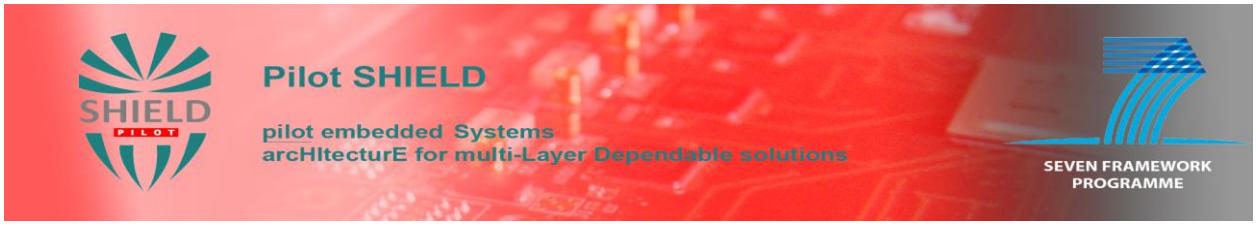


Project no: 100204

**p-SHIELD**

pilot embedded Systems architecture for multi-Layer Dependable solutions

Instrument type: Capability Project

Priority name: Embedded Systems (including RAILWAYS)

# D5.1: pSHIELD Semantic Models (Prototypes)

Due date of deliverable: M15 (30th August 2011)
Actual submission date: M15 (15th September 2011)

Start date of project: 1st June 2010 Duration: 19 months

Organisation name of lead contractor for this deliverable: TRS

Revision [Draft C]

| **Project co-funded by the European Commission within the Seventh Framework Programme (2007-2012)** | | |
|---|---|---|
| **Dissemination Level** | | |
| **PU** | Public | |
| **PP** | Restricted to other programme participants (including the Commission Services) | |
| **RE** | Restricted to a group specified by the consortium (including the Commission Services) | X |
| **CO** | Confidential, only for members of the consortium (including the Commission Services) | |

| Document Authors and Approvals | | | |
|---|---|---|---|
| **Authors** | | **Date** | **Signature** |
| **Name** | **Company** | | |
| Andrea Fiaschetti | Univ. “La Sapienza” | | |
| Andrea Taglialatela | TRS | | |
| Sergio Wanderlingh | TRS | | |
| Giuseppe Fusco | TRS | | |
| Andrea Morgagni | Elsag Datamat | | |
| Renato Baldelli | Elsag Datamat | | |
| Andi Palo | Univ. “La Sapienza” | | |
| Vincenzo Suraci | Univ. “La Sapienza” | | |
| Mohammad Chowdhury | CWIN | | |
| | | | |
| **Reviewed by** | | | |
| **Name** | **Company** | | |
| Josef Noll | Movation | | |
| Sarfraz Alam | CWIN | | |
| Andrea Fiaschetti | Univ. “La Sapienza” | | |
| **Approved by** | | | |
| **Name** | **Company** | | |
| Andrea Taglialatela | TRS | | |

| Modification History | | |
|---|---|---|
| **Issue** | **Date (DD/MM//YY)** | **Description** |
| **Draft A** | 15/03/2011 | First issue for comments |
| **Draft B** | 30/06/2011 | Second issue for comments |
| **Draft C** | 26/07/2011 | Third Issue for comments |
| **Version 1** | 15/09/2011 | First version |

# Contents

# Figures

# Tables

# Acronyms

| Acronym | Meaning |
|---|---|
| ESD | Embedded System Device |
| ESs | Embedded Systems |
| L-ESD | Legacy Embedded System Device |
| MS | Middleware Service |
| MwA | Middleware Adapter |
| NC | Node Capability |
| NoA | Node Adapter |
| NS | Network Service |
| NwA | Network Adapter |
| pS-ESD | pSHIELD Embedded System Device |
| pS-MS | pSHIELD Middleware Service |
| pS-OS | pSHIELD Overlay Service |
| pS-P | pSHIELD Subsystem |
| pS-P | pSHIELD Proxy |
| pS-SPD-ESD | SPD Embedded System Device |
| SPD | Security Privacy Dependability |
| CC | Common Criteria |
| FUA | Faults with Unauthorized Access |
| HMF | Human-Made Faults |
| NFUA | Not Faults with Unauthorized Access |
| NHMF | Nonhuman-Made Faults |
| SoC | System on Chip |

- This page intentionally left blank -

# 1 Executive Summary

Semantic technologies are becoming more and more attractive as they enable exploiting the benefits of the explicit knowledge of a domain in an effective way.

