Abstract—Service Oriented Architectures represent a huge step towards providing simpler, more dynamic and cheaper integration solutions. But SOA cannot and it is not meant to solve all the heterogeneity problems inherent to enterprise integration tasks. Semantics is coming to offer the tools to enable scalable, efficient and cost effective solutions to some of these problems. Using ontologies and semantics, services can be unambiguously described, from data, functionality and behavior point of view. The purpose of this paper is to raise awareness on semantics and semantic interoperability, to review the main concepts regarding this topic such as semantic spectrum as well as the models and patterns that can be used in order to achieve semantic interoperability.

Keywords—semantics, interoperability, semantic interoperability, service oriented architecture

I. INTRODUCTION

In an environment characterized by a huge variety of dynamic, independent data sources, with a wide geographical distribution, accurate access to relevant information is becoming increasingly complex. This complexity is exacerbated by the evolving system, semantic and structural heterogeneity of these potentially global, cross-disciplinary, multicultural and rich-media technologies. Clearly, solutions to these challenges require addressing directly a variety of interoperability issues.

Several classification frameworks have been proposed to group together different types and aspects of interoperability.

One of these classifications supposes that in order to ensure interoperability of ICT systems the following aspects must be taken into account

a) Interoperability at the organizational level: involves the process by which different organizations work together to achieve mutual benefits by using specific ICT resources;
b) Interoperability at the semantic level: involves ensuring a common understanding of concepts, services, data;
c) Interoperability at the technical level: it covers technical elements to enable interconnection and interoperability of computer systems and services.

Another classification differentiates between types of interoperability based on the perspective on heterogeneity in computing systems.

According to this classification we can identify four different levels of interoperability as shown in Fig. 1.

Out of the four types mentioned above, semantic interoperability is the one category that has been most neglected while significant progress has been achieved in system, syntactic, and structural/schematic interoperability. This is mostly due to the fact that developers tend to take semantic interoperability for granted especially because the semantic interpretation, mapping and transformation are so ingrained with home grown applications, Enterprise
Application Integration (EAI) and Enterprise Information Integration (EII) [1].

To achieve semantic interoperability, systems must be able to exchange data in such a way that the precise meaning of the data is readily accessible and the data itself can be translated by any system into a form that it understands.

II. SEMANTIC INTEROPERABILITY IN SOA CONTEXT

A. General consideration

Service Oriented Architectures represent a huge step towards providing simpler, more dynamic and cheaper integration solutions. By having services to encapsulate discrete piece of functionality that can be later discovered and consumed is without a doubt the critical step in moving from a patchwork of legacy products, monolithic off-shelf applications and proprietary integration to a strongly decoupled, robust yet flexible software architecture.

But SOA cannot and it is not meant to solve all the heterogeneity problems inherent to enterprise integration tasks. Such heterogeneity problems could be induced for example by the services (the key players of SOA) themselves or by the environment the enterprises are acting in. In the former case it is natural that services deployed by different providers to have specific requirements regarding data formats or communication protocols.

Semantics is coming to offer the tools to enable scalable, efficient and cost effective solutions to these problems. Using ontologies and semantics, services can be unambiguously described, from data, functionality and behavior point of view.

Semantic interoperability in a Service Oriented Architecture context refers to the ability of service consumers and providers to exchange data that makes sense in a consistent, flexible way no matter what type of information is involved. Without semantic interoperability there are no guarantees that the data embedded within messages exchanged by different services are interpreted by consumers and providers as representing the same concepts, relations or entities, and so it can be misinterpreted and ultimately have negative effects on the business.

According to [3] besides the obvious reasons behind achieving comprehensive solutions to semantic interoperability there are several trends in software technologies and organizationally complex information infrastructures that provide further motivation such as:

a) progress in techniques to model, capture, represent and reason about semantics; graduate progress in attention from data to information, and increasingly knowledge;

b) challenges in dealing with non-traditional (esp. visual) data that cannot be easily handled with well known IR and traditional database techniques;

c) attention to the issue of interoperability in various domains and research areas (e.g., bibliographic data, digital libraries, geographic and environmental data, space and astronomy data, etc.) and the improved technological ability to develop more challenging applications (e.g., digital earth, digital human) involving wider variety of users and perspectives over shared information resources;

d) support to the evolving concepts of virtual organizations and adversories – and concomitant requirement for flexible semantic interoperability to interpret the available information in light of new market contingencies and the variety of intra- and cross-disciplinary forms of collaboration scientific or otherwise.

B. Semantic spectrum

In order to understand semantic interoperability we need to familiarize ourselves with the concepts, methodologies and techniques that support it. These concepts, methodologies and techniques aimed at creating increasingly precise definitions for data elements in knowledge representation are grouped together into the Semantic spectrum [11].

