					
		Plan	Source	Make	Deliver	Return	
Process Type	Planning	P1	P2	P3	P4	P5	Process Category
	Execution		S1 - S3	M1 - M3	D1 - D4	S/DR1 - S/DR3	
	Enable	EP	ES	EM	ED	ER	

Fig. 20. SCOR Configuration Toolkit

The supply chain is configured when relevant process categories are chosen. On the basis of the manufacturing strategy, three different configurations are possible:

- S1, M1, D1, D4 – Make-to-stock
- S2, M2, D2 – Make-to-order
- S3, M3, D3 – Engineer-to-order

The full description of the level 2 of SCOR model is illustrated on the **Fig. 21**.

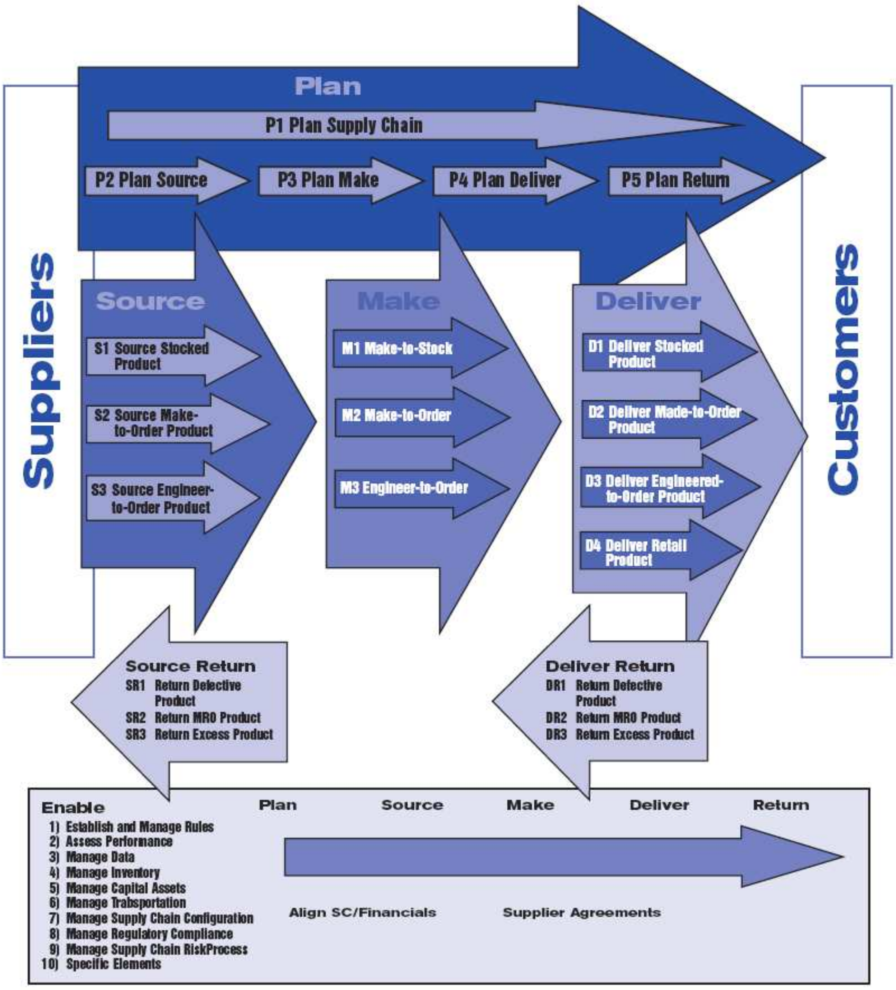

Fig. 21. SCOR Level 2

Each of the process categories is defined by the set of process elements which basically forms a workflow. **Fig. 22** shows the process elements of the S1 process category (Source Stocked Product), with a focus to S1.2 process element (Receive product), showing its relationships with other process elements.

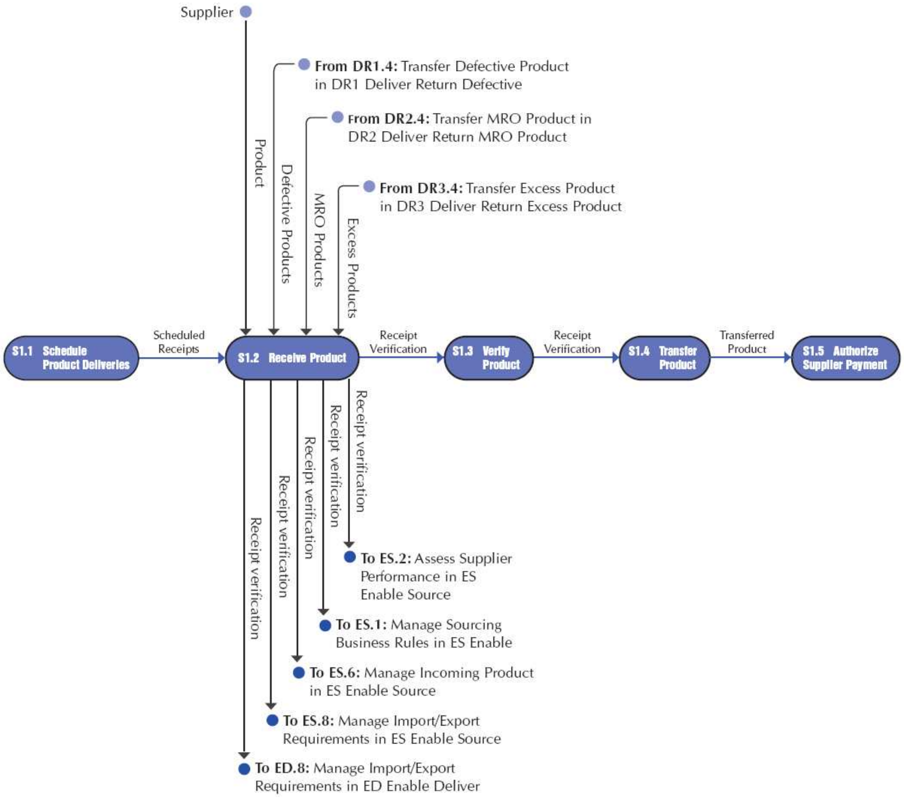

Fig. 22. Example of Level 3 representation of SCOR model

Besides process elements and their relationships with other elements, one process category is also defined by inputs and outputs for each of the elements, metrics, best practices and recommended capabilities of the systems and systems themselves, which facilitate implementation of those practices.

The performances of the SCOR processes are measured by using metrics where each of this metrics is related to one of the five core attributes of the supply chain performance. Metrics is also structured at the levels. Level 1 metrics are known as strategic metrics and Key Performance Indicators (KPI). They are determined by using the lower level metrics. For example, each of the process categories is assigned with a set of Level 2 metrics which is used to determine the performance of those process categories. Similarly, Level 3 metrics are assigned to process elements.

Three of the performance attributes are related to external relationships of the enterprise and these are: Reliability, Responsiveness and Agility. Other two attributes, Costs and Assets address internal performances of the processes. The detailed description of core performance attributes follows:

– The Reliability attribute addresses the ability to perform tasks as expected. Reliability focuses on the predictability of the outcome of a process. Associated KPI is

Perfect Order Fulfilment. Typical metrics for the reliability attribute include: on-time, the right quantity, the right quality.

- The Responsiveness attribute describes the speed at which tasks are performed, and is measured by Order Fulfilment Cycle Time KPI.
- The Agility attribute describes the ability to respond to external influences and the ability to change. External influences include: Non-forecasted increases or decreases in demand; suppliers or partners going out of business; natural disasters; acts of (cyber) terrorism; availability of financial tools (the economy); or labor issues. The agility is evaluated by following KPIs: Upside supply chain Flexibility, Upside supply chain Adaptability and Downside supply chain Adaptability.
- The Cost attribute describes the cost of operating the process. It includes labour costs, material costs, and transportation costs, and is evaluated by using Supply Chain Management Cost and Cost of Goods Sold KPIs.
- The Asset Management Efficiency ("Assets") attribute describes the ability to efficiently utilize assets. Asset management strategies in a supply chain include inventory reduction and in-sourcing vs. outsourcing. The Key Performance Indicators are Cash-to-Cash Cycle Time, Return on supply chain Fixed Assets and Return on Working Capital. Metrics include: inventory days of supply and capacity utilization.

2.3 Existing work in semantic enrichment of SCOR

Like most of the other reference models, SCOR is a form of knowledge organization system. The key feature of these models is subjectivity, or context-dependent determination (Hodge, 2000). They are not developed with the intent to be semantically rich or precise, but to provide human-understandable knowledge on the specific domain. However, their implicitness is considered as an obstacle for a machine-based interpretation. SCOR lacks semantic precision. SCOR's Input/Output entity entails all resources exchanged between process elements and actors - physical or non-physical, states, events, documents, etc. System entity includes information systems, modules, capabilities, approaches or volume of use, integration levels, etc.

Also, sometimes, reference models do not provide enough expressivity for a complete formal model. In the case of SCOR, this is evident from the lack of relationships between metrics and systems, which could point out to the source of information needed for performance measurement.

So far, there were only a few attempts to ontologize SCOR model.

SCOR+[42] is directed towards overcoming the limitations of the basic SCOR model through an ontology based tool. This tool enables an automated and comprehensive definition of the supply chain at four of its distinctive levels: supply chain level, the enterprise level, the elements level, and the interaction level. It enables generation of generic explicit views and models that represents the four levels. Unfortunately, SCOR+ is a proprietary product and details on the formalization approach are not accessible.

[42] Productivityappex website. http://www.productivityapex.com/products/scor.asp

Lin (2008) extended the SCOR model by generalizing existing elements to 3A concepts (Activity, Artefact, Actor-Role), defined in GPO (General Process Ontology). Also, she used the model for development of the goal ontology, by modelling SCOR performance attributes as general soft goals and deriving domain specific goals from attributes' metrics.

Vegetti et al (2005) used SCOR to develop the SCOntology. They extended SCOR with the notions of an enterprise model, with aim to provide the foundations for the specification of information logistics processes in extended supply chains associated to process industries.

Lu et al (2010) extended the ONTO-PDM Product Ontology developed by Tursi et al (2009) with the SCOR model. The resulting ONTO-SCOR ontology is then defining product-centric supply chain ontology for facilitating the interoperation between all enterprise's applications involved in an extended supply chain.

On basis of the analysis of the contribution of the SCOR model to the alignment of business processes and information systems, Millet et al (2009) proposed the extended reference model, including the structure of information exchanged between processes. This model is proposed in response to the identified weaknesses of the current SCOR model, in specific, lack of important process dependencies.

In addition, there are many relevant papers with reported work on other reference models' formalization, addressing the semantics of RosettaNet (Haller et al, 2007), UNSPSC (Hepp, 2006), AIAG and STAR (Aničić et al, 2005), EDI (Foxvog and Bussler, 2006), etc. Presented results, methods, tools and gained experiences were extremely useful in setting up and implementing the proposed approach.

3 Ontologies and models of the formal framework

Important role in the implementation of the interoperable systems is given to a domain ontology – explicit representation of the specific domain knowledge (e.g. about Supply Chain Management), namely its concepts and logical relations between those. Domain ontology ensures the correctness of the inference on the meaning of the information which is being exchanged. Thus, it has to be: a) expressive (to contain all concepts from one domain and all their relations); b) explicit (to uniquely define all concepts and their relations); c) neutral (to define all concepts objectively, independently from the specific context); and d) relevant (in the sense that there is a consensus in the domain community about used conceptualization). Given that domain ontology is a facilitator of arbitrary EISs' interoperability, it is obvious that its relevance is the most important feature.

Lack of relevance is a weakness of all existing efforts in definition of the supply chain ontology, such as TOVE, The Enterprise Ontology, IDEON, etc. All these ontologies are created in isolation, by applying an inspirational approach, from the scratch, while their verification is performed only in small number of cases. In the approach to the development of a formal framework for supply chain networks, this issue is addressed by using a widely accepted SCOR reference model framework as a source of implicit semantics. Then, this semantics is made explicit in the process of synthesis of the general enterprise notions and explained, in a process of mapping of those notions to the common enterprise knowledge, formalized in the selected domain ontologies.

The consensus on the collective knowledge is extremely hard to reach (Hepp, 2007), particularly when very expressive (or richly axiomatized) ontologies with large number of concepts are involved. In response to this problem, the approach of collaborative conceptualization is proposed and applied on the case of electronic product catalogues integration (Guo, 2009).

The proposed formal framework for supply chain networks is illustrated at . An approach to its development is based on a premise that domain knowledge changes and evolves at the highest rate in the lower levels of abstraction, in domain community interaction, because consensus on the specific notions is easier to be reached than agreement on the generalizations and abstractions. It is a fact that this level is often characterized by the implicit semantics of the standards, reference models, database structures, etc., as they are often described by using a natural language.

However, this is not necessarily an issue. Under condition that this implicit semantics can be formalized by using a language based on Description Logics, such as OWL, in a native form, and mapped accordingly to an expressive, neutral domain model, having an implicit model within the framework may be considered as a valuable asset.

Namely, it can be used to bridge the gap between formal domain theories and EISs which are using those implicit models. Thus, the coherence between creation, evolution and use of specific, highly contextualized knowledge and development of formal expressive models is considered as a very important factor for usability of the models.

The advantages of the bottom-up type of approach are already discussed before. First, it is usually built upon the implicit, but common, widely accepted knowledge (in this case, SCOR model). It is important to emphasize that, in contrast to some other approaches (Millet et al, 2009), proposed formalization approach does not aim to extend the semantics of SCOR, but only to improve its expressiveness. Second, the development time is shorter, because the process of ontology engineering is reduced to semantic analysis of the documents standards. Third, the evaluation problem can be only reduced to consistency checking and completeness assessment.

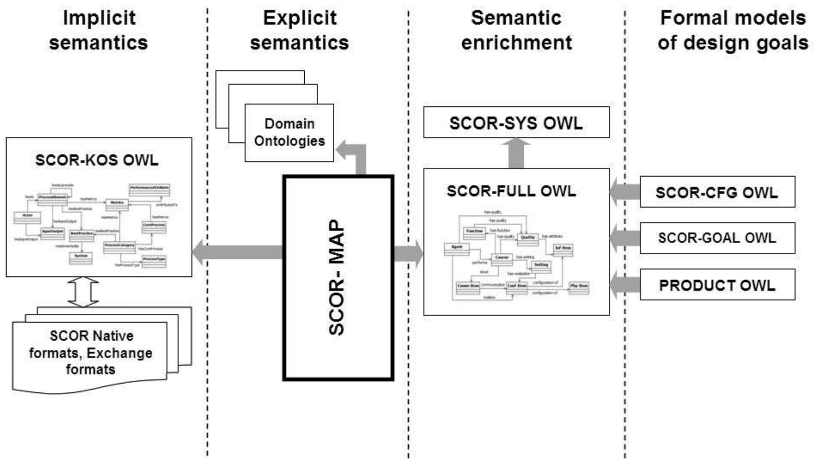

Fig. 23. Formal framework of supply chain operations.

The core of the formal framework for supply chain networks consists of two models which describe the same thing – supply chain operations, but they do that at two different layers of abstraction. First layer models implicit semantics of SCOR elements and stores actual knowledge on supply chain operations by using semantically weak knowledge structure. In this layer, SCOR elements are represented in a native – natural language form, and related accordingly. This representation is described in detail in Section 3.1 of this Chapter. Second layer represents SCOR's semantic enrichment (presented in Section 3.2) - it identifies common enterprise notions, maps those to SCOR entities and classifies them into more general inter-related concepts. Both layers are then represented by OWL models – SCOR-KOS (SCOR Knowledge Organization System) and SCOR-Full.

SCOR-Full may be considered as a micro-theory which identifies and classifies common enterprise concepts in the context of supply chain operations. It is developed by semantic analysis of SCOR Input/Output elements, identification of core terms and their generalization into notions of Course, Setting, Quality, Function and Resource. It extends the SCOR-Sys ontology, which formalizes the SCOR System element. It is then extended by the SCOR-Goal ontology, which semantically maps its concepts to SCOR Performance Metrics element. SCOR-Full ontology is mapped to an implicit knowledge model of SCOR (SCOR-KOS). Hence, any structural changes in the underlying model, such as introducing new process dependencies (Millet et al, 2009) will be reflected immediately on the system that is using the SCOR-Full ontology.

It is important to emphasize that SCOR-Full is only an intermediary model, in the sense that it only classifies common enterprise notions in the context of the supply chain, while their semantics is defined externally. Different enterprise formalizations, contexts and views of existing architectures and other conceptualizations need to be used as sources of specifications of enterprise semantics, and mapped accordingly to the enterprise notions in SCOR-Full ontology. Currently, SCOR-Full ontology is mapped to TOVE organizational and foundational ontology (in fact, to its OWL representation). The approach to this mapping and some correspondences are described in Section 4 of this Chapter.

SCOR-Full is exploited by the different application models, which formalize specific design goals. Namely, besides ontologies, the formal framework consists also of semantic applications, which are shared resource of all enterprises in the supply chain network. Their role is to support the collaborative activities and functions of the network, such as the management of inter-organizational processes, partner selection, management of use of shared resources, etc. For fulfilment of these roles, each of the semantic applications exploits the individual application (or problem) ontology – formal representation of the individual problem.

For example, SCOR-Cfg OWL model is used to develop a semantic web application for supply chain process configuration (Zdravkovic et al, 2011). While Product OWL model is used to develop a semantic web application for acquisition of product requirements in the inter-organizational settings (Zdravkovic and Trajanovic, 2009), SCOR-Goal OWL (Zdravkovic and Trajanovic, 2011) model drives the performance measurement of supply chain operations.

In the context of the conditions for expressiveness, explicitness, neutrality and relevance of the domain ontology, described at the top of this section, the following important considerations regarding the described formal framework are made:

1. It is expressive, because it formalizes the widely accepted industrial standard – SCOR reference model;
2. It is explicit, because implicit SCOR elements are synthesized to the common enterprise notions in SCOR-Full;
3. It is neutral to the extent of the neutrality of the concepts of the domain ontologies which are used for descriptions of the semantics of those notions;
4. It is relevant because it reflects the industry practice (SCOR reference model).

3.1 SCOR-KOS OWL Model

According to proposed methodology, SCOR reference model is used as a starting point for building the fully expressive SCM semantics, as it reflects the community consensus. Because of the SCOR’s weak semantics, in the first step, it is modelled as a knowledge organization system (KOS). In order to make this system interoperable with other components of the framework, the semantic tools are used to represent this model in a computable language – OWL language. **Fig. 24** shows entities of SCOR-KOS OWL model and relationships between them.

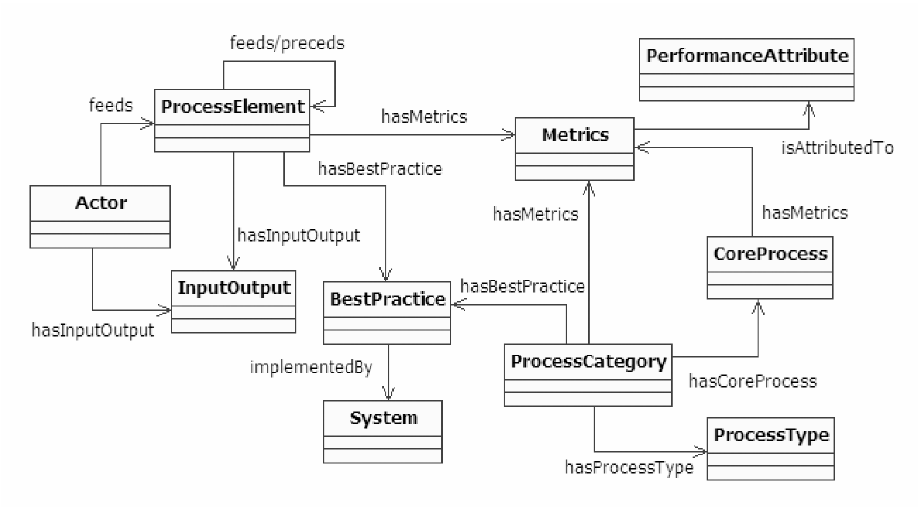

Fig. 24. Entities of SCOR-KOS OWL model

The items of the SCOR model are represented as instances of SCOR-KOS OWL concepts: SCOR_CoreProcess, SCOR_ProcessCategory, SCOR_ProcessType, SCOR_ProcessElement, SCOR_PerformanceAttribute, SCOR_Metrics (with child concepts of SCOR_Asset_Metrics, SCOR_Reliability_Metrics, SCOR_Asset_Metrics and SCOR_Responsiveness_Metrics), SCOR_BestPractice, SCOR_System, SCOR_Actor i SCOR_Input-Output.

Then, following relationships are asserted between those instances:

```
hasProcessElement(SCOR_ProcessCategory,
SCOR_ProcessElement);
hasMetrics(SCOR_ProcessCategory, SCOR_Metrics);
hasProcessType(SCOR_ProcessCategory, SCOR_ProcessType);
hasBestPractice(SCOR_ProcessCategory, SCOR_BestPractice);
hasBestPractice(SCOR_ProcessElement, SCOR_BestPractice);
hasCoreProcess(SCOR_ProcessCategory, SCOR_CoreProcess);
preceeds (SCOR_ProcessElement, SCOR_ProcessElement);
feeds(SCOR_ProcessElement, SCOR_ProcessElement);
hasInput(SCOR_ProcessElement, SCOR_Input-Output);
hasOutput(SCOR_ProcessElement, SCOR_Input-Output);
hasMetrics(SCOR_ProcessElement, SCOR_Metrics);
hasMetrics(SCOR_CoreProcess, SCOR_Metrics);
hasProcess(SCOR_CoreProcess, SCOR_CoreProcess);
feeds(SCOR_Actor, SCOR_ProcessElement);
hasInput(SCOR_Actor, SCOR_ProcessElement);
hasOutput(SCOR_Actor, SCOR_ProcessElement);
implementedBy(SCOR_BestPractice, SCOR_System);
isAttributedTo(SCOR_Metrics, SCOR_PerformanceAttribute)
```

Following relationships are inferred as inverse of the corresponding asserted properties:

```
suceeds(SCOR_ProcessElement, SCOR_ProcessElement);
measure(SCOR_Metrics, SCOR_ProcessElement);
```

Also, axiomatic definition of the SCOR_Asset_Metrics, SCOR_Flexibility_Metrics, SCOR_Reliability_Metrics and SCOR_Responsiveness_Metrics concepts facilitates automatic classification of the metrics of different types:

```
SCOR_Asset_Metrics •
(( isAttributedTo.{Assets}) • SCOR_Metrics)
SCOR_Reliability_Metrics •
(( isAttributedTo.{Reliability}) • SCOR_Metrics)
SCOR_Asset_Metrics •
(( isAttributedTo.{Assets}) • SCOR_Metrics)
SCOR_Responsiveness_Metrics •
(( isAttributedTo.{Responsiveness}) • SCOR_Metrics)
```

SCOR-KOS OWL model is developed as OWL-DL ontology, by using Protégé (Knublauch et al, 2004) tool (see **Fig. 25**), on basis of the semantic analysis of version 6 of SCOR reference model[43]. It contains 418 of the metrics elements, 166 process elements, 25 process categories, 164 best practices, 282 Input/Output elements and 108 system elements. For inference about the properties of the SCOR-KOS OWL, Pellet 1.5 reasoner (Sirin et al, 2007) is used.

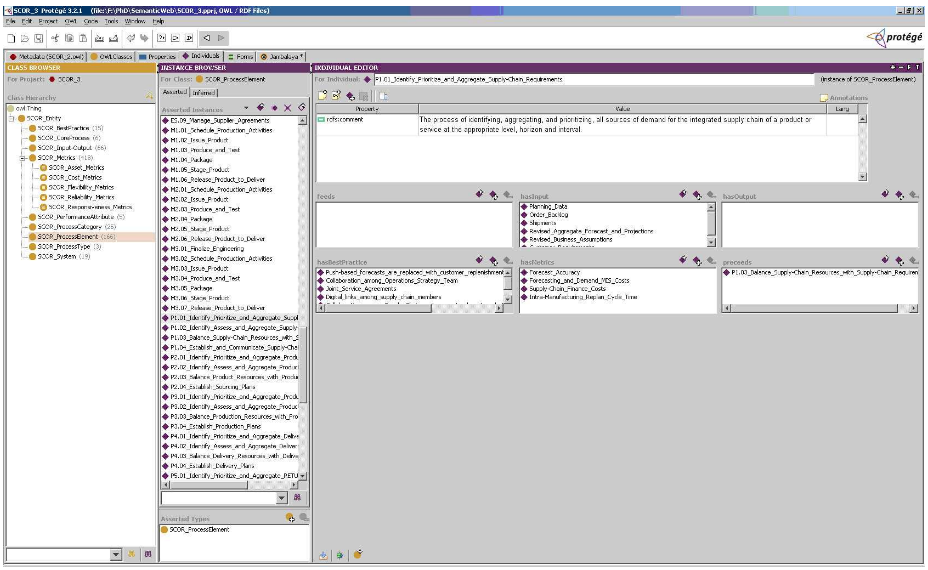

Fig. 25. Protégé tool with SCOR-KOS OWL model

[43] Supply-Chain Operations Reference-model SCOR Version 6.0, 2003, supply chain Council

In SCOR-KOS OWL, process flows are asserted by using a property precedes(SCOR_ProcessElement, SCOR_ProcessElement). Consequently, they may be inferred by its inverse proeprty: succeeds(SCOR_ProcessElement, SCOR_ProcessElement). In addition, property feeds(SCOR_ProcessElement, SCOR_ProcessElement) is used for establishment of flows between process elements of the different processes.

This property is also used for assertion of the flows between process elements and actors of the SCOR processes. Actor of the SCOR process is supplier, buyer, or any organizational unit of the enterprise from which perspective processes of a supply chain are managed, such as sales, marketing or legal unit.

Competency of SCOR-KOS OWL model

Aim of the literal OWL specification is to preserve the classification approach of SCOR. It represents SCOR model's concepts and properties and thus it enables the use of a resulting SCOR-KOS model for the original purpose. This purpose can be formalized by the competency questions, used for the validation of resulting model.

Competency of a SCOR-KOS OWL model is validated by using following questions:

1. Which process elements constitute one SCOR process and in which order?
2. What are the input and output resources for the selected process element?
3. What are the metrics and best practices for the selected process element?
4. Which systems can facilitate the improvement of the selected process element and/or process category?

In the remainder of this Section, the competency of the SCOR-KOS OWL model is argued and it is shown how competency questions may be answered.

The actual order of process elements is determined by executing SPARQL queries against asserted "precedes" (meaning direct precedence) triples. SPARQL (Prud'hommeaux and Seaborne, 2008) (a recursive acronym for SPARQL Protocol and RDF Query Language) is an RDF query language, able to retrieve and manipulate data stored in Resource Description Framework format.

The great most of the process categories are characterized by the simple linear flows, with exception of P1, P2, P3 and P4 planning process categories, where concurrent process elements exist.

The definition of concurrency in a SCOR-KOS OWL model is used only for the determination of flows branching and hence, it is not semantically correct. Concurrency is inferred on basis of "isConcurrentWith" relation and modelled by property chain axioms, on basis of asserted "precedes" and inferred (inverse) "succeeds" property:

```
precedes o succeeds => isConcurrentWith
```

, or by using RDQL query:

```
∃precedes.(2 succeeds)
```

Fig. 26 shows the example of concurrent process elements A and B, preceding process element C (asserted relationships). Inferred relationships "succeeds" and "isConcurrentWith" are shown as dashed lines.

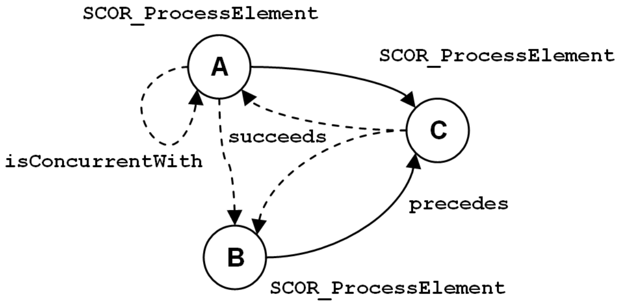

Fig. 26. Approach to modeling concurrency of process elements

Flows of input and output resources are determined by SPARQL queries, which return instances of "SCOR_InputOutput" concept from domain of asserted triples of "hasInput" and "hasOutput" properties. The source of these properties is determined from the domain of "fedBy" property, inverse of "feeds", which is used to assert connections between process elements from different process categories. **Fig. 27** shows input and output resources of D1.08 process element. The visualization of the D1.08 process element is generated by the developed tool which uses SCOR-KOS OWL model to illustrate the asserted and inferred properties of its elements.

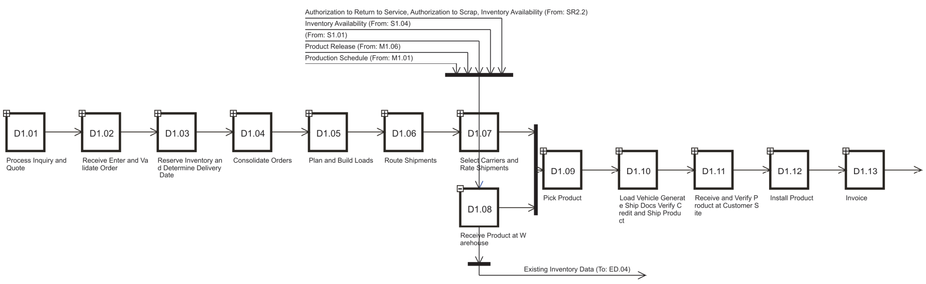

Fig. 27. Visualization of D1 process category, with selected focal element D1.08

In response to competency question 3, the tool also show elements of metrics and best practices, asserted by using "hasBestPractice" and "hasMetrics" properties. **Fig. 28** shows the map of "P1. Plan supply chain" process category, with additional metrics and best practices layers turned on.

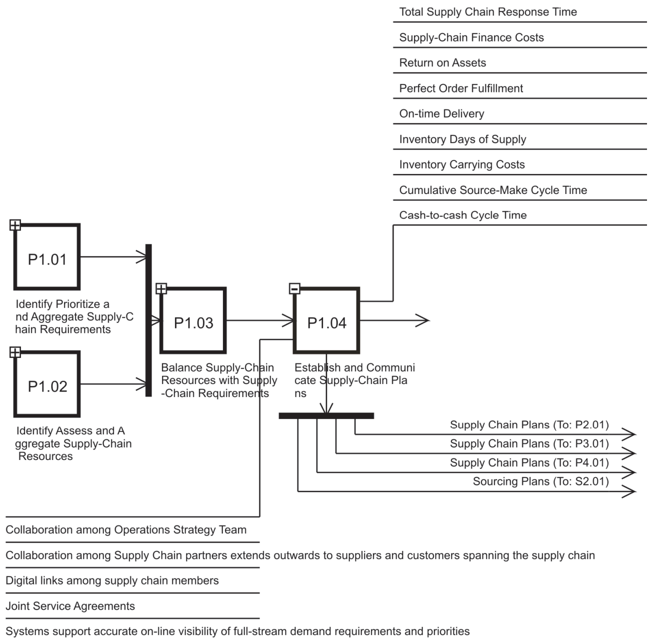

Fig. 28. Visualization of P1 process category with displayed metrics and best practices layer

Inference of systems which can facilitate improvement of selected process elements (categories) is achieved by implementing properties:

```
implements(SCOR_System,SCOR_BestPractice)
```

, and:

```
facilitates (SCOR_BestPractice,SCOR_ProcessElement),
```

as inverse to “implementedBy” and “hasBestPractice”, used for the assertion of relationships between process elements, best practices and systems. The properties above are defined as sub-properties of transitive property “enable”, hence, enabling reasoning of relationships between “SCOR_System” and “SCOR_ProcessElement” concepts (see **Fig. 29**, below).

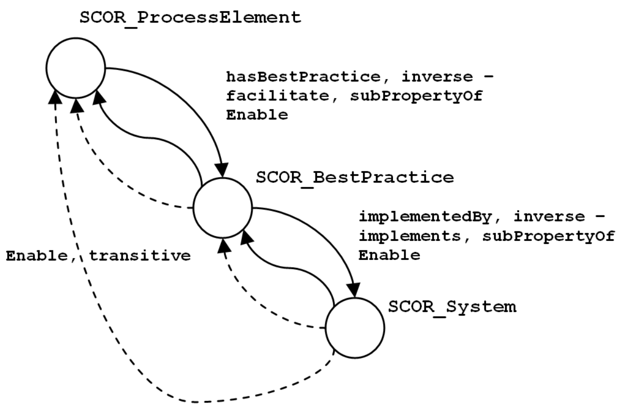

Fig. 29. Asserted and inferred relationships between instances of SCOR_System, SCOR_BestPractice and SCOR_ProcessElement concepts

After establishing these relationships, for „B2B Integration and Application Server System", the following relevant statements can be inferred:

```
enable P1.01_Identify_Prioritize_and_Aggregate_Supply-
Chain_Requirements
enable P1.02_Identify_Assess_and_Aggregate_Supply-
Chain_Resources
enable P1.04_Establish_and_Communicate_Supply-Chain_Plans
enable
P4.01_Identify_Prioritize_and_Aggregate_Delivery_Requirem
ents
```

Namely, it can be concluded that the implementation of B2B integration and the application server system can influence the improvement of performances of the 4 abovementioned process elements.

Since best practices are related also to process categories, it is possible to infer the impact of a system or system capability to process categories. For example, for instance „supply chain Event Management Software", following relevant statement can be inferred on the basis of initial assertions:

```
enable P2_Plan_Source
```

Namely, it can be concluded that implementation of the system for event management in the supply chain may positively affect the performance of the process of sourcing of purchase planning.

By defining inverse property "enabledBy", the inference on relationships between systems and process elements (categories) becomes possible in the opposite direction. Thus, it is possible to identify systems which can improve the performance of a selected process element and/or category. This last conclusion is the response to the last competency question of the SCOR KOS OWL model.

SCOR-KOS OWL is used for the development of the web application for browsing and visualization of the SCOR framework. Main features of the web application are:

- display of the selected process category map,
- display of the input/output resources (including sources/destinations) for selected process element,
- display of the best practices and metrics for selected process element and
- customization of the display, including layering of different levels of detail and customization of the resulting schemes' geometries.

Fig. 30 shows the web application's work area, with displayed output resources, best practices and metrics for „P4.04. Establish delivery plans" process element of „P4. Plan Deliver" process category.

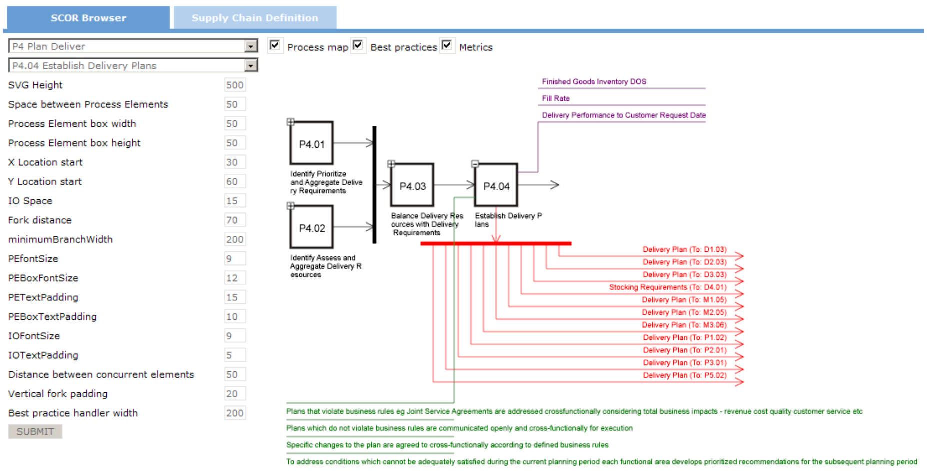

Fig. 30. „P4. Plan Deliver" process category

The tool is developed by using RAP (RDF API for PHP) (Oldakowski et al, 2005) application programming interface for parsing, querying, manipulation and serialization of RDF models. Some features of the RAP API are: support of RDF, RDFS, N3, N-Triple and OWL models, serialization of the ontologies to MySQL databases, engine for processing RDQL and SPARQL queries, some limited inference support and others. For visual representation, SVG (Scalable Vector Graphics) format is used.

3.2 SCOR-Full Model

Although it is developed as semantic enrichment of SCOR reference model, SCOR-Full can be considered as domain ontology – a micro theory for representation and management of knowledge of the supply chain operations. It formalizes core concepts of supply chain operations, embedded in SCOR model definitions. It is developed by semantic analysis of SCOR Input/Output elements, identification of core terms and their categorization. It extends SCOR-Sys ontology, which formalizes the SCOR System element. It is extended by SCOR-Goal ontology (Zdravkovic and Trajanovic, 2011), which semantically maps its concepts to SCOR Performance Metrics element. Latter two ontologies are not in the scope of this work and will not be elaborated.

SCOR-Full ontology does not aim at formalizing the supply chain, but only to resolve semantic inconsistencies of a SCOR reference model. Thus, its scope is strictly limited to using the common enterprise notions for expressing the existing elements of SCOR model.

Central notion of the SCOR-Full ontology (as it is the case for SCOR model) is a generalization of process, in the sense that it acts as the main context for semantic definition of other concepts in the ontology.

Main concepts of the SCOR-Full ontology are: Agent, Course, Resource Item (and its sub-concepts: Information Item, Physical Item, Configured Item and Communicable Item), Function, Quality and Setting. **Fig. 31** shows the main concepts of SCOR-Full ontology and relationships between them.

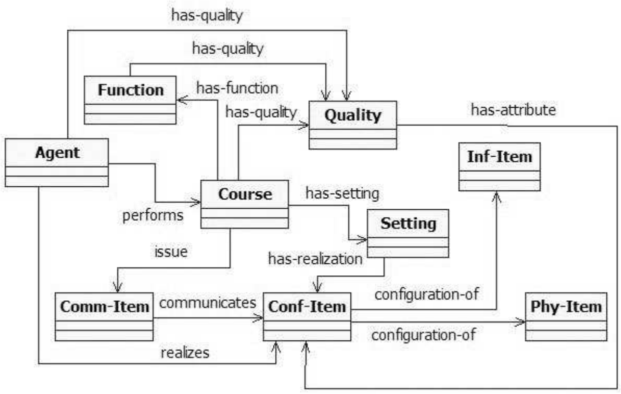

Fig. 31. Top-level concepts of SCOR-Full ontology and relationships between them

Agent (see **Fig. 32**) is the concept which describes an executive role and entails all entities which perform individual or set of tasks within the supply network, classified with the concepts of equipment, organization, supply chain, supply chain network, facility and information system.