In pSHIELD, semantic technologies shall address the interoperability issues among different SPD technologies by enabling their composition. This goal shall be accomplished by building a framework that, in a consistent fashion, will provide a **methodology** for ontology building and verification, a suitable **meta-model** of the SPD in Embedded Systems, and a set of elementary functional components for **semantic management** of systems.

This framework will lay a groundwork to build, as a second step, an ontology of the SPD modules, capabilities and interfaces, that thanks to a semantic-aware model of all the exchanged information and control flows between the node, network, middleware and overlay layer, will allow an effective way to represent and reason about all the relevant entities by means of a common (shared) and consistent schema.

The structure of WP5, as conceived in the Technical Annex, is summarized in Figure 1.1, where one of the main novelty addressed by these activities (i.e. the composability mechanism) is represented as a closed loop system. In this context, Task 5.1 provides the enabling (semantic) technologies for system modeling, measuring and control.

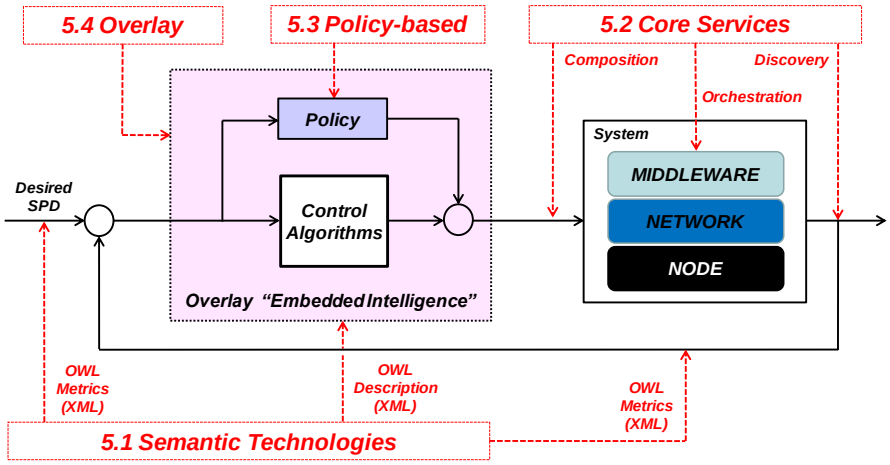


**Figure 1.1 - Work Package 5 Structure**

The document is structured as follows: in Section 4 an overview of the Semantic Tecnologies is provided; in Section 5 the Ontology Formalization procedure is described as part of the prototype; then in section 6 the pSHIELD model is punctually described, while in Section 7 some consideration about the potential inference mechanism underlying this models are reported, with particular focus on semantic composition.

The purpose of the present document is to introduce and describe the pSHIELD Semantic Model Prototype developed in Task 5.1. This prototype will be enriched with technological and background analysis in Deliverable 5.3, so the union of D5.1 and D5.3 will cover the whole effort for this activity (Task 5.1).

# 2 Introduction

## 2.1 Semantic interoperability

Over the last few years much work has been conducted in regards to the research topic of fully interoperability. The use of specific data models implies that making a whole system from its components is a process which is very time consuming and prone to the introduction of artificial layers that cause performance and overall control decay

The advantages of a successful model integration are obvious for many reasons:

- Quality improvement of model due to the availability of *large and complete information*.
- Improvement of *existing analysis* and application of the *new analysis*.
- Cost reduction resulting from the *multiple use of existing information sources*.
- Avoidance of redundant data and conflicts that can arise from *redundancy*.