When talking about defining a data element we must refer to both the content of the element (data) as well as the description of the element (metadata). The elements contained in the Semantic Spectrum can be classified from these two perspectives as shown in Fig. 2.

---

**Fig. 2 – Data/metadata classification of Semantic Spectrum elements**

A glossary is a lesser-known list of words encountered in a text, accompanied by explanations in order to facilitate reading.

A controlled vocabulary provides a way to organize knowledge for subsequent retrieval by grouping different ways of describing a concept under a single word or phrase. Phonebook Yellow Page listings are arranged using controlled vocabulary. For example, a search for "Car Dealers" leads you to a note to “see Automobile Dealers.” At a basic level, this is how a controlled vocabulary system works.

A data dictionary is a centralized repository of information about data such as meaning, relationships to other data, origin, usage and format.
A data model refers to an abstract representation of the structure of data elements. The main aim of data models is to support the development of information systems by providing the definition and format of data.

A taxonomy deals with the establishment of classification and systematization laws for complex structured fields. An enterprise taxonomy is organized into a hierarchical structure to convey the parent-child relationship of terms and concepts often associated with content management, knowledge management and search technologies.

Ontology defines the terms used to describe and represent an area of knowledge.

C. Ontologies

An ontology deals with properties of objects and categories of objects from a given area. Currently, ontology is the usual way by which semantics is modeled by expressing the meaning and the concepts that define a basis between different informatics systems.

Structured ontology provides vocabulary and semantics, defining a common vocabulary for those who are interested in exchanging information in a given field. It includes definition of basic concepts and relations between them. Among various other classification schemes and structures, including the before mentioned controlled vocabularies and taxonomies, ontologies are often viewed as allowing more complete and precise domain models.

Ontology types are defined depending on the level of formalization and generalization. Depending on the level of formalization there are three types of ontology:

- a) informal ontology, the simplest, involving a set of names of concepts organized in an hierarchy;
- b) ontology terminology, characterized by a hierarchy of concepts described by definitions in natural language;
- c) formal ontology, involving axioms and definitions created in a formal language.

Depending on the level of generalization, there are six types of ontologies:

- d) high-level ontology, which includes very general concepts like space, time, object event, independent of a particular area;
- e) general ontology which includes concepts related to fundamental human knowledge;
- f) domain ontology, which includes concepts associated with a specific interest areas;
- g) task ontology, with concepts relating to the execution of certain tasks or activities;
- h) application ontology, with key concepts of a particular application;
- i) basic ontology or meta-ontology, including concepts common to various domains, which can be further developed to serve a particular area of interest.

Developing ontologies involves the following phases:

- defining concepts, classes, properties and attributes;
- organizing concepts into an hierarchical structure;
- creating detailed definitions of concepts using logical description and defining constraints and restrictions.

One challenging issue in supporting semantic interoperability is how to allow both users and providers to subscribe to existing ontologies of their choice or create a new one.

In the context of a service oriented architecture framework we can say that any service description can be regarded as an independent ontology. As such, each service provider can develop its own ontology thus making it impossible to make assumptions regarding consistency of message interpretation.

Interpreting someone else’s ontology is a conjecture unless a standard set of predicates is adopted, shared, understood, and consistently interpreted by every node in the distributed environment.[2] The availability and acceptance of semantic standards is an enabling condition for widespread adoption of service-oriented infrastructures.

III. MODELS FOR SEMANTIC INTEROPERABILITY, PATTERNS AND ANTI PATTERNS

A. Models

[5] proposes a series of models as solutions for semantic interoperability that are classified based on two fundamental dimensions: choosing one of two possible ways to set up integration mapping and choosing whether the integration logic is executed in a single, distinguished node or the execution is distributed among multiple, functionally equivalent nodes.

Basically, there are two alternatives to set up the schema integration mappings, that is any-to-any or any-to-one, and two ways to execute the integration logic: centralized on a single node or distributed among several nodes (decentralized).

By combining these options four interoperability models are obtained:

- a) Any-to-any centralized model - services are interpreted and mapped to others without resorting to ontologies. This model is used in the specification of business processes, supported by standardized languages such as BPEL, which are used in mature production environments. This model is really the traditional one used by system integrators. It is a one-off, high level-of-effort, non-generalized and non-repeatable model of integration that only works in closed environments.

![Fig. 3 - Any-to-any centralized model as diagrammed by [5]](image)

- b) Any-to-one centralized model - input/output data are mapped to a single ontology that is a superset of all contributing systems managed by a specialized application
or service. This interoperability model, too, is quite costly in that all suppliers (or service providers) must conform to the single ontology.