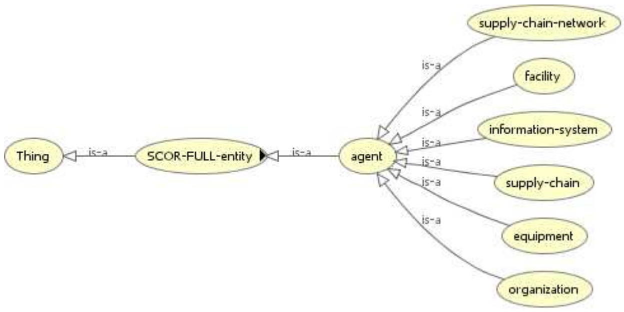

Fig. 32. Taxonomy of "agent" concept

Although semantically described as roles, agents do not have explicit definition of functions. Functionality is defined as a property of a course, performed by an agent. Hence, agents are functional in a context of a course they execute. The basic formal consequence of the assumptions above is that agents do not exist if they do not perform some course of doable things. Hence, the necessary condition for an agent is to perform some course. In other words, the concept of Agent in the SCOR-Full ontology is a child of anonymous class:

$$\forall a\ (agent(a))\ \exists c\ (course(c) \wedge performs(a,c))$$

Child concepts of the Agent concept, such as equipment, facility, information-system, organization, supply-chain and supply-chain-network inherit this anonymous class.

Course (see **Fig. 33**) classifies prescriptions or descriptions (independent of the time dimension) of ordered sets of tasks: activity, process, method, procedure, strategy or plan, at the same level of abstraction. The notion of course generalizes "doable" or "done" things with common properties of environment (corresponding to the enabling and resulting states, constraints, requirements, etc.), quality (cost, duration, capacity, performance, etc.) and organization (agent and business function). The first necessary condition for the classification of instances of Course type is that they are functional, in the sense that there is some general purpose why some ordered set of tasks is performed (or is expected to be performed). The second necessary condition for a Course is that it has some impact to the environment (a goal, objective or state) and/or it receives some feedback from the environment or it considers some of its features (such as constraint, requirement, rule or assumption). In other words, the course must have its own setting.

Hence, the concept of Course inherits two anonymous classes:

$$\forall c\ (course(c))\ \exists f\ (function(f) \wedge has\text{-}function(c,f))$$
$$\forall c\ (course(c))\ \exists s\ (setting(s) \wedge has\text{-}setting(c,s))$$

Although the concept of a course may be associated with a different attributes (qualities), executing or responsible actor and communication items, those are not its defining features.

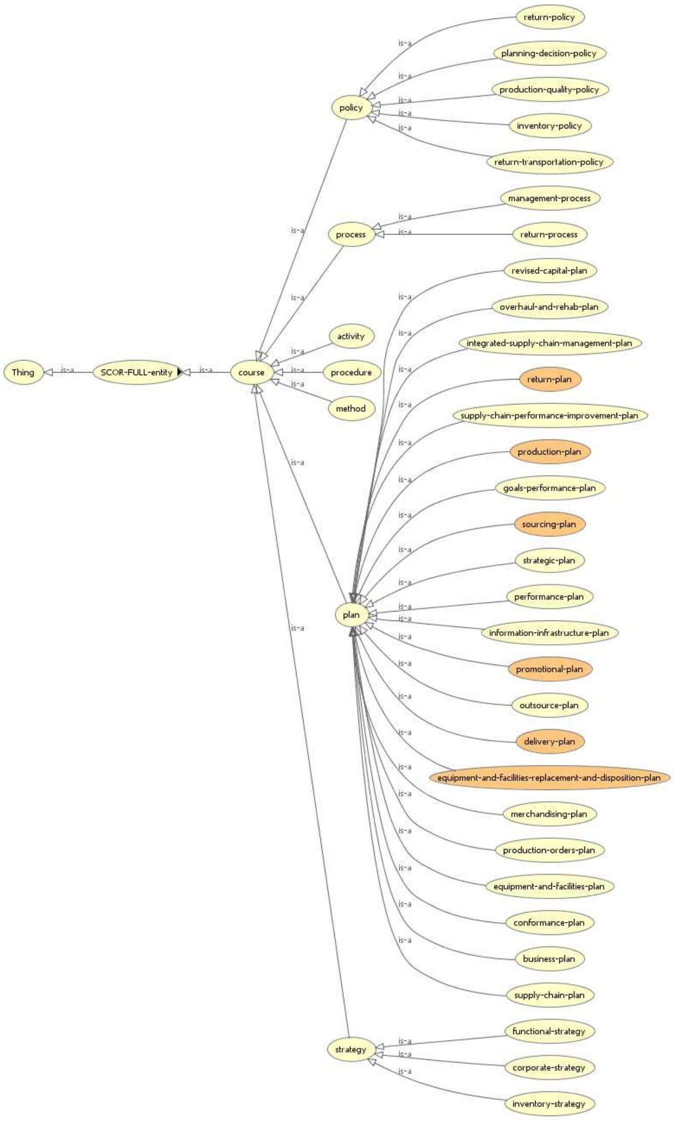

Fig. 33. Taxonomy of "course" concept

Setting (see **Fig. 34**) concept provides the description of environment of a course. It aggregates semantically defined features of the context in which course take place – its motivation, drivers and constraints. Thus, it classifies rules, metrics, requirements, constraints, objectives, goals or assumptions of a prescribed set of actions. While the instances of the Setting concept are semantically described by their classification into some of its sub-concepts, they also must correspond to some quantifiable notions which describe the specific values or states. Otherwise, they would be only of abstract nature. So, the necessary condition for a setting is to be realized by some configured item (to be described later):

$$\forall s\ (\text{setting}(s))\ \exists ci\ (\text{configured-item}(ci) \wedge \text{has-realization}(s,ci))$$

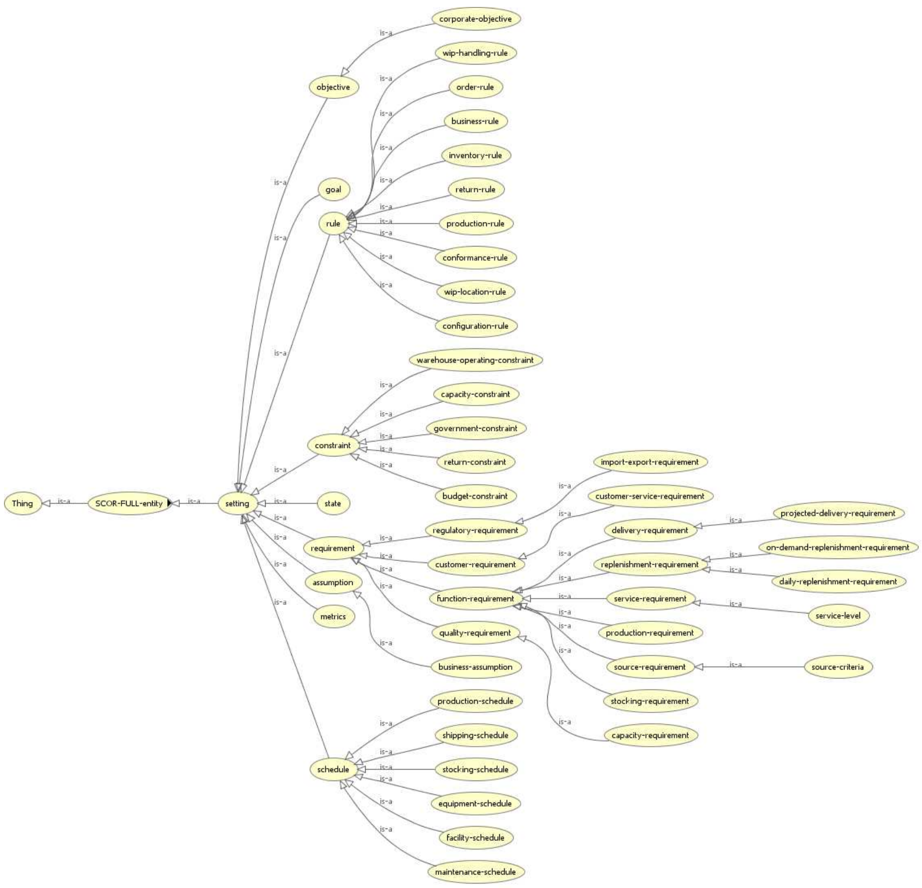

Fig. 34. Taxonomy of "setting" concept

Quality (see **Fig. 35**) is the general attribute of a course, agent or function which can be perceived or measured, e.g. capability, capacity, availability, performance, cost or time/location data.

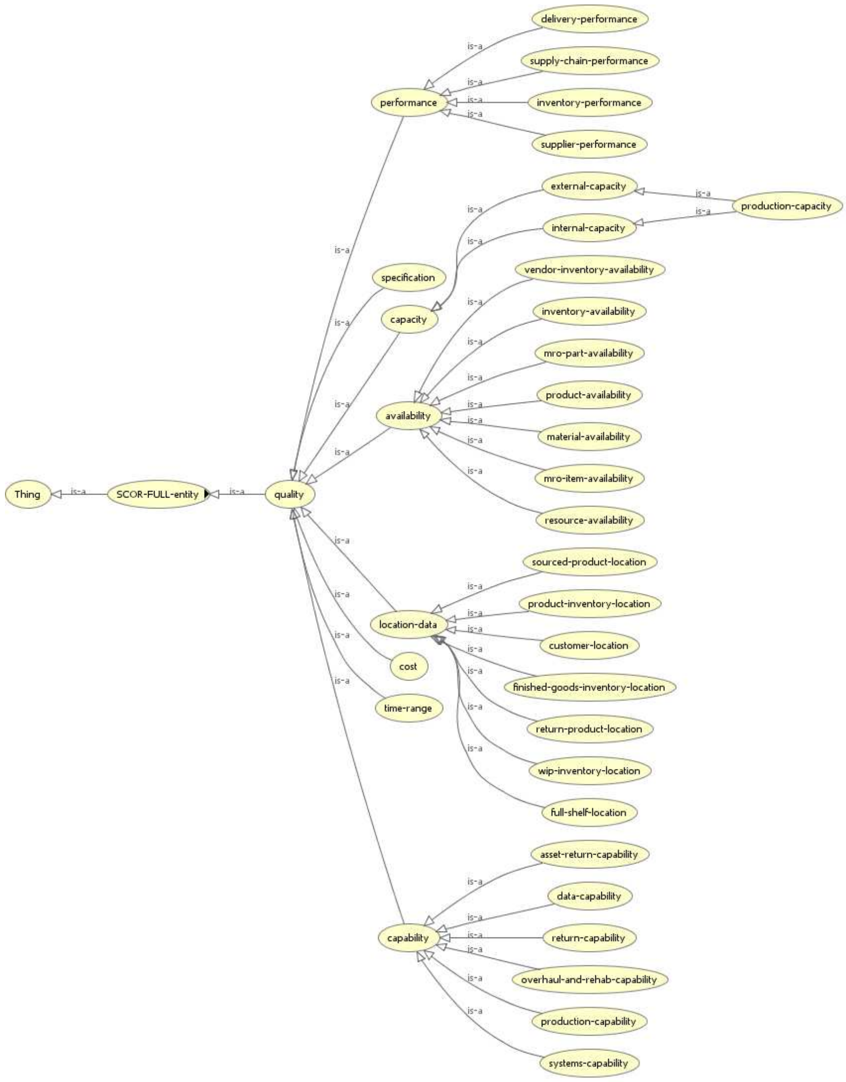

Fig. 35. Taxonomy of "quality" concept

Like in the case of Setting concepts, those attributes are only semantically described abstract categories. Hence, they need to be mapped to the actual specific values or states. The necessary condition for the instances of the Quality concept is that they need to be associated to at least one instance of the "configured-item" concept:

$$\forall q\ (quality(q))\ \exists ci\ (configured\text{-}item(c) \wedge has\text{-}attribute(q,ci))$$

The anonymous class which is inherited by the quality concept is coded, by using Manchester OWL syntax, as below:

```
has-attribute min 1 configured-item
```

Function (see **Fig. 36**) concept entails elements of the horizontal business organization, such as stocking, shipping, control, sales, replenishment, return, delivery, disposition, maintenance, production, etc. Although it may have some qualities associated, the concept of function is an abstract concept, which basic purpose is to semantically define the context of the course.

Fig. 36. Taxonomy of "function" concept

Resource items of SCOR-Full ontology

Instead of representing process flows, SCOR-Full is used to model enabling and caused states of the relevant activities. These states are represented by the concept of configured item (Conf-Item), the range of the "has-postcondition" and "has-precondition" properties of Course and its sub-concept – Activity.

A resource item (see **Fig. 37**) is a general term which encloses communicated (Comm-Item, e.g. Notification, Response, Request) and configured (Conf-Item, with defined state) information items (Inf-Item), such as Order, Forecast, Report, Budget, etc., and physical items (Phy-Item). Where information items are the attributes of a Quality (of Function, Agent or a Course), their configurations are realizations of the rules, metrics, requirements, constraints, goals or assumptions of a course.

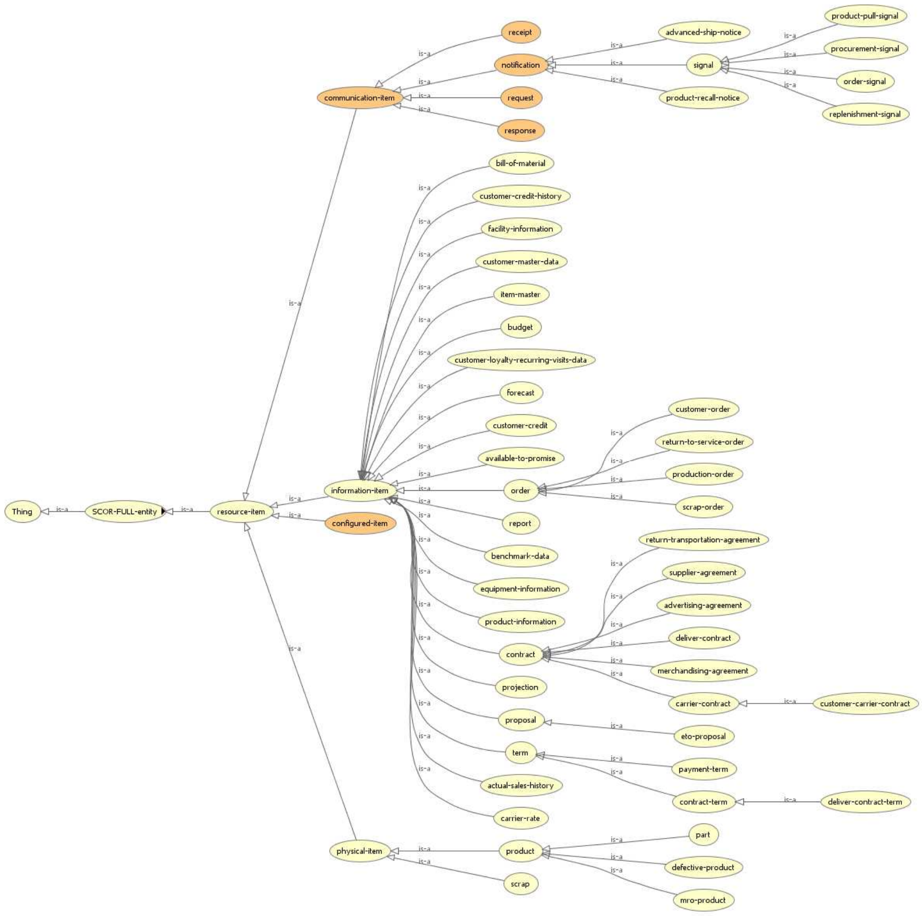

Fig. 37. Taxonomy of "resource-item" concept

Configured items model state semantics of the resource – physical or information item, the notions which are used to aggregate the atomic, exchangeable objects in enterprise environment. Examples of information items are Order, Forecast, Budget, Contract, Report, Proposal, Bill-Of-Material, etc. Their structure is not addressed by SCOR-Full ontology – from this perspective, these are the atomic concepts which can be semantically defined when mapped to other enterprise ontologies. Physical items are Product (MRO-Product, Defective-Product and Part) and Scrap. Configured items are characterized by one or multiple states of information or a physical item, assigned numerical (textual or date) value or realized by another configured item:

```
(Inf-Item(?x) ∧ (has-numerical-value(?x, decimal) ∨ has-
text-value(?x, string) ∨ has-date-value(?x, dateTime) ∨
(information-item(?i) ∧ has-realization(?x, ?i)))) ∨
```

```
((Phy-Item(?x) ∨ Inf-Item(?x)) ∧ has-
state(?x,state(?y))) ⇒ Conf-Item(?x)
```

Thus, information items become configured when at least one of their properties is defined or configured, whether this property can be described by numerical, textual or date information; or the state. Sometimes, it is not possible to "configure" the information item with a simple object, such as data type or state. Hence, information item can also be "realized" with a configured item, as a complex property.

Basically, like all other concepts, information item is also an abstract one and is only a placeholder for instantiation. Typically, information items inherit the anonymous classes which determine how they are realized. For example, in case of the production-schedule-item sub-concept of information item, these anonymous classes are defined as (Manchester OWL syntax):

```
has-product-information exactly 1 product-information
has-production-end-date exactly 1 dateTime
has-production-start-date exactly 1 dateTime
```

where "has-production-end-date" and "has-production-start-date" are sub-properties of "has-date-value" data property. "Has-product-information" is a sub-property of "has-realization property". Hence, necessary conditions for having one production schedule item are: 1) to have exactly one product associated; 2) to have a production start date for this product; and 3) to have a production end date for this product.

Similarly, "product-information" information item is configured (hence, its realization is used in the range of first necessary condition above) by having exactly one product id associated:

```
has-product-id exactly 1 string
```

Available states are identified in the analysis of SCOR model and include 25 possible attributes of the configured item, which can be associated to different information and physical items. Some of the examples of the states are: Adjusted, Approved, Authorized, Completed, Delivered, Installed, Loaded, Planned, Released, Returned, Updated, Validated, etc. Many implicit terms of SCOR-KOS OWL model correspond to the configured items of SCOR-Full ontology. For example, following statements (rules) define the relationships between some of the SCOR terms and concepts of SCOR-Full, namely, their states:

```
customer-credit(?x) ∧ in-state(?x, Adjusted) ⇒ SameAs
(?x, Adjust_Customer_Credit)

return-to-service(?x) ∧ in-state(?x, Authorized) ⇒
SameAs (?x, Authorization_to_Return_to_Service)

product(?x) ∧ in-state(?x, Consolidated) ⇒ SameAs (?x,
Consolidated_Product)
```

```
contract(?x) ∧ in-state(?x, Approved) ⇒ SameAs (?x, Ap-
proved_contract)

item-master(?x) ∧ in-state(?x, Approved) ⇒ SameAs (?x,
Approved_Item_Master)

contract(?y) ∧ in-state(?y, ?x) ⇒ SameAs (?x, Con-
tract_Status)
```

Where Inf-Item defines the semantics of the relevant resource, Conf-Item describes its dynamics. Note that SCOR-Full asserts the semantic relation ("realizes (Agent, Conf-Item)") which can be used to infer which Agent is responsible for a particular state of the resource, although this specific information cannot be extracted from the original SCOR model. SCOR-Full will rely on the external enterprise knowledge to fill this and other gaps.

For the expressive process model, it is crucial to define how resources are communicated among activities and their corresponding actors. This knowledge is embedded (explicitly or implicitly) in original SCOR model (in natural language) and is used by SCOR-Full ontology to formalize abstract communicated item (Comm-Item) which aggregates specific concepts of Notice (or its child concept - Signal), Request, Response and Receipt. SCOR model does not provide explicit information about who communicates configured items but this can be inferred by using external knowledge when property chain of

```
performs(Agent, Course)o issue(Course, Comm-Item)
```

is exploited, where former relation is inferred on basis of the mappings with external ontologies and latter – from SCOR-KOS OWL. Necessary conditions for a Comm-Item are that it is issued (requested, responded, notified or received) by a course and that it communicates a configured item:

```
Course(?x)∧Conf-
Item(?y)∧issue(?x,?z)∧communicates(?z,?y) ⇒ Comm-
Item(?z)
```

More specific axioms are set for the sub-concepts of Comm-Item, by using the sub-properties of issue (Course,Comm-Item) property, namely: issue-request(Course, Request), issue-response(Course, Response), issue-notice(Course, Notice) and issue-receipt(Course, Receipt).

Currently, SCOR-Full ontology has 212 concepts and 33 properties and is semantically mapped to the SCOR Input/Output elements.

Mappings of SCOR-KOS OWL and SCOR-Full concepts

SWRL (Semantic Web Rule Language) (Horrocks et al, 2004) is a proposal for a Semantic Web rules-language, combining sub-languages of the OWL Web Ontology

Language with those of the Rule Markup Language (RuleML)[44]. Rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

In order to increase the flexibility of semantics framework, SWRL rules are used for mapping the SCOR-Full concepts to SCOR-KOS OWL instances.

For example, all instances of the business-rule class from SCOR-Full ontology are the same as SCOR Input/Output concept “Business_Rules_For_Return_Processes”, if there exists a return process in SCOR-Full ontology which has a business rule from above, as a setting:

```
business-rule(?x) ∧ return-process(?y) ∧ has-rule(?y,
?x) ⇒ SameAs(?x, Business_Rules_For_Return_Processes)
```

Similar correspondences between implicit terms of SCOR-KOS OWL model and concepts of SCOR-Full ontology are established in following examples:

```
available-to-promise(?x) ∧ time-range(?y) ∧ has-
quality(?x, ?y) ⇒  SameAs (?y, Avail-
able_to_Promise_Date)

capability(?x) ∧ return-process(?y) ∧ has-quality(?y,
?x) ⇒  SameAs (?x, Capabilities_of_the_Return_Processes)
```

Semantic mappings between SCOR-Full and SCOR-KOS enable characterization of supply chain operations managed by using SCOR-Full ontology, in context of SCOR reference model. For example, based on the first above SWRL implication, it can be inferred that a business rule, which is asserted in SCOR-Full ontology as a setting for an instance of the return process, is an output of the SCOR process element ER.01 Manage Business Rules for Return Processes. In the opposite direction, relevant inferences of SCOR-KOS OWL model can result with a formal semantics of the selected SCOR element. Similar implications can be derived from other two examples of the “Available_to_promise_date” and “Capabilities_of_the_Return_processes” SCOR concepts.

Explication of SCOR-Full concepts

SCOR shows lack of expressivity for a complete formal model. One of the evidences is the lack of relationship between metrics and systems, which could point out to the source of information needed for performance measurement. This is obvious limitation of the reference model and it can not be addressed in the process of semantic enrichment, as this relationship does not exist.

However, semantically enriched model facilitates the establishment of the references between formalized systems, system capabilities, intended uses, etc., and goals,

[44] http://ruleml.org/

mapped to the metrics of the SCOR model, by using the external knowledge, formalized in various domain ontologies.

Namely, if there exist systems S_1 and S_2, driven by the ontologies O_1 and O_2 (external knowledge), and if there exist alignment between these ontologies O1≡O2, the competence of O_1 will be improved and S_1 will be enabled to make more qualified conclusions about its domain of interest.

For example, in TOVE organization ontology, the concept of Communication-Link (cl) captures the notion of benevolent communication in which organization agents voluntarily provide information that they believe are relevant to other agents. TOVE organization ontology can be extended with a property chain axiom of the new information-provided-by(inf,oa) relationship, established between the concepts of Organization-Agent (oa) and Information (i):

```
information-provided-by(inf,oa) • inverse(inverse(has-
sending-agent(cl,oa)) o will-volunteer(cl,i))
```

Assertions of the above TOVE relationships can be exploited for inference of the sources of information relevant for measuring the performance of the process elements if the following assumptions hold true: 1) Organization agent is an abstraction of an information system concept; 2) The correspondences between TOVE Information and SCOR-Full Inf-Item instances are established or inferred. 3) SCOR-Full Inf-Item are configured (Conf-Item) and these configurations are mapped to the corresponding goal concepts.

Alignment of SCOR-Full ontology with other relevant ontologies make all the research efforts based on these ontologies complementary with this one, thus, improving the competence of the SCOR-Full ontology. For example, mapping of Location instances to GIS (Geographic Information Systems) ontologies can provide routing services for the shipment companies. Mapping of Product instances and corresponding identifiers to UNSPSC or eClass ontologies can enable customers to identify the suppliers of the substitutable or alternative parts or assemblies. Mapping of Process elements to Partner Interface Process instances in RosettaNet ontology can enable the collaboration between two companies using different standards for modeling and tracking their supply chain processes.

In order to improve the expressivity of the SCOR-Full ontology, it is mapped to OWL representations of TOVE ontologies (resource, organization and underlying activity-state-time ontologies).

TOVE Resource ontology sets semantic relations (and constraints - axioms) between the notions of resource and activity. These relations enable the inference on the commitment of the resources to specific activity, their consumption and availability at given time. Thus, it becomes possible to exploit the above mappings to improve the competence of SCOR-Full ontology and ask additional questions about SCOR activities, such as: Which resources are committed (or available for commitment) to a process element at given time? Is there an alternative to an unavailable resource, to be used by a process element at a given time? Or, more specific: Can the unplanned order for manufacturing of the 10 hydraulic pumps, to be delivered until September 2010, be accepted (in context of available resources)?

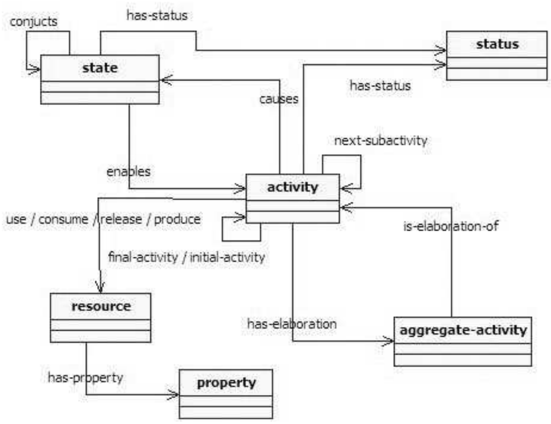

Fig. 38. TOVE Activity-State-Time ontology

Alignment with TOVE Activity-State-Time ontology enables SCOR-Full to infer about the resources associated to an activity, or a process element, by exploiting use(a,r), consume(a,r), release(a,r) and produce(a,r) relations between an activity (a) and a resource (r) concept. These relations represent so-called terminal states and can also be used to imply the pre-conditions and post-conditions of the SCOR-Full activity. Namely, where Conf-Item concept is used to describe a change of the properties of a resource, including their existence and/or a quantity, the above relations represent the type of this change. „Consume“ state is equivalent to a change of the resource, which is used by the activity and will not exist once the activity is completed. „Use“ state imply some (or none) change of the properties in the course of performed action. Both states are classified as enabling states since they are the preconditions for the activities. While „Release“ state of the resource is caused state of the activity whose enabling state is „Use“ of this resource, „Produce“ state indicates that an Resource-Item, that did not exist prior to the performance of the activity, has been created by the activity. These two states are caused states of the activities, and are equivalent to SCOR-Full post-conditions.

Finally, TOVE ontologies can provide information on the contents of the SCOR activity (and potentially provide guidelines for selection of corresponding SCOR Level 4 tasks) by exploiting “conjucts” relation of the enabling and caused states. This relation facilitates the definition of the sub-states of the given state and thus, it defines the conditions-of-the-condition.

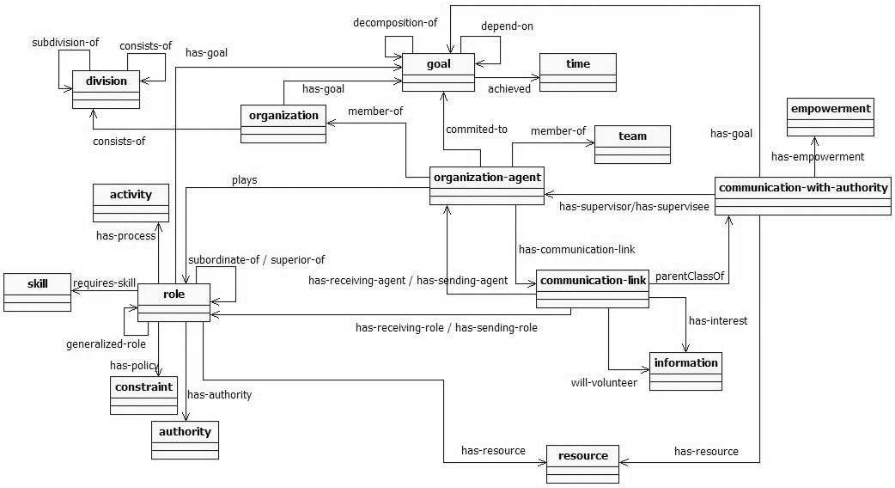

Fig. 39. The portion of TOVE Organization ontology

TOVE organizational ontology (see **Fig. 39**) links the structure and behaviour of the enterprise by using the concept of empowerment. Empowerment is defined as the right of an organizational agent to perform status changing (of a state or an activity) actions. Mappings between TOVE and SCOR-Full facilitate the usage of the external enterprise knowledge to infer about who can define or change the state of Resource-Item and thus, fulfil the precondition of the activity, or about who has authority to perform that activity.

In addition, mappings with TOVE Organization ontology facilitates the improvement of the structural and behavioural (in context of organizational goals) competence of the SCOR-Full model. For example, answers to the following questions may become available: Whose permission (if any) is needed in order to perform the specific task of selected process element (activity)? Who has authority to verify the receipt of the sourced part? Which communication link can be used to acquire specific information?, etc.

4 Semantic Interoperability of systems in supply chain environment

SCOR-Full ontology is expected to support knowledge management in supply chain operations. It classifies concepts and relevant data objects, which can be used in collaborative systems. It enables lookup of data objects, required for consistent and complete definition of supply chain operations concepts. It provides a roadmap for implementation of SCOR reference model. It does not improve the expressivity of SCOR, because it only uses common enterprise notions and proposed generalizations to formalize core concepts of supply chain operations, embedded in SCOR model definitions. However, these generalizations enable alignment of SCOR-Full model with relevant enterprise models, such as TOVE ontology and thus, exploitation of its knowledge for improving the competence of SCOR. Last, and most important, SCOR-Full ontology is expected to facilitate the semantic interoperability of systems, relevant for supply chain networks management.

While SCOR-KOS provides implicit semantics of the supply chain operations by using a semantics representation language, SCOR-Full (and its corresponding mappings with the domain ontologies and SCOR-KOS itself) makes this semantics explicit. Objective conceptualization and corresponding explicit representation of domain knowledge is considered as a main condition for making the relevant systems semantically interoperable. In this Section of the thesis, an approach to the semantic interoperability in supply chain networks which exploits the defined formal framework is elaborated. Also, some practical impacts of the semantically interoperable systems to Supply Chain Management are discussed.

4.1 Description of the approach

Fig. 40 shows the extended view of the formal framework for supply chain operations, presented in this Chapter. The formal framework is developed on the different levels of abstraction. Hence, it results with modular ontologies, which are classified into the layers of implicit and explicit semantics, semantic enrichment and the formal models of design goals – application or problem semantics (see).

In the approach of semantic interoperability of systems in supply chain environment, two application layers are added to the formal framework (see **Fig. 40**). First layer represents EISs, namely the sources of isolated, local islands of semantics, owned by the individual enterprises. The second layer represents Semantic Applications, which are typically implemented by Virtual Breeding Environments (VBE) with aim to support some of its integrative functions, where all individual enterprises from VBE may have benefits from performing those functions or from providing the relevant data for their performance.

Semantic applications actually exploit the correspondences which are established between the formal framework for supply chain operations and islands of the local semantics, represented by the local ontologies, for the joint benefit of VBE and fulfilment of the cooperative goals.

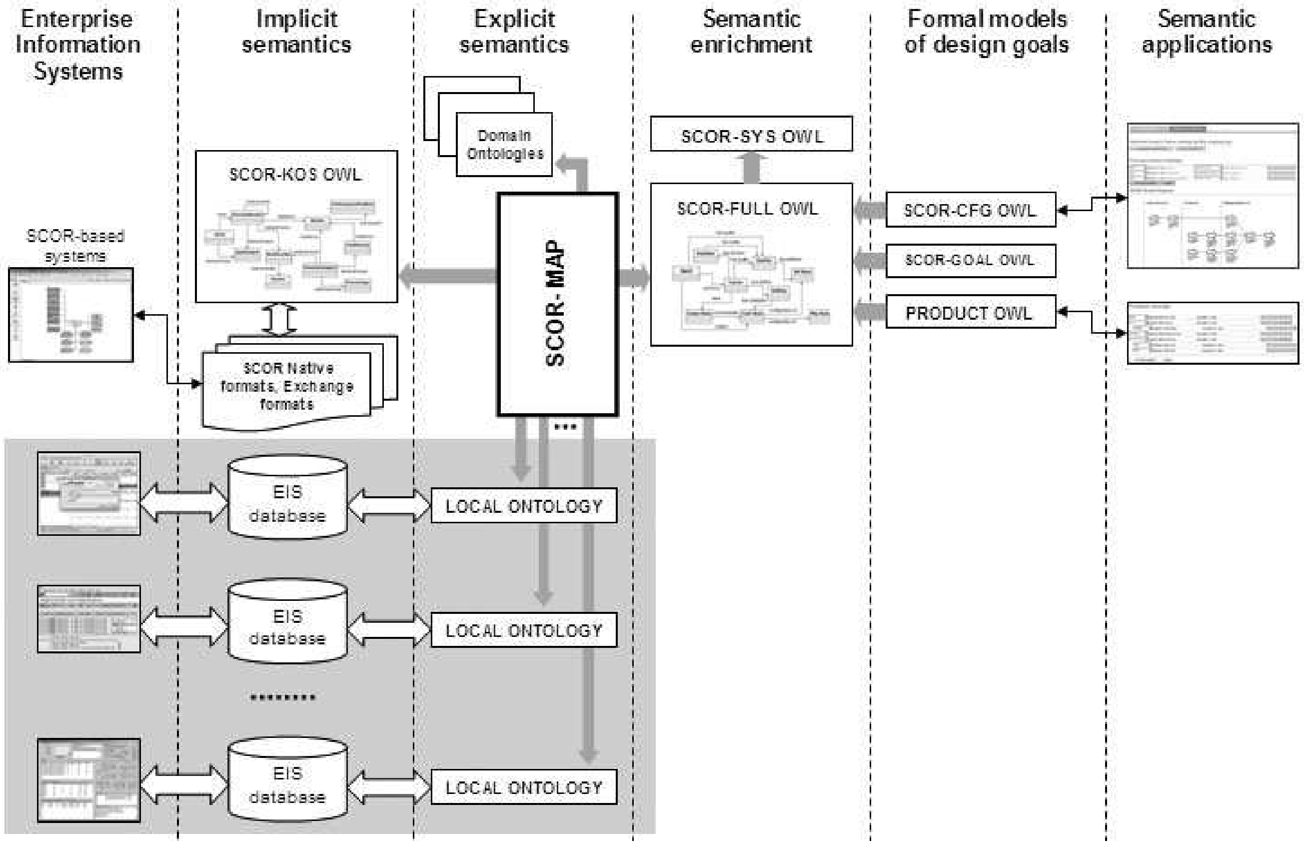

Fig. 40. Semantic interoperability of systems in supply chain network

The local ontologies formalize the implicit data from the heterogeneous sources in order to facilitate the semantic interoperability of the systems which store this data. In order to cope with the implicitness of semantics of the enterprises' realities, it is assumed that: 1) these realities are represented by the corresponding EISs, and 2) enterprise message models are based on EISs' data models, represented implicitly in their databases. The proposed approach aims at making this representation - explicit.

The database-to-ontology method is employed in order to transform implicit Entity-Relationship (ER) models to explicit OWL representations, namely, local ontologies. Then, these local ontologies are mapped to a common, shared knowledge of the enterprise collaboration environment, namely, formal framework for supply chain operations, where different contexts may be added. Each of the contexts corresponds to a domain ontology, whose concepts are logically related to the concepts of the local ontologies. Thus, domain ontology becomes a dictionary – a common knowledge of particular enterprise perspective one can use to query the hidden, implicit knowledge stored in EISs. Hence, single, integrated access to the multiple contexts of the particular enterprise concept will become possible.

Sometimes, Entity-Relationship models, namely database schemas, do not capture the semantics of the application functionality and underlying data models. When information systems are highly generic, the application semantics is actually captured in the populated table rows. For example, in Business Process Management systems, the structure of the enterprise processes, namely activities, associated data structures (messages), compensation and error handling blocks, etc. are defined by a system user and are not expressed by the database schema. In these cases, the intervention of the domain expert in enriching the conceptual model may be useful. Some research is tackling this issue by providing some tools to automatically or semi-automatically discover the semantics buried into existing data patterns (Astrova, 2004).

The above assumptions are made for the purpose of making the process of local ontology creation – automatic. Otherwise, the precondition for this process would be a detailed analysis of the involved EISs. Example of the work which follows this approach can be found in the work of Castano and Antonellis (1998). They "analyzed the process descriptions for the aspects related to information and operation similarity, to evaluate semantic correspondences between processes and identify activity replication and overlapping, as well as for the aspects related to interaction/cooperation, to evaluate the degree of coupling between processes and identify the type and the nature of exchanged information flows".

In this work, the range of semantic interoperability is clearly set to Enterprise Information Systems. The interoperability of the enterprises is considered as more complex problem and is not addressed in this research. The conceptualization of their information systems is made on basis of the business logic, which is hidden in the actual code, in most cases, and data model, represented by the corresponding relational database structure.

The EIS's databases are considered as legitimate starting point for building a relevant local ontology. Obviously, business logic which is encapsulated in the EIS' will remain hidden – only underlying data model is exposed by ontology. The exceptions are database's triggers, which can be considered as business rules, if they are not implemented only to enforce referential integrity of the database.

4.2 Benefits and impact

The stack of the semantic technologies, consisting of informal dictionaries or formal ontologies, their representations, inference engines and semantic applications, provides the means for development and implementation of a new layer of the enterprise systems architecture. The main role of this layer is to make the implicit semantics of the different existing enterprise systems (or underlying reference models) – explicit, and consequently, mutually correspondent. Thus, the layer is expected to enable the semantic interoperability of these systems and facilitate better integration of the heterogeneous environments, such as supply chain networks.