As we shift from the problem of information integration to those of model integration, however, difficulties arising from organizational, competence questions and many other technical problems have to be solved. We distinguish different integration levels, that need to be solved in order to achieve complete integrated access to information:

- Syntactic Integration: Many standards have evolved that can be used to integrate different information sources. Beside classical database interfaces such as ODBC, web-oriented standards such as HTML and XML are gaining importance.
- Structural Integration: The first problem that passes a purely syntactic level is the integration of heterogeneous structures. This problem is normally solved by mediator systems defining mapping rules between different information structures.
- Semantic Integration: In the following, we use the term semantic integration or semantic translation, respectively, to denote the resolution of semantic conflicts, that make a one to one mapping between concepts or terms impossible.

Our approach provides an overall solution to the problem of information integration, taking into account all three levels of integration and combining several technologies, including standard markup languages, mediator systems and ontologies. In order to overcome the obstacles mentioned earlier, it is not sufficient to solve the heterogeneity problems separately. It is important to note that these problems can only be solved with a system taking all three levels of integration into account.

- **Syntactic Integration**: The typical task of syntactic data integration is, to specify the information source on a syntactic level. This means, that different data type problems can be solved (e. g. short int vs. int and/or long). This first data abstraction is used to re-structure the information source. The standard technology to overcome problems on this level are wrappers. Wrappers hide the internal data structure model of a source and transform the contents to a uniform data structure model.
- **Structural Integration**: The task of structural data integration is, to re-format the data structures to a new homogeneous data structure. This can be done with the help of a formalism that is able to construct one specific information source out of numerous other information sources. This is a classical task of a middleware which can be done with CORBA on a low level or rule-based mediators on a higher level. Mediators provide flexible integration of several information systems since it combines, integrates, and abstracts the information provided by the sources. Normally the sources are encapsulated by wrappers. Popular implementations of mediators have are rule driven: Usually, the rules in the mediator describe how information of the sources can be mapped to the integrated view. In simple cases, a rule mediator converts the information of the sources

into information on the integrated view. The mediator uses the rules to split the query, which is formulated with respect to the integrated view, into several sub-queries for each source and combine the results according to query plan. A mediator has to solve the same problems which are discussed in the federated database research area, i. e. structural heterogeneity (schematic heterogeneity) and semantic heterogeneity (data heterogeneity) Structural heterogeneity means that different information systems store their data in different structures. Semantic heterogeneity considers the content and semantics of an information item. In rule-based mediators, rules are mainly designed in order to reconcile structural heterogeneity. Where as discovering semantic heterogeneity problems and their reconciliation play a subordinate role. But for the reconciliation of the semantic heterogeneity problems, the semantic level must also be considered. Contexts are one possibility to describe the semantic level. A context contains ”meta data relating to its meaning, properties (such as its source, quality, and precision), and organization”. A value has to be considered in its context and may be transformed into another context (so-called context transformation).

- **Semantic Integration**: The semantic integration process is by far the most complicated process and presents a real challenge. As with database integration, semantic heterogeneities are the main problems that have to be solved within real time and embedded systems.
    - Generic semantic heterogeneity: Heterogeneity resulting from field- and object-based databases.
    - Contextual semantic heterogeneity: Heterogeneity based on different meanings of concepts and schemes.

In this project, we will focus on contextual semantic heterogeneity which is based on different semantics of the local schemata. In order to discover semantic heterogeneities, a formal representation is needed. Lately, WWW standardized markup languages such as XML and RDF have been developed by the W3C community for this purpose (W3C, 1998), (W3C, 1999). We will describe the value of these languages for the semantic description of concepts and also argue that we need more sophisticated approaches to overcome the semantic heterogeneity problem. Actually, we pursue an intergration/ interoperation process based on Ontologies.