Fig. 4 - Any-to-one centralized model as diagrammed by [5]

c) Any-to-any decentralized model - those where services manage their integration with each other without resorting to any overall business model or integrator component. The integration logic is distributed, and there are not shared ontologies. This is a peer-to-peer system, sometimes known as P2P information integration systems or 'emergent semantics.

Fig 5 - Any-to-any decentralized model as diagrammed by [5]

d) Any-to-one decentralized model - the integration logic is distributed by any service or component implementation, based on a shared ontology. In this model Endpoints directly connect and exchange information with one another, but service descriptions are mapped to (or directly taken from) a common shared business model.

B. Patterns

Another approach in achieving semantic interoperability is proposed by [1] and consists of studying patterns and best practices as well as anti-patterns, the later cautioning the developers in avoiding traps.

Although there are numerous patterns for achieving semantic interoperability they can be roughly classified by the following:

- Pattern one: Point-to-point semantic integration - each data source has its own proprietary semantic meaning, and semantic translation is performed in a point-to-point manner. If data definition in one data source is changed, the impact to other systems is multiplied and often unpredictable thus this particular pattern is messy and quite difficult to manage when dealing with growing data. However, point-to-point integration is not necessarily a bad thing. It can be used selectively to ensure high performance and create a "fast path".

- Pattern two: Hub-and-spoke semantic integration - each system has its own proprietary semantic meaning, but is mapped to a logical data model which can be instantiated as a physical federated model or a canonical message model. Semantic interoperability is achieved within an enterprise via a hub-and-spoke topology, which reduces the redundancy and maintenance cost of point-to-point integration.

- Pattern three: Master data management (MDM) pattern – it emerges as a pattern of semantic interoperability responding to data silos produced by departmental solutions. Today, many versions of truth exist in a typical enterprise information management system. A MDM system connects heterogeneous information sources and produces a single version of truth on key information such as customers or products for Online Transaction Processing (OLTP) and Operational Data Store (ODS) systems. The key information could be either a data instance, such as a particular customer, or metadata, such as specifications of products. A MDM system liberates data from individual business applications, package vendors and is based on open standards. As a result, data is truly treated and reused as a corporate asset. With MDM systems, companies gain many proven benefits, such as improved customer relationships, reduced time to introduce new products to market, data integrated with legacy systems and enabling asset reuse.

- Pattern four: Industry information model - in order to encourage interoperability within an industry, vertical industry standardization groups develop industry-specific information models, which often include XML messages and message schema, also known as Domain Information Models (DIMs). DIMs are typically XML-based and used to exchange messages in a business-to-business (B2B) environment. The members of industry standard groups agree to follow the specifications, and they are often required to certify for compliance.

- Pattern five: The Semantic Web – it cuts across the boundaries of applications, enterprises and industries. The Semantic Web links and relates elements of the data model to a common ontology. It uses the Resource Description Framework and the Web Ontology Language to allow data to be shared and reused on the Web [8].
We can notice from analyzing the patterns and models presented that although they appear to have been designed starting from different premises they share common points: pattern one – point to point semantic presentation with the any to any decentralized model, pattern two hub-and-spoke semantic integration with the any to one centralized model.

C. Anti Patterns

Anti-patterns refer to elements that get in the way of semantic interoperability such as: semantic chaos, business terms and definitions chaos or lack of information integration logic reuse.

Semantic chaos refers to the fact that everybody defines their own schemas and vocabularies, does not follow any information standards, and does not consider semantic interoperability with the rest of the systems. People use their own terms and have difficulty on understanding each other.

Business terms and definitions chaos - frequently, business users will make requests from information technology professionals to provide data to help them with their business analysis using business terms and definitions that may be quite different in different departments.

Regarding the lack of information integration logic reuse – there are three major information integration patterns to achieve semantic interoperability: data federation, data consolidation and Enterprise Application Integration (EAI). In reality, different groups within enterprises are good at using one pattern. Without cross-team coordination and enterprise-level, long term strategy, it is very easy to create a new set of integration silos. As the result of a siloed approach, each group may have slightly different semantic integration and data processing logic.

IV. Conclusion

As we move forward into the era of information the challenge of semantic interoperability will only increase and this particular area of interest will hopefully move more toward the spot-light.

In this paper we have tried to raise awareness on the issue of semantic interoperability by placing it in the context of one of the most interesting trends of the moment – Service Oriented Architecture. We have presented the main concepts used in relation with semantic interoperability and then we have analyzed some of the models and patterns that can be used to achieve it. Also we presented some of the factors or anti-patterns that developers should keep in mind as problems related with the topic at hand.

REFERENCES