In this scenario, EISs will be represented in the semantic layer by local ontologies – semantically weak representations (OWL models) of the implicit knowledge related to the enterprise, and typically stored in relational databases of the relevant systems and in other data sources. Semantic matching techniques and tools can facilitate the contextualization and explicitation of the individual representations, by helping to establish the correspondences between these representations and relevant formal micro-theories, such as SCOR-Full. Consequently, semantic mappings between SCOR-Full notions and other domain and problem ontologies can be exploited for applying an integrated approach to solving some of the supply chain networks issues.

For example, the partner selection problem can be associated with the definition of the individual semantic query which expresses the sufficient and necessary conditions, regarding the capability, capacity, cost, availability, etc. (SCOR-Full notion of quality) of a specific resource or an agent (among resources and agents of the whole network). Mappings between those notions, used in a query and defining correspondences between concepts in the different local ontologies, expressed as logical functions, ensure that the single query is interpreted correctly in each of the network partners' systems and corresponding data storage facilities. Thus, it becomes possible to use a single query (expressed in a formal semantics) to explore the whole supply chain network, despite the heterogeneity of used systems and their data sources (syntax, data modelling patterns, etc.).

Collaborative process management can be facilitated by monitoring the state (configurations) of the resource items in the semantic layer (by using a software agent), and triggering appropriate actions (e.g. initiating SCOR process elements, or equivalently, launching the process activities) when desired configurations are established. Hence, desired configurations of the resource items, whose parameters are stored as semantic annotations of the process models (generated by the process modelling tool) are continuously compared with the specific entities of the relevant local ontologies, and logically related with those items. Once they become logically same, the software agent would assert a new individual of the Activity type, assign an agent to this individual and set other necessary properties. This change will also be propagated backwards, by assertion of the logically equivalent concepts of corresponding local ontologies and consequently, update of the relevant database(s). Thus, appropriate EISs will be affected by automatic insertion of the work order, web service invocation, issuing of the request for approval (authorization) or similar action.

More details on how above-mentioned semantic layer should be implemented are given in the Chapter 4 of this thesis.

Chapter 4: Implementation issues of the formal framework for semantic interoperability in supply chain networks

Abstract. In the previous Chapter, the conceptual description of the formal framework for semantic interoperability in supply chain networks is presented. It includes the formal models and relationships between those models and technical components of the framework: enterprise information systems, local ontologies and semantic applications. In this Chapter, based on this conceptual description, the implementation issues and corresponding sets of functionalities are identified and elaborated. Then, the service-based approach to commoditizing these functionalities as so-called Semantic Interoperability Service Utilities (S-ISU) is proposed and described. Meta-model of the resulting architecture (S-ISU Ontology) is developed and presented. Special focus is given to the functionalities of translation between implicit semantics of the Enterprise Information Systems and explicit local ontologies; and processing of the semantic queries in the framework.

1 Introduction

In the last section of the Chapter 3, an extended view of the formal framework for semantic interoperability in supply chain networks is presented and described. This view explains the role of the formal models for supply chain operations in achievement of the semantic interoperability within the supply chain. It aligns these formal models with the realities of the enterprises, namely implicit semantics of its EISs.

Scheme at **Fig. 40** in Chapter 3 illustrates this extended view – it lists and relates the main entities of the architecture for achieving the semantic interoperability of systems in the supply chain. Here, the formal models make explicit the common knowledge of the supply chain network. The backbone of this knowledge is SCOR reference model. Then, this knowledge is related to the common knowledge about the enterprises or other perspectives of the supply chain. This knowledge is represented by the domain ontologies. Next, the common knowledge of the supply chain network is contextualized by using problem or application ontologies, which formalize some specific, integrative, shared, commonly used functions of the network in a whole. Finally, individual enterprises are represented in the formal framework by the local ontologies.

The conceptual description of the formal framework above is considered as a starting point for definition of the technical architecture for semantic interoperability of systems in supply chain networks. On basis of this description, the implementation issues are identified and elaborated. The technical architecture consists of two sets of functionalities: first is associated to the VBE of the supply chain network; the second one is related to the individual enterprises. The implementation issues correspond to those sets of functionalities and they are, as follows:

1. Automatic or semi-automatic transformation of the implicit semantics of the EISs to formal local ontology;
2. Scalability of the framework, namely, the process of adding new local ontologies (enterprise registration) and new domain ontologies (dictionary registration, increase of expressivity) to the framework;
3. Automatic or semi-automatic reconciliation of the added local or domain semantics with the existing knowledge;
4. Robust and reliable distributed reasoning;
5. Single point of access to the knowledge framework – the facility which can process "Ask" and "Tell" semantic queries, take corresponding actions and return results in the form of ontology, if relevant.

The implementation issues above correspond to 5 sets of functionalities or capabilities of the formal framework for semantic interoperability in supply chain networks: translation, registration, reconciliation, reasoning and query processing. Each of these sets may be directly or indirectly used by any or all members of the VBE. Thus, it is considered as very important to commoditize those functionalities, namely, to make those sets uniform, accessible and affordable, and thus, easy to utilize by any or all members of the VBE.

Continuous utilization and commoditization has been a basic feature of technology advancement in any field. In the field of IT, this means that the basic functionalities of IT should be made available to all enterprises comprehensively and non-discriminately. In the recent years, it became evident that this will be achieved by providing the IT functionalities by using Software-as-a-Service (SaaS) paradigm. This paradigm revolutionized delivery of software, by developing and introducing new business models, such as rent-a-software or pay-per-use. Hence, it enabled a wide range of choices in the way one enterprise (or any user) are leveraging specific computing (or storage) asset. Technologies of Cloud Computing or Cloud Networking[45] went even one step further, by providing infrastructural services, such as virtual machines or dedicated networks.

Recently, research community launched systematic approach (Li et al, 2006) to commoditization of the Enterprise Interoperability functionalities. The approach is based on a premise that Enterprise Interoperability functions should be delivered as services, in the form of so-called Interoperability Service Utilities (ISU). The general vision of the approach is to have interoperability of enterprises becoming a part of the basic IT functionality, so it can become a fundamental premise that all enterprises can leverage.

The term of Interoperability Service Utility (ISU) (Li et al, 2006) is used to denote the overall system that provides enterprise interoperability as a utility-like capability. That system comprises a common set of services for delivering basic interoperability to enterprises, independent of particular IT solution deployment. The utility metaphor

[45] GENI Project of NSF

is used to indicate that enterprises should be able to expect and afford basic, interoperable IT as a critical infrastructure for operation, just as water or electricity.

In this Chapter, the concept of Interoperability Service Utility is elaborated and customized with purpose to address the implementation issues of the formal framework for semantic interoperability of systems in supply chain networks. First, the concept of ISU and current approaches to ISU development are described. Then, from the aspect of implementation issues above, the functional requirements of Semantic Interoperability Service Utilities (S-ISU) are defined. These requirements were used to elaborate the proposal of the architecture of the S-ISU technical framework. Each of this architecture's components is described and these descriptions are formalized by S-ISU ontology – a meta-model of the architecture for semantic interoperability of systems.

2 Interoperability Service Utilities

ISU is envisaged as a transparent, scalable, vendor-independent infrastructure (system), built upon the paradigm of SaaS, leveraging open standards, available and accessible to all and by all. It is considered as implementation of the capability that is:

- available at low cost,
- accessible in principle by all enterprises (universal or near-universal access),
- guaranteed to a certain extent and at certain level in accordance with a set of common rules,
- not controlled or owned by any single private entity.

The ISU is conceived to be a basic "infrastructure" that supports information exchange between diverse knowledge sources, software applications, and Web Services.

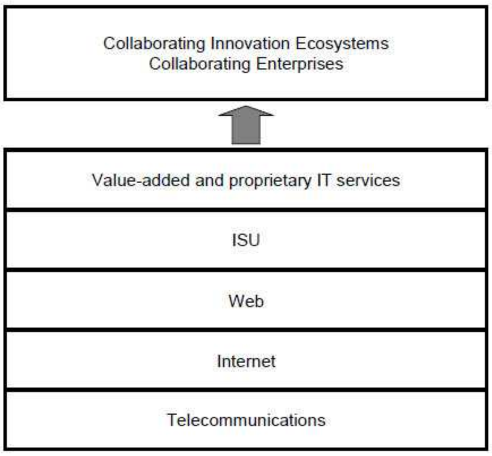

Fig. 41. Conceptual view of ISU.

It will make use of the new generation of Web technologies and enable knowledge-oriented collaboration. Conceptually, the ISU constitutes the next "layer" of open cyberspace, as depicted in **Fig. 41**.

The ISU implementation assumes several principles, of which the most important is functional decentralization. It implies peer-to-peer communication and intelligent end-points. Latter is a proposed solution for the assumption that precise location of services and the means to access them will not be pre-determined. Second, ISU must leverage open standards and specification and its architecture will be based on the modular software blocks, avoiding hierarchical layering. Third, its architecture must be transparent, so it is possible to build additional, value-added capabilities on the top. Fourth, the environment and conditions in which the message transactions between services occur must be clearly defined, predictable and uniform. Fifth, key feature of the ISU architecture is scalability. This concerns reliable information propagation across multiple systems to a growing number of end-points but also inter-working with and transitioning from existing systems.

The ISU aims to provide and guarantee accessible interoperability infrastructure to the enterprises. This infrastructure includes services, such as basic information exchange over the Internet, transparent semantic reconciliation, handling quality of service, etc. Some potential ISU services are:

- Services that facilitate real-time information sharing and collaboration between enterprises such as reasoning, searching, discovery, composition, assembly, and automatic delivery of semantics;
- Services that leverage emerging Web technologies for enabling a new generation of information-based applications that can self-compose, self-declare, self-document, self-integrate, self-optimize, self-adapt, and self-heal;
- Services that support knowledge creation, management, and acquisition to enable knowledge sharing between virtual organizations;
- Services that help connect islands of interoperability by federating, orchestrating, or providing common e-business infrastructural capabilities such as digital signature management, certification, user profiling, identity management, and libraries of templates and interface specifications;
- Services that support the next generation of e-business services such as verification of credentials; reputation management; assessment of e-business capabilities; assessment of collaboration capabilities; facilities for data sourcing, integrity, security and storage; contracting; registration and labelling; and payment facilities, among others.

2.1 Current approaches to ISU development

Although the challenge of Interoperability Service Utilities has been set in 2006, the literature analysis shows that there are only a few attempts to address this challenge and describe and deliver practical interoperability solutions on the top of this concept.

In this thesis, four researches are shortly described and presented, indicating the current understanding of ISU concepts in the research community and setting the first cases and consequent evidence on their applicability. Those researches correspond to the works on: 1) developing ISU platform for automobile Supply Chain Management; 2) developing iSURF ISU for semantic mediation of planning documents; 3) developing e-mail-based ISU for small and medium enterprises; and 4) using ISU paradigm to define the corresponding issues of ATHENA interoperability framework implementation. With the exception of the first work, all others are performed in the scope of the FP7 projects, funded by EC, respectively: iSURF, COMMIUS and COIN.

ISU platform for automobile Supply Chain Management

Zhang et al (2008) developed an Interoperability Service Utility platform for automobile Supply Chain Management. In the platform, interoperability is considered to be a utility-like capability and delivered in the form of SaaS. Zhang et al specified these ISUs and proposed an interactive framework which is used to establish interoperability between two of those, namely Supply Business Management (SBM) and Advanced Planning and Optimization (APO). SBM service is expected to help the assembly factory to deal with businesses related to suppliers such as bill inquiry, inventory management, and payment management, etc. With APO, companies can optimize their supply chains to reduce costs, improve product margins, lower inventories, and increase manufacturing throughput. APO necessitates deciding when to build each order, in what operation sequence, and with what machines to meet the required due dates. In real world, most of the SMEs can't afford the expensive software systems with the same function of SBM and APO. Therefore, the goal of the ISU platform is to facilitate SMEs' participation to collaborative Supply Chain Management processes by invoking SBM service and APO service on the fly.

Fig. 42 illustrates the architecture of ISU platform. The work of the services is facilitated by the data layer, while portal serves as the presentation layer and single point of a user access to the Virtual Breeding Environment. ISU services layer is the most important in the platform and it currently contains SBM, APO, SMS and conferencing services. In the composite services layer, the existing services are dynamically composed according to the identified or occurred circumstances of production.

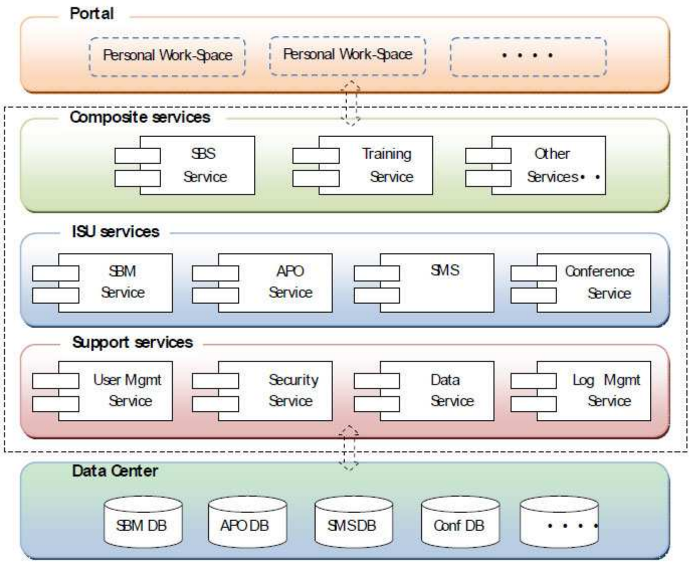

Fig. 42. Architecture of the ISU platform.

The associated methodology allows establishing interoperability by: (1) constructing a Virtual Enterprise by identifying and involving various actors and stakeholders; (2) dynamically composing available ISUs according to identified requirements; (3) evaluating and improving the interoperability solution in practice.

ISU for semantic mediation of planning documents.

The main functionality of iSURF ISU (Dogac et al, 2008) is to perform the semantic mediation of planning documents across enterprises by using a common denominator, OASIS (Organization for Advancing open Standards for the Information Society) UBL documents.

Universal Business Language (UBL) is a framework consisting of library of standard electronic XML business documents, such as purchase orders and invoices and customization methods. In order to provide semantic interoperability and mediation, iSURF ISU has the capability of translating UBL documents (Yarimagan and Dogac, 2009) of one enterprise to another. The assumption is interconnection of EISs with the collaborative planning environment, where legacy applications are wrapped as semantically annotated web services. In order to avoid the bottlenecks of the centralized architectures, iSURF ISU is designed to perform semantic mediation on a distributed architecture. Namely, the tasks are distributed among multiple servers for balancing the workload.

The infrastructure of iSURF Interoperability Service Utility is illustrated in the **Fig. 43**. The most critical building block in iSURF ISU is the semantic mediator. Since UBL customizations of partners in the supply chain are independent from each other and might be industry specific; different planning document structures are created in each enterprise through the iSURF semantic UBL Customization Tool. In order to provide semantic interoperability and mediation of these documents, iSURF ISU can translate UBL documents of one enterprise to another. In the semantic mediation of UBL documents, iSURF ISU uses intelligent algorithms and description logic reasoning services, based on the semantic annotations made in the UBL customization phase. In order to avoid the bottlenecks of the centralized architectures, iSURF ISU is designed to perform semantic mediation on a distributed architecture in which the tasks are distributed among multiple servers for balancing the workload.

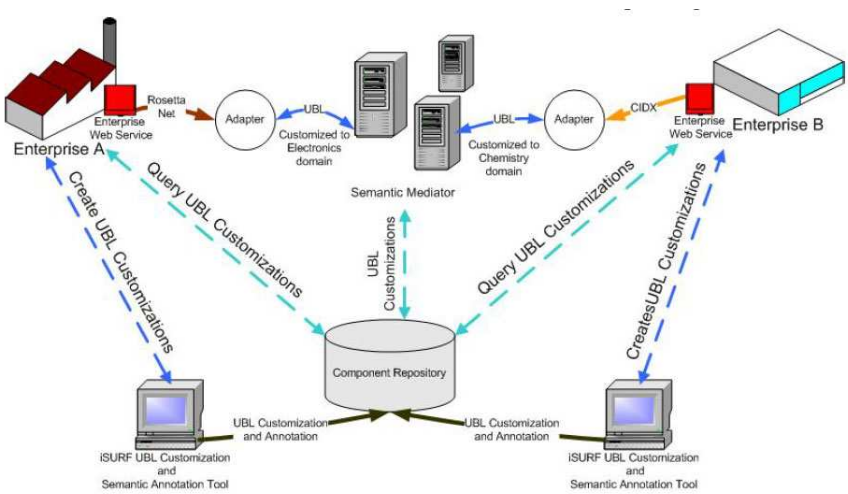

Fig. 43. iSURF Interoperability Service Utility architecture.

In the architecture, web services are used for the communication between the enterprises for document exchange. Achieving the communication via web services contributes to the platform independence for the enterprises in the supply chain. Existing services of the enterprises are designed to be exposed as web services so that the legacy applications will not have to be re-implemented or modified. Furthermore, the web service operations are also semantically annotated to facilitate discovery of the services.

iSURF ISU platform is extended (Kabak et al, 2009) to provide interoperability services to all CCTS (UN/CEFACT Core Components Technical Specification) standards based documents. CCTS provides a methodology to identify a set of reusable building blocks, called Core Components to create electronic documents, such as UBL, GS1 XML and OAGIS. Universal Business Language was the first implementation of the CCTS methodology in XML. CCTS based document standards are not interoperable (because they apply the CCTS methodology differently) and it still requires experts to discover the correspondences between document artefacts and to map them. So, semantics of CCTS is defined through a formal, machine processable language as ontology and the Web Ontology Language (OWL). This is considered as “upper” ontology. Other developed “upper” ontologies are GS1 XML, OAGIS and

UBL (standards derived from CCTS). Then, on basis of the "upper" ontologies, document schema ontologies are developed, for different schemas. Finally, from this ontological framework, relationships between different artefacts of different standards can be inferred.

Recently, the platform is upgraded to support CPFR (Collaborative Planning, Forecasting, and Replenishment) guidelines (Kabak et al, 2009) and thus, facilitate semantic reconciliation of the CPFR related documents and enterprise planning applications formats. Collaborative Planning, Forecasting, and Replenishment (CPFR) guidelines[46] describe collaborative business practices which enable the trading partners to have visibility into one another's critical demand, the order forecasts and the promotional forecasts through a systematic process of sharing planning information, exception identification and resolution. The main objective of CPFR is to increase the accuracy of demand forecasts and replenishment plans, necessary to lower inventories across the supply chain and attain high service levels by making right products available at right locations. CPFR proposes a planning process which involves a number of transactions between partners exchanging planning documents with each other. Although CPFR provides guidelines, there is no machine processable process templates defined. Also, CPFR does not mandate any technology to implement the CPFR approach. iSURF ISU achieves the semantic reconciliation of the planning and forecasting business documents exchanged between the companies according to different standards. Now, these standards include CPFR.

Finally, iSURF is also associated to the resolution of the systems' visibility gaps. In the supply chain, the internal planning and scheduling systems base their decisions on inaccurate and out-to-date data which results in sub-optimal decision-making in the whole supply chain. Thus, RFID technologies are employed. Smart Product Infrastructure (SPI) enables SMEs to collect real-time product visibility events from massively distributed RFID devices; filter, correlate and aggregate them in order to put them into business context. Through the integration of iSURF with Smart Product Infrastructure (Dogac et al, 2009), most of the manual operations related to the visibility of the product in the supply chain are expected to be eliminated, by applying the product coding reconciliation system. Some applications include RFID based auto inventory system, non-intrusive anti-theft system, item tracking system and a system facilitating automatic match of order document with transport document.

E-mail-based ISU

The main motivation of the e-mail-based ISU (Truong et al, 2009) development is the fact that SMEs cannot afford to have skilled IT to manage a complex network of SMEs or to take a long learning curve to master complex interoperability solutions. SMEs need simple, almost zero-cost solutions.

To implement the utility-like capability, this work relies on email communications. The rationale is that email systems, based on SMTP and with diverse email clients available, are widely used in most SMEs and are Internet-scale. The architecture of the Commius framework is designed as an open, secured and customizable system,

[46] "VICS, CPFR-An Overview", 2004

supporting networks of cooperative SMEs to perform their daily business based on emails and Web. Examples of the tasks in which this framework may help are an invoice asking for a specific document, a simple reply to requests or surveys, order to services and hardware suppliers, etc. Such tasks require interoperability solutions spanning from system to data/semantic to process layers and everything is done via emails.

Fig. 44 illustrates the architecture of Email-based ISU for SMEs. User tools rely on existing email tools and Web browsers and do not require any modification or plugins for email tools and Web browsers. Email Gateway Plugin is responsible for intercepting and post-processing emails. Modules and Module Management include system and business-related modules that handle interoperability tasks to fulfill the request of emails passed through the Commius.

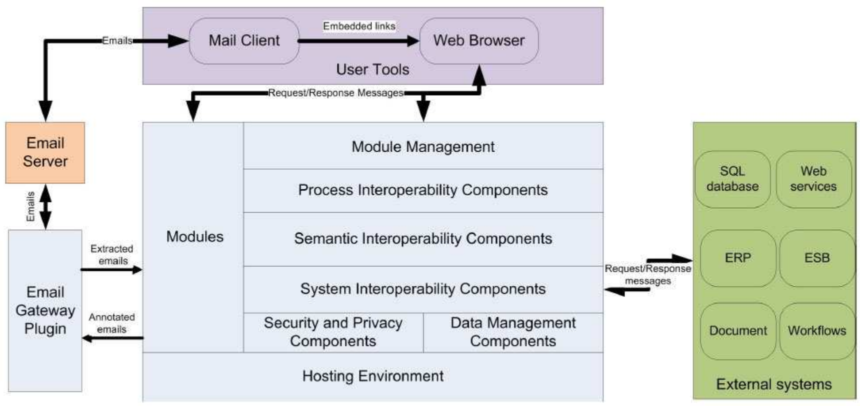

Fig. 44. Architectural overview of Email-based Interoperability Service Utility.

System Interoperability components support functionalities for achieving system interoperability, such as providing a basic interoperability infrastructure over SMTP to extract information from emails and annotate emails with new information and to integrate Commius with external systems in order to access legacy/external information and services. Semantic Interoperability components include components providing functionalities for achieving semantic interoperability, such as providing facilities to achieve semantic alignment, facilitate concept negotiation, and to annotate messages with meta-data to embed semantics within them. Process Interoperability components offer functionalities for supporting process interoperability, such as features to configure process modules, to match and adapt business processes.

The core entities of the Commius can interact with external systems, which are not part of the Commius. External systems include common/specific legacy systems in SMEs as well as other services supporting the business of SMEs.

When an email arrives, the Email Gateway Plugin will extract metadata from the email and its attachments, such as sender and receiver information, existing Commius embedded header information, and relevant keywords. The extraction is performed by applying regular expression rules which are predefined and updated regularly. The extracted metadata, a set of keys and values, is then enriched with other metadata related to business activities. This enrichment is achieved by using semantic and process interoperability components to analyze the extracted metadata.

Based on the enriched metadata, the Module Manager performs a matching process to select the right module to handle the email. This matching process utilizes various sources of information, including extracted keywords, module description, historical data, rules, and components, and in particular, semantic information. For example, based on pre-defined rules and metadata a module can be selected. This happens when the user knows for sure which modules should process which emails. When selecting a module, the Module Manager can also interact with the user, in case it cannot decide the right module (e.g., due to rule conflict or missing information). In this case, the Module Manager will inform the user with an email including embedded links.

When a module is identified, the email and enriched metadata are forwarded to the module which processes the request. Here, in the matching process, the Module Manager treats modules as black boxes. Then it obtains the results from the modules and passes the result to the Email Gateway Plugin which combines and prepares the result in a right format before sending the resulting email(s) to appropriate recipients.

ATHENA Enterprise Interoperability Services

Previous three referenced works are development-oriented. Moreover, a bottom-up approach is applied, in the sense that the developed frameworks and tools are fitted into the paradigm of Interoperability Service Utilities.

The work of Elvesæter et al (2008) is considered as top-down approach. They identified Enterprise Interoperability Services on the basis of the ATHENA Interoperability Framework and defined the functional requirements for their architecture.

Their approach uses ATHENA framework's dimensions, to classify interoperability services into:

1. Model-driven interoperability
   - Model Transformation Service Engine, based on MDA technologies, which will provide functionality for storing, searching and executing model-to-model and model-to-text transformations, in order to overcome the incompatibilities between different modeling formalisms,
2. Enterprise modelling interoperability
   - Enterprise Model Interchange Service based on POP* meta-model - a flexible intermediate language that facilitates model exchange between different enterprise modelling tools;
   - Enterprise Interoperability Maturity Assessment Service to assess and improve the level of interoperability,
3. Business process interoperability
   - Cross-Organizational Business Process (CBP) Modelling Service;
   - Semantic Business Process Modelling Service that deals with enrichment of existing business process models with semantic annotations;

– Semantic Business Process Management Service manages the life-cycle of deployed business process models independently of the underlying process engines actually executing the model,

4. Service interoperability (WSMX),
5. Semantic mediation interoperability
   - Semantic annotation service;
   - Semantic mediation and reconciliation service;
   - Semantic mapping assessment service
6. Information and data interoperability
   - Transactional Data Interoperability Service concerns the exchange of information between two distinct actors;
   - Massive Data Interoperability Service concerns the exchange of information among multiple actors.

3 Semantic Interoperability Service Utilities (S-ISU)

One of the design principles of ISU is that it exploits services. Thus, in general, interoperability becomes intentionally restricted and partial because it depends on their scope and functionality. The restrictions of service-oriented approach can be considered in two aspects.

First aspect is related to the scope and availability of existing enterprise services which is a precondition for ISU implementation. Second aspect of the conditionality is related to variety and diversity of interoperability services.

For example, in the work of Zhang et al, ISU services layer is organized in a functional way, where SBM (Supply Business Management) and APO (Advanced Planning and Optimization) ISU services are considered as the most important. It's questionable whether those two services should be considered as interoperability services, at all. Namely, SBM and APO expose certain business functions and processing capabilities to the public (actually, to the Virtual Breeding Environment), but they do not provide the interoperability capabilities. Although SBM and APO may be used in the platform to build composite services, those composite services would be of restricted functionality, due to the lack of variety of the building blocks.

Although the interoperability restrictions are not direct implication of the functional organization approach, it is obvious that, in this case, the scope of interoperability between two systems will depend on the variety of available functional interoperability services.

The scope restriction is even more evident in the work of iSURF ISU development. iSURF platform is based on the document models. Hence, its purpose can be considered more like syntax than semantic interoperability. Semantic mediation is the only ISU service in iSURF. It reconciles the models of very low level of abstraction. This, bottom-up type of approach contradicts to a usual practice of ontology engineering. However, bottom-up approach also has many advantages, which are discussed previously in this thesis.

Commius approach of using SMTP infrastructure to automatically interpret and consequently process email messages does not actually address systems interoperability. It provides one solution to achieving interoperability within business communities, such as VBEs, where human communication is dominating over systems' collaboration.

In this thesis, an approach to developing the formal and technical architecture for semantic interoperability of systems is proposed. This proposal goes beyond the existing work in the following aspects:

- The proposed architecture takes into account the restrictions of the functional approach and it assumes that enterprises should take their own decision (based on their interests, needs and requirements) on which part of their semantics should be made interoperable;
- This semantics is described by the local ontologies. The main objective of the framework for semantic interoperability of systems is to make those ontologies interoperable;
- Minimum technical pre-requirements are foreseen for each enterprise which wants to take part in the interoperable world of the Virtual Breeding Environment;
- The formal framework is not associated with some storage facility; the formal framework facilitates delivery of the information by combining their sources (namely, local ontologies). Only meta-information (other than a formal framework - common ontologies) about the interoperable systems is kept centrally;

In this section, the requirements for semantic interoperability of systems are analyzed and Semantic Interoperability Service Utilities are identified. Also, certain design decisions about the conceptual architecture are elaborated. Then, identified service utilities are described in the architectural context and inter-related. This description is formalized in the S-ISU Ontology for semantic interoperability of EISs. Finally, each of the service utilities is described in detail, with general focus on the services for local ontology generation and semantic querying of the overall platform.

3.1 The functional analysis of Semantic Interoperability Service Utilities (S-ISU) framework

The focal problem of semantic interoperability of systems is identification of the logical correspondences between two models, where one of those models is implicit representation of the enterprise (or one of its contexts) knowledge and the another is explicit model of the enterprise or some of its functions.

Hence, the most important service in the S-ISU architecture is Semantic Reconciliation Service. The process of recognition and, in some cases, assertion of the relations between the concepts and individuals of two ontologies, corresponds to the ontology operations: merging, mapping, alignment, refinement, unification, integration or inheritance. Those tasks are difficult and cannot be performed automatically in non-trivial cases. Typical reasons are usage of very expressive languages which may result with undecidability or insufficient specification of conceptualizations for finding similarities between those. Obviously, Semantic Reconciliation Service must be coupled with client software, which needs to facilitate review and approval of suggested generated mapping axioms, as well as manual assertions.

In semantic interoperability architecture, an enterprise is introduced by its local ontology or ontologies. The local ontology may be any formal model of the enterprise or any of its contexts, which describes some reality of an enterprise, and with which the enterprise wants to be represented in the interoperable world. Introduction of the enterprise is enabled by the Registration Service. It facilitates declaration of the local ontology (or ontologies) location and rules for semantic queries handling. Namely, enterprise may decide to unconditionally restrict access to specific information (subgraph) in the local ontology. Or, enterprise may want to be capable to manage access to particular information per request in the process of query execution. It is important to note that, in latter case, the process of semantic querying will become asynchronous. Registration Service is also used for registering the domain ontologies. These ontologies describe different perspectives to an enterprise or one of its contexts. For example, they may be used to specify the conceptualizations of the standard dictionaries' implicitly defined concepts.

The local ontology is representation of the implicit semantics of an enterprise. If we assume that the realities of an enterprise are stored in the corresponding EISs, we can identify their relational databases and other data storage facilities as valid sources of this semantics. Some arguments for this assumption are described in Section 4.1 of Chapter 3. These databases need to be exposed in a certain way, in order to enable the transformation of the implicit enterprise knowledge they contain to a valid local ontology. Thus, the Transformation Service Utility is identified as an element of S-ISU architecture. This utility is already developed and described in detail in Section 3.3 of this Chapter. The approach enables the complete (from the aspect of OWL expressivity) explicitation of the implicit semantics of the ER model, as well as full correspondence between semantic and database queries. This correspondence is exploited in the design of the Semantic Query Service.

The Semantic Query Service is considered with "Ask" and "Tell" interfaces, enabling extraction of relevant instances and assertion of new ones in designated local ontologies. Semantic Query Service is a single point of access to the overall knowledge of the interoperable world. Its "Ask" interface accepts semantic (e.g. DL – Description Logics) queries in the form of a pair (O, C), where O is a set of concepts which need to be inferred and C - a set of restrictions to be applied on their properties, namely value and qualified cardinality restrictions, and cardinality constraints. When mappings between registered local and domain ontology(s) are consistent and complete, one can use the dictionary of the domain ontology(s) to build semantic queries, without any knowledge on the underlying semantics of the enterprise local ontologies (Zdravkovic et al, 2011). The "Tell" interface of the SQS, accepts semantic queries in the form of a triple (A, C, U), where A is a set of assertion statements, C - a set of conditions represented by a common dictionary(s) concepts and U – identifier of the local ontology where assertions need to be made. More details about how the Semantic Query Service Service works can be found in Section 3.4 of this Chapter.

In a distributed environment, a reasoner may be viewed as a self-contained component. It is used when semantic queries are issued or in the process of semantic reconciliation. A first step towards the provision of reasoners that can be deployed in a distributed architecture is the Description Logics Implementation Group's (DIG) specification of the DIG Interface (Bechhofer and Patel-Schneider, 2006). The DIG Interface provides an implementation-neutral mechanism for accessing DL reasoner functionality and is supported by the most of the frequently used reasoners. It accepts HTTP requests and responds accordingly with the content defined by an XML schema. Since DIG is simply a protocol that exposes the reasoner, it does not support stateful connections or authorization. Hence, a Semantic Reasoning Service is anticipated in the architecture, to be implemented on the top of the DIG interface with provision of functionality which is not inherently supported. Almost all the work on semantic reasoning still assumes a centralized approach where all inferences are carried out on a single system. However, transfer of the complete model to a central reasoner takes time and reasoning systems have limited performance (Schlicht and Stuckenschmidt, 2009). There are different strategies (Bonacina, 2000) for parallelizing logical inference, which can be used for its implementation. Thus, Semantic Reasoning Service is envisaged as distributed service.

On the basis of above analysis, the architecture for achieving the semantic interoperability of the EISs, namely, S-ISU architecture is proposed. It consists of the ontological and utility frameworks, located and exploited centrally or locally. Here, the terms "central" and "local", imply distributed component infrastructure, where some of its assets are located behind the enterprises firewalls, while others are shared by the pool of enterprises, or owned by its broker. **Fig. 45** shows the component view of the S-ISU interoperable world's architecture.

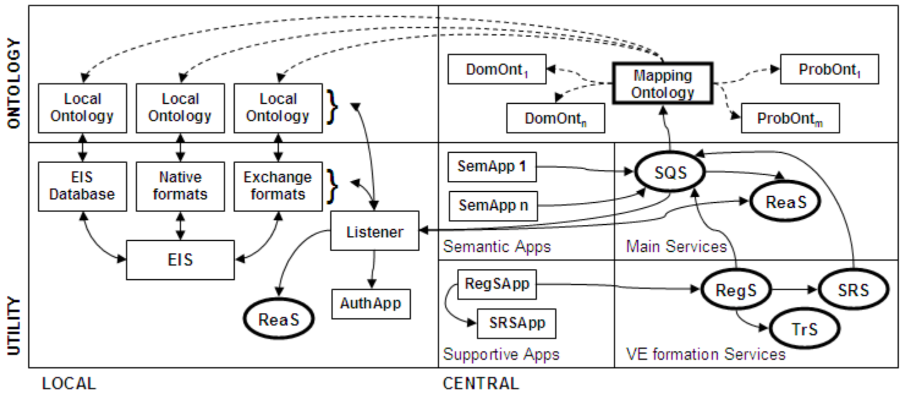

Fig. 45. Component view of the S-ISU architecture.

Locally, enterprises introduce their implicit semantics, residing in the EISs' databases, native and exchange formats, etc., to the interoperable world, by using local ontologies. They are mapped then to an arbitrary number of centrally stored domain ontologies ($DomOnt_{1\text{-}n}$), which formalize the dictionaries, so one can query the local ontologies of unknown structure, by using terms from known models.

At the central level, so-called application, or problem ontologies ($ProbOnt_{1-m}$) are introduced. They are used to formalize specific, integrative functions of the Virtual Enterprises, e.g. collaborative business process management or biding. Problem ontologies are used then by the shared semantic applications which facilitate these functions ($SemApp_{1-m}$).

On **Fig. 45**, ontologies are mutually related by import relations (dashed lines). Other relations between components are of “uses” type. In this architecture, we distinguish between the services which are used during the lifecycle of the Virtual Enterprise, namely Semantic Querying Service (SQS) and Reasoning Service (ReaS) and those which are used only once, in the process of the Virtual Enterprise’s formation, namely, Registration Service (RegS), Semantic Reconciliation Service (SRS) and Transformation Service (TrS). As mentioned before, the supportive applications (RegSApp and SRSApp) are introduced in order to facilitate a human involvement in the processes of registration and semantic reconciliation. Their inner workings are considered as trivial, so they will not be discussed in detail.

Single point of access to an interoperable world is provided by Semantic Querying Service, namely its “Ask” and “Tell” interfaces. They accept the semantic queries, where these queries may be built by a user, a semantic application or another service. Upon receive, the “Ask” query (built by using one of the registered dictionaries) is interpreted “in the languages” of each of the registered local ontologies. This translation is done by the Reasoning Service, based on the mappings between used dictionary and the local ontologies. Then, the local queries are launched concurrently. Local query execution is performed by the listeners, local components of the S-ISU architecture. They accept the individual requests for information and launch the queries. If the enterprise decides to host a dedicated reasoner, then it is used for inference of the query results. Otherwise, a central reasoning service is invoked. Based on the access rules, the results (OWL triples) enter the approval procedure (facilitated by Authorization Semantic application – AuthApp, to approve or deny requests) or are immediately delivered back to the Semantic Querying Service. In a former case, process of semantic querying is asynchronous. Hence, it is performed by the Semantic Querying Service in different “request” and “receive” threads.

3.2 S-ISU meta-model

The architecture described above is formalized by the S-ISU ontology. The S-ISU ontology is illustrated on **Fig. 46**.

Fig. 46. S-ISU Ontology

The main concept of S-ISU ontology is a Component, which classifies Interface, Data-Container and Utility concepts. Other top level concepts are Actor, Process, Data and Function (only used to aggregate natural language descriptions of the functions).

In the context of interoperability, an Interface is the main functional component of S-ISU. Data-Container is any component which involves some kind of data persistence, asserted by "stores" relationship, and aggregates the concepts of Database, File and Ontology. Utility is an abstract concept which subtypes are Enterprise-Information-System, Listener, Semantic-Application and Service and their instances are expected to be directly asserted to S-ISU ontology. An Actor is defined as something that uses some utility. It classifies employees, departments, enterprises, Virtual Breeding Environments (VBE) and Virtual Enterprises (VE), while additional properties describe relationships between those. These relationships may be used to infer the accessibility of a particular utility by specific actor, based on the ownership and collaboration properties.

More important, the relationships can point out where interoperations between enterprises in a VE take place. Namely, a VE is considered as a set of processes, configured by simple precedence relations. Then, VE is assembled of the enterprises which implement its processes. Thus, partnership relation of the enterprise in specific VE is inferred as a property chain:

```
implements-process(Enterprise, Process) o is-process-
of(Process, VE)
```

Each of the processes is assigned to an individual enterprise in the process of VE formation, while additional assertions are made to declare which EISs, owned by the enterprise facilitate the specific process.