XML and RDF have been developed for the semantic description of information sources. In order to overcome the purely visualization-oriented annotation provided e. g. by HTML, XML was proposed as an extensible language allowing the user to define his own tags in order to indicate the type of it’s content. Therefore, it followed that the main benefit of XML lies actually in the opportunity to exchange data in a structured way. Recently, this idea has been emphasized by introducing XML schemata that could be seen as a definition language for data structures. In the following paragraphs we sketch the idea behind XML and describe XML schema definitions and their potential use for data exchange. A data object is said to be XML document if it follows the guidelines for wellformed XML documents provided by the W3C community. The specification provide a formal grammar used in well-formed documents. In addition to the general grammar, the user can impose further grammatical constraints on the structure of a document using a document type definition (DTD). A XML document is valid if it has an associated type definition and complies to the grammatical constraints of that definition. A DTD specifies elements that can be used in an XML document. In the document, the elements are delimited by a start and an end tag. It has a type and may have a set of attribute specifications consisting of a name and a value.The additional constraints in a DTD refer to the logical structure of the document, this especially includes the nesting of tags inside the information body that is allowed and/or required. Further restrictions that can be expressed in a DTD concern the type of the attributes and default values to be used when no attribute value is provided.

An XML schema itself is, an XML document defining the valid structure of an XML document in the spirit of a DTD. The elements used in a schema definition are of the type ’element’ and have attributes that are defining the restrictions already mentioned above. The information in such an element is a list of further element definitions that have to be nested inside the defined element. Furthermore, XML schema have some additional features that are very useful to define data structures such as:

- Support for basic data types.
- Constraints on attributes such as occurrence constraints.

- Sophisticated structures such as type definition derived by extending or restricting other types.
- A name-space mechanism allowing the combination of different schemata.

We will not discuss these features at length. However, it should be mentioned that the additional features make it possible to encode rather complex data structures. This enables us to map data-models of applications from whose information we want to share with others on an XML schema. From this point, we can encode our information in terms of an XML document and make it (together with the schema, which is also an XML document) available over the internet. This procedure has a big potential in the actual exchanging of data. However, the user must to commit to our data-model in order to make use of the information. We must point out that an XML schema defines the structure of data providing no information about the content or the potential use for others. Therefore, it lacks an important advantage of meta-information.We argued that XML is designed to provide an interchange format for weakly structured data by defining the underlying data-model in a schema and by using annotations, from the schema, in order to clarify the role of single statements. Two things are important in this claim from the information sharing point:

- XML is purely syntactic/structural in nature.
- XML describes data on the object level.

Consequently, we have to find other approaches if we want to describe information on the meta level and define its meaning. In order to fill this gap, the RDF standard has been proposed as a data model for representing meta-data about web pages and their content using an XML syntax. The basic model underlying RDF is very simple, every kind of information about a resource which may be a web page or an XML element is expressed in terms of a triple (resource, property, value). Thereby, the property is a two-placed relation that connects a resource to a certain value of that property. This value can be a simple data-type or a resource. Additionally, the value can be replaced by a variable representing a resource that is further described by nested triples making assertions about the properties of the resource that is represented by the variable. Furthermore, RDF allows multiple values for a single property. For this purpose, the model contains three builtin data types called collections, namely an unordered lists (bag), ordered lists (seq), and sets of alternatives (alt) providing some kind of an aggregation mechanism. A further requirement arising from the nature of the web is the need to avoid name-clashes that might occur when referring to different web-sites that use different RDF-models to annotate meta-data. RDF defines name-spaces for this purpose. Name-spaces are defined by referring to an URL that provides the names and connecting it to a source id that is then used to annotate each name in an RDF specification defining the origin of that particular name: source id:name

A standard syntax has been developed to express RDF-statements making it possible to identify the statements as meta-data, thereby providing a low level language for expressing the intended meaning of information in a machine processable way. The very simple model underlying ordinary RDF-descriptions leave a lot of freedom for describing meta-data in arbitrary ways. However, if people want to share this information, there has to be an agreement on a standard core of vocabulary in terms of modeling primitives that should be used to describe meta-data. RDF schemes (RDF/S) attempt to provide such a standard vocabulary. Looking closer at the modeling components, reveals that RDF/S actually borrows from frame systems well known from the area of knowledge representation. RDF/S provides a notion of concepts (class), slots (property), inheritance (SubclassOf, SubslotOf) and range restrictions (Constraint Property). Unfortunately, no well-defined semantics exist for these modeling primitives in the current state. Further, parts such as the re-identification mechanism are not well defined even on an informal level. Lastly, there is no reasoning support available, not even for property inheritance.