Interoperations between two enterprises occur when a process, owned by one enterprise, precedes (or succeeds) the process of another. Hence, enterprise interoperation relationships may be inferred by using SWRL rule:

```
Process(?p1), Process(?p2), Enterprise(?e1), Enter-
prise(?e2), implements-process(?e1,?p1), implements-
process(?e2,?p2), precedes(?p1,?p2), Different-
From(?e1,?e2)->interoperate-with(?e1,?e2).
```

Key concepts and properties for making this inference are presented at **Fig. 47**a. **Fig. 47**b shows example processes (with asserted precedence relationships) of the VE for snow making facility engineering, assembled of three enterprises, where implements-process property is illustrated by the pattern of the enterprise and process individuals.

Based on a rule above, following inferences are made:

```
interoperate-with('Pumps-Inc', 'Snow-Solutions-Inc'),
interoperate-with('Lenko-Snow-Inc', 'Snow-Solutions-
Inc'),
interoperate-with('Snow-Solutions-Inc', 'Lenko-Snow-
Inc'),
interoperate-with('Snow-Solutions-Inc', 'Pumps-Inc').
```

The last top-level concept, Data is considered as anything that is exchanged between the utilities, in specific, their interfaces and stored in some Data-Container.

A functional unit of the service utility is its interface. Thus, service may be multi-functional, depending on the interface(s) it implements. In that sense, role of the service in S-ISU architecture is attributed, not given and is inferred as:

```
Service and has-interface some (has-function value
'<literal>')
```

, where literal describes the function (‘function-reasoning’, ‘function-transformation’, etc.).

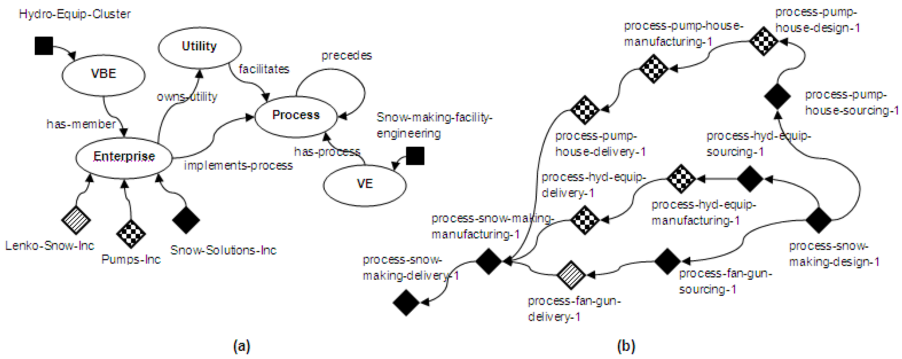

Fig. 47. Organizational view of an example S-ISU Ontology

While **Fig. 47** illustrates portion of the organizational view of S-ISU ontology, component architecture is described by its asset view, generated by dependency relationships inference. Dependency analysis is generated by inferring “uses” relationships between the components of S-ISU, on basis of the asserted sub-properties of transitive 'uses' property, such as imports(Ontology, Ontology), uses-data-container(Utility, Data-Container) and uses-utility(Utility or Actor, Utility). Where latter is used to assert the interoperation relations between generic utilities in S-ISU, interoperation between registered (asserted) EISs is inferred by using SWRL rule:

```
EIS(?u1), EIS(?u2), Process(?p1), Process(?p2), Enter-
prise(?e1), Enterprise(?e2), facilitates(?u1, ?p1), fa-
cilitates(?u2, ?p2), implements-process(?e1,?p1), imple-
ments-process(?e2,?p2), precedes(?p1,?p2), Different-
From(?e1,?e2)->system-interoperate-with(?u1,?u2).
```

It is important to note that interoperation properties are not symmetric, because the semantic interoperability of systems is considered as unidirectional.

Dependency analysis is demonstrated on the example of the snow making facility manufacturing supply chain, where ontological framework for semantic interoperability, based on SCOR model is applied. Asset view of the S-ISU architecture in this case is shown on **Fig. 48**. The illustration distinguishes between asserted components (depicted by rhombs of different patterns, depending on the ownership of the corresponding components) and generic components of S-ISU (depicted by squares), both individuals of S-ISU ontology.

Also, membership of the individuals to S-ISU concepts (Oval symbols) is asserted and shown on the figure (solid line). Asset perspective of the S-ISU architecture on **Fig. 48** is illustrated by example supply chain, where three enterprises are interoperating in the organizational context, shown on **Fig. 47**. In this case, enterprises expose ERPNext’s MySQL and OpenERP’s PostgreSQL databases and EasySCOR system native format to Virtual Enterprise for snow making facility manufacturing, by using local ontologies: ERPNext-1-Ont, OpenERP-1-Ont and SCOR-KOS OWL, respectively.

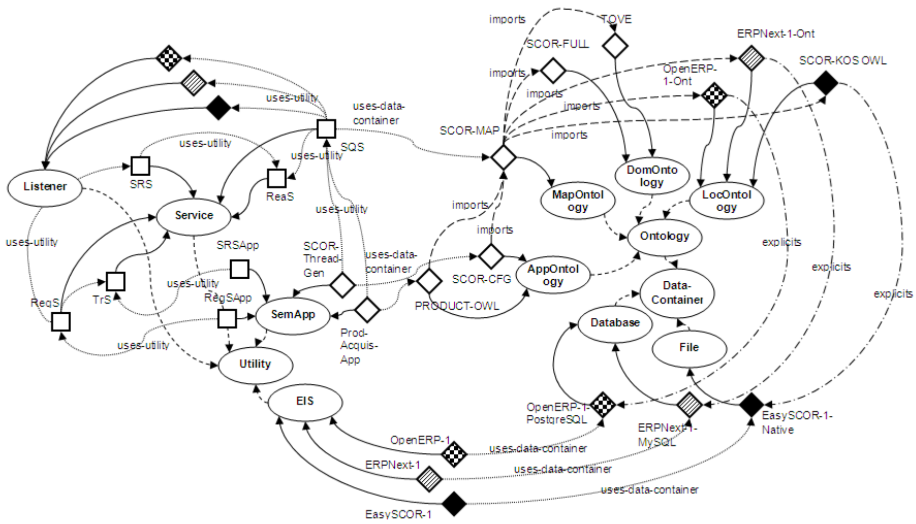

Fig. 48. Asset view of the S-ISU architecture.

In this example, two shared semantic applications are facilitating the VE's lifecycle, namely, SCOR-Thread-Gen, for supply chain process configuration; and Prod-Acquis-App for acquisition of product requirements, where respective problems are modelled by two application ontologies: SCOR-Cfg OWL and PRODUCT-OWL. Both applications are using Semantic Querying Service to assert to or infer about the implicit knowledge in local ontologies, by using two dictionaries: TOVE Enterprise Ontology and SCOR-Full – semantic enrichment of the SCOR reference model.

In the next sub-sections, each of the services from the S-ISU architecture is described. The emphasis is made on already developed and implemented services – Transformation Service and Semantic Querying Service.

3.3 Transformation Service

Database to ontology mapping is a process in which the implicit semantics of a database schema is correlated to the explicit and formal knowledge structure of the ontology. In this thesis, the database schema is used to generate this formal structure, while the logical mappings between ER meta-model and generated local ontology are preserved. These mappings will enable the translation of semantic to database queries.

Generation process consists of 4 phases:

1. data import and classification of ER entities;
2. classification (inference) of OWL types and properties;
3. lexical refinement;
4. generation of local ontology.

The process is illustrated on **Fig. 49**. It is supported by a web application, developed by using previously described RAP API. Web application consists of modules for data import/assertion of ER meta-model instances, lexical refinement and transformation of classified OWL types and properties to a local ontology.

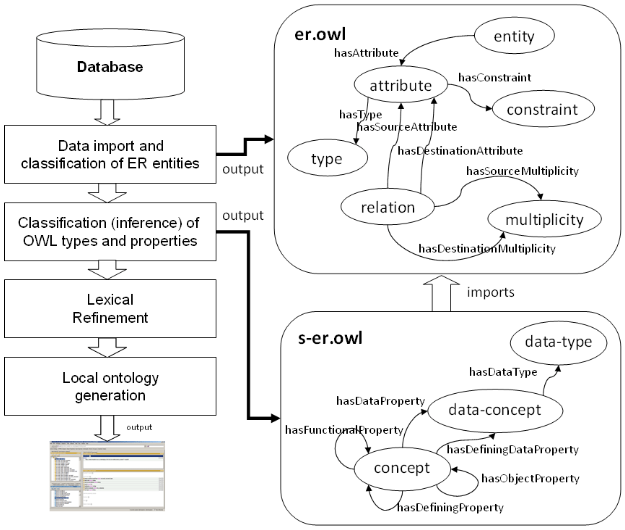

Fig. 49. Approach to database-to-ontology mapping.

In a first step of the transformation process, database schema is investigated and OWL representation of the ER model is constructed. This is realized by developed application, which connects to the database, uses introspection queries to discover its structure and asserts the relations between the artifacts by using proposed ER formalization (er.owl). Following assertions are made for each field of the corresponding table: hasAttribute (entity, attribute), hasType(attribute, type) and hasConstraint(attribute,'not-null') and/or hasConstraint(attribute,'unique') (if applicable). Following assertions are made for each relation: hasDestinationAttribute (relation, attribute), hasSourceAttribute(relation, attribute).

Second, resulting (serialized) OWL representation of the database ER-model is imported into meta-model (s-er.owl), which classifies future OWL concepts (axiom Ax_1, below) and domains and ranges of the object and data properties, according to defined axioms (axioms Ax_2 and Ax_4, below). Although specification of object and data properties may impose the unnecessary restrictions on the resulting ontology, those are considered as important for improving the efficiency of mapping or alignment process, which is critical for the semantic interoperability.

Another reason for the assertion of object properties in OWL representation of database ER-model is that object properties of the resulting local ontology will be annotated with the URI's of the respective relations, in order to enable the correspondence between the ontology and database representation, for the benefit of query transformation.

On the other hand, existential constraints from the ER-model are associated to an explicit semantics of the resulting ontology, namely, necessary conditions for inferring of the entailments. Thus, the meaning of the concepts can be attributed to these necessary conditions. This approach to a conceptualization is referred to as intensional, and is considered as equivalent to a human thinking (Guarino, 1997), in contrast to extensional approach, which implies that the elements of the mental image of the specific domain are simply enumerated or listed.

According to above constraints, axioms for intensional conceptualization (inherited anonymous classes) for particular entity are identified by inferring ranges of hasDefiningProperty(concept, concept) and hasDefiningDataProperty(concept, data-concept) relations (axioms $Ax_{2.2}$ and $Ax_{4.2}$, below).

Finally, the approach takes into account the functionality of the properties (owl:FunctionalProperty). Functional property is property that can have only one (unique) value y for each instance x. They are classified when relation one-to-one is identified between two concepts (axiom $Ax_{2.3}$, below).

Classification of future OWL concepts is inferred by exploiting following axioms:

Ax_1. Concepts are all entities of the ER model's OWL representation, except the entities whose all attributes are relation sources (corresponding to intermediary tables, connecting two tables with many-to-many relationship).

```
er:entity(x) ∧ not (er:hasAttribute only (er:attribute ∧
(er:isSourceAttributeOf some er:relation))) ⇒ s-
er:concept(x)
```

$Ax_{2.1}$. Domains and ranges of the object properties are inferred by using the rule below.

```
er:entity(x) ∧ er:entity(y) ∧ er:relation(r) ∧
er:hasAttribute(x, a1) ∧ er:hasAttribute(y, a2) ∧
er:isDestinationAttributeOf(a2, r) ∧
er:isSourceAttributeOf(a1, r) ⇒ s-
er:hasObjectProperty(x, y)
```

$Ax_{2.2}$. Domains and ranges of the defining properties (necessary conditions of the concept) are inferred by using the rule below. Defining property is a sub-property (rdfs:subPropertyOf) of the object property (hence, simplified representation of the rule below).

```
s-er:hasObjectProperty(x, y) ∧ er:hasConstraint(a1,'not-
null') ⇒ s-er:hasDefiningProperty(x, y)
```

$Ax_{2.3}$. Domains and ranges of the functional properties are inferred by using the rule below. Functional property is a sub-property (rdfs:subPropertyOf) of the defining property (hence, simplified representation of the rule below).

```
s-er:hasObjectProperty(x, y) ∧ er:hasConstraint(a1,'not-
null') ⇒ s-er:hasDefiningProperty(x, y)
```

Ax$_3$. Data concepts are all attributes of the ER model's OWL representation which are not at the source of any relation.

```
er:attribute and not (er:isSourceAttributeOf some
er:relation) ⇒ s-er:data-concept
```

Ax$_{4.1}$. Domains and ranges of the data properties are inferred by using the rule below. Ranges of the data properties are data types, corresponding to the simple types from XML schema.

```
er:type(x) ⇒ s-er:data-type(x)
s-er:concept(c) ∧ er:attribute(a) ∧ er:type(t) ∧
er:hasAttribute(c, a) ∧ er:hasType(a, t) ⇒ s-
er:hasDataProperty(c, t)
```

Ax$_{4.2}$. Domains and ranges of the defining data properties are inferred by using the rule below. Defining data property is a sub-property (rdfs:subPropertyOf) of the data property (hence, simplified representation of the rule below).

```
s-er:hasDataProperty(c, t) ∧ er:hasConstraint(a,'not-
null') ∧ er:hasConstraint(a,'unique') ⇒ s-
er:hasDefiningDataProperty(c, t)
```

The above conversion rules are specified in s-er.owl by using SWRL. Below are some examples of SWRL representations of the axioms (conversion rules).

```
(R1) entity(?e), hasAttribute max 0 attribute(?a), is-
SourceAttributeOf some relation(?r) -> concept(?e)

(R2.1) entity(?e1), entity(?e2), relation(?r), attrib-
ute(?a1), attribute(?a2), hasAttribute(?e1,?a1), hasAt-
tribute(?e2,?a2), isDestinationAttributeOf(?a2,?r), is-
SourceAttributeOf(?a1,?r)->hasObjectProperty(?e1,?e2)

(R2.2) entity(?e1), entity(?e2), relation(?r), attrib-
ute(?a1), attribute(?a2), hasAttribute(?e1,?a1), hasAt-
tribute(?e2,?a2), isDestinationAttributeOf(?a2,?r), is-
SourceAttributeOf(?a1,?r), hasConstraint(?a1,"not-null")-
>hasDefiningProperty(?e1,?e2)
```

Rules above classify instances of the OWL representation of the database ER model (er.owl) into a meta-model (s-er.owl). Inferred triples can be edited in a simple web application, which also launches the process of local ontology generation. In this process, meta-model entities are transformed into corresponding OWL, RDF and RDFS constructs – a resulting local ontology. Concepts of the generated local ontology are annotated with URI's of the corresponding ER entities from er.owl model. Thus, translation of semantic to SQL queries becomes possible.

It is evident that database-to-ontology transformation is not a novel concept. However, the most of the approaches are not considered as suitable for generation of the local ontology, which can be used in the formal framework for supply chain networks, for at least three reasons.

First, and most important, they do not interpret the semantics of all ER constructs and patterns. Similarly, a remark can be made that the existing approaches do not use the full expressivity of the OWL language. The above statements are argued in the Section 5.3 in Chapter 2 of this thesis. Second, approaches to instance population are not fully appropriate for use in the collaborative enterprise settings. Finally, although some of the researches claim that they provide a method for translation of semantic to SQL queries, the detailed information about this method is not present in the papers. **Table 3** shows the comparative analysis of the selected approaches, including the approach presented in this thesis.

	DB2OWL	**Relational.OWL**	**D2OMapper**	**Vis-A-Vis**	**This approach**
Main feature	Create new ontology from existing data-base	Create new ontology from existing database	Create mappings between database schema and existing ontology	Create mappings between data-base schema and existing ontology	Create new ontology from existing database
Semantic interpreta-tion of ER patterns	Semantics of existential constraints of the columns and cardinality of relations is not taken into account	Resulting model is (weakened) replica of the database schema and no attempt is made to interpret its semantics.	Based on the five heuristic rules. The remark for DB2OWL stands here, too.	Not relevant. The level of conceptualiza-tion (if any) is a choice of an expert, since the mapping is done manually.	See Axioms for classifica-tion of OWL concepts, in this section.
Data mapping process	Query-driven population	Massive dump	Query-driven population	Query-driven population	Query-driven, temporary, per request
Correspondences between database and ontology	XML document with mappings	Not known.	XML document with mappings	Not known.	Using URIs of er.owl “rep-lica” model concepts to annotate local ontology concepts

Table 3. Analysis of the selected approaches to database-to-ontology mapping.

Another challenge for the development of local ontologies is related to instance population, namely, on how and when database data is represented in the local ontology. As it is mentioned before, two types of the approaches are applied in the reported work.

Massive dump assumes that all data is represented as individuals in the process of ontology generation (or mapping of existing ontology with database schema). Besides obvious maintenance related difficulties, this type of approach is unacceptable mainly because of the size of the resulting ontology and the mapping document and consequently, performance issues related to reasoning processes.

Query-driven population approach assumes that individuals are asserted to ontology during exploitation, upon execution of the semantic query. Here, some kind of query rewriting mechanism is involved to transform the semantic to SQL query or queries which are executed in the database; result-sets are then represented as logical statements which are finally asserted to local ontology.

For many purposes, existing query-driven approach to population seems as a good candidate. However, when semantic interoperability between diverse and heterogeneous EISs is discussed, there are some concerns, mostly related to complexity of inferences when modular ontological framework is queried and handling of data access rights. Those concerns are elaborated in the next section.

One of the benefits of the semantically interoperable systems is the possibility to use the single criterion (or criteria) to infer the statements that hold true in all these systems, despite their heterogeneous structure. Namely, specific semantic query executed against the local ontology O_{Li} would normally infer triples of information from the database of S_i. However, if mappings (or logical function of mappings) between O_{Li} and O_{Lj} exist, inferred triples will also include information from the database of S_j. For example, in supply chain networks, a single semantic query can be used to find out the availability of specific resource or competence, of all - owned and used by the enterprises from the VBE (for the benefit of VE formation process). Precondition for this scenario is to enable the reasoning with local ontologies, namely, translation of semantic to SQL queries.

3.4 Semantic Querying Service

Semantic interoperability of systems enables a single point of access to the overall knowledge of the “interoperable world”. Not only that it makes possible to use a single semantic query to extract and combine relevant information from the multiple sources of implicit data, but it also enables usage of the different dictionaries for writing this query.

Fig. 50 illustrates how the data is extracted from heterogeneous sources by using three different approaches: 1) simple use of EISs; 2) merging the relevant result-sets from the databases; and 3) executing semantic queries. In the first case, one can use (USE_i) the EISs’ data exchange facilities to export data files (F_i) and then transform each of the files to a common format and merge. In the second case, the SQL queries (SQL_{Qi}) are executed against EISs’ databases to get relevant result-sets (R_{Si}) and then merge.

In the case of semantic queries data extraction, and if the assumption that logical mappings between local and domain ontologies are consistent and complete holds true, a single DL query (DL_{Qi}) can be constructed by using any dictionary, formalized by the domain ontologies, to extract the same data. Thus, whichever dictionary is used to build the query, the result of its execution is the union of identical sets of triples (S_{Ti}).

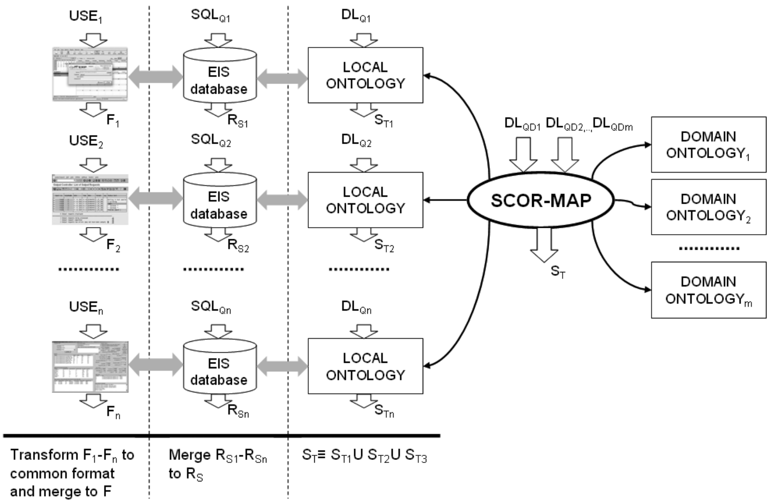

Fig. 50. Extraction of data from heterogeneous sources.

In this section, the method for instance assertions to local ontology on basis of the semantic query results is described. Method is illustrated at **Fig. 51** and it consists of the following steps:

1. decomposition and analysis of the semantic query;
2. data extraction and instance assertions and;
3. reasoning.

Semantic query can be considered as a pair (O, C), where O is a set of concepts which need to be inferred and C - a set of restrictions to be applied on their properties, namely value (owl:hasValue and qualified cardinality restrictions, owl:allValuesFrom, owl:someValuesFrom) and cardinality constraints (owl:cardinality, owl:minCardinality, owl:maxCardinality). This consideration corresponds to a simplified representation of a SQL query which includes tables (and fields) and comparison predicate, namely restrictions posed on the rows returned by a query. In addition, different types of property restrictions correspond to different cases (or patterns, where complex semantic query is mapped) of SQL queries.

Since relevant entailments can be reasoned only by property domain and range inferences, a set C may be considered as necessary and sufficient for representation of the semantic query. For example, in the local ontology of OpenERP EIS (see Chapter 5 of this thesis), a DL query "hasAccountAccountType some (hasCode value 3)" returns all instances of account_account concept whose type's code is exactly 3.

This kind of query representation (only by using properties restrictions) may produce unpredictable and misleading results when the restrictions are posed on the common lexical notions of different concepts, such as "name", "type", "id", etc. Ambiguity of the corresponding properties is reflected on the relevant ontology in the sense that their domains are typically defined as union of large number of concepts.

For example, in OpenERP ontology, domain of the “hasName” data property is union of 170 concepts.

However, this ambiguity may be considered as an advantage in some cases. Value restrictions on ambiguous data properties may produce relevant inferences and thus, they can facilitate semantic querying without a need to have extensive knowledge on the underlying ontology structure. This kind of query is mapped to a set of SQL queries made on the each element of the property domain, with the WHERE statement corresponding to the relevant rows restrictions. For example, in a mapping process (in the example of OpenERP ontology), DL query “hasName value ‘Derek Porter’” is first used to infer all 170 possible entailments (property domains), which are, then, used to assemble qualified (O,C) pairs, e.g. “res_users and hasName value ‘Derek Porter’”.

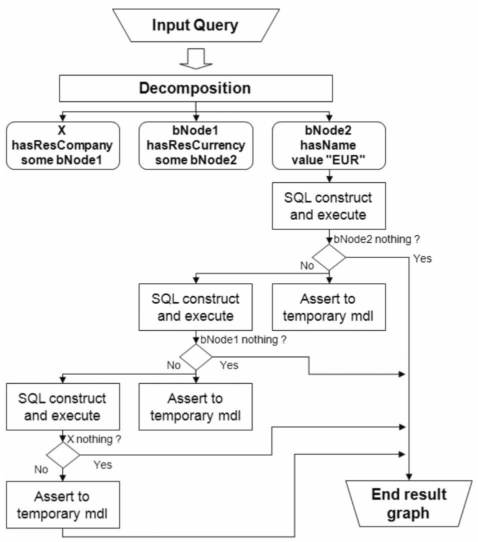

Fig. 51. Execution of the example semantic query in local ontology.

In the first step of the method, decomposition and semantic analysis of the input query is performed. The 4-tuplets in forms of (subject predicate some|only|min n|max m|exactly o bNode) and (subject predicate value {type}) are extracted from the input query. In case of the DL query which returns all concepts which are related to a company whose primary currency is EURO, following DL query is used:

```
hasResCompany some (hasResCurrency some (hasName value
"EUR"))
```

Here, following 4-tuplets are identified:

```
X hasResCompany some bNode1
bNode1 hasResCurrency some bNode2
bNode2 hasName value "EUR"
```

In some cases, more complex queries may be needed to define the requirements of the user. This occurs when multiple restrictions on a desired object are given, so intersection of two or more sets, corresponding to these restrictions, is taken into account. For example, all payable accounts for companies whose primary currency is EURO are inferred by using DL query:

```
hasAccountAccountType value "Payable" and hasResCompany
some (hasResCurrency some (hasName value "EUR"))
```

In this case, following 4-tuples are identified:

```
X hasAccountAccountType value "Payable"
X hasResCompany some bNode1
bNode1 hasResCurrency some bNode2
bNode2 hasName value "EUR"
```

In the next step of semantic query execution, a database connection is established and sets of SQL queries are constructed and executed for each element of 4-tuplet, in reverse order, as a result of analysis described above. Each query returns data which is used to generate OWL statements which are asserted to a temporary model. Each set of the OWL statements corresponds to a sub-graph whose focal individual is an instance of the concept, inferred on basis of the 4-tuplet's property domain or returned result (label). Other individuals or values correspond to defining properties of this concept (inherited anonymous classes). In case of ambiguity, resulting blank nodes are represented as the sets, which are filtered as a result of range inference of the parent 4-tuplet, in a final stage of the method.

As it is shown on **Fig. 51**, the output of the process of semantic querying of local ontology is a set of OWL triples which formalize the parts of the local ontology, asserted with individuals whose properties match the restrictions, defined by DL query.

Obviously, a query-driven population is applied in this case. As it is mentioned before, this approach separates data from the meta-model and hence, it enables better performance of the reasoning processes. However, at the moment, query-driven population cannot be applied in more complex environment of inter-related ontologies, such as the scenario of semantic interoperability of systems. In the remainder of this section, two main arguments for this statement are elaborated.

Semantic reasoning still assumes a centralized approach where all inferences are carried out on a single system. The consequence of this approach is that all ontologies that need to interoperate (typically inter-related by “imports” relations) have to be loaded by the reasoner software before inference is even started. In semantic interoperability scenario, the reasoner uses asserted logical correspondences between the local ontologies and domain ontology to infer about the individuals of the local ontologies by using the language of the domain ontology. Since all ontologies need to be loaded into memory space of the reasoner, it is not possible to apply query-driven approach because database is not accessible. This issue may be resolved by customizing inference engines or by enabling more flexible and dynamic imports, where, for example, imported local ontologies are populated by the dynamic services, capable to process restrictions from the semantic query executed in the parent ontology. At the moment, there are no known efforts of the scientific community to tackle this problem.

Another issue of the query-driven population of local ontologies in inter-organizational settings is data security, namely access authorization. In massive dump population approach, specific export and synchronization rules may be implemented to publish only some parts of the system's database to a local ontology. However, query-driven population, as explained above is done at the runtime, when query itself is executed. Hence, it is very difficult to implement and manage access rules. Even, more complex, but realistic scenario can be imagined, where enterprise wants to manage access to particular information per request in the process of query execution. It is important to note that, in this case, the process of semantic querying will become asynchronous. Again, it seems that no relevant work on this topic has been done so far.

Despite the fact that above concerns are serious, query-driven population is still considered as better candidate approach for application in semantically interoperable EISs than massive dump. The problems of static and restricted imports and access rights are mainly related to technical challenges, which are expected to be faced more likely than performance issues of DL-based reasoners.

Above results are mostly related to how "Ask" interface of the Semantic Querying Service works. The issues of the "Tell" interface are not elaborated in this thesis, or implemented in the case study, presented in Chapter 4. Those issues include ontology versioning, commit and rollback functions, etc. The exception is the basic use of "Tell" interface, where the ontology which needs to be queried is submitted to the service.

3.5 Reasoning Service

In S-ISU architecture, a reasoning service is auxiliary service which facilitates semantic query answering (see **Fig. 45**).

A semantic reasoner is software which is capable to infer logical consequences from a set of asserted facts or axioms, where the inference rules are commonly specified in some ontology. Reasoners are typically distinguished by different features, which correspond to: capability for OWL-DL entailment, supported expressivity for reasoning, reasoning algorithm, capability to check consistency, DIG support and Rules support. **Table 4** shows the basic comparison[47] between most commonly used reasoners, made by using the above-listed features.

Standard set (Sirin et al, 2007) of Description Logics inference services include:

- Consistency checking, to ensure that ontology does not contain any contradictory facts. Basically, this service takes ontology as an input and returns one of the three words: Consistent, Inconsistent or Unknown.

[47] http://en.wikipedia.org/wiki/Semantic_reasoner

- Concept satisfiability, to check if it is possible for a class to have any instances. If class is unsatisfiable, then defining an instance of the class will cause the whole ontology to be inconsistent.
- Classification, to compute the subclass relations between every named class and hence, to create the complete class hierarchy.
- Realization, to find the most specific classes that an individual belongs to; or in other words, to compute the direct types for each of the individuals. Realization can only be performed after classification since direct types are defined with respect to a class hierarchy.
- Query answering, to return instances from the ontology, based on RDQL or SPARQL query.

	Pellet	**KAON2**	**Jena**	**FaCT++**	**HermiT**
OWL-DL Entailment	Yes	Yes	No complete reasoner included with standard distribution	Yes	Yes
Supported expressivity for reasoning	SROIQ(D)	SHIQ(D)	varies by reasoner (incomplete for nontrivial DL)	SROIQ(D)	SHOIQ+
Reasoning algorithm	Tableau	Resolution & Datalog	Rule-based	Tableau	Hypertableau
Consistency checking	Yes	Yes	Incomplete for OWL DL	Yes	Yes
DIG Support	Yes	Yes	Yes	Yes	No
Rule Support	Yes (SWRL-DL Safe Rules)	Yes (SWRL-DL Safe Rules)	Yes (Own rule format)	No	Yes (SWRL-DL Safe Rules)

Table 4. Reasoner comparison.

The basic reasoning services can be accessed by querying the reasoner. Generally, such queries are supported by the reasoner's API or bindings and support for common toolkits, such as Jena (Carroll et al, 2004), DIG interface (Bechhofer et al, 2003), and others.

The DIG Description Logic Interface is a standard, proposed with objective to allow client tools to interact with different reasoners in a standard way, by using a common standard interface. The "Client Application" and the "DL Reasoner" communicate via XML-HTTP (see **Fig. 52**). The exchanged XML messages must comply to the DIG schema. The DIG interface allows the client: to introspect, namely, to query the reasoner to determined its capabilities; to "tell", namely to assert new facts into the reasoner and; to "ask", namely, to query the reasoner.

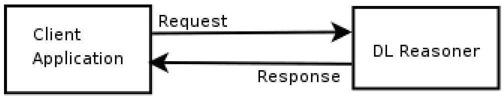

Fig. 52. XML-DIG Interface.

Basically, DIG interface facilitates the interoperability between the semantic reasoner and semantic application. Thus, it allows those to be developed by using different programming languages (reasoners are often implemented in LISP) and to be implemented in different platforms. Moreover, using DIG interface means that reasoning engine can be substituted with another one, without any impact to semantic application. It is obvious that, with standard interface, reasoning becomes a commoditized utility in the architecture.

The most practical service of the reasoner is query answering. This service relies upon and invokes other services (classification or realization), in order to return the expected results. It is based on a query engine, which typically:

1. analyzes the query and determine if it consists of independent sub-queries;
2. splits the query into multiple simple queries;
3. examines each of the queries and sorts the patterns and variables to improve efficiency;
4. decides which query engine (typical architecture includes one query engine per query type) will generate the answer and executes;
5. combines the results of each of the simple queries into final output.

As it is illustrated in **Table 4**, there exist many reasoners which are capable to provide the defined services. However, there are still some practical features which are not yet embedded in any of them, where incremental reasoning, distributed reasoning and combination with other logical formalisms are considered as the most important.

Incremental reasoning feature addresses the problem of the reasoners' performance. Many semantic applications assume repeated changes in OWL ontology in a relatively short period of time. For these applications, it is critical for the reasoner to re-compute as little as possible after each update, in order to achieve a better performance. The requirement of combining the reasoning of DL and other logical formalisms is derived from the needs of many semantic applications, such as multi-media systems, to have the ability to reason with space, time and motion. Currently, Pellet reasoner is being extended with various spatio-temporal representation and reasoning functionalities. Finally, as it is already mentioned in Section 3.1, a centralized approach to semantic reasoning, which assumes that all inferences are carried out on a single system, takes time and reasoning systems have limited performance (Schlicht and Stuckenschmidt, 2009). Hence, a distributed reasoning where different strategies (Bonacina, 2000) for parallelizing logical inference is applied, must be considered as the feature of the reasoner service.

3.6 Registration Service

Main purpose of the Registration Service is to enable the logical assertions to S-ISU ontology, which describes the inter-organizational environment (VBE), VEs and enterprises themselves, as well as associated assets, such as knowledge assets (local ontologies, domain ontologies, application ontologies) and functional common assets (semantic applications). Those logical assertions are made by using “Tell” interface of Semantic Querying Service.

Registration Service is invoked in following cases:

- Registration of Enterprise to interoperable environment, namely, VBE;
- Registration of a domain ontology;
- Registration of a semantic application; and
- Registration of a VE.

Fig. 53 illustrates the case of an enterprise registration to VBE. This case describes the steps involved when the enterprise is using the Registration Application, as a common asset of the interoperable environment, to apply for participation in the VEs. Main steps in the registration process are as follows:

1. Request to register;
2. Generation of the local ontology;
3. Establishment of the logical correspondences between the generated local ontology and ontological framework of VBE;
4. Access rights definition;
5. Confirmation.

Each of the registration requests need to be approved by the broker of the VBE. Once, this is done, Registration Service generates a local component – a listener, which is delivered to an enterprise. Listener is installed behind the firewall and configured locally (to access the EIS database) by the enterprise.

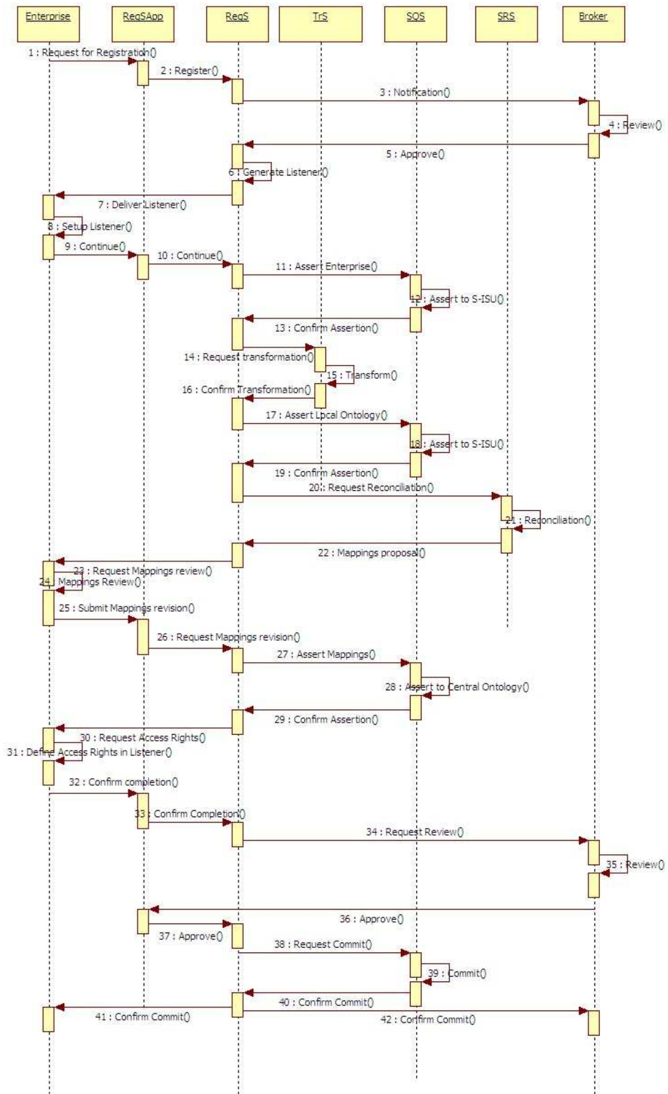

Fig. 53. Registration Process UML sequence diagram.

When this is confirmed, a new release of S-ISU ontology is created and enterprise concept is asserted to this release. Then, Registration Service requests from Transformation Service to communicate with local listener, to employ database to ontology method and to generate a local ontology. URI of this local ontology is asserted to a new release of S-ISU ontology. All assertions are made by the Semantic Querying Services, on basis of the request, submitted to its “Tell” interface. Next, Registration Service requests from Semantic Reconciliation Service to propose the mappings between the concepts of the existing ontological framework in VBE and a new local ontology. Once the mappings are proposed, they need to be reviewed and revised by the enterprise. This revision is then asserted to a new release of central ontology. The last action step is to define the access rights to a local ontology. The access rules are stored in a local listener component.

Once the registration is reviewed by the broker and approved, all changes (new releases of the S-ISU and Central ontology) are committed.

Registration of domain ontology is performed in a similar way, by VBE broker. This process includes assertion to S-ISU ontology, invocation of the Reconciliation Service and proposal, review, revision of the mappings and their submission to a new release of the Central ontology.

The process of VE registration is illustrated on **Fig. 54**.

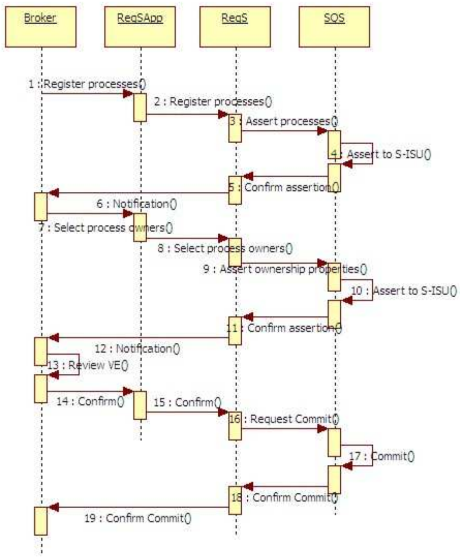

Fig. 54. Virtual Enterprise registration process UML sequence diagram.

The illustration shows the simplified process of VE creation, where broker is responsible for registration of the processes, needed for the completion of VE project task, where the processes are identified by using SCOR framework. In addition, Broker assigns the ownerships for each of the processes, based on available information. Obviously, selection problem is not handled by this scenario. Typically, the selection of partners (actors) is expected to be supported by dedicated semantic application.

3.7 Semantic Reconciliation Service

The only task of the Semantic Reconciliation Service is to propose the mappings between two submitted OWL ontologies, based on similarities between their concepts and properties. This service is invoked only during the registration of new domain ontology or new enterprise, when mappings between its local ontology and the concepts of the ontological framework are proposed.

Extensive literature related to the area of semantic or schema matching has been published so far[48].

The mappings are considered (Euzenat, 2004) as 5-tuples (id, e, e', n, R), where:

- id is a unique identifier of the given mapping element;
- e and e' are the entities (concepts) of the first and the second schema/ontology respectively;
- n is a confidence measure in some mathematical structure (typically in the [0,1] range) holding for the correspondence between the entities e and e';
- R is a relation (e.g., equivalence (=); more general (⊒); disjointness (⊥); overlapping (⊓)) holding between the entities e and e'.

An alignment is a set of mapping elements. The matching operation determines the alignment (A') for a pair of schemas/ontologies (o and o'). There are some other parameters which can extend the definition of the matching process, namely: 1) the use of an input alignment (A) which is to be completed by the process; 2) the matching parameters, p (e.g., weights, thresholds); and 3) external resources used by the matching process, r (e.g., thesauri);

It is important to distinguish between schema matching and ontology matching approaches to semantic reconciliation. While former are trying to guess the meaning encoded in the schemas, the latter primarily try to exploit knowledge explicitly encoded in the ontologies.

The problem of semantic matching is strictly associated to explicitness and completeness of the input ontologies. In other words, more explicit and complete ontologies are, better results are expected from the semantic matching service. Database schemas often do not provide explicit semantics. It is usually specified at design-time but frequently it is not becoming the part of database specification, and therefore is not available (Noy and Klein, 2002). Approach to transformation of database schemas to local ontologies, presented at Section 3.3 partially addresses this problem. It aims at discovering these design-time decisions by formalizing what exist in the specification, namely, database schema. Hence, it also increases the potential of the semantic matching service to deliver better results.

Semantic Reconciliation Service is not implemented in the S-ISU architecture. Only some conceptual requirements and its roles are anticipated at this moment. Ontology matching is considered as the most difficult problem in the implementation of the semantic web. Although there are some proposed solutions, such as QOM (Ehrig and Staab, 2004) or S-Match (Giunchiglia et al, 2004), some human intervention in the reconciliation of two semantic models is inevitable. This is more evident in the cases where two models, developed with different conceptualization approaches need to be reconciled.

---

[48] http://www.ontologymatching.org/publications.html

Chapter 5: Case study - Interoperability issues in a Virtual Enterprise for custom implant manufacturing

Abstract. In previous two chapters of this thesis, an approach to developing and implementing a formal framework for semantic interoperability in supply chain networks is described. In Chapter 3, a multi-level ontological framework is proposed, based on SCOR reference model. In Chapter 4, an implementation view is presented, with emphasis on the process of introducing the enterprise information systems to an interoperable environment. In this chapter, some evidences on the feasibility and usability of the proposed framework and implementation method are given. Two case studies of using the proposed approach are presented. First case study shows how the approach can be exploited to support one of the common functions of the Virtual Breeding Environment – a setup of supply chain processes. The second case study illustrates how the S-ISU architecture is used to get the relevant information from the local ontologies with a single query, facilitating the collaborative production planning in Virtual Breeding Environment.

1 Introduction

The original vision of the semantic web was to provide a new layer at the top of the existing World Wide Web, which annotates the content of the web pages, so the computers can understand them. However, since its beginning, the development of semantic web has been considered as an academic exercise, rather than a practical technology. There are many technical, scientific and business challenges for implementing the scenarios of semantic web. It is a fact that this development relies on the semantic annotation of the implicitly written and unstructured information, by using RDF, RDFS and OWL dictionaries and ontologies. Thus, it considers that bottom-up approach is needed, where authors of the individual web pages need to put the additional effort in making their pages compatible with the future web – semantic web.

Obvious problem with such a scenario is the fact that no incentive is provided to the authors for doing that, especially because currently there are no tools which leverage such annotated information and provide added-value to the web site owners.

The framework for semantic interoperability of EISs, proposed in this thesis, aim to provide this incentive by enabling the enterprises to act in a flexible way, while they concurrently participate in much more supply chains then it was possible with a conventional integration technologies. Thus, it becomes possible to get closer to the organizational paradigms of Virtual Enterprise and Virtual Breeding Environment and to achieve more dynamic behavior of the enterprises.

Interoperability is one of the main consumer benefits of the semantic web vision. Although it was addressed as such by the academic community and businesses, it appears that the role of semantics in the current efforts in making two systems interoperable is superficial. Namely, in the conventional approaches, semantics is only associated to some kind of structured information, by annotating this information, whether this information represents XML messages, database rows or even enterprise models' entities. This is exactly the characteristic of the bottom-up approach, mentioned earlier.

At least two negative consequences of such an approach are derived from the fact that one needs a lot of effort (and knowledge) to semantically annotate static, implicit information. First, the amount of this effort typically restricts the scope of this task. Hence, semantically facilitated interoperability is achieved in a relatively small number of critical cases of systems collaboration and it involves many preconditions, related to mostly manual reconciliation of two implicit (specific) semantic models. Consequently, no indirect effects or wider outreach are expected. Second and more important negative consequence of the bottom-up approach is related to the implication that more restricted is the set of messages that need to be annotated; the "weaker" conceptualization is applied. In this case, it is likely that extensional conceptualization will dominate over intensional approaches. Moreover, there is no obvious interest to work on establishing correspondences between the outcomes of this conceptualization and formal upper ontologies, because direct mappings between concepts of two systems is more reasonable approach, more cost-effective in a short term. The end result is that messages are annotated with implicit concepts, still represented by the languages of Semantic Web. It does the job, but it does only that job.

The proposed framework for semantic interoperability in supply chain networks takes more pragmatic approach by combining the consumer-orientation of top-down and efficiency of bottom-up paradigm.

Here, the diversity and level of consumer benefits (and hence, scope of interoperability) directly corresponds to the amount of explicit knowledge which one enterprise voluntarily introduces into shared collaboration environment. In the interoperability scenario, annotated messages are replaced with logical statements (see Chapter 4, Section 3.4), where each instance of the used concepts is annotated with data. Hence, the reasoning is much simpler and more efficient, since increased amount of semantics in the environment and consequently, increased number of relationships improves the quality and usefulness of the inferences. This variety of semantics is exploited by the possible variety of semantic applications, which are directly related to the consumer value of the semantic technologies. The semantic applications can be easily introduced in the framework, because they implement presentation layer and only exploit application and storage layer (corresponding to the conventional architecture of multi-tier applications). Business logic can be considered as defined in domain and local ontologies (at least generic, common business rules) while data is still stored in their natural environment – databases, and accessed to by using local ontologies.

The top-down approach implies that universal (at least universal to the domain) ontological framework is used when correspondences between different semantic representations of the domain realities are established. In this way, it is ensured that implicit semantics of the micro-realities, such as specific enterprise, information system or reference standard are correlated consistently to the explicit domain reality. The challenge is to make one domain reality explicit. Many models and ontologies are developed over time with objective to address this challenge. Some of them are discussed in Chapter 2 of this thesis. However, they are built by using inspirational approach and they are not validated in the way which would ensure their wide adoption and self-sustainable improvement. Thus, they lack consumer value. In order to resolve this issue, the proposed approach takes the industry-adopted process reference model (SCOR), represents it by using semantic languages in an implicit way (see Chapter 3, Section 3.1) which can be easily mapped to native formats of SCOR-based applications and make its implicitly defined concepts correlated with respective, explicitly defined notions of SCOR's semantic enrichment (see Chapter 3, Section 3.2). Hence, it becomes possible to extend the supply chain process model because now, the correspondences between these explicit notions and respective concepts of other domain ontologies may be established.

It is very important to make clear that universal interoperability is far from the reality. The proposed framework focuses on the supply chain problems and by proposing the solution to these problems, it extends its conventional understanding and organization to new organizational forms, such as Virtual Enterprises and Virtual Breeding Environments. These problems include, but are not restricted to collaborative planning for the benefit of efficient selection of the enterprises in the process of Virtual Enterprise formation and supply chain processes configuration. In this Chapter, it is shown how these two problems can be addressed by the proposed framework for semantic interoperability in the case of the manufacturing of the orthopaedic implants.

2 Case analysis – Orthopaedic implants manufacturing

The success of the orthopaedic predictive, preventive, diagnostic and therapeutic activities depends on the variety of factors, such as: availability of data about patient's condition, physician's knowledge and experience and availability of tangible resources (instruments, devices, fixtures, implants, software, etc.). Surgeons are often not able to perform those activities efficiently. Efficiency and effectiveness of the above activities is achieved when right decisions are made fast, based on complete and updated information on the patient's condition. Those decisions assume the selection of appropriate actions, performed by exploiting the appropriate resources in appropriate manner, and is typically facilitated by the information systems.

Conventional bone implants have been successfully used for more than 30 years and they are associated to the most common orthopaedic implant surgeries performed around the world (Harrysson et al, 2007). In most of the cases, conventional implants provide satisfactory results.

However, sometimes standard implant components are not sufficient because of abnormal joint anatomy or possible risks of postoperative complications (Keenan et al, 2000), such as aseptic loosening. The typical reason for aseptic loosening is uneven stress distribution on the bone surface. This problem can be addressed by custom design process in which the design of the implant is accommodated to the specific features of the patient's anatomy.

The research of custom orthopaedic implants manufacturing is typically focused to direct fabrication technologies (Gibson and Harrysson, 2006). New rapid-manufacturing equipment and techniques provide far greater efficiency in small-volume or one-of-a-kind runs for producing a finished custom implant. The most commonly used method for direct manufacturing of high-strength materials, such as titanium is electron-beam melting (EBM). However, the complexity of the services and products associated to the manufacturing and other relevant processes implies that many other (e.g. organizational) challenges need to be considered in order to bring custom implants to everyday use.

Two important issues are identified in the daily practice in orthopaedics surgery: information interoperability and manufacturing of highly customized products.

A wide set of information need to be combined fast by the physicians in order to make possible to decide about the actions following the diagnosis. This set of information include the patient's health record, diagnosis, the domain models (which formalize the knowledge needed for taking decisions about the therapeutic actions), information about the availability of the resources needed for the therapeutic actions, availability of staff, etc. All this information must be uniquely accessed and processed in order to make relevant decisions. This is typically achieved by making the systems which store this information - interoperable.

The second issue for orthopaedics surgery is related to the manufacturing of orthopaedic implants. The orthopaedic implants may be highly complex and custom products, which need to be manufactured on basis of the above information in a shortest possible time. The key factors of the custom orthopaedic implants manufacturing are the degree of customization and time of delivery of the final product. Higher degrees of implants customization reduce the duration of the operation and increase their reliability. Hence, the period of patient's recovery and overall cost of treatment can be decreased. Also, the risks of possible complications are reduced; the costs of treatment of only one complication can be as high as costs of many successful operations.

2.1 Barriers to customization

Two of the most critical non-technical barriers to customization are: 1) lack of efficiency of traditional manufacturing enterprise to handle low-series or one-of-a-kind production tasks; and 2) lack of efficiency in transfer of multi-disciplinary knowledge, required for the design of custom implant.

In general, manufacturing enterprises refine their designs for simplicity and cost; they design their workflows for volume manufacturing. Hence, by default, they are not capable to handle one-of-a-kind manufacturing tasks efficiently. In traditional settings, the workflow for manufacturing of custom implants includes many human analysis and decisions, such as interpretation and analysis of CT scans, review of the wax prototypes, mechanical analysis, collecting inputs and approvals, etc. The lack of efficiency to adapt their traditional workflows to these activities becomes even more critical when enterprises are required to subcontract the different parts or services suppliers.

All this human involvement includes a number of interactions between different experts in which functional (medical), mechanical, organizational and other perspectives to the custom manufacturing are discussed. Hence, efficient design elaboration and mutual understanding on the complex variety of issues require involvement of experts with multi-disciplinary skills and knowledge.

In order to overcome the barriers above, the systems and models infrastructures are proposed. The models infrastructure formalizes the knowledge required for the manufacturing of custom orthopaedic implants and thus, it facilitates its exchange. While systems replace humans in decision making process and hence, increase the customization efficiency, the models are also considered as enablers for systems interoperability. They use different agreed formalisms to explicitly represent a domain of interest (relevant to the system) and hence, explain its inner workings, parts, use cases, etc.

2.2 Models infrastructure

The custom orthopaedic implants manufacturing is based on a multi-level computer model of human osteo-articular (bone & joint) system, consisting of design, behaviour and production perspective (see **Fig. 55**). The model of certain level is used to solve specific problems and it is associated with the appropriate tools for its creation and use.

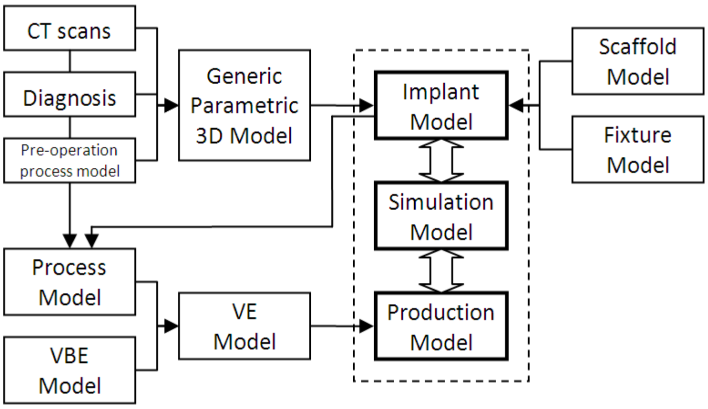

Fig. 55. Models infrastructure

The models represent different views to the orthopaedic implant products. They aim at enabling the representation of the relevant knowledge and inference in the field of decision support in treatment and pre-operation planning, as well as VE configuration, technology planning and business process management.

Generic parametric 3D model of the selected bone represent the surfaces and volumes whose geometry is determined by mathematical and logical relations, established between some key parameters. This model is constructed on the basis of data obtained from CT (Computed Tomography) scans of bone-joint system, and is required for digital reconstruction of a traumatized bone. Then, based on this model, a scaffold is designed to replace the missing part of a bone. The implant typically consists of the scaffold, fixture (used for heavily loaded bones) and bio-degradable, osteo-fixation material.

Simulation models facilitate prediction and optimization of mechanical behaviour of the implants under realistic load conditions, by using Finite Element Analysis (FEA). The simulation models also elaborate future exploitation of the bone implant, including surgery simulation.

While the upper two groups of models are related to the design and behaviour of the implant, production models establish references between their elements and concepts and:

1. processes, involved in the implant manufacturing and implementation (process model);
2. capabilities and resources, required for the implant manufacturing, configured in particular way (VE model).

Production models also include the formal domain models, such as enterprise, collaborative enterprising and supply chain. These are the parts of the common knowledge infrastructure of the VBE.

2.3 Example product

Depending on nature of the bone trauma, the custom orthopaedic implant can be assembled of some of different types and designs of fixtures and scaffolds. In addition, some services may be associated to the product manufacturing and/or implementation, such as: pre-operation planning, reposition simulation, digital reconstruction, remodelling, analysis of biomechanical properties of the implant, sterilization, ethical review, product certification and others.

For example, in case of bone cancer of tibia (larger of the two bones in the leg, below the knee), the missing part of the bone is replaced with the scaffold, which is enforced with the inner fixture. The scaffold is designed on the basis of bone geometry, which is digitally reconstructed from CT scans. Geometry and topology of inner fixture is designed on the basis of diagnosis and pre-operation plan, developed by surgeon. The process of manufacturing of the custom part is associated also with review of the design by the clinics ethical committee and analysis of biomechanical properties.

It is considered that the manufacturing of the implant parts and provision of the associated services is executed within the VBE, which consists of the enterprises, qualified, certified and competent to deliver a manufactured product and/or to provide associated services. VBE is organized as a cluster and technically coordinated by the brokering enterprise (broker). It supplies orthopaedic implants and services to clinical centres. Each case of supply of the product and associated services is considered as a case of VE.

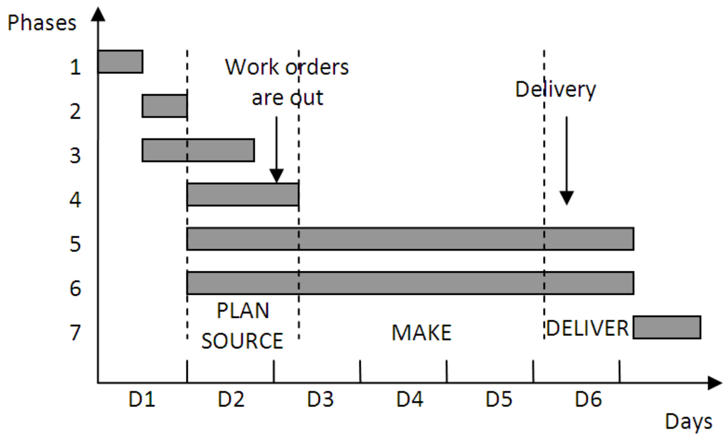

Fig. 56. Lifecycle of the Virtual Enterprise for custom orthopaedic implants manufacturing

The broker takes following actions:

1. receives and negotiates orders;
2. sets up a contract, Service Level Agreements (SLA) and certification processes;
3. provides immediate support to clinical centres in requirements definition and, consequently, instantiates appropriate product infrastructure;
4. launches the VE, by selecting the enterprises and by setting up the process configuration;
5. ensures that delivery of the product and services are in accordance with SLAs and product-associated certificates;
6. manages contract throughout the whole lifetime of VE;
7. takes appropriate actions to dissolve VE.

Illustration at **Fig. 56** shows the above mentioned actions and corresponding phases of the VE lifecycle, in the described case of bone trauma.

2.4 IT Infrastructure

The required agility of VE is achieved through extensive use of IT systems for coordinating and executing involved processes. Hence, the IT environment for manufacturing of orthopaedic implants is considered before the interoperability issues are identified and analyzed. This environment is shortly elaborated in this subsection from the perspectives of IT environment of the customer, namely a clinical centre and shared, common applications of VBE. This is illustrated on **Fig. 57**.

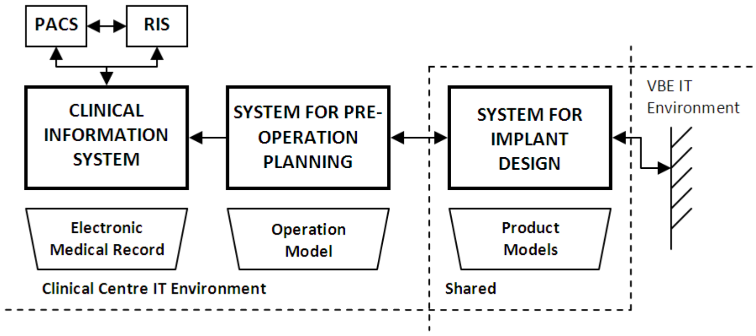

Fig. 57. IT Infrastructure

Diagnosis and pre-operation planning are carried out on the basis of Electronic Medical Record (EMR) of the patient, including CT scans and other associated information. This information is stored in the Clinical Information System (CIS), an integrated suite of IT tools, designed to manage medical, administrative, financial and legal aspects of healthcare provision. Relevant subsystems of CIS are Radiology Information System (RIS) and Picture Archiving and Communication System (PACS). RIS is used by radiology departments to store, manipulate and distribute patient radiological data and imagery. PACS is a medical imaging technology which provides storage and convenient access to images (scans) from multiple source machine types.

The System for pre-operation planning (PRE-OP-Sys) is used for making the most important decisions, with regard to significant features of the custom implant. For example, based on CT scan, namely bone and joint system and interpreted features of trauma, PRE-OP-Sys is capable to suggest position, size and orientation of the cut, geometry of missing parts of a bone, critical features of the implant and/or fixture, repositioning of the displaced bone parts, etc (Vitkovic et al, 2011).

The System for implant design (IMPL-D-Sys) is a software application, used by VBE (represented by broker) and surgeons. Some features of the system are design of product topology, facilitation of the decisions on characteristic geometric features and decisions on associated services.

2.5 Interoperability issues in orthopaedic implants manufacturing

Interoperability is considered as one of the main enablers of the VE, because it facilitates the flexible collaboration; it reduces the time needed for its setup, as well as discontinuation. Since production lead time is critical for the custom orthopaedic implants manufacturing, it is important to review the interoperability issues in its environment and to assess the potential for efficiency increase.

Interoperability issues can be considered in two aspects. The first aspect assumes inter-relationships between knowledge, assets and processes of the medical (clinical centers) and production (VBE) environment. The second aspect is related to the interoperability issues within VBE environment. In more traditional settings, this consideration corresponds to distinguishing between customer-manufacturer and manufacturer-suppliers interoperability.

The main outcome of the customer-manufacturer collaboration is instantiation of the product (models) infrastructure, namely, product requirements definition. Given the high requirements for efficiency and quality, it is of outmost importance to introduce some degree of automatization, mostly by removing all preconditions for this collaboration and needs for any kind of previous agreements. Thus, it is necessary to review the issues of interoperability between the clinical centres environment and VBE. The short review of interoperability issues is provided below, at the levels of domain, data, organization and systems.

Domains (conceptual) interoperability concerns the semantic correspondences between the domain of surgery and manufacturing, affecting models interoperability (between pre-operation process model and VBE process model). The problem of custom orthopaedic implants manufacturing is cross-disciplinary. Hence, the issue of domains interoperability can be addressed by providing formal domains vocabularies. Also, some correspondences between the models from different domains can be established in order to enable automatic processing of knowledge at the intersection of the manufacturing and orthopedics disciplines.

Data interoperability issues are related to exchange of different formats between different systems. This exchange occurs between CIS and PRE-OP-Sys and; between PRE-OP-Sys and IMPL-D-Sys. Former case assumes reconciliation of the data formats of the EMR and operation process model. It also assumes interpretation of the key geometry features on the basis of CT scans. In the case of exchange between PRE-OP-Sys and IMPL-D-Sys, data formats corresponding to the pre-operation process model (including key geometric features) and parametric 3D model need to be reconciled.

Organizational interoperability issues affect administrative procedures, related to review of the proposed product infrastructure by the clinics ethical committee.

Finally, systems interoperability issues are related to the interoperation between:

- CIS and PRE-OP-Sys. PRE-OP-Sys need to have capability to access the EMR details and CT scans, stored in PACS, by exploiting the references between CIS and PACS;
- PRE-OP-Sys and IMPL-D-Sys. Since design of the implant is directly related to the operation plan, the interoperability of these two systems is critical;
- IMPL-D-Sys and VBE IT infrastructure. In this case, interoperability issues are related to exploiting the correspondences between product model and: 1) CAD systems, which store parametric 3D model; 2) FEA systems, which store mechanical analysis models; and 3) Common VBE applications for VE partners' selection and process configuration.

When semantic interoperability is discussed, the issues of three above-mentioned system interoperability axes may be interpreted as follows.

Pre-operation planning is based on the location and the arrangement of anatomical structure parts within the human body, expressed in quantitative or qualitative way (by using spatial orderings such as superior, anterior, lateral, etc). This arrangement can be formalized by appropriate anatomical ontology (Burger et al, 2008). When operation is planned, the relevant spatial features are used to determine the features of the micro-steps which are carried out during the surgery, such as bone screw entry angles, fixture-bone assembly contact locations, etc. Hence, relevant properties of the spatial relations can be exploited for automated reasoning (Schultz and Hahn, 2001), which can assist pre-operation planning process. In order to make this possible, two preconditions need to be fulfilled. First, CT scans need to be semantically annotated with characteristic geometric features of the relevant bones, where anatomical ontology concepts are used for the annotation. Second, PRE-OP-Sys must be capable to infer the spatial relations and corresponding micro-steps features, by exploiting previously established logical correspondences between anatomical ontology and pre-operation process ontology (model).

Another consideration of CIS and PRE-OP-Sys interoperability is important when pre-operation planning involves also anaesthesia planning. In this case, it is obvious that the direct relations between the choices of the anaesthesia treatment and medical history and condition of the patient (readable from EHR) may be used for the benefit of risk reduction and more efficient surgery.

Above-mentioned spatial relations are also relevant when semantic interoperability between PRE-OP-Sys and IMPL-D-Sys systems is discussed. They provide formal definitions of the geometry restrictions which are typically considered when design of the orthopaedic implant is determined. For example, the angle between distal and proximal part of the inner fixture depends on the specific arrangement of bones and joints.

Finally, the third perspective to semantic interoperability issues for orthopaedic implants manufacturing is related to interoperations of IMPL-D-Sys system and IT environment of VBE. The main objective of these interoperations is to manufacture and deliver the most relevant and reliable, customized orthopaedic implant in the most efficient and effective way, on the basis of the product's conceptual description. These processes and proposed semantic interoperability infrastructure for their execution are elaborated in the following section.

3 Supply chain processes' configuration in the case of orthopaedic implants manufacturing

As it is mentioned before in the case analysis, one of the aspects of the interoperability issues consideration is related to the VBE environment. This consideration corresponds to manufacturer-suppliers interoperability. In the first case study of this Chapter, it is shown how the developed formal framework for semantic interoperability in supply chain networks can be exploited for the purpose of supply chain configuration.

The case study presents the demonstration of the use of the SCOR-KOS OWL model for supply chain process configuration, namely, the inference and presentation of a SCOR thread diagram for the described case of VE for orthopaedic implants manufacturing. SCOR thread diagram is illustration of the specific configuration of source, make and deliver processes, designed on basis of the product topology, participants and production strategies for each component.

The compliance to industry (community) standards is a competitive advantage of a single enterprise, especially if it depends on multiple supply chains. It is beneficial for dealing with horizontal integration, interoperability of systems and flexible governance, as critical success factors for collaborative enterprises. However, these benefits are realistic only if the compliance is managed in a manner which enables a seamless acquisition, storage, effective use and re-use and continuous evolution of knowledge, relevant to the standards themselves. In other words, it is necessary to establish and maintain the semantic relations between the reference models entities and enterprise knowledge. In order to make this possible, two preconditions need to be fulfilled.

First, the reference models need to be semantically enriched, as a condition for the establishment of the meaningful mappings between their entities and the concepts of the enterprise knowledge. In this thesis, this is achieved by conceptualizing the individual entities of the SCOR model and hence, developing SCOR-Full ontology.

Second, the enterprise knowledge must be stored in an explicit form. The methodology, presented in this thesis proposes the local ontologies as a candidate form, associated with the methods for making the implicit enterprise knowledge – explicit (by transforming the Entity-Relationship schemas of the EISs) and for semantic querying of the local ontologies.

The case study, presented in this section is an example of the usefulness of the proposed methodology of layering the semantics in the different levels of abstraction, where semantic applications, which deal with a specific problem (typically common problem of the VBE) use application (problem) ontologies (mapped to the implicit reference models, in this case SCOR-KOS OWL model) to resolve this problem.

The specific problem addressed is the supply chain configuration, namely, the generation of the process maps, relevant for production of the specific product in the context of a supply chain.

3.1 SCOR-based modelling of supply chain

The motivation for supply chain modelling of one enterprise may be the strategy development, expansion, optimisation and re-engineering of processes, standardization of reorganization of the enterprise, start-ups, making decisions about outsourcing or benchmarking or processes. SCOR reference model considers four different types of models, which can be developed for different purposes, on different levels of detail: Business Scope diagram, Geo map, Thread diagram and Workflow or Process model.

Business Scope diagram defines the framework of the supply chain. Namely, it identifies its participants and represents basic flows of information and material between those participants. Geo map (see **Fig. 58**, below) is used to represent material flows in geographical context. In contrast to the Business Scope diagram, a new level of detail is introduced – basic subjects in the modelling process are not enterprises or their departments, but process categories.

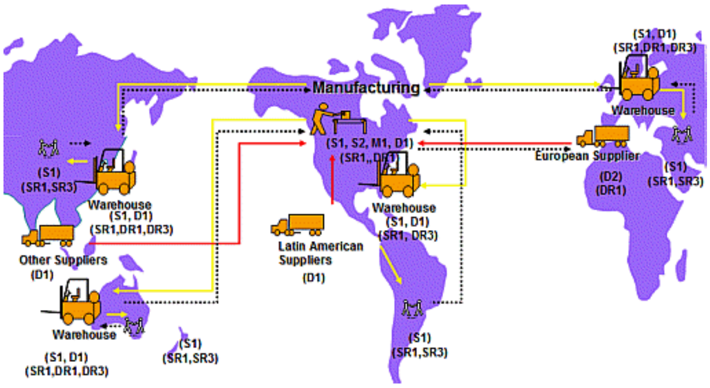

Fig. 58. Example of SCOR Geo map

SCOR Thread diagram (see **Fig. 59**) establishes the relationships between process categories, which are previously assigned to enterprises which implement those process categories.

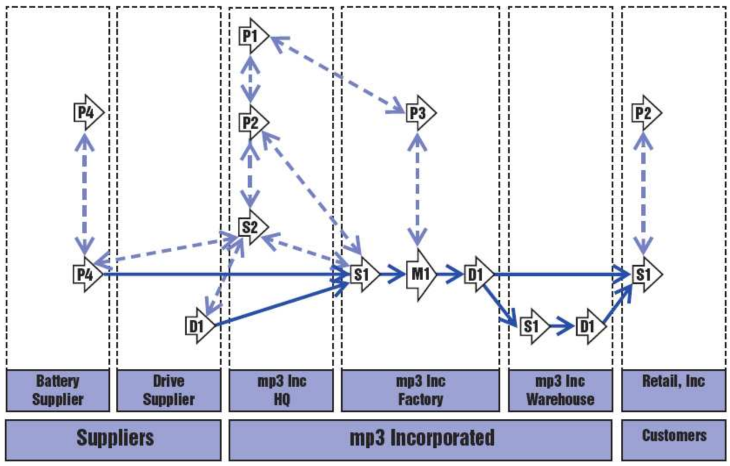

Fig. 59. Example of SCOR Thread diagram

Finally, a Workflow model (see **Fig. 60**) illustrates a detailed representation of the Level 3 processes (process elements), all relationships between those process elements and enterprises (or departments) which implement those processes.

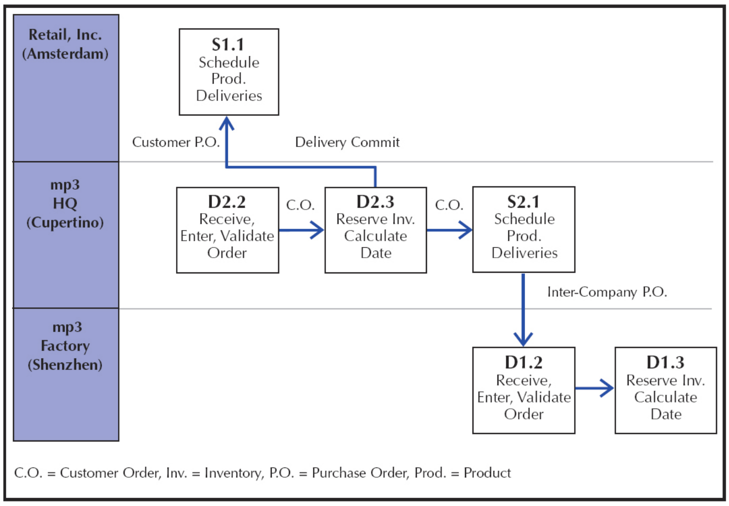

Fig. 60. Example of SCOR Workflow model

3.2 SCOR-Cfg Ontology

In the development of the semantic framework for supply chain processes configuration, the first step was to develop a conceptualization of the problem. Hence, the case of the VE for the manufacturing of orthopedic implants is considered as a project, which is owned by a focal partner of the VE (or broker). The main project objective is to produce (or to engineer) a principal product (to stock or to order), consisting of other products (parts), which are produced and delivered by the selected partners from VBE with different strategies (such as made-to-stock, made-to-order or engineer-to-order).

The conceptualization above is used to design problem (application) ontology – SCOR-Cfg OWL model. This model is illustrated on **Fig. 61**. It consists of following concepts: Project, Product, Production-Type, Process (with child concepts, corresponding to different process types) and Participant.

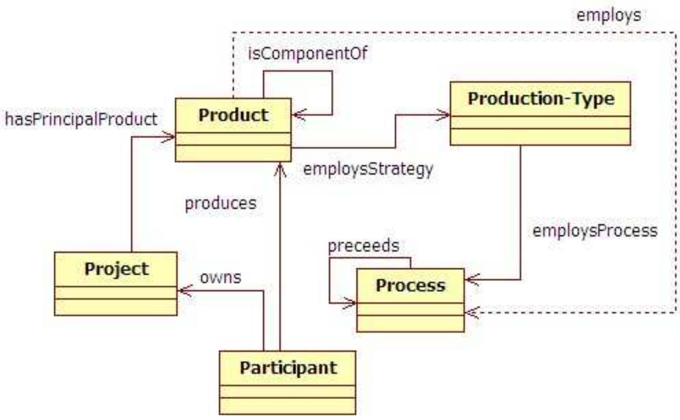

Fig. 61. SCOR-Cfg ontology

Relations between the concepts are established by following properties:

```
hasPrincipalProduct(Project, Product)
isComponentOf(Product, Product)
employsStrategy(Product, Production_Type)
employsProcess(Production_Type, process)
owns(Participant, Project)
preceeds(Process, Process)
produces(Participant, Product)
```

Initially, the SCOR-Cfg OWL model is asserted with instances of production strategies and processes. Hence, Production-Type concept has following individuals: engineering-to-order, made-to-order and made-to-stock. The concept of Process (or its child concepts – Deliver-Process, Make-Process, Plan-Process and Source-Process) has following individuals:

```
D1._Deliver_stocked_product
D2._Deliver_made-to-order_product
D3._Deliver_engineered-to-order_product
M1._Make-to-stock
M2._Make-to-order
M3._Engineer-to-order
P1._Plan_supply_chain
P2._Plan_source
P3._Plan_make
P4._Plan_deliver
S1._Source_stocked_product
S2._Source_make-to-order_product
S3._Source_engineered-to-order_product
```

Properties “employsStrategy” and “employsProcess” are defined as sub-properties of the transitive property “employs”. These relations enable the inference of source, make and delivery processes involved in manufacturing of the component of specific strategy.

3.3 Semantic application for supply chain processes’ configuration

For the purpose of supply chain processes’ configuration, a semantic web application is developed. This application exploits the application ontology – SCOR-Cfg OWL model. Use of the application involves:

1. assertion of the product configuration, namely principal product topology and manufacturing strategies for each of the components and
2. invocation of the algorithm (semantic querying) for rendering SCOR thread diagram.

Different process patterns (and roles) are applied in each of the three possible manufacturing strategies: made-to-stock, made-to-order or engineered-to-order.

As in the case of SCOR-KOS OWL model browsing application, the semantic application is developed by using RAP application programming interface. For visual representation, SVG (Scalable Vector Graphics) format is used. For inference, related to semantic queries, Pellet 1.5 reasoner (Sirin et al, 2007) is used.

The functionality of the semantic application is demonstrated on the case of orthopaedic implants manufacturing. In specific, the example product of custom orthopaedic implant for the case of bone cancer tibia is considered.

In this case, the missing part of the bone is replaced with the scaffold, which is enforced with the inner fixture. In addition, the product manufacturing is also associated with the services of digital reconstruction of traumatized bone and analysis of biomechanical properties of the implant. From the perspective of supply chain configuration, these two services are considered as parts of Bill of Material of custom orthopaedic implant. Hence, they need to be sourced, implemented (corresponding to SCOR Make process) and delivered, exactly like in the case of tangible parts, namely, inner fixture and scaffold.

In the process of supply chain configuration by using the semantic application, product information is acquired by using a software module for product acquisition in inter-organizational networks (Zdravkovic and Trajanovic, 2009). **Fig. 62** shows the basic interface for the definition of custom orthopaedic implant product topology, with entered information about custom orthopaedic implant product topology.

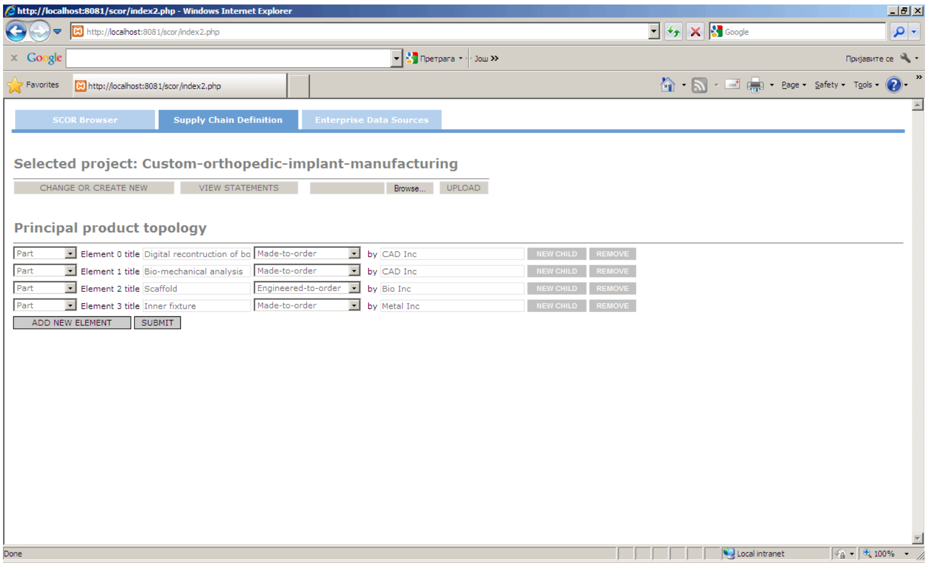

Fig. 62. Interface for definition of principal product topology

The submission of the product topology form or definition of the product configuration triggers the assertion of the new statements in SCOR-Cfg OWL model. These statements are related to type assignments and properties' assertions. **Fig. 63** illustrates the partial view to the SCOR-Cfg OWL model, after the assertions.