After introducing the W3C standards for information exchange and meta-data annotation we have to investigate their usefulness for information integration with reference to the three layers of integration. Firstly, we previously discovered that XML is only concerned with the issue of syntactic integration. However, XML defines structures as well, except there are no sophisticated mechanism for mapping different structures. Secondly, RDF is designed to provide some information on the semantic level, by enabling us to include meta-information in the description of a web-page. In the last section we mentioned, RDF in it's current state fails to really provide semantic descriptions. Rather it provides a common syntax and a basic vocabulary that can be used when describing this meta-data. Fortunately, the designers of RDF are aware that there is a strong need for an additional 'logical level' which defines a clear semantics for RDF-expressions and provides a basis for integration mechanisms.

Our conclusion about current web standards is that using XML and especially XML schemata is a suitable way of exchanging data with a well defined syntax and structure. Furthermore, simple RDF provides a uniform syntax for exchanging meta-information in a machine-readable format. However, in their current state neither XML nor RDF provides sufficient support for the integration of heterogeneous structures or different meanings of terms. There is a need for semantic modeling and reasoning about structure and meaning. Promising candidates for semantic modeling approaches can be found in the areas of knowledge representation, as well as, in the distributed databases community. We will discuss some of these approaches in the following sections.

Recently, the use of formal ontologies to support information systems has been discussed. The term 'Ontology' has been used in many ways and across different communities. If we want to motivate the use of ontologies for information integration we have to define what we mean when we refer to ontologies. In the following sections, we will introduce ontologies as an explication of some shared vocabulary or conceptualization of a specific subject matter. Further, we describe the way an ontology explicates concepts and their properties and finally argue for the benefit of this explication in many typical application scenarios. In general, each person has an individual view on the world and the things he/she has to deal with every day. However, there is a common basis of understanding in terms of the language we use to communicate with each other. Terms from natural language can therefore, be assumed to be a shared vocabulary relying on a (mostly) common understanding of certain concepts with very little variety. This common understanding relies on specific idea of how the world is organized. We often call these ideas a conceptualization of the world. These conceptualizations provide a terminology that can be used for communication between people. The example of our natural language demonstrates, that a conceptualization cannot be universally valid, but rather a limited number of persons committed to that particular conceptualization. This fact is reflected in the existence of different languages which differ even more (English and Japanese) or much less (German and Dutch). Confusion can become worse when we are considering terminologies developed for a special scientific or economic areas. In these cases, we often find situations where one term refers to different phenomena.

The use of the term 'ontology' in philosophy and in computer science serves as an example. The consequence of this confusion is, a separation into different groups, that share terminology and its conceptualization. These groups are then called information communities. The main problem with the use of a shared terminology according to a specific conceptualization of the world is that much information remains implicit. When a mathematician talks about a binomial normal he is referring to a wider scope than just the formula itself. Possibly, he will also consider its interpretation (the number of subsets of a certain size) and its potential uses (e. g. estimating the chance of winning in a lottery). Ontologies set out to overcome this problem of implicit and hidden knowledge by making the conceptualization of a domain (e. g. mathematics) explicit. This corresponds to one of the definitions of the term ontology most popular in computer science (Gruber, 1993): An ontology is an explicit specification of a conceptualization. An ontology is used to make assumptions about the meaning of a term available. It can also be viewed an explication, of the context a term, it is normally used in. Lenat (Lenat, 1998) for example, describes context in terms of twelve independent dimensions that have to be know in order to understand a piece of knowledge completely.