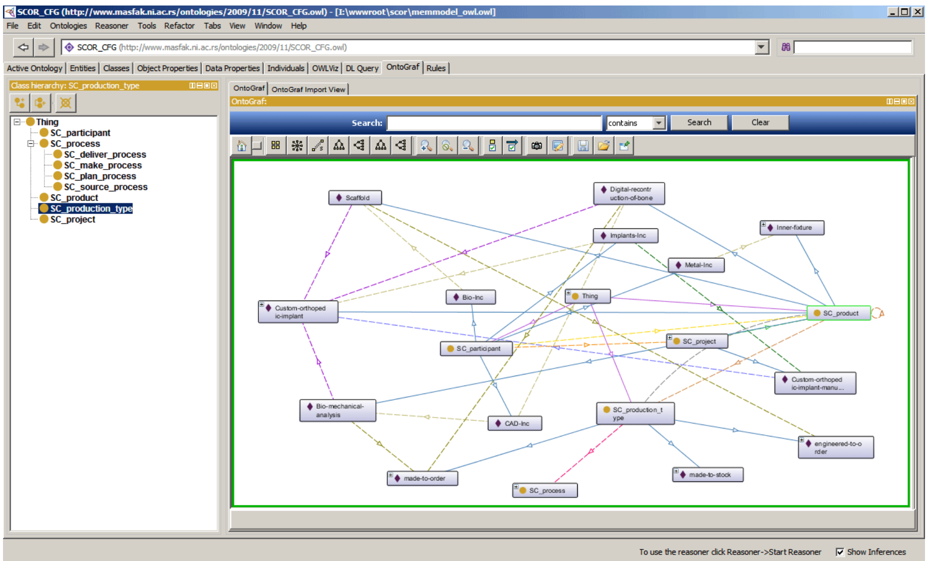

Fig. 63. (Partial) Visualization of the asserted statements (in Protege)

The generation of a SCOR thread diagram is done by selecting (and rendering) participants of supply chain project, its products (components) and, finally, processes, in exact order. Participants of selected supply chain project are inferred by using a following DL query:

```
(produces some
(isComponentOf some
(isPrincipalProductOf value <selected_project>)))
or
(produces some
(isPrincipalProductOf value <selected_project>))
```

In order to enable inference of participants in the infinite number of levels of supply and demand from the main participant, “isComponentOf” property is defined as transitive. Next, for each participant, its components of a principal product are inferred by semantic query:

```
producedBy value <participant>
and (
isComponentOf some
(isPrincipalProductOf value <selected_project>)
or
(isPrincipalProductOf value <selected_project>))
```

Employed processes are inferred on basis of asserted “precedes” relations, which determine possible transitions of SCOR process categories, within participants (S_x-M_y-D_y) or between them (D_1-S_1, D_2-S_2, D_3-S_3). The latter, cross-participant asserted transitions are valuable for the inference of source processes, on basis of principal product topology. For the selected product, employed processes are inferred by query:

```
SC_process and
(((preceededBy some
(employedBy some
(isComponentOf value <product>)))
and SC_source_process)
or
(employedBy value <product>))
```

Fig. 64 shows the main outcome of the semantic application - SCOR thread diagram. It is generated by the application script, on basis of data collected from SCOR-Cfg OWL file by DL queries above.

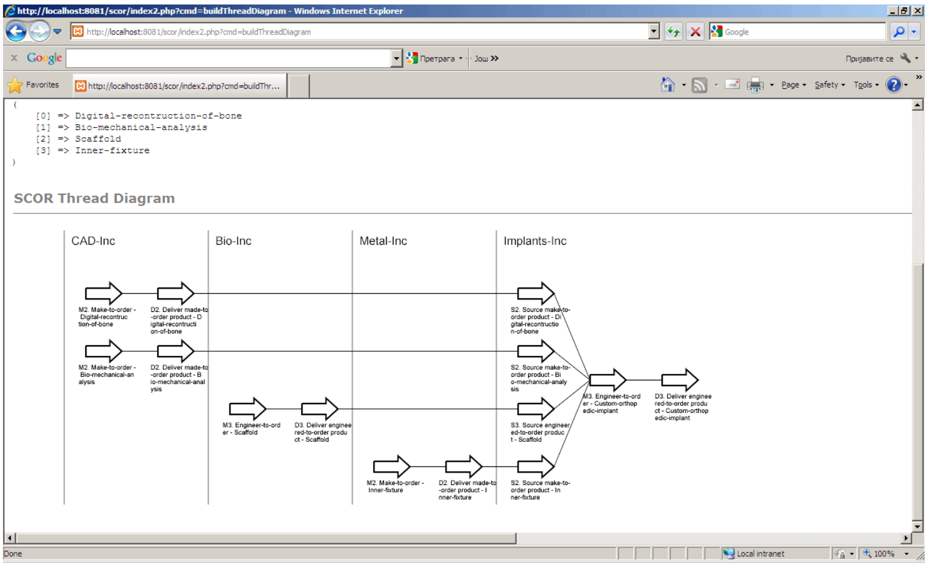

Fig. 64. Generated SCOR thread diagram

The SCOR thread diagram is a conceptual map of supply chain, built on basis of rules, asserted in SCOR-CFG OWL, prescribed by the SCOR framework. It enables the visual representation of high-level processes (process categories), roles and basic flows of information and resources between supply chain participants. Some of the features of the presented application are: inference of complex thread diagrams, generation of process models and workflows and generation of implementation roadmaps. These are elaborated below.

The case above shows only interactions and collaborations between two levels of a supply chain: principal customer and its first-tier suppliers. However, the number of visualized levels depends on the submitted product topology: if detailed product topology is entered (where principal product topology is represented as a bill of material, with a full structure to the level of parts, or raw material), full supply chain would be represented by the SCOR thread diagram, with the number of tiers corresponding to the depth of a principal product topology. Hence, the focal partner of the VE would be capable to gain full and detailed insight into progress of the project implementation. In the similar manner, for the purpose of better tracking of the production project, horizontal organization of individual supply chain actors (VE actors) can be represented in more detail, by inferring additional participants for different manufacturing strategies: warehouses (owning Deliver and Source processes), plants (owning Make processes) and headquarters (owning Plan processes).

Alignment relations between the SCOR-KOS and SCOR-Cfg OWL models also provide opportunities for the generation of detailed process implementation roadmaps, consisting of proposed best practices, relevant systems (or their modules, capabilities, intended use, etc.) for their execution, resource tracking (SCOR Inputs and

Outputs), environment for measuring the performance of a supply chain, by using the SCOR metrics and identification of the process interoperability issues.

3.4 Reasoning about process interoperability issues

The process interoperability may be considered as the interoperability between systems which implement the processes.

For example, the semantic application for supply chain configuration and SCOR-KOS OWL model enables the inference of the relationships between individual process elements, namely, the flows of the tangible and intangible assets between activities of the processes, identified in the generation of SCOR thread diagram. Figure **Fig. 65** illustrates the exchange of these assets for the case of engineered-to-order production, namely, between P2. Plan Source, S3. Source Engineered-to-Order Product, M3. Make Engineering-to-Order product and D3. Deliver Engineered-to-Order product. This process also involves following process categories: EP. Enable Plan; ES. Enable Source; EM. Enable Make; ED. Enable Deliver and P3. Plan Make. Only "P3. Plan Make" process category from the last group of categories is illustrated on **Fig. 65**, because of the visual representation complexity.

This example corresponds to the collaboration between the focal partner (Implants-Inc) and scaffold or inner fixture supplier (Bio-Inc or Metal-Inc, respectively) of the example supply chain for custom orthopaedic implants manufacturing.

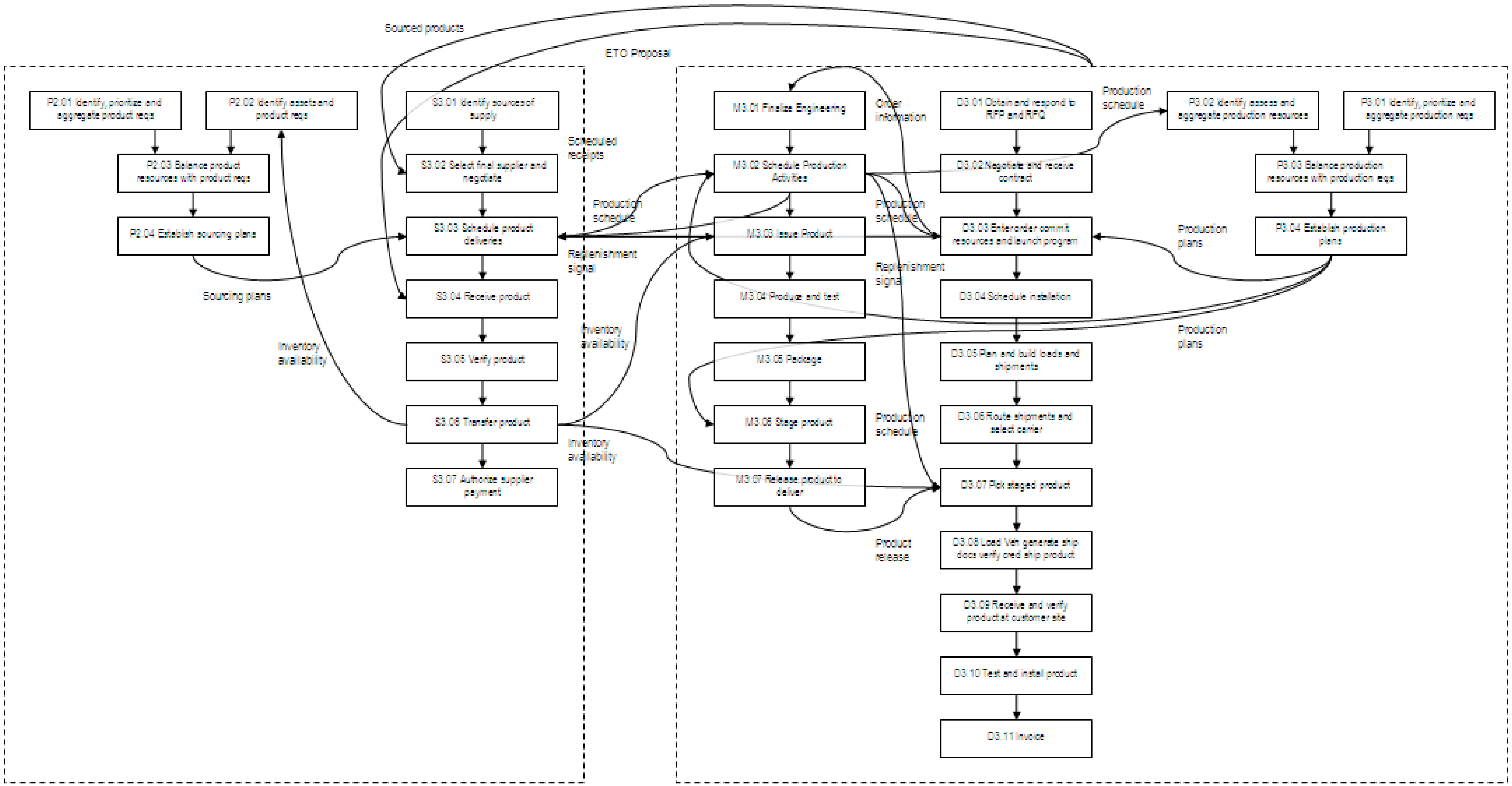

Fig. 65. The example of the assets flows between process elements for engineered-to-order production type

If we assume that both partners are using the ERP systems, these systems are considered as interoperable (in context of exchange information between SCOR processes), if they are capable to transmit and understand the information which is exchanged between following process categories:

- `S3_Source_Engineer-to-Order_Product` and `M3_Engineer-to-Order`
- `S3_Source_Engineer-to-Order_Product` and `D3_Deliver_Engineered-to-Order_Product`
- `M3_Engineer-to-Order` and `S3_Source_Engineer-to-Order_Product`
- `D3_Deliver_Engineered-to-Order_Product` and `S3_Source_Engineer-to-Order_Product`

Since interoperability is considered as unidirectional capability of the EISs, two different queries are needed to infer the concepts exchanged between two systems, where the first one implements Source process (of the focal partner) and the second one – Make and Deliver processes (of the supplier).

Information which needs to be sent from the focal partner's ERP system to supplier's ERP system and interpreted by the latter can be inferred by using following DL query:

```
(isOutputFrom some (isProcessElementOf value
S3_Source_Engineer-to-Order_Product)) and (isInputFor
some (isProcessElementOf value M3_Engineer-to-Order)) or
(isOutputFrom some (isProcessElementOf value
S3_Source_Engineer-to-Order_Product)) and (isInputFor
some (isProcessElementOf value D3_Deliver_Engineered-to-
Order_Product))
```

The above query results with following SCOR Input-Output elements:

```
Scheduled_Receipts
Inventory_Availability
```

In the opposite direction, following DL query is used:

```
(isOutputFrom some (isProcessElementOf value M3_Engineer-
to-Order)) and (isInputFor some (isProcessElementOf value
S3_Source_Engineer-to-Order_Product)) or
(isOutputFrom some (isProcessElementOf value
D3_Deliver_Engineered-to-Order_Product)) and (isInputFor
some (isProcessElementOf value S3_Source_Engineer-to-
Order_Product))
```

The above query results with following SCOR Input-Output elements:

```
Replenishment_Signal
```

```
Production_Schedule
```

The illustration at **Fig. 66** shows the interoperability requirements for two ERP systems which implement the corresponding SCOR processes, according to the generated SCOR thread diagram (the illustration shows only interactions between the ERPs of focal partner and selected supplier) and inferences related to the exchange of assets between those. It is very important to emphasize that inferred assets are relevant only when above mentioned SCOR processes environment is considered.

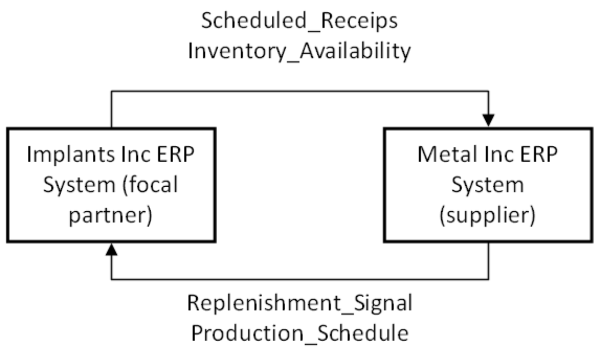

Fig. 66. The example of interoperability requirements

Scheduled receipts are the units (of any component or part) which are already scheduled to come in at a particular time. They are planned by the focal partner and delivered to the supplier on the basis of the supplier's production schedule – a plan for production, staffing, inventory, etc.

According to SCORs semantic enrichment - the SCOR-Full ontology (see Chapter 3, Section 3.2), production schedule is considered as sub-concept of "setting" notion and is represented explicitly by the concept "production-schedule", sub-concept of "function-schedule"->"schedule". Thus, the sameness of the instances of "production-schedule" concept of SCOR-Full and "Production_Schedule" instance of SCOR-KOS OWL (of SCOR_Input_Output type) is inferred by the following simple SWRL rule:

```
production-schedule(?x) ⇒ SameAs (?x, Produc-
tion_Schedule)
```

In SCOR-Full, a setting is defined as a circumstance of any type which affects some course of actions. It is associated with some state or configuration of the tangible (physical-item) or intangible (information-item) resources, namely, with an instance of "configured-item":

```
∀s (setting(s)) ∃ci (configured-item(c)∧ has-
realization(s,ci))
```

The production schedule "setting" is configured by the realization of "production-schedule-item" sub-concept of "information-item". Hence, "production-schedule" concept inherits the anonymous class, defined as (Manchester OWL syntax):

```
has-realization some production-schedule-item
```

"Production-schedule-item" concept inherits anonymous classes, defined as (Manchester OWL syntax):

```
has-product-information exactly 1 product-information
has-production-end-date exactly 1 dateTime
has-production-start-date exactly 1 dateTime
has-product-quantity exactly 1 float
```

,where "has-production-end-date" and "has-production-start-date" data properties are sub-properties of "has-date-value" data property, and "has-product-quantity" is sub-property of has-numerical-value data property. "Has-product-information" is a sub-property of "has-realization property". Hence, necessary conditions for having one production schedule item are: 1) to have exactly one product associated; 2) to have a production start date for this product; and 3) to have a production end date for this product (for more details, see Section 3.2 of Chapter 3).

Similarly, "product-information" information item is configured (hence, its realization is used in the range of first necessary condition above) by having exactly one product id associated:

```
has-product-id exactly 1 string
```

In addition, "function-schedule" concept also inherits the anonymous class:

```
∀fs (function-schedule(s)) ∃f (function(f)∧ sched-
ules(fs,f))
```

For the concept of "production-schedule", this condition is specialized to:

```
schedules some production
```

As shown above, the SCOR-Full ontology semantically describes the concept of production schedule. This description is mapped to the corresponding instance of the SCOR-KOS OWL model, so it can be used in the context of SCOR processes and in this case, for inference of the interoperability issues of the corresponding EISs.

These issues may be considered as resolved if the semantic descriptions of the concepts – messages which are exchanged between systems, are matched with the corresponding concepts of the corresponding systems. The elaboration and proposed solution to this problem is presented in Section 4 of this Chapter.

3.5 S-ISU formal model for semantic interoperability of systems in the Virtual Enterprise for custom orthopedic implants manufacturing

A SCOR thread diagram is not a process map. In fact, it is just a representation of supply chain configuration. This configuration can be considered as a closed-loop business process model if Plan processes and precedence relations between the process elements are added. A full process model can be generated by adding and exploiting rules for configuration of the SCOR Plan activities and other activities, needed for the process model, which are added manually. For example, in the case of custom orthopaedic implants manufacturing, following processes need to be considered:

1. Pre-operation planning, shared by the surgeon and VBE broker;
2. Design of the implant, shared by surgeon and VBE broker. In this process, a product infrastructure is launched. It contains the exhaustive information about the product itself, including the information, relevant for its production. Hence, this process also include planning of the sourcing, manufacturing and delivery sub-processes, related to scaffold and inner fixture products and services of analysis of biomechanical properties and digital reconstruction of traumatized bone;
3. Sourcing, manufacturing and delivery of scaffold and inner fixture. While sourcing sub-processes are owned by VBE broker or focal partner, manufacturing and delivery are assigned to the selected partners (as the outcome of sourcing process);
4. Sourcing, implementation and delivery of services of digital reconstruction of traumatized bone and analysis of bio-mechanical properties of the implant. Ownership of the sub-processes is assigned in a similar manner like in the previous bullet point;
5. Sourcing and implementation of the ethical committee review and delivery of review results.

The above processes include the activities, generated by the semantic application for supply chain configuration and are also formalized by S-ISU ontology (see Chapter 4, Section 3.2) in the formation phase of the VE. The goal of the formalization process is to identify the interoperation requirements between the systems, involved in the VE and to structure the assets, required for achievement of these requirements.

A VE for custom implant engineering is considered as a set of processes, configured by simple precedence relations. Then, VE is (formally) assembled of the enterprises which implement its processes. Partnership relation of the enterprise in specific VE is inferred as a property chain:

```
implements-process(Enterprise, Process) o is-process-
of(Process, VE)
```

Each of the processes is assigned to an individual enterprise in the process of VE formation, after selection.

Interoperations between two enterprises occur when a process, owned by one enterprise, precedes (or succeeds) the process of another. Hence, enterprise interoperation relationships may be inferred by using SWRL rule (see Chapter 4, Section 3.2):

```
Process(?p1), Process(?p2), Enterprise(?e1), Enter-
prise(?e2), implements-process(?e1,?p1), implements-
process(?e2,?p2), precedes(?p1,?p2), Different-
From(?e1,?e2)->interoperate-with(?e1,?e2)
```

Key concepts and properties for firing these rules are presented at **Fig. 67**. **Fig. 68** shows example processes (with asserted precedence relationships) of the VE for orthopaedic implants manufacturing, assembled of four enterprises, where implements-process property is illustrated by the pattern of the enterprise and process individuals.

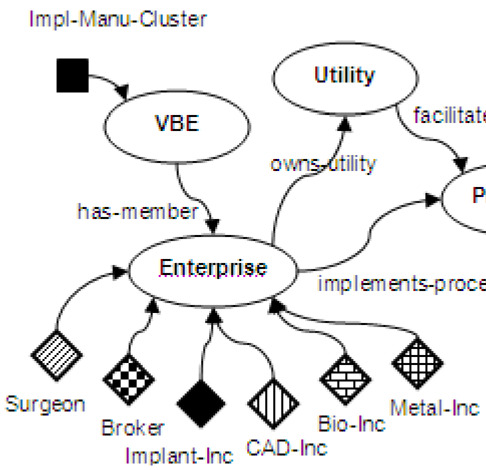

Fig. 67. Relevant assertions in organizational view of S-ISU ontology

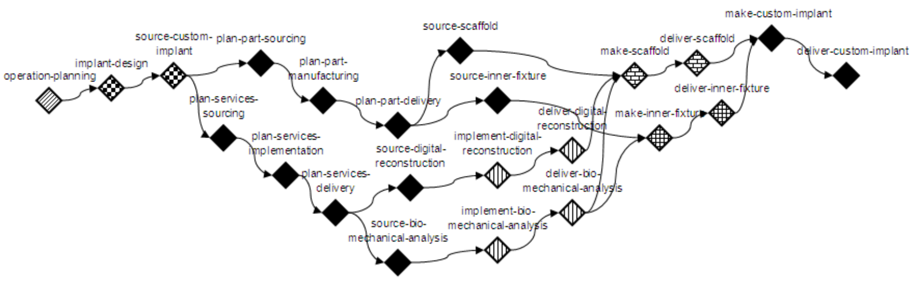

Fig. 68. Organizational view of S-ISU ontology in the case of orthopaedic implants manufacturing

Based on a rule above, following inferences are made in the presented case:

```
interoperate-with('Surgeon', 'Broker')
interoperate-with('Broker', 'Implant-Inc')
interoperate-with('Implant-Inc', 'Bio-Inc')
interoperate-with('Implant-Inc', 'Metal-Inc')
interoperate-with('Implant-Inc', 'CAD-Inc')
interoperate-with('CAD-Inc', 'Bio-Inc')
interoperate-with('CAD-Inc', 'Metal-Inc')
interoperate-with('Bio-Inc', 'Implant-Inc')
interoperate-with('Metal-Inc', 'Implant-Inc')
```

While **Fig. 68** illustrates the portion of the organizational view of S-ISU ontology, IT component architecture is described by its asset view, generated by dependency relationships. Dependency analysis is generated by inferring “uses” relationships between the components of S-ISU, on basis of the asserted sub-properties of transitive 'uses' property, such as imports(Ontology, Ontology), uses-data-container(Utility, Data-Container) and uses-utility(Utility or Actor, Utility).

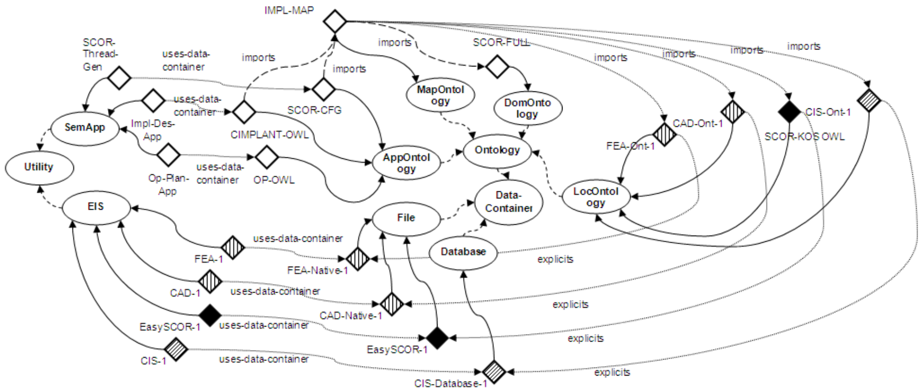

Fig. 69. Partial view of the asset perspective of S-ISU ontology in the case of orthopedic implants manufacturing

On **Fig. 69**, the concepts of the asset perspective of S-ISU ontology are represented with ovals (is-a relations are indicated by dashed lines) and instances with diamond symbols (their types are illustrated with full arrow lines). Object properties (imports, exploits and uses-data-container) are illustrated with dashed lines. Figure does not show the services of S-ISU architecture.

The asset perspective of S-ISU ontology is used to assert the existing systems and their data-containers; to identify the local ontologies; to assert the semantic applications, necessary for handling common, shared problems and respective application ontologies. **Fig. 69** is only partial illustration of the asset perspective and it also does not show the EISs of the involved partners (e.g. ERP systems), their data containers and local ontologies.

The instantiated model corresponds to the described case of bone cancer of tibia. It exploits the semantics background, which is formalized by the local ontologies (corresponding to data formats and database schemas of CAD, FEA, SCOR and CIS systems) and one domain ontology – SCOR-Full, the semantic enrichment of SCOR model (see Chapter 3, Section 3.2). The concepts of these local ontologies are mapped by using logical axioms which are stored in a central – IMPL-MAP ontology. Then, this exhaustive knowledge is exploited for the common purposes, by shared semantic applications for implant design (Impl-Des-App), pre-operation planning (Op-Plan-App) and process configuration (SCOR-Thread-Gen). These common purposes are formalized by respective problem ontologies – CIMPLANT-OWL, OP-OWL and previously described SCOR-Cfg semantic application for supply chain process configuration.

4 Retrieving knowledge from the Enterprise Information Systems

In the previous section, the alignment of implicit reference model and application ontology is exploited for supply chain process configuration in the case of the manufacturing of custom orthopaedic implants. Then, this configuration is used to infer the process interoperability issues and to identify information which needs to be transmitted and interpreted by the systems which implement the configured processes. This information (SCOR Input Output elements) is semantically enriched by using SCOR-Full ontology. Finally, it is shown how S-ISU ontology can be used to describe the organizational and asset perspective of the VE for custom orthopaedic implants manufacturing.

The presented case represents the top-down approach in the sense that it exploits the reference model and its semantic enrichment for the purpose of providing common service for the benefit of all members of VBE. It aims at delivering the consumer value of the semantic interoperability of systems, because it provides the tools which increases the efficiency of VE configuration and thus, reduces time and associated costs for its launch.

However, this consumer value is not significant, unless the semantic applications are capable of making specific conclusions (in a real-time) regarding functions they provide, in the context of the knowledge of the partnering enterprises – the members of the VBE. In other words, an approach is needed, first, to make this knowledge compatible or complementary and, second, to use this knowledge to make concrete, specific conclusions about the common function, e.g. VE partners selection, collaborative planning, collaborative product design, etc.

From the abstraction perspective, this approach is considered as bottom-up, because it involves: 1) explicitation of the enterprises' semantics; 2) establishment of the correspondences between these explicitations (formulated by the local ontologies), domain models (such as SCOR-Full, aligned with SCOR-KOS OWL) and application models (such as SCOR-Cfg); and 3) reasoning with an integrated ontological environment. The approach used in the step of explicitation is described in Chapter 4, Section 3.3, where it is shown how implicit sources of information about the enterprises, such as relational databases can be automatically transformed to formal, local ontologies. In Chapter 4, Section 3.4, it is shown how the semantic queries can be used to extract the instances of the local ontologies.

In this section, these steps are demonstrated on the case of the collaboration between two members of the VE for orthopaedic implants manufacturing. It is shown how the knowledge about specific EISs can be transformed to formal local ontology and how the production schedule can be extracted from this local ontology, by using semantic enrichment of SCOR – SCOR Full ontology.

4.1 Case description and motivation

In Section 3.4, it is presented how the interoperability issues related to the communication between the focal partner of VE for orthopaedic implants manufacturing and supplier of implant parts, can be inferred. Then, the information, which is exchanged between two systems of the focal partner and supplier, is semantically enriched, by using SCOR-Full ontology. The formal definition of production schedule can be, now used to extract semantically equivalent notions, namely production schedules, from these two systems.

One of the benefits of the semantically interoperable systems (see **Fig. 13**) is the possibility to use the single criterion (or criteria) to infer the statements that hold true in all these systems, despite their heterogeneous structure.

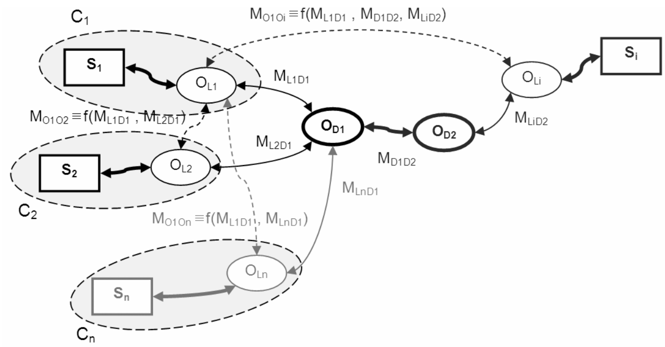

Fig. 70. Semantic interoperability of systems.

Namely, specific semantic query executed against the local ontology O_{Li} would normally infer triples of information from the database of S_i. However, if mappings (or logical function of mappings) between O_{Li} and O_{Lj} exist, inferred triples will also include information from the database of S_j.

In the case of the VE for implants manufacturing, a semantic query, which is made of the notions of SCOR-Full (or other) domain ontology, can be used to find the production schedule for a specific product, manufactured by the supplier of implant parts. This production schedule is stored in the database of the EISs of the supplier, but is formalized by the appropriate local ontology. As different EISs store different structures of the same data, the concepts of these local ontologies need to be appropriately mapped to the concepts of domain ontology which is used to extract the required information.

4.2 Generating local ontology of the OpenERP system

The approach to generation of local ontology is implemented on the case of OpenERP EIS. In the case of manufacturing of the custom implant, it is assumed that the system is owned by Metal Inc. enterprise, a supplier of the inner fixture part of the custom implant. The local ontology is generated with objective to facilitate resolution of process interoperability issues (see Section 3.4), so focal partner's system can easily access the production schedules of the supplier.

OpenERP is an open source suite of business applications including sales, CRM, project management, warehouse management, manufacturing, accounting and human resources. According to author, it is an open source alternative for SAP ERP and Microsoft Dynamics. OpenERP is a client-server suite, where the client communicates with the server by using XML-RPC interfaces. It uses PostgreSQL relational database for data storage.

OpenERP database schema is transformed to a local ontology by using the web application which implements the described method (see Chapter 4, Section 3.3).

With all modules installed, OpenERP database counts 238 tables. In the first step of database import into er.owl model, namely, instantiation of the OWL representation of ER model, 3806 individuals are created (2633 of "attribute" type, 238 of "entity" type and 934 of "relation" type) and 7999 object property assertions are made. These individuals and their asserted properties directly correspond to the structure of OpenERP database schema and they are their literal OWL representation.

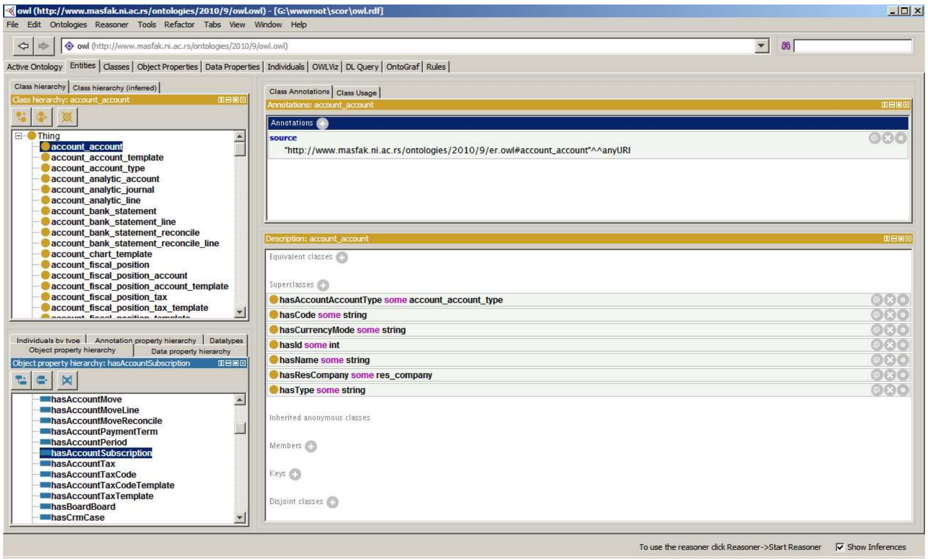

Fig. 71. OpenERP local ontology in Protege.

In the second step of transformation process, classification of OWL concepts and properties is done by the reasoner and s-er.owl model is generated. 193 concepts, 493 data-concepts and 2779 properties are inferred, on the basis of the SWRL rules, presented in section 3.3, executed on the literal OWL representation produced in a former step. All inferences are stored in a separate OWL file, which is considered as meta-model of OpenERP database schema, in order to reduce the processing requirements for the final step.

In the final step of local ontology generation, application transformed classified instances of the meta-model of the OpenERP database to the corresponding OWL concepts and properties (see **Fig. 71**).

Resulting OWL file is considered as the output of described database-to-ontology transformation process. In the case of OpenERP, additional work on lexical refinement is not necessary because the database developers used natural language to describe the entities and their attributes.

Resulting conceptualization, namely, generated local ontology fully corresponds to user perspective of OpenERP system. This is demonstrated below, in the description of the manufacturing module of OpenERP system.

Manufacturing module of OpenERP EIS facilitates management of master data about products, master Bill of Materials, work centers and routings; it automates procurements management, manufacturing and purchase scheduling; it facilitates management of the manufacturing and delivery orders and after-sales services. **Fig. 72** displays the fragment of the UML representation of OWL concepts and relations (from generated local ontology) which describe the manufacturing module of OpenERP.

The basis for manufacturing management in OpenERP is management of master data, namely, Bills of Materials, work cells and routings. Bills of materials ("mrp_bom" concept on **Fig. 72**) describe the single or multi-level structure of the product ("product_product" concept) to be manufactured – sub-assemblies or raw material, each of which can be moved from stock or manufactured or purchased (determined by "hasType" functional property of "mrp_bom" concept). Work cells ("mrp_work_center") represent units of production (machines or human resources, determined by "hasType" functional property), capable of doing material transformation operations, with certain production capacity, expressed in cycles (for machines) or hours (for human resources). Routings ("mrp_routing") define the manufacturing operations to be done in work cells to produce certain product. They are associated to bills of materials.

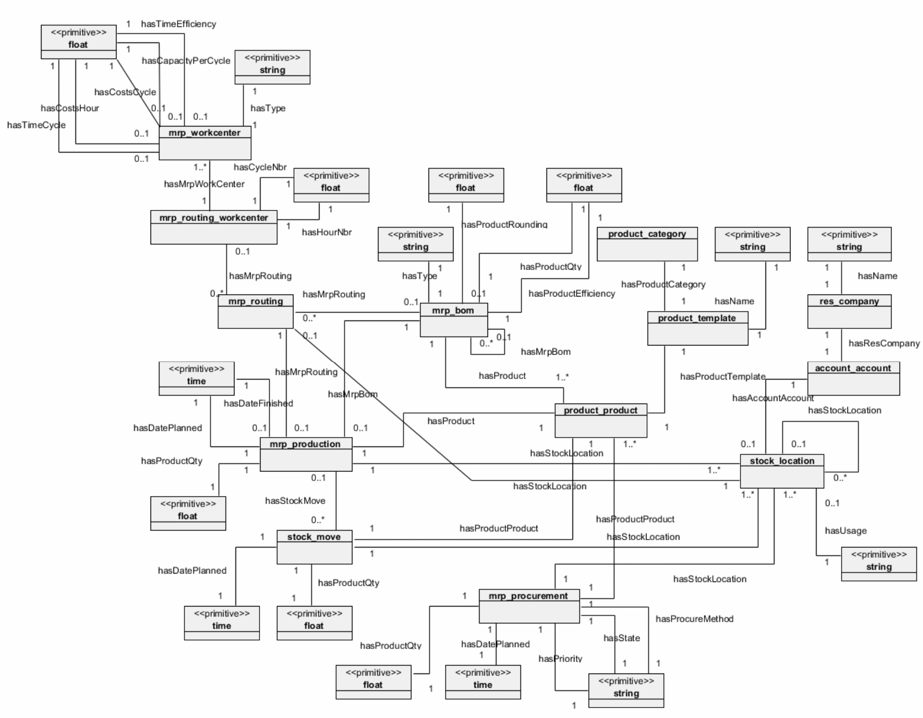

Fig. 72. Fragment of UML representation of OpenERP local ontology.

Once the master data is defined, the system can automatically generate the production schedule (schedule of generation of production – "mrp_production", and procurement – "mrp_procurement" orders) by using make-to-order rules, minimum stock (for make-to-stock production) rules or production plan (based on forecasts). For make-to-order production, orders are computed on the basis of quantity of the ordered product, bill of material and delivery date. For each of the product's elements which are supplied, a procurement order is generated. Planned dates ("hasDatePlanned" property) for the orders are calculated on the basis of a delivery date and manufacturing and purchase lead times for the product elements. For make-to-stock production, instead of the delivery date, minimum stock rules are used for production scheduling. In this case, orders are launched when minimum stock thresholds are reached.

The logistics of production is managed on the basis of stock moves ("mrp_stock_move" concept). OpenERP supports three types of stock locations: physical stock locations (warehouses), partner locations (customers' and suppliers' stocks) and virtual locations. The notion of stock location is used to define pull and push flows and to manage all types of storage places, including internal, supplier, customer, production and others. It is used to manage manufacturing logistics, since each of the manufacturing operations (described by "mrp_routing" concept) can be associated to a single stock location.

The above description of how OpenERP system works with manufacturing management semantically corresponds to this domain's conceptual model, illustrated at **Fig. 72**. However, although the principles above are used to manage production in many other (if not all) ERP systems, they are all realized by the different database schemas. The differences in conceptualization approaches of the ERP systems designers have negative effect on the capabilities of these systems to cooperate. This problem is resolved by applying the reconciliation methods of different semantic models, such as different explicit representations of the implicit realities of two systems, namely, local ontologies and conceptual models of a specific domain. In the next subsection, the semantic differences between the concepts representing the notion of production schedule in the domain ontology (SCOR-Full) and local ontology of OpenERP system are elaborated.

4.3 Semantic correspondences between the concepts of OpenERP and SCOR-Full ontologies

In Section 3.4 of this Chapter, the semantic description of the concept production-schedule of SCOR Full domain model is presented. This description corresponds to the illustration on **Fig. 73**.

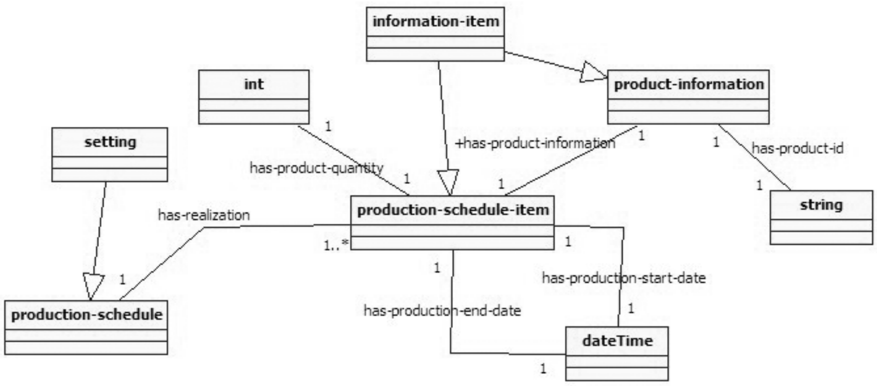

Fig. 73. Semantic description of production-schedule concept in SCOR-Full ontology.

In OpenERP local ontology, the production schedule is described by the "mrp_production" concept. This concept inherits following anonymous classes, which correspond to the necessary conditions for the inference of the concept:

```
hasDatePlanned some time
hasId some int
hasName some string
hasProductProduct some product_product
hasProductQty some float
```

```
hasStockLocation some stock_location
```

In addition, ‘mrp-production” concept is in range of the following properties:

```
hasDateFinished(mrp_production, time)
hasStockMove(mrp_production, stock_move)
hasMrpBom(mrp_production, mrp_bom)
hasMrpRouting(mrp_production, mrp_routing)
```

Given the semantic descriptions of “production-schedule” concept of SCOR-Full and “mrp-production” of OpenERP ontology, it can be easily concluded that intensional definition of the “production-schedule-item” concept semantically corresponds to the “mrp-production” concept of OpenERP local ontology. Thus, these two concepts are considered as logically equivalent.

Hence, the following SWRL rules can be asserted to the mapping ontology:

```
(1) production-schedule-item(?x) ⇒ SameAs (?x, mrp-
    production)
(2) mrp-production(?x) ⇒ SameAs (?x, production-
    schedule-item)
```

The SCOR-Full concept of “product-information” corresponds to OpenERP’s “product_template”, which is associated to a “product_product” concept by “hasProductTemplate” property of the “product_product” concept. The anonymous superclasses of the “product_template” concept of OpenERP local ontology are:

```
hasCostMethod some string
hasId some int
hasMesType some string
hasName some string
hasProcureMethod some string
hasProductCategory some product_category
hasStandardPrice some decimal
hasSupplyMethod some string
hasType some string
```

Finally, following SWRL rules can be asserted with goal to enable semantic querying of production schedules for specific product on the local ontology.

```
(3) product-information(?x) ⇒ SameAs (?x, prod-
    uct_template)
(4) product-template(?x) ⇒ SameAs (?x, prod-
    uct_information)
(5) product-information(?pi) ∧ has-name(?pi, ?n) ⇒ has-
    Name(?pi, ?pid)
```

```
(6) production-schedule-item(?psi) ∧ has-product-
    information(?psi, ?p) ∧ hasProductProduct(?psi, ?pp)
    ⇒ hasProductTemplate(?pp, ?p)
```

Rules (3) and (4) establish the logical equivalence relationship between SCOR-Full concept of “product-information” and OpenERP local ontology’s concept of “product_template”. Rule (5) establishes the logical equivalence of the properties “has-name” (of SCOR-Full) and “hasName” (of OpenERP), in the case that “production-information” concept (or equivalent “product_template”) is in the domain of has-name” property. The last rule (6) establish the logical relationships between “has-product-information” property of “production-schedule-item” (of SCOR-Full) and “hasProductProduct” and “hasProductTemplate” properties of OpenERP local ontology.

Semantic correspondences between the concepts of local and domain ontology are established with a main objective to enable reasoning on the local ontology where the concepts of domain ontologies are used for building a semantic query. Hence, it becomes possible to use the terms of domain ontology to infer on the multiple local ontologies, representing different partners in a Virtual Enterprise. This is elaborated in the next sub-section.

4.4 Execution of semantic queries

Once the local ontology of OpenERP system is generated and correspondences between this ontology and domain ontology (e.g. SCOR-Full) are established, the method for semantic querying of the local ontologies, described in Chapter 4, Section 3.4 can be applied to facilitate extraction of the relevant information, such as a production schedule for a specific product.

For example, in Section 3.4 of this Chapter, reasoning about process interoperability issues is elaborated. Those issues are related to interoperation between the ERP systems of two partners of VE for custom implant manufacturing (Fig. 12). In this case, it is inferred that the production schedule information is exchanged between these two systems. Here, the focal partner can use a semantic query to extract the production schedule for a given part from the database of the ERP system of a supplier (inner fixture F12). This semantic query can be written by using a common dictionary of the VE, in this case – SCOR-Full ontology, and is as follows:

```
has-realization some (production-schedule-item and has-
product-information some (has-name value "Custom inner
fixture F12"))
```

Assuming that the semantic correspondences between SCOR-Full and OpenERP local ontologies are established as elaborated above, this semantic query is expected to return all instances of SCOR-Full “production-schedule” concept which satisfy the condition of being associated with products whose name is “Custom inner fixture F12”, where those instances are formal representations of data from the database of OpenERP system.

The queries like the one above are executed against the central, mapping ontology, which imports all ontologies and stores SWRL rules which define logical correspondences between their concepts. Two scenarios of this execution are possible, depending on the approach to instance population (see Section 3.3. of Chapter 4). Massive dump approach to instance population assumes that all data is represented as individuals in the process of ontology generation (or mapping of existing ontology with database schema). In this case, the semantic query above would normally return all relevant instances.

Query-driven population approach (see **Fig. 74**) assumes that individuals are asserted to ontology at the time of the semantic query execution. This approach is used in the development of S-ISU environment. In this approach, two types of query rewriting mechanisms are needed; first one needs to be capable to transpose the semantic query, written by using the language of domain ontology, into another semantic query, which can be executed then on the local ontology.

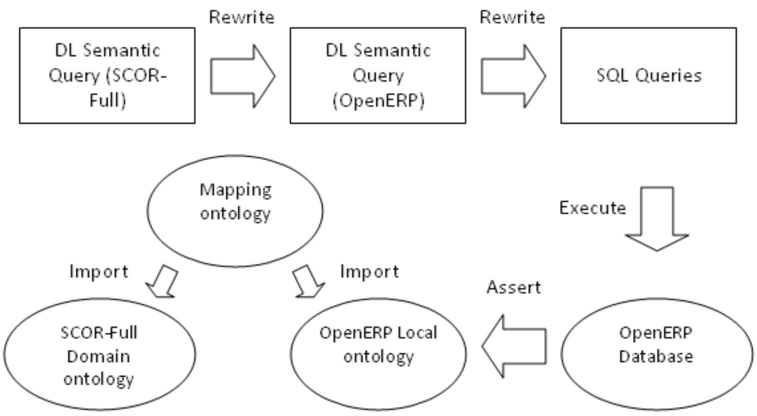

Fig. 74. Semantic querying of query-driven populated ontology.

The second type of query rewrite mechanism is needed to transform the semantic query to SQL query or queries which are executed in the database; result-sets are then represented as logical statements which are finally asserted to local ontology. This, second type is developed in the scope of research presented in this thesis and is explained in detail in Section 3.4 of Chapter 4.

The DL query which returns the production schedule for the product (part) with name "Custom fixture F12" from the local ontology of OpenERP system is:

```
mrp_production and hasProductProduct some (hasProductTem-
plate some (hasName value "Custom inner fixture F12"))
```

According to the method, in the first step of semantic query execution, the query is decomposed to following 4-tuplets:

```
X hasProductProduct some bNode1
```

```
bNode1 hasProductTemplate some bNode2
bNode2 hasName value "Custom fixture F12"
```

In the next step, SQL queries are generated for each of the 4-tuplet, from bottom up. The domain of "hasName" property of OpenERP ontology is the union of 170 sets – concepts, each of which corresponds to a data table. Hence, the resulting SQL query is an array of 170 SELECT queries.

The SQL queries, generated by the module for semantic query execution for the last 4-tuplet are, as follows:

```
(1) SELECT * FROM account_account_template WHERE
    name='Custom fixture F12'
(2) SELECT * FROM account_account_consol_rel WHERE
    name='Custom fixture F12'
....
....
(65) SELECT * FROM product_template WHERE name='Custom
    fixture F12'
....
....
(170) SELECT * FROM wkf_workitem WHERE name='Custom fix-
    ture F12'
```

The queries are executed and resulting datasets are transformed into logical statements which are, then, asserted to a temporary model.

The query (65) above returns the product template description, matching the given criteria. The result-set is then transformed into the logical statements, which describe an instance of "product_template" concept and its necessary conditions.

```
custom-fixture_f12 type product_template
custom-fixture_f12 hasCostMethod 'Average price'
custom-fixture_f12 hasId 1332
custom-fixture_f12 hasMesType 'Measure type'
custom-fixture_f12 hasName 'Custom fixture F12'
custom-fixture_f12 hasProcureMethod 'Make to Order'
Inner-Fixtures type product_category
Inner-Fixtures hasName 'InnerFixtures'
Inner-Fixtures hasId 12
custom-fixture_f12 hasProductCategory Inner-Fixtures
custom-fixture_f12 hasStandardPrice 540.00
custom-fixture_f12 hasSupplyMethod 'Produce'
custom-fixture_f12 hasType 'Product type'
```

These logical statements are then asserted into temporary model (stored in memory space of the semantic querying engine).

It is important to emphasize that a query execution procedure is recursive. The query is expected to extract from the database and assert all necessary conditions for a given concept. When the result set includes a field which is at the destination of one-to-many schema relationship, the algorithm signals the occurrence of another concept (not a basic data type) as a necessary condition. In this case, another SQL query is executed to extract the result set which corresponds to this concept. In the above example, for the definition of necessary conditions of “product_template” concept, the instance of the “product_category” concept needs to be constructed and asserted to a temporary model.

In the next iteration of the query execution, next 4-tuplet is transformed into a set of SQL queries. As it is shown above, value restrictions are transformed to SQL queries in a simple way, where basic data-types (in this case, strings) are used as criteria. In this iteration, the criterion is defined with an instance(s) of the ontology (in this case, bNode2 array). In the example above, only one instance is asserted into local ontology, as a result of a first iteration. Thus, in the second iteration, following statement is transposed to SQL queries:

```
bNode1 hasProductTemplate custom-fixture-f12
```

When existential restrictions are used, SQL WHERE statements are interpreted as the values of the functional data properties of this instance:

```
custom-fixture_f12 hasId 1332
```

Given the fact that the domain of “hasProductTemplate” property is a union of three concepts (“product_pricelist_item”, “product_product” and “product_supplierinfo”) in OpenERP local ontology, following set of SQL queries is generated:

```
(1) SELECT product_pricelist_item.* FROM prod-
    uct_pricelist_item, product_template WHERE prod-
    uct_pricelist_item.product_template_id=product_templa
    te.id AND product_template.id='1332'
(2) SELECT product_product.* FROM product_ product, prod-
    uct_template WHERE product_ prod-
    uct.product_template_id=product_template.id AND prod-
    uct_template.id='1332'
(3) SELECT product_ supplierinfo.* FROM product_ sup-
    plierinfo, product_template WHERE product_ supplier-
    info.product_template_id=product_template.id AND
    product_template.id='1332'
```

In this example, only the second SELECT query returns some value, because custom fixture product is engineered to order, so no pricelist or supplier information is relevant for its description. Similarly like in the case of the first iteration, a result set is transformed into a set of logical statements, which describe the instance of “product_product” concept of OpenERP local ontology, by using its necessary conditions:

```
custom-fixture_f12_p type product_product
custom-fixture_f12_p hasId 67
```

```
custom-fixture_f12_p hasProductTemplate custom-
fixture_f12
```

These logical statements are also asserted into temporary model. In the last iteration, a domain of “hasProductProduct” property is determined for a given range (“custom-fixture_f12_p” instance). Then, the value of functional property of a criterion instance is used to generate SQL query. This set has 22 SELECT queries because the domain of the “hasProductProduct” property is union of 22 classes:

```
(1) SELECT account_analytic_line.* FROM ac-
    count_analytic_line, product_product WHERE ac-
    count_analytic_line.product_id=product.id AND prod-
    uct.id='67'
...
(7) SELECT mrp_production.* FROM mrp_production, prod-
    uct_product WHERE
    mrp_production.product_id=product.id AND prod-
    uct.id='67'
...
(22) SELECT stock_warehouse_orderpoint.* FROM
    stock_warehouse_orderpoint, product_product WHERE
    stock_warehouse_orderpoint.product_id=product.id AND
    product.id='67'
```

In contrast to a previous iteration, in this step, the instances of more than one concept of OpenERP local ontology are returned – all instances to which the custom fixture product is associated (the domain of “hasProductProduct” property), such as account_invoice_line, delivery_carrier, mrp_bom, and others. Then, the result sets are transformed to logical statements which are asserted to a temporary model. Some relevant statements are:

```
custom-fixture_f12_prod_sched type mrp_production
custom-fixture_f12_prod_sched hasDatePlanned '2012-02-15
23:59:59'
custom-fixture_f12_prod_sched hasId 67
custom-fixture_f12_prod_sched hasName 'Production sched-
ule for Custom fixture F12'
custom-fixture_f12_prod_sched hasProductProduct custom-
fixture_f12_p
custom-fixture_f12_prod_sched hasProductQuantity 3.0
custom-fixture_f12_prod_sched hasDateFinished '2012-02-17
23:59:59'
stock_location_w2 type stock_location
stock_location_w2 hasAllocationMethod ''
stock_location_w2 hasChainedAutoPacking ''
stock_location_w2 hasChainedLocationType ''
```

```
stock_location_w2 hasId 8
stock_location_w2 hasName ''
stock_location_w2 hasUsage 'Warehouse 2'
custom-fixture_f12_prod_sched hasStockLocation
stock_location_w2
```

At this time, all instances required for the semantic representation of the query result are stored in a temporary model, in the memory of the inference engine. A second step of the semantic query execution method – query execution and assertions can be considered as completed.

In the third, last step of the method; a semantic DL query is executed on the temporary model, in order to filter only relevant instances. Namely, as it is shown above in the description of the third iteration of the query execution step, the property domain inferences may result with some excessive information which is not relevant for the case. Also, in case where the complex semantic queries (with multiple restrictions on the desired instance, see Section 3.4. of Chapter 4) are executed, the intersection of the resulting instances' sets, each corresponding to the individual restrictions, need to be inferred. Finally, this filtered model is returned as an end outcome of the semantic query execution. The representation of the outcome of the production schedule querying for the product "Custom fixture F12" is illustrated on **Fig. 75** (data properties are not displayed).

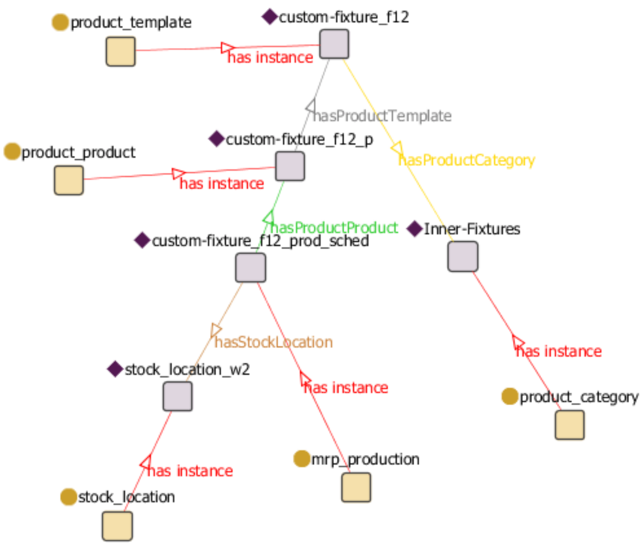

Fig. 75. Visual representation of the production schedule for example product "Custom fixture F12".

The resulting graph is a semantic representation of the production schedule concept and is delivered after the semantic query is transformed to a set of SQL queries which are executed in the database of OpenERP system. Now, its concepts and instances can be mapped to the domain models and, hence, more advanced reasoning may be enabled. More important, a production schedule concept of OpenERP local ontology may become logically equivalent to the corresponding concepts of other systems' local ontologies. Thus, these systems will become capable to logically interpret messages which encapsulate different production schedules.

Chapter 6: Conclusions

of research and development of the Formal framework for semantic interoperability in supply chain networks

Abstract. In this thesis, the methodology and process of development and verification of a formal framework for representation and reasoning of knowledge in supply networks is presented. As a conclusion, a discussion of the presented results is given in this Chapter. This discussion includes a critical view to the state-of-the-art in the relevant scientific areas, main features and restrictions of the presented methodology and the resulting ontological and technical framework, elaboration of the possible impact and identified gaps. The latter is used to set future research directions, important for improving the usability and applicability of the presented methodology. The discussion also includes explicit answers to the research questions, set in Chapter 1 of this thesis.

1 The impact of the state-of-the-art research to semantic interoperability in supply chain networks

Despite the potential decrease in operational costs and complexity, introduced by the homogenous systems, it is a fact that enterprises will continue to have mixed ICT environments for the foreseeable future. The main reason is leveraging the existing investments and specific requirements, which cannot be addressed by the "standard" architectures. It is even expected that, due to increase of the data complexity (related to increased enterprises' demand for automation) and further ICT developments (particularly related to future internet technologies), the rate of the heterogeneity in the systems architecture will increase. Thus, interoperability is expected to become more critical feature of the EISs. This assumption raises the important question on the readiness of the existing research of enterprise interoperability to deliver the practical results and related benefits.

The great most of the relevant works in this area is based on the results of IDEAS, ATHENA and INTEROP NoE projects. Discussion in Section 2.3 of Chapter 2 shows that these works are strongly compatible. While IDEAS focuses on structuring interoperability issues, ATHENA adopts IDEAS framework and it seeks for solutions for those issues. Then, EIF goes one step back to define the interoperability barriers and to discuss those within each of the ATHENA's interoperability levels. One should consider that the theoretical background of the enterprise interoperability as a scientific topic is already set in the scope of the above projects. The content of the EU FP7 work programs shows that enterprise interoperability is currently researched at the implementation level, where mostly technical paradigms are developed on the top of the current Internet infrastructure, to enable the adaptation and implementation of the conceptual frameworks, developed in the past.

However, despite many efforts in development of interoperability frameworks, the enterprise interoperability related concepts are not yet sufficiently defined. Formal statements of interoperability domain and interoperability domain ontology are needed (Chen et al, 2008). Some initial work to elaborate ontology of interoperability has been performed within ATHENA and INTEROP NoE, but the development has not reached a sufficient maturity. Furthermore, current results are, to some extent, inconsistent with existing definitions of the enterprise interoperability.

Interoperability is related to the federated approach, which implies that systems must accommodate on the fly in order to interoperate – no pre-determined assets are assumed. However, this is not the case with the existing frameworks. The use of semantics (ontologies) is suggested to enable reconciliation between two systems on different levels. But, the structuring of the interoperability problem into proposed levels poses a serious constraint in this case. Namely, individual level, as proposed by the frameworks, cannot be semantically analyzed (by implementing a full ontological commitment) in isolation from the others. This type of approach poses the technical difficulties, which, on the other hand, increase the amount of technical requirements to be fulfilled so two systems can become interoperable.

In conclusion, it is the author's opinion that enterprise systems should not be exposed to the interoperable environment by the levels or any other conceptual categories, but by ontologies. Then, ontologies should be used in reconciliation and semantic querying process as an asset to determine or evaluate the level of interoperability. So, the only pre-determined asset, which is needed so two system can interoperate is a common semantics. In a way, the discussion on the basic architecture of the semantically interoperable systems in Section 4.2 of Chapter 2 shows that this kind of technical independence can be achieved by making two systems semantically interoperable.

When considering the principles of the semantic interoperability of systems, described in Section 4 of Chapter 2, it can be concluded that it is unconditional and universal. It is not structured by the levels, nor does it assume the particular kind of architecture for its implementation (by using, for example, SaaS paradigm). Restrictions may occur, but they can be only related to: a) incompleteness and lack of validity of logical correspondences between two ontologies; b) expressiveness of the implicit models, namely local ontologies; c) expressiveness of the languages, used to formalize those models; or d) restricted access to some of the information, modelled by the parts of local ontology.

The above position on the structuring of interoperability implies one important assumption: the problem of enterprise interoperability should be reduced to the problem if interoperability of their information systems. Although very relevant for the enterprise interoperability, the problem of alignment of the organizational and ICT perspectives of one enterprise is not an interoperability problem. It is dealt by the scientific topic of enterprise architecture and MDA paradigm. Also, this topic is expected to provide the (weak) formalisms which can be used as a context of semantic interoperability of systems.

In the field of enterprise architecture, Chen et al (2008) identified some major research issues that need to be tackled in the future, for the benefit of the enterprise interoperability. They argue on the need to put more efforts in development of Type 1 reference architectures at higher level of abstractions. These would facilitate more efficient process of enterprise engineering and integration. In the opposite direction, enterprise architectures need to be associated with higher abstraction formalisms, the languages or ontologies for representing enterprise architectural structure, features and properties at earlier stages of design. Also, the exploitation of the enterprise architectures may be enhanced if architecting principles and evaluation methods are more developed and the existing architectures – justified, from the conceptual (requirements, purpose) and economical (benefits for the enterprise) view. Last, but not the least, continuous alignment of business and IT architecture is considered as one of the greatest challenges for implementing enterprise architecture in industry. This alignment addresses the problems occurred during the processes of IT change management and evolution of the enterprise architectures and is expected to be implemented by using MDA paradigm.

When considering the semantic interoperability of systems, the approaches to conceptualization used to develop systems and consequently, the extraction of this semantics are still important issues because of the different, often contextual understanding of tacit knowledge embedded into enterprise systems. These issues are typically driven by the misbalance of the needed ontological commitment and epistemological dimension in the conceptualization process. In this sense, the task of the EIS conceptualization is not really to conceptualize the EIS models, but to make the assumptions on the mental models of the information systems' designers, which they then expressed as Entity-Relationship models, and to introduce the ontological commitments by making those models fully or partially equivalent to the real world semantics. The analysis of existing database-to-ontology approaches, presented in Section 4.5 of Chapter 2 shows that this objective is not yet achieved. The presented approaches suffer from serious weaknesses with regard to lack of full interpretation of the ER models, mainly related to lack of logical implications of the cardinality of relationships and existential constraints (mandatory elements).

Still, there are some more general issues of restricting the domain of the conceptualization to database schema. Sometimes, ER models, namely database schemas, do not capture the semantics of the application functionality and underlying data models; when information systems are highly generic, the application semantics is actually captured in the populated table rows. For example, in Business Process Management systems, the structure of the enterprise processes, namely activities, associated data structures (messages), compensation and error handling blocks, etc. are defined by a system user and are not expressed by the database schema. This issue is evident even in trivial cases. For example, attribute of "type" is often used by database developers to describe some entity. It is typically transformed to hasType(string) property. In this case, the meaning of this property is unknown, because of the ambiguity of the linguistic term of "type". Similar remark can be made also for often used notion of “status”. However, sometimes this meaning can be determined if the list of associated data (strings) in database rows is semantically analyzed in the context of domain (entity) of the property above. For example, if OWL is used as a formalism, “hasType some bNode” construct may be used to model this property, where bNode is anonymous class that contains enumerated (owl:oneOf) elements which correspond to data associated to the attribute. In a more formal approach, the values of those attributes may be considered as classifiers of the subsumed classes. For example, the property hasType(string) of the concept Machine tool, asserted with one of the following values: “turning”,”milling” and ”drilling” may enable inference of the respective subconcepts of the Machine tool concept - Lathe, Mill and Drill.

In above cases, the intervention of the domain expert in enriching the conceptual model is inevitable. Some research is tackling this issue by providing the tools to automatically or semi-automatically discover the semantics buried into existing data patterns (Astrova, 2004). Anyway, it is highly unlikely that the ideal of fully automated process of local ontologies development will be reached in a near future. Generated local ontologies should always be considered as intermediary models, which need to be refined by the domain or EIS experts.

The final research question asked in Chapter 2 is: what are the practical benefits of the enterprise (semantic) interoperability; or what is the impact of the relevant results of the scientific topics summarized in this Chapter to the way business is actually done?

Section 6 of Chapter 2 provide the answers on how the current issues of the traditional supply chains will be resolved in the future and what are the directions for establishment of what is considered as new organizational forms. Although significant innovation is made in this topic, the essence of the supplier-customer relationships remains the same as in what is considered as traditional supply chains. The economic phenomena, such as globalization, outsourcing, increased demand for customization and specialization do not change this essence. This is the reason why the title of this thesis still refers to the supply chains, and not to the new terms of Virtual Enterprise or Collaborative Networked Organization. However, it is a fact that these new circumstances of doing business, as well as new requirements for flexibility and rapid market response, have big impact on how the supplier-customer relationships are established and facilitated. It is expected that corresponding new methods and ICT facilities will directly benefit from the advances in the topic of semantic interoperability of systems. Namely, its main feature and advantage over conventional interoperability or integration is lesser technical preconditions needed for systems interoperation and thus, lesser operational costs and shorter time needed for the implementation. The latter is considered as critical for new dynamic supply chains, which are created for the individual opportunities.

2 The language of interoperability: an approach to formalization of supply chain operations

Semantic interoperability can be easily explained by using the basics of the human communication. When enterprises are exchanging messages during a collaborative process, their information systems are responsible for articulating, transmitting and interpreting these messages. By default, information systems are not built with the purpose to cooperate. Hence, the “language” they understand is a local language and it is not useful for communication. This is the main issue of EISs interoperability. In this thesis, this issue is addressed by proposing a formal framework for collaboration in a supply chain – a basic form of collaboration between two or more enterprises. The main principles of this formalization process are described in the remainder of this Section. Provided elaboration of the methodology applied in the formalization processes is intended to answer on the following research questions:

- What are the main principles for the development of a formal model which may facilitate a semantic interoperability in a supply chain environment?
- What are the most suitable method and/or approach to its development?
- How will this model fit into the formal description of the semantic interoperability of systems?

When all local languages are translated to universal domain knowledge, this domain knowledge is then used as a facilitator for the communication. The pre-condition for implementing the above scenario is to have all local languages and domain knowledge - formally described, by using the same formalism. When same formalism is used for all those formal descriptions, it is also possible to define correspondences between the notions of the local languages and domain knowledge. Now, domain knowledge can be considered as advanced dictionary, which is used to formally define meanings of all terms of the exchanged messages.

The meaning is formally defined because it is intended to be computable or inferred by the different agents for the different purposes. This formal definition aims at bringing closer the symbols, used to formally describe a particular object, to its typical mental representation. With regard to this, the logical positivists strongly argued that the meaning is nothing more or less than the truth conditions it involves. Here, the meaning is explained by using the references to the actual existing (possibly also logically explained) things in the world. The process of the representation of such meanings is called intensional conceptualization.

In linguistics, meaning is what the sender expresses, communicates or conveys in its message to the receiver (or observer) and what the receiver infers from the current context (Akmajian et al, 1995). The diversity of the contexts in which the same message is inferred may easily lead to different interpretations of the meaning of this message. The pragmatic meaning considers the contexts that affect the meaning and it distinguishes two of their primary forms: linguistic and situational. The linguistic context refers to how meaning is understood, without relying on intent and assumptions. The situational context refers to non-linguistic factors which affect the meaning of the message.

The linguistic context of the meanings depends on the expressivity of the vocabulary used to describe those meanings and a level of abstraction applied in its development. Both factors significantly influence the capability of the receiver to understand the transmitted messages. The expressivity of vocabulary basically refers to the number and diversity of the concepts (and their properties) used to describe one domain of knowledge. The higher levels of expressivity are important for the cases of very specific communication about highly focused issues of the domain. In most cases, it is very likely that the outside listener will not understand the communication between two domain experts.

The level of abstraction has more profound impact. The human reasoning of an unknown term is done by attempting to refer to the known related concepts (or truth conditions). When this is not enough to classify a term, humans reduce or eliminate some truth conditions in attempt to infer a more general, more abstract, known term, which may help in understanding the initial one. Sometimes, even more truth conditions are added so the unknown term is specialized to a known one. Hence, existence of the different levels of abstraction of similar terms or groups of terms may certainly help in understanding the domain knowledge.

Typically, higher level of abstraction used in development of one vocabulary, implies lesser expressivity and vice-versa. However, the advantages of both factors can be combined by developing different vocabularies whose concepts are referenced to each other. Hence, highly abstract, less expressive knowledge may be related to a very specific one. If we consider the above-mentioned communication between two experts on the focused domain issues, it is clear that the references to the known generalizations of the specific terms would certainly help the outside listener to understand this communication.

However, one question still remains - which knowledge to use to make these vocabularies? There are many efforts related to the conceptualization of the enterprise knowledge, including architectures, frameworks and ontologies. Some of the most important work on this topic is shortly presented and referenced in Section 5 of Chapter 2. The main problem of these knowledge models is exactly the lack of balance in layering the levels of abstraction. The crucial reason for this issue is the use of inspirational, top-down approach in their development. In order to address this issue, the development approach proposed in this thesis, is based on using the real-life knowledge about the domain as a starting point, where SCOR model is selected as a natural choice. The implicit knowledge on the supply chain operations, captured by SCOR is described by using the selected formalism. Then, in the induction and synthesis processes, this knowledge is made explicit, as its terms are logically mapped to a corresponding terms of new domain ontology – SCOR-Full. This method overcomes the limitations of the existing top-down approaches to ontology development (presented in Section 2.3 of Chapter 3) and thus, it is a candidate for real industry application. Still, it takes into account the possible advantages of bottom-up approaches because it allows having the concepts of the framework associated to the concepts of some upper ontology. This statement concludes the answer on the second research question.

Given the discussion above and in response to the first research question, mentioned in this Section, following main principles for the development of a formal model which may facilitate a semantic interoperability in a supply chain environment are identified:

- OWL-DL is used as formalism for describing the meanings and contexts of the formal framework for supply chain operations. OWL-DL provides maximum possible expressiveness while retaining computational completeness, decidability and availability of practical reasoning methods.
- The linguistic context of the meanings is provided by the concepts of domain ontologies, formal descriptions of the knowledge domain.
- Linguistic and situational context of the meanings and a meaning itself are described by using assumptions on the intent of the sender. The intent is typically conceptualized by providing the truth conditions for a given concept.
- The central domain ontology is developed as a semantic enrichment of the industry standard model – reference process model.
- Additional contexts are provided by other domain ontologies. Thus, the vocabulary for EISs is extended, as well as the competence of the domain knowledge.
- The situational context of the meanings is provided by the concepts of problem or application ontologies - the formal descriptions of specific problems which are addressed by the semantic applications which use the formal framework.

The principles above are used to develop the formal framework for semantic interoperability in supply chain networks. This framework is presented and described in Section 4 of Chapter 3.

In the architecture for achieving the semantic interoperability of systems in the supply chain, the formal models make explicit the common knowledge of the supply chain network. The backbone of this knowledge is SCOR reference model. Then, SCOR formal model is related to the common knowledge about the enterprises or other perspectives of the supply chain. This knowledge is represented by the domain (or even upper) ontologies. Next, the common knowledge of the supply chain network is contextualized by using problem or application ontologies, which formalize some specific, integrative, shared, commonly used functions of the network in a whole. Finally, individual enterprises are represented in the formal framework by the local ontologies.

Exactly these local ontologies are formal descriptions of the local languages used by the EISs to collaborate each with another. Instead of the exchange of the information between systems, the formal definition of the semantic interoperability of systems, presented in Section 4.3 of Chapter 2 considers that inference of the logical statements, based on the exchanged data is done. Thus, it specialize the general notion of enterprise interoperability to establishment of the logical correspondences between the islands of the enterprises' semantics. Consequently, it demonstrates that the latter implies the former. In other words, enterprise systems may be considered as interoperable (or more specific, semantically interoperable), if their semantic representations are mutually correspondent. These correspondences are facilitated by using the common vocabularies – domain ontologies.

3 Experiences from the implementation of formal framework for semantic interoperability of systems for Supply Chain Management

In Section 1 of Chapter 5, a consumer value of semantic web technologies is discussed from the perspective of the conceptualization stand. The conceptualization stand is considered as one of the most important success factors for achievement of the wider outreach of development of ontologies for semantic interoperability of systems. It is also identified as a possible bottleneck in defining the interoperability infrastructures, because poor conceptualization decisions may easily lead to serious restrictions and dependencies in real-life applications. The methodology for defining the formal framework for semantic interoperability in supply chain networks takes pragmatic approach by combining the consumer-orientation of top-down and efficiency of bottom-up paradigm. More detailed discussion on the main features of this combined approach, in the context of the expected consumer value is provided in Section 1 of Chapter 5.

While above-mentioned Section describes how the expected rate of utilization is affected by the conceptualization stand, now it is time to bring the final arguments for the architectural choices, made during the development of the implementation view of the semantic interoperability framework proposed in this thesis, by using the same criteria. The arguments correspond to the answers to the following research questions, proposed by this thesis:

- Which software services, applications, components and associated assets must be developed in order to become possible to exploit the formal framework for semantic interoperability of the systems in supply chain?
- How they will be configured?
- What is the level of human involvement in the process of making two systems semantically interoperable?

The main architectural choice in implementing the framework is posed by the trend of continuous utilization and commoditization of IT technology. This trend implies that the basic functionalities of IT will be made available to all enterprises comprehensively and non-discriminately, by providing the IT functionalities by using Software-as-a-Service (SaaS) paradigm. SaaS paradigm revolutionized delivery of software, by developing and introducing new business models, such as pay-per-use. Hence, it enabled a wide range of choices in the way one enterprise is leveraging specific computing asset. It has been even used to commoditize interoperability, by Interoperability Service Utilities (ISU), which are: 1) available at low cost; 2) accessible by all; 3) guaranteed to a certain level; 4) not controlled or owned by a single entity. ISU architecture and selected applications and approaches in its implementation are described in detail in Section 2 of Chapter 3.

Unfortunately, the review of the current results in implementation of ISU architecture, presented in Section 3 of Chapter 4 shows that its main principles and requirements are not yet satisfied. First, the fact that ISU exploits services is intentional restriction of the overall architecture. Namely, certain level of diversity of enterprise services is a precondition for setup of interoperability services. Most likely, interoperability levels will strongly depend on their variety. Hence, a functional, vertical approach to interoperability of only specific business functions commoditized by the corresponding services will be taken. Second, related to above, enterprise services must be semantically annotated in order to improve more efficient work of ISU infrastructure, e.g. improved matching between supply and demand of services. Third, although there are some attempts to define common interoperability services (Elvesæter et al, 2008), those efforts still did not produce some tangible results.

Considering the above findings, an architectural view for the semantic interoperability of systems in supply chain environment is based on the following principles:

- The proposed architecture takes into account the restrictions of the functional approach and it assumes that enterprises should take their own decision (based on their interests, needs and requirements) on which part of their ontologies should be made interoperable;
- This semantics is described by the local ontologies. The core unit of the interoperability in networked enterprise environment is local ontology, not a service. Main objective of the framework for semantic interoperability of systems is to make those ontologies interoperable;
- The local ontologies are explicit formal representations of the semantics of the EISs, where ER schemas of their databases are assumed as a starting point in the explicitation process;
- Minimum technical pre-requirements are foreseen for each enterprise which wants to take part in the interoperable world of the Virtual Breeding Environment;
- The formal framework is not associated with some storage facility; the formal framework facilitates delivery of the information by combining their sources (namely, local ontologies). Only meta-information (other than a formal framework - common ontologies) about the interoperable systems is kept centrally;

In the process of implementation of the above principles, and in response to first research question in this Section of the thesis conclusion, five main Semantic Interoperability Service Utilities (S-ISU) are identified and analyzed in Section 3.1 of Chapter 4:

1. Semantic Reconciliation Service for automatic or semi-automatic identification of the logical correspondences between two domain and/or local ontologies;
2. Registration Service for declaration of the local or domain ontology (or ontologies) location and rules (e.g. access rights) for semantic queries handling;
3. Transformation Service for explicitation of the implicit semantics of the ER schemas, and for facilitating full correspondence between semantic and database queries;
4. Semantic Query Service for extraction of relevant instances from the designated local ontologies;
5. Reasoning Service for accessing DL reasoner functionality.

In response to the second research question, namely, in order to elaborate on how the services are configured, S-ISU architecture is analyzed from the component and organizational perspective, by using a meta-model – S-ISU ontology. Finally, two main services of S-ISU architecture – Transformation Service and Semantic Query Service are developed and implemented in the provided architectural context. Their inner workings are presented in Sections 3.3 and 3.4, respectively, of Chapter 4.

Transformation Service implements principles of proposed database-to-ontology mapping process. It clearly outperforms the existing work in this area, since the current efforts do not interpret the semantics of all ER constructs and patterns or use the full expressivity of the OWL language. This is illustrated in the comparative analysis in Table 1 of Section 3.3 of Chapter 4. Still, it is strongly emphasized that local ontologies which are generated by the Transformation Service should be considered only as intermediary results of the process of conceptualization of one EISs. The main argument for needed human intervention is that weak assumption is made that ER schema of the EISs represent the semantics of their data models. There are obvious limitations introduced by this assumption, related to semantics coverage and even correctness (because it is more correct to say that ER schemas are conceptual models of the developers' intents rather than actual systems' databases). However, the case study of generating local ontology from the OpenERP system (Section 4.2 of Chapter 5) shows that Transformation Service provide the exhaustive semantics landscape by fully interpreting semantics of ER underlying schema, by using full OWL/DL expressivity, automatically. As such, this landscape can be improved in the following human intervention which may consider business rules, ambiguous types (see Section 1 of this Chapter) and more sophisticated semantic relations.

In the process of Semantic Query Service development, described in Section 3.4 of Chapter 4, so-called instance population approach is taken. It assumes that the local ontology is only considered as meta-ontology; it does not store database data but only the semantics of their schema. Hence, database information is interpreted as local ontology instances at the runtime of semantic query execution. A query rewriting method is implemented to transform semantic DL query to a set of SQL queries which extract the relevant data from the database. Based on the semantic relations between S-ER and ER meta-models, the result-sets are then converted to local ontology instances and returned as the outcome of the semantic query. The resulting graph may be asserted to a local ontology for the further processing (e.g. inference) or it may just be interpreted for the given purpose, as a memory model. The selection of instance population approach is made for two reasons. First, the alternative massive dump approach would certainly introduce performance problems, related to the size of the resulting local ontologies, populated with instances which represent database information. Second, massive dump approach would introduce privacy problems, because of the full exposition of the data of one EIS. In this case, the VE actor would not be capable to restrict access to the parts of their semantic models, as it is envisaged by the S-ISU principles.

Once configured, the proposed architecture is expected to facilitate the semantic interoperability of systems. The main condition for the semantic interoperability is to increase the amount of the explicit semantics. Enterprise knowledge discovery, transformation of implicit to explicit semantics and reconciliation of different explicit models are related to extremely difficult manual work, due to diversity, complexity and size of enterprise data. Thus, some level of automation in these processes is involved, as it is shown in this thesis. However, it must be noted that the quality of the results of the automatic tools will vary a lot and will be directly related to the expressiveness of the implicit data which is used as an input. For example, all the output results of the transformation services must be considered as intermediary, as they would still need a human intervention in semantic refinement and enactment.

In Chapter 5, some evidences on the feasibility and usability of the proposed framework and implementation method are given. Two case studies of using the proposed approach are presented. First case study shows how the approach can be exploited to support one of the common functions of the Virtual Breeding Environment – a setup of supply chain processes (Section 3 of Chapter 5). It is estimated that by resolving the interoperability issues, the proposed infrastructure should reduce the lifecycle of the VE for the manufacturing of custom orthopaedic implant to 4-8 days, for the implants of complexity similar to the one described in the case. This is considered as acceptable period for many cases of trauma. In the traditional settings of the enterprise collaboration, due to more human decisions and lack of interoperability, custom orthopaedic implant manufacturing may need even up to three months of lead time (Christensen and Chen, 2008). The estimation is based on the fact that integrated semantic framework practically automates the process configuration phase of VE lifecycle and exchange of information between relevant systems. Thus, it significantly reduces the time typically needed for supply chain planning. In contrast to traditional supply chains and volume manufacturing, the planning for one-of-a-kind manufacturing in VBE is not based on the forecasts. It depends on the timely access to information about available capacities, raw materials and other assets.

The second case study illustrates how the S-ISU architecture is used to get the relevant information from the local ontologies with a single query, facilitating the collaborative production planning in Virtual Breeding Environment (Section 4 of Chapter 5). While the first case study represents model verification, the second one demonstrates the practical usability of the transformation and semantic querying services of S-ISU.

4 Future research directions

In the final Section of this thesis, some of the specific future research directions are listed and arguments for their selection are summarized. The scope for their selection is related to the achievement of the following objective: “The semantic interoperability framework for supply chain operations is operational infrastructure which can be implemented in the industry settings”. Hence, the list of future research directions does not represent a result of the critical analysis of the relevant state-of-the-art (see Section 1 of this Chapter) or a general discussion, but only a result of informal gap analysis, derived from the experiences in performing this research.

In the following list, these gaps are represented as topics for the future research work which could significantly improve the potential of the semantic interoperability framework, described in this thesis. The topics are classified, as follows:

1. General Semantic interoperability

- Implementing method for evaluating semantic interoperability of two systems;
- Further development of theoretical background for semantic interoperability, by following the principles of human communication;

2. Formal model for supply chain operations

- Further explication of the SCOR-Full domain model by mapping with relevant and/or complementary domain models, such as RosettaNet , UNSPSC , AIAG and STAR , EDI , etc;
- Development of new application models and ontologies which directly exploits SCOR-Full domain model;
- Top-down validation of SCOR-Full domain model by semantic analysis of the logical correspondences with relevant upper ontologies, such as DOLCE;

3. S-ISU Transformation and Semantic Querying Service

- Analysis of data patterns with goal to discover the semantics of the ambiguous notions of the local ontologies (e.g. type or status);
- Semi-automatic classification of the concepts of local ontologies by analysis of necessary conditions for different concepts;
- Developing universal method for semantic query rewriting, where source and destination queries are using the concepts of two ontologies, logically interrelated by using SWRL rules;
- Developing method and tools for execution of "Tell" semantic queries;

4. General Semantic web tools

- Implementing distributed reasoning capabilities for modular ontologies with dynamic imports;
- Implementing security and access control levels to the parts of ontologies in distributed ontological frameworks;
- Advance in performance and quality of ontology matching tools.

The proposed topics will contribute to establishment of semantic interoperability as a scientific discipline and thus, to increased attention of the scientific community (Topic 1). The improved relevance of SCOR-Full micro-theory (Topic 2) is expected to contribute to its standardization and hence, to its increased usage. It will also motivate further research of SCOR reference model and resolution of some of its issues (such as missing dependencies, conceptual inconsistencies, etc.). Finally, improvement of S-ISU services and Semantic Web infrastructure (Topics 3 and 4) refers to some development challenges and validation of the proposed conceptual directions which will produce significant benefits for the practical implementation of the approach proposed in this thesis.

References

Akmajian, A., Demers, R., Farmer, A., Harnish, R. (1995). Linguistics: an introduction to language and communication, 4th edition. 1995. Cambridge: MIT Press

AMICE (1993). CIMOSA - Open System Architecture for CIM, 2nd edition, Springer-Verlag, Berlin, 1993

Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawana, S. (2003). Business Process Execution Language for Web Services, Version 1.1

Angeles, P.A. (1981). Dictionary of Philosophy. Harper Perennial. New York, 1981.

Aničić, N., Ivezić, N., Jones, A. (2005). An architecture for semantic enterprise applica-tion integration standards. In: Proceedings of The First International Conference on the Interoperability of Enterprise Software and Applications (I-ESA'05)

Ashton, K. (2009). That 'Internet of Things' Thing. RFID Journal. 22 July 2009

Astrova, I., 2004. Reverse Engineering of Relational Databases to Ontologies. In Lecture Notes in Computer Science, LNCS 3053 (2004) 327-341

Bateman, J.A. (1995). On the relationship between ontology construction and natural language: a socio-semiotic view. International Journal of Human-Computer Studies. 43 (5–6) (1995) 929–944

Batini, C., Lenzerini, M., Navathe, S.B. (1986). A comparative analysis of methodologies for data-base schema integration. ACM Computing Surveys. 18 (4) (1986)

Bechhofer, S., Moller, R., Crowther, P. (2003). The DIG description logic interface. In: Proceedings of the International Description Logics Workshop (DL 2003)

Bechhofer, S., Patel-Schneider, P.F. (2006). DIG 2.0: The DIG Description Logic Interface Overview, http://dig.cs.manchester.ac.uk/overview.html

Bernus, P., Nemes, L., Schmidt, G. (Eds.) (2003). Handbook on Enterprise Architecture. Springer-Verlag, 2003

Berre,A.J., Elvesæter,B., Figay,N., Guglielmina, C.J., S. G., Karlsen, S.G. (2007). The ATHENA Interoperability Framework. In: Proceedings of the 3rd International Conference on Interopera-bility for Enterprise Software and Applications (I-ESA'07). Springer (2007) 569-580

Berre, A.J., Hahn, A., Akehurst, D., Bezivin, J., Tsalgatidou, A., Vernaut, F., Kutvonen, L., Linington, P.F. (2004). State-of-the art for Interoperability architecture approaches. Interop Network of Excellence - Contract no.: IST-508 011, Deliverable D9.1

Berry, A.J. (1994). Spanning traditional boundaries: organization and control of embedded operations. Leadership & Organization Development Journal. 15 (7) 4-10

Bolstorff, P., Rosenbaum, R. (2003). Supply Chain Excellence: A Handbook for Dramatic Improvement Using the SCOR Model. AMACOM

Bonacina, M.P. (2000). A taxonomy of parallel strategies for deduction. Annals of Mathematics and Artificial Intelligence. 29(1-4) (2000) 223–257

Browne, J. and Zhang, J. (1999). Extended and Virtual Enterprises – similarities and differences. International Journal of Agile Management Systems. 1(1) 30-36

Burger, A. Davidson, D. Baldock, r. (Eds.) (2008). Anatomy Ontologies for Bioinformatics. Principles and Practice. Series: Computational Biology, Vol. 6. 1st Edition., 2008, XVI, 356 p. 73

Camarinha-Matos, L. M., Afsarmanesh, H., Galeano, N., Molina, A. (2009). Collaborative networked organizations – Concepts and practice in manufacturing enterprises. Computers & Industrial Engineering, 57 (1) 46-60

Camarinha-Matos, L. M., & Afsarmanesh, H. (2006). Collaborative networks: Value creation in a knowledge society. In: Proceedings of PROLAMAT, 2006 IFIP international conference on knowledge enterprise – New challenges. Shanghai, China, Boston: Springer.

Camarinha-Matos, L. M., & Afsarmanesh, H. (2008). Collaborative networks – Reference modeling. New York: Springer

Camarinha-Matos, L.M., Afsarmanesh, H. (2008). On reference models for collaborative networked organizations. International Journal Production Research. 46 (9) (2008) 2453–2469

Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K. (2004). Jena: Implementing the semantic web recommendations. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004)

Castano, S., De Antonellis, V. (1998). A framework for expressing semantic relationships between multiple information systems for cooperation. Information Systems. 23 (3-4) (1998) 253-277

Chen, D., Doumeingts, G. (1996). The GRAI-GIM reference model, architecture and methodology. In: P. Bernus, et al. (Eds.), Architectures for Enterprise Integration, Chapman & Hall, London, 1996

Chen, D., Daclin, N. (2006). Framework for Enterprise Interoperability. In: Proceedings of IFAC TC5.3 Workshop EI2N06, Bordeaux, France, March 21, 2006

Chen, D., Doumeingts, G., Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry. 59 (2008) 647–659

Chen, D. Vernadat, F. (2004). Standards on enterprise integration and engineering - a state of the art. International Journal of Computer Integrated Manufacturing. 17 (3) (2004) 235–253

Childe, S.J. (1998). The extended enterprise – a concept of co-operation, Production Planning and Control. 9 (4) 320-327

Choi, Y., Kang, D., Chae, H., Kim, K. (2006). An enterprise architecture framework for collaboration of Virtual Enterprise chains. International Journal of Advanced Manufacturing Technology

Christensen, R., Curbera, F., Meredith, G., Weerawana, S. (2001) Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, http://www.w3.org/TR/wsdl

Christensen, A., Chen, D. (2008). Personalizing Orthopedic Implants. MDDI Online. http://www.mddionline.com/article/personalizing-orthopedic-implants

C4ISR (1998), Architecture Working Group (AWG), Levels of Information Systems Interoperability (LISI), March 30, 1998

COMPTIA (2004). European Interoperability Framework, White Paper, Brussels, February 18, 2004

CROSI (2005). Semantic Integration Technology Survey, http://eprints.ecs.soton.ac.uk/10842/1/crosi-survey.pdf

Cullot, N., Ghawi, R. and Yetongnon, K. (2007). DB2OWL: A Tool for Automatic Database-to-Ontology mapping. In: Proceedings of the 15th Italian Symposium on Advanced Database Systems. June, 17-20. 2007, Torre Canne di Fasano (BR), Italy.

De Laborda, C.P. and Conrad, S. (2005). Relational.OWL: a data and schema representation format based on OWL. In: Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling. January 30 - February 4, 2005. Newcastle, Australia.

Desai, N., Mallya, A.U., Chopra, A., Singh, M. (2006). OWL-P: A Methodology for Business Process Development. Lecture Notes in Computer Science. 3529 (2006) 79-94

Diez, A.B.G. (2004). First Version of State of the Art in Enterprise Modelling Techniques and Technologies to Support Enterprise Interoperability. Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application (ATHENA) - Contract no.: 507849, Deliverable D.A1.1.1

Doan, A., Halevy, A.Y. (2005). Semantic Integration Research in the Database Community: A Brief Survey. AI Magazine. 26 (83) (2005)

Dogac, A., Jardim-Gonçalves, R. (2009). iSurf: RFID enabled collaborative supply chain planning environment. In: Proceedings of IFIP TC9 - 4th MEDITERRANEAN CONFERENCE ON INFORMATION SYSTEMS

Dogac, A., Laleci, G.B., Okcan, A., Olduz, M., Sesana, M., Canepa, A. (2008). iSURF: An Interoperability Service Utility for Collaborative supply chain Planning across Multiple Domains, Proc. of the eChallanges 2008 Conference

Elvesæter, B., Taglino, F., Del Grosso, E., Benguria, G., Capellini, A. (2008). Towards Enterprise Interoperability Service Utilities. In: Proceedings of the International Workshop on Enterprise Interoperability (IWEI 2008), 15-19 September 2008, Munich, Germany

Engelbart, D.C. (1962). Augmenting human intellect: a conceptual framework. In: Menlo Park, CA: Stanford Research Institute.

Ehrig, M., Staab, S. (2004). QOM: Quick ontology mapping. In: Proceedings of the International Semantic Web Conference (ISWC), 683–697

Euzenat, J. (2004). An API for ontology alignment. In: Proceedings of the International Semantic Web Conference (ISWC), 698–712

Falkenhainer, B., Forbus, K.D. (1991). Compositional modeling: finding the right model for the job. Artificial Intelligence. 51 (1-3) 95-143

Fayez, M., Rabelo, L. and Mollaghasemi, M. (2005). Ontologies for supply chain simulation modelling. In: Proceedings of the 37th Winter Simulation Conference. Orlando, USA, December 04-07, 2005

Felix T.S. Chan, H.K. (2005). The future trend on system-wide modelling in supply chain studies. International Journal of Advanced Manufacturing Technology. 25 (7-8) (2005) 820-832

Forrester, J.W. (1961). Industrial Dynamics, The M.I.T. Press, Cambridge, MA, 1961

Fox, M.S., Barbuceanu, M. and Gruninger, M. (1996). An organization ontology for enterprise modelling: preliminary concepts for linking structure and behaviour. Computers in Industry. 29 (1-2) 123–134

Foxvog, D., Bussler, C. (2006). Ontologizing EDI semantics. In: Proceedings of the Workshop on Ontologising Industrial Standards. pp. 301-311. Springer, Tucson, AZ, USA, 2006

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A. and Schneider, L. (2002). Sweetening Ontologies with DOLCE. Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, Lecture Notes in Computer Science, 2473 (2002) 223-233

Gangemi, A. (2005). Ontology Design Patterns for Semantic Web Content. Lecture Notes in Computer Science. 3729 (2005) 262-276

Genesereth, M.R., Nilsson, N.J. (1987). Logical Foundation of Artificial Intelligence, Morgan Kauffman, Los Altos, CA, 1987

GERAM (1999). IFAC–IFIP Task Force, GERAM: Generalized Enterprise Reference Architecture and Methodology, Version 1.6.3, IFAC–IFIP Task Force on Architecture for Enterprise Integration, 1999. http://www.ict.griffith.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/GERAMv1.6.3.pdf

Giunchiglia, F., Shvaiko, P., Yatskevich, M. (2004). S-Match: an algorithm and an implementation of semantic matching. In: Proceedings of the European Semantic Web Symposium (ESWS), 61–75

Gibson,I., Harrysson,O.L.A. (2006). Direct Fabrication of Custom Orthopedic Implants Using Electron Beam Melting Technology. In: Advanced Manufacturing Technology for Medical Applications: Reverse Engineering, Software Conversion and Rapid Prototyping. Wiley

Ghawi, R. and Cullot, N. (2007). Database-to-Ontology Mapping Generation for Semantic Interoperability. In: Proceedings of 3rd International Workshop on Database Interoperability. September, 24. 2007, Vienna, Austria.

Goldman, S.L., Nagel, R.N., Preiss, K. (1995). Agile Competitors and Virtual Organizations – Strategies for Enriching the Customer, Van Nostrand Reinhold, New York

Grefen, P., Mehandjiev, N., Kouvas, G., Weichhart, G., Eshuis, R. (2009). Dynamic business network process management in instant virtual enterprises. Computers in Industry. 60 (2) (2009)

Grenon, P., Smith, B. (2004). SNAP and SPAN: Towards Dynamic Spatial Ontology. Spatial Cognition and Computation 4 (1) 69-103

Grabot, B. (2008). Process Alignment or ERP Customisation: Is There a Unique Answer?, Springer Series in Advanced Manufacturing, ERP Systems and Organisational Change, Springer, London, (2008) 139-156

Grubic, T., Fan, I.-S. (2010). Supply chain ontology: Review, analysis and synthesis. Computers in Industry. 61 (8) 776-786

Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. International Journal of Human-Computer Studies. 43 (5–6) (1995) 625–640

Guarino, N. (1997). Understanding, building and using ontologies. International Journal of Human-Computer Interaction. 46 (2-3) (1997) 293-310

Guarino, N. (1998). Formal Ontology in Information Systems. IOS Press

Guarino, N., and Giaretta,P. (1995). Ontologies and knowledge bases – towards a terminological clarification. In: Mars, N., ed. Towards very large knowledge bases: knowledge building and knowledge sharing. Amsterdam: IOS Press, 25-32.

Guo, J. (2009). Collaborative conceptualisation: towards a conceptual foundation of interoperable electronic product catalogue system design. Enterprise Information Systems. 3 (1) (2009) 59–94

Haller, A., Gontarczyk, J., Kotinurmi, P. (2007). Towards a complete SCM Ontology – The Case of ontologising RosettaNet. In: Proceedings of the 2008 ACM Symposium on Applied Computing (SAC08), March 16-20, 2007, Fortaleza, Ceara, Brazil

Hamel, G. (1999). Bringing Silicon Valey Inside. Harvard Business Review, September – October, pp.71-84

Harrysson.O.L., Hosni,Y.A., Nayfeh,J.F. (2007). Custom-designed orthopedic implants evaluated using finite element analysis of patient-specific computed tomography data: femoral-component case study. BMC Musculoskeletal Disorders. 8 (91) (2007)

Hepp, M. (2007). Ontologies: State of the art, business potential and grand challenges. In Hepp, M., De Leenheer, P., de Moor, A. and Sure, Y. (eds), Ontology Management – Semantic Web, Semantic Web Services and Business Applications (pp.3-22), Berlin/Heidelberg: Springer.

Hepp, M. (2006). Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. International Journal on Semantic Web and Information Systems. 2 (1) (2006) 72-99

Hodge, G. (2000). Systems of Knowledge Organization for Digital libraries - Beyond traditional authority files, Washington, DC: the Council on Library and Information Resources, http://www.clir.org/pubs/reports/pub91/contents.html

Holsapple, C.W., Joshi, K.D. (2002). A collaborative approach to ontology design. Communications of the ACM. 45 (2) (2002) 42-47

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M. (2004). SWRL: A Semantic Web Rule Language - Combining OWL and RuleML. W3C Member Submission 21 May 2004. http://http://www.w3.org/Submission/SWRL/

Huber, C., Plüss, A. (2003). Vertrauenskultur – Erfahrungen aus dem Produktionsnwtzwerk Virtuelle Fabrik Nordwestschweiz/Mittelland. In Vernetzt planen und produzieren, TU Chamnitz, Sonderforschungsbereich 457, Tagundgsband, pp. 143–147

IDEAS (2002) Thematic Network, IDEAS: Interoperability Development for Enterprise Application and Software—Roadmaps, Annex 1—DoW, May 13, 2002

IEEE. (1990). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. Institute of Electrical and Electronics Engineers. 1990

Izza, S. (2009). Integration of industrial information systems: from syntactic to semantic integration approaches. Enterprise Information Systems 3 (1) (2009) 1-57

Jagdev, H.S., Thoben, K.D. (2001). Anatomy of enterprise collaborations, Production Planning and Control. 12 (5) 437-451

Jespersen, B.D., Larse, T.S. (2006). Supply Chain Management - in Theory and Practice. Copenhagen:Business School Press.

Jiang, G., Cybenko, G., Hendler, J.A. (2006). Semantic interoperability and information fluidity. International Journal of Cooperative Information Systems. 15 (1) (2006) 1-21

Jones, A., Ivezic, N. and Gruninger, M. (2001). Toward self-integrating software applications for Suppy Chain Management. Information Systems Frontiers. 3 (4) 403-412

Kabak, Y., Dogac, A., Ocalan, C., Cimen, S., Laleci, G. B. (2009). iSURF Semantic Interoperability Service Utility for Collaborative Planning, Forecasting and Replenishment. In: Proceedings of the eChallenges Conference, October 2009, Istanbul, Turkey.

Kalfoglou, Y., Schorlemmer, M. (2003). Ontology mapping: the state of the art

Katzy, B.R., Dissel, M. (2001). A toolset for building the Virtual Enterprise. Journal of Intelligent Manufacturing. 12 (2) 121-131

Katzy, B.R., Loeh, H., Zhang, C. (2004). Virtual Organising Scenarios. Collaborative Networked Organizations, Springer US, pp.27-40

Katzy, B., Ma, X. (2002). A research note on virtual project management systems. In: K. Pawar, F. Weber, & K. Thoben (Eds.) Proceedings of the 8th international conference on concurrent enterprising ICE (pp. 517–520). Rome: University of Nottingham

Keenan, J., Chakrabarty, G., Newman, J.H. (2000). Treatment of supracondylar femoral frac-ture above total knee replacement by custom made hinged prosthesis. Knee 2000, 7 (2000) 165-170

Klein, M., Ding, Y., Fensel, D., Omelayenko, B. (2002). Ontology management - Storing, aligning and maintaining ontologies

Klein, M. (2001). Combining and relating ontologies: an analysis of problems and solutions. In: Proceedings of IJCAI2001 Workshop on Ontologies and Information Sharing

Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. The Semantic Web – ISWC 2004 In The Semantic Web – ISWC 2004, Vol. 3298 (2004), pp. 229-243

Konstantinou, N., Spanos, D., Chalas, M., Solidakis, E. and Mitrou, N. (2006). VisAVis: An approach to an intermediate layer between ontologies and relational database contents. In:

Proceedings of International Workshop on Web Information Systems Modeling. June, 6, 2006, Luxembourg.

Lamber, D.M., Knemeyer, A.M. (2004). We're in This Together. Harvard Business Review on Supply Chain Management

LaOngsri, S. (2009). Semantic Extensions and a Novel Approach to Conceptual Modelling, Ph.D. Thesis. School of Computer Science, Engineering and Mathematics, The Flinders University of South Australia

Lee, J.Y., Kim, K. (2007). A distributed product development architecture for engineering collaborations across ubiquitous Virtual Enterprises. International Journal of Advanced Manufacturing Technology. 33 (1-2) 59–70

Lee, J.L., Madnick, S.E., Siegel, M.D. (1996). Conceptualizing semantic interoperability: A perspective from the knowledge level. International Journal of Cooperative Information Systems. 5 (4) (1996) 367-393

Li, M.S., Cabral,R., Doumeingts,G., Popplewell, K. (eds) (2006). Enterprise Interoperability Research Roadmap, 31 July 2006, http://cordis.europa.eu/ist/ict-ent-net/ei-roadmap_en.htm

Lima, C., Vallejos, R. V., Varvakis, G. (2004). Applying knowledge management to support a virtual organization of mould and die makers. In: L. M. Camarinha-Matos & H. Afsarmanesh (Eds.), Processes and foundations for virtual organizations (pp. 351–358). Boston: Kluwer.

Lin, H.K., Harding, J.A. (2007). A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. Computers in Industry. 58 (5) 428-437

Lin, Y. (2008). Semantic Annotation for Process Models. PhD thesis, Norwegian University of Science and Technology

Lu, Y., Panetto, H., Gu, X. (2010). Ontology Approach for the Interoperability of Networked Enterprises in supply chain Environment. On the Move to Meaningful Internet Systems: OTM 2010 Workshops (Meersman R., et al.) (Eds)), 5th IFAC/IFIP Workshop on Enterprise Integration, Interoperability and Networking (EI2N'2010), Hernissou, Crete, Greece, October 27-28, Springer Verlag, Lecture Notes in Computer Science, LNCS 6428, 229-238

MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., Metz, R. (eds) (2006). OASIS Reference Model for Service Oriented Architecture 1.0. Public Review Draft 2, 31 May 2006

Madni, A.M., Lin, W., Madni, C.C. (2001). IDEON^{TM}: An Extensible Ontology for Designing, Integrating and Managing Collaborative Distributed Enterprises. Systems Engineering. 4 (1) 35-48

Maedche, A., Motik, B., Silva, N., Volz, R. (2002). MAFRA – A Mapping FRAmework for distributed ontologies. Proceedings of EKAW 2002.

McBrien, P., Poulovassilis, A. (1998). A Formalisation of Semantic Schema Integration. Information Systems. 23 (5) (1998) 307-334

Miller, R.J., Ioannidis, Y.E., Ramakrishnan, R. (1994). Schema equivalence in heterogeneous systems: Bridging theory and practice. Information Systems. 19 (1) (1994) 3-31

Millet, P.A., Schmitt, P. and Botta-Genoulaz, V. (2009). The SCOR model for the alignment of business processes and information systems. Enterprise Information Systems. 3 (4) (2009) 393-407

Miles, R.E., Snow, C.C. (1984). Fit, failure and hall of fame, California Management Review. 26 (3) 10-28

Motwani, J., Subramanian, R., Gopalakrishna, P. (2005). Critical factors for successful ERP implementation: exploratory findings from four case studies. Computers in Industry, 56 (6) (2005) 529-544

NEHTA (2005). Towards an Interoperability Framework, Version 1.8, August 21, 2005

Noy N.F. (2004). Semantic integration: a survey of ontology-based approaches. ACM SIGMOD Record Special section on semantic integration. 33 (4) 65-70

Noy N., Klein, M. (2002). Ontology evolution: Not the same as schema evolution. Knowledge and Information Systems, 2002.

Obitko, M. (advisor Vladimir Marik) (2007). Translations between Ontologies in Multi-Agent Systems. Ph.D. dissertation, Faculty of Electrical Engineering, Czech, Technical University in Prague, 2007

Obrst, L. (2003). Ontologies for semantically interoperable systems. In: Proceedings of the 12th International Conference on Information and Knowledge Management. November 3-8, 2003, New Orleans, USA

Oldakowski, R., Bizer, C., Westphal, D. (2005). RAP: RDF API for PHP. Workshop on Scripting for the Semantic Web (SFSW 2005) at 2nd European Semantic Web Conference (ESWC 2005), Heraklion, Greece, May 2005

Panetto, H., Molina, A. (2008). Enterprise integration and interoperability in manufacturing systems: Trends and issues. Computers in Industry. 59 (7) (2008) 641–646

Pathak, S.D., Nordstrom, G. and Kurokawa, S. (2000). Modelling of supply chain: a multi-agent approach. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. 3 2051 -2056

Perrin, O., Wynen, F., Bitcheva, J., Godart, C. (2003). A Model to Support Collaborative Work in Virtual Enterprises, Lecture Notes in Computer Science, Springer Berlin / Heidelberg. 2678 (2003) 104-119

Peñaranda, N., Galeano, N., Romero, D., Mejía, R., Molina, A. (2006). Process improvement in a virtual organization focused on product devel-opment using collaborative environments. In Proceedings of IFAC symposium on information control problems, in manufacturing (INCOM'06). Mat 17–19th, 2006, Saint-Etienne, France

Phelps, T. (2006). SCOR and Benefits of Using Process Reference Models. In: Proceedings of Supply Chain International Conference, Taiwan, 2006

Prud'hommeaux,E., Seaborne,A. (Eds) (2008). SPARQL Query Language for RDF, W3C Recom-mendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/

Rahm, E., Bernstein, P.A. (2001). A survey of approaches to automatic schema matching, VLDB Journal. 10 (2001) 334–350

Sánchez, N.G., Apolinar, D., Zubiaga, G., Atahualpa, J., González. I. and Molina, A. (2005). Virtual Breeding Environment: A First Approach to Understanding Working and Sharing Principles. In: Proceedings of the 1st International Conference on interoperability of Enterprise Software and Applications. February 23-25, 2005, Geneva, Switzerland

Sari, B., Se, T., Kilic, E.S. (2006). Formation of dynamic Virtual Enterprises and enterprise networks, International Journal of Advanced Manufacturing Technology. 34 (11-12) (2006) 1246-1262

Scheer, A.-W. (1994). Business Process Engineering. Reference Models for Industrial Enterprises, 2nd edition, Springer-Verlag, Berlin, 1994

Schlicht, A., Stuckenschmidt, H. (2009). Distributed Resolution for Expressive Ontology Networks, In Proceedings of the 3rd International Conference on Web Reasoning and Rule Systems, Springer-Verlag Berlin, Heidelberg 2009

Schulz, S., Hahn, U. (2001). Parts, Locations, and Holes - Formal Reasoning about Anatomical Structures, Artificial intelligence in medicine, Lecture Notes in Computer Science, 2001, 2101 (2001) 293-303

Sheth, A., Larson J. (1990). Federated Database Systems. ACM Computing Surveys. 22 (3) (1990) 183-236

Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web. Software Engineering and the Semantic Web. 5 (2) (2007) 51-53

Smith, B. (2003). Ontology and Information Systems. In: Zalta, E.N., ed. The Stanford Encyclopaedia of Philosophy. Stanford, CA

Sowa, J. (2000). Knowledge Representation : Logical, Philosophical, and Computational Foundations, CA:Brooks/Cole Publishing Co.

Sprott, D. (2000) Componentizing the enterprise application packages, Communications of the ACM. 43 (2000) 63–69

Stewart, G. (1997). Supply-chain operations reference model (SCOR): the first cross-industry framework for integrated supply-chain management. Logistics Information Management. 10 (2) 62–67

Themistocleous, M., Irani, Z., O'Keefe, R.M. (2001). ERP and application integration: exploratory survey. Business Process Management Journal. 7 (2001) 195–204

Tursi, A., Panetto, H., Morel, G., Dassisti M. (2009). Ontological approach for Products-Centric Information System Interoperability in Networked Manufacturing Enterprises. IFAC Annual Reviews in Control. 33 (2) (2009) 238-245

Truong, H.L., Morten, E., Laclavik, M., Burkhart, T., Carpenter, M., Dorn, C., Gkouvas, P., Konstantinos, K., Lopez, D.L., Marín, C.A., Melchiorre, C., Pinuela, A., Seleng, M., Werth. D. (2009). Email-based Interoperability Service Utilities for Cooperative Small and Medium Enter-prises. In: Cordeiro,J., Filipe.J. (Eds.): ICEIS 2009 - Proceedings of the 11th International Conference on Enterprise Information Systems, Volume SAIC, Milan, Italy, May 6-10, 2009. 95-101

Uschold, M., King, M., Moralee, S. and Zorgios, Y. (1998). The enterprise ontology. Knowledge Engineering Review. 13 (1) 31-89

Vanderhaeghen, D., Loos, P. (2007). Distributed model management platform for cross-enterprise business process management in Virtual Enterprise networks. Journal of Intelligent Manufacturing. 18 (5) 553-559

Vegetti, M., Gonnet, S., Henning, G., Leone, H. (2005). Towards a supply chain ontology of information logistics within process industry environments. In: Proceedings of The Fourth Mercosur Congress on Process Systems Engineering, August, 14-18, 2005, Rio de Janeiro, Brazil

Vernadat, F.B. (2002). Enterprise Modeling and Integration (EMI): Current Status and Research Perspectives. Annual Reviews in Control. 26 (2002) 15-25

Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I.S., Mazura, M., Harrison, M., Eisenhauer, M., Doody, P. (2009). Internet of Things Strategic Research Roadmap. Internet of Things Initiative

Visser, P. R. S., Jones, D. M., Bench-Capon, T. J. M., Shave, M. J. R. (1997). An analysis of ontological mismatches: Heterogeneity versus interoperability. In: Proceedings of AAAI 1997 Spring Symposium on Ontological Engineering, Stanford, USA, 1997

Vitković, N., Trajanović, M., Milovanović, J., Korunović, N., Arsić, S., Ilić, D. (2011). The geometrical models of the human femur and its usage in application for preoperative planning in orthopedics. In: Proceedings of the 1st International Conference on Internet Society Technology and Management - ICIST (pp. 13) Kopaonik

Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hubner, S. (2001). Ontology-Based Integration of Information – Survey of Existing Approaches, In: Proceedings of IJCAI-01, 2001

W3C OWL Working Group, (2009). OWL 2 Web Ontology Language Document Overview. W3C Recommendation 27 October 2009

William, W.S., Johannesson, P., Bubenko Jr. J.A. (1996) Semantic similarity relations and computation in schema integration, Data Knowl Eng. 19 (1) (1996) 65-97

Williams, T.J. (1994). The Purdue enterprise reference architecture. Computers in Industry 24 (2/3) (1994) 141–158

Williamson, O. (1985). The Economic Institutions of Capitalism: Firms, Markets, Relational Contracting. New York, The Free Press

Xu, Z., Zhang, S. and Dong, Y. (2006). Mapping between Relational Database Schema and OWL Ontology for Deep Annotation. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. 18-22. December, 2006, Hong Kong.

Xu, Z., Cao, X., Dong, Y. and Su, W. (2004) Formal Approach and Automated Tool for Translating ER Schemata into OWL Ontologies. Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science. 3056 (2004) 464-475.

Yan, T. (2007). Practical Issues in Ontology Engineering. In: Arabnia, H.R., Yang, M.Q. Yang, J.Y, eds. Proceedings of the 2007 International Conference on Artificial Intelligence, ICAI 2007, Volume II, June 25-28, 2007, Las Vegas, Nevada, USA. pages 730-736, CSREA Press, 2007

Yarimagan, Y., Dogac, A. (2009). A Semantic-Based Solution for UBL Schema Interoperability. Journal IEEE Internet Computing. 13 (3) (2009)

Ye, Y., Yang, D., Jiang, Z. and Tong, L. (2008). Ontology-based semantic models for Suppy Chain Management. International Journal of Advanced Manufacturing Technology. 37 (1112) 1250-1260

Zachman, J. (1996). The Framework for Enterprise Architecture: Background, Description and Utility. Zachman Institute for Advancement, 1996, http://www.zifa.com

Zdravković, M., Trajanović, M. (2009). Integrated Product Ontologies for Inter-Organizational Networks, Computer Science and Information Systems (COMSIS). 6 (2) (2009) 29-46

Zdravković. M., Panetto, H., Trajanović, M. (2011). Local ontologies for semantic interoperability in supply chain networks. In: Proceedings of the 13th International Conference on Enterprise Information Systems (ICEIS 2011), 8-11. June 2011, Beijing, China, pages 22-31, SciTePress

Zdravković. M., Panetto, H., Trajanović, M., Aubrey, A. (2011). An approach for formalising the supply chain operations. Enterprise Information Systems. 5 (4) (2011) 401-421

Zdravković. M., Trajanović, M. (2011). Ontological framework for performance measurement of supply chain operations. In: Proceedings of 6th International Working Conference - Total Quality Management Advanced & Intelligent Approaches, 7-11. June 2011, Belgrade, Serbia

Zeffane, R. (1995). The widening scope of inter-organizational networking: economic, sectoral and social dimensions. Leadership & Organization Development Journal. 16 (5) (1995) 26-33

Zhang, Y., Lio, S., Jiao Y. (2008). An interoperability service utility platform for automobile Suppy Chain Management. In: the Proceedings of the 12th Enterprise Distributed Object Computing Conference Workshop, 2008 , pp. 230 - 237, Munich

Zhao, H., Ram, S. (2007). Combining schema and instance information for integrating heterogeneous data sources, Data Knowledge Engineering. 61 (2) (2007) 281-303

Table of Contents

Table of Figures

List of Acronyms

AHP	Analytic Hierarchy Process
AIAG	Automotive Industry Action Group
AIM	ATHENA Interoperability Methodology
API	Application Program Interface
APO	Advanced Planning and Optimization
ARCON	A Reference model for Collaborative Networks
B2B	Business-to-Business
BOM	Bill of Material
BPEL	Business Process Execution Language
BPM	Business Process Management
CAD	Computer Aided Design
CIMOSA	Computer Integrated Manufacturing Open System Architecture
CIS	Clinical Information System
CN	Collaborative Network
CNO	Collaborative Networked Organization
CPFR	Collaborative Planning, Forecasting, and Replenishment
CT	Computed Tomography
DIG	Description Logics Implementation Group
DL	Description Logic
DoDAF	Department of Defense Architecture Framework
DSL	Domain-Specific Language
ER	Entity/Relationship
EA	Enterprise Architecture
EC	European Commission
EDI	Electronic Data Interchange
EE	Extended Enterprise
EIF	Enterprise Interoperability Framework
EIS	Enterprise Information System
EM	Enterprise Modeling
EMR	Electronic Medical Record
ERP	Enterprise Resource Planning
EUP	Enterprise Unified Process
FEA	Finite Element Analysis
FInES	Future Internet Enterprise Systems
FOL	First-Order Logic
GERAM	Generalized Enterprise-Reference Architecture and Methodology
GIS	Geographic Information System
GPO	General Process Ontology
HTTP	Hypertext Transport Protocol
IDE	Integrated Development Environment
IDEAS	Interoperability Development for Enterprise Application and Software

IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IFAC	International Federation of Automatic Control
IFIP	International Federation for Information Processing
Interop NoE	Interoperability Network of Excellence
ISO	International Organization for Standardization
ISU	Interoperability Service Utilities
KIF	Knowledge Interchange Format
KPI	Key Performance Indicator
LAMP	Linux, Apache, MySQL and PHP/Perl/Python
LISI	Levels of Information Systems Interoperability
MDA	Model-Driven Architecture
MDE	Model-Driven Engineering
MRP	Material Requirements Planning
NSF	National Science Foundation
OIL	Ontology Interchange Language
OMG	Object Management Group
OWL	The Web Ontology Language
PACS	Picture Archiving and Communication System
PDM	Platform-Definition Model
PDM	Product Data Management
PERA	Purdue Enterprise-Reference Architecture
PERT	Program Evaluation Review Technique
PIM	Platform-Independent Model
PSM	Platform-Specific Model
RDF	Resource Description Framework
RDFS	RDF Schema
RDQL	RDF Query Language
RFID	Radio Frequency Identification
RIS	Radiology Information System
RPC	Remote Procedure Call
SaaS	Software-as-a-Service
SBM	Supply Business Management
SCC	supply chain Council
SCM	Supply Chain Management
SCOR	Supply Chain Operations Reference
SCOR-KOS	SCOR Knowledge Organization System
S-ISU	Semantic Interoperability Service Utilities
SLA	Service Level Agreement
SME	Small or Medium Enterprise
SMTP	Simple Mail Transport Protocol
SOA	Service Oriented Architecture
SPARQL	SPARQL Protocol and RDF Query Language
SPI	Smart Product Infrastructure
SPM	Synchronization Point Model

SQL	Structured Query Language
SVG	Scalable Vector Graphics
SWRL	The Semantic Web Rule Language
TOVE	Toronto Virtual Enterprise
UBL	Universal Business Language
UDDI	Universal Description Discovery and Integration
UML	Unified Modeling language
UNSPSC	The United Nations Standard Products and Services Code
URI	Uniform Resource Identificator
VBE	Virtual Breeding Environment
VE	Virtual Enterprise
VECCF	Virtual Enterprise Chain Collaboration Framework
VEM	Virtual Enterprise Methodology
VO	Virtual Organization
WSDL	Web Services Description Language
WSMX	Web Service Execution Environment
XML	eXtensible Markup Language