There are many different ways in which an ontology may explicate a conceptualization and the corresponding context knowledge. The possibilities range from a purely informal natural language description of a term corresponding to a glossary up, to a strictly formal approach, with the expressive power of full first order predicate logic or even beyond (e. g. Ontolingua (Gruber, 1991)). Jasper and Uschold (Jasper and Uschold, 1999) distinguish two ways in which the mechanisms for the specification of context knowledge by an ontology can be compared:

- Level of Formality: The specification of a conceptualization and its implicit context knowledge, can be done at different levels of formality. As already mentioned above, a glossary of terms can also be seen as an ontology, despite its purely informal character. A first step to gain more formality, is to describe a structure to be used for the description. A good example of this approach is the standard web annotation language XML (see section ). The DTD is an ontology describing the terminology of a web page on a low level of formality. Unfortunately, the rather informal character of XML encourages its misuse. While the hierarchy of an XML specification was originally designed to describe a layout, it can also be exploited to represent sub-type hierarchies, (van Harmelen and Fensel, 1999) which may lead to confusion. Fortunately, this problem can be solved by assigning formal semantics to the structures used for the description of the ontology. An example of this is the conceptual modeling language CML (Schreiber et al.,

1994). CML offers primitives that describe a domain which can be given a formal semantic in terms of first order logic (Aben, 1993). However, a formalization is only available for the structural part of a specification. Assertions about terms and the description of dynamic knowledge is not formalized which offers total freedom for a description. On the other, there are specification languages which are completely formal. A prominent example is the Knowledge Interchange Format (KIF) (Genesereth and Fikes, 1992) which was designed to enable different knowledge-based systems to exchange knowledge. KIF has been used as a basis for the Ontolingua language (Gruber, 1991) which supplies formal semantics to that language as well.

- Extend of Explication: The other comparison criterion is, the extend of explication that is reached by the ontology. This criterion is strongly connected with the expressive power of the specification language used. We already mentioned DTD's which are mainly a simple hierarchy of terms. Furthermore, we can generalize this by saying that, the least expressive specification of an ontology consists of an organization of terms in a network using two-placed relations. The idea of this goes back to the use of semantic networks in the seventies. Many extensions of the basic idea examined have been proposed. One of the most influential ones was, the use of roles that could be filled out by entities showing a certain type (Brachman, 1977). This kind of value restriction can still be found in recent approaches. RDF schema descriptions (Brickley and Guha, 2000), which might become a new standard for the semantic descriptions of web-pages, are an example of this. An RDF schema contains class definitions with associated properties that can be restricted by socalled constraint-properties. However, default values and value range descriptions are not expressive enough to cover all possible conceptualizations. A more expressive power can be provided by allowing classes to be specified by logical formulas. These formulas can be restricted to a decidable subset of first order logic. This is the approach of description logics (Borgida and Patel-Schneider, 1994). Nevertheless, there are also approaches that allow for even more expressive descriptions. In Ontolingua for example, classes can be defined by arbitrary KIF-expressions. Beyond the expressiveness of full first-order predicate logic, there are also special purpose languages that have an extended expressiveness to cover specific needs of their application area. Examples are; specification languages for knowledge-based systems which often including variants of dynamic logic to describe system dynamics.

Ontologies are useful for many different applications, that can be classified into several areas. Each of these areas, has different requirements on the level of formality and the extend of explication provided by the ontology. We will review briefly common application areas, namely the support of communication processes, the specification of systems and information entities and the interoperability of computer systems. Information communities are useful because they ease communication and cooperation among members with the use of shared terminology with well defined meaning. On the other hand, the formalization of information communities makes communication between members from different information communities very difficult. Generally, because they do not agree on a common conceptualization. Although, they may use the shared vocabulary of natural language, most of the vocabulary used in their information communities is highly specialized and not shared with other communities. This situation demands for an explication and explanation of the use of terminology. Informal ontologies with a large extend of explication are a good choice to overcome these problems. While definitions have always played an important role in scientific literature, conceptual models of certain domains are rather new. Nowadays systems analysis and related fields like software engineering, rely on conceptual modeling to communicate structure and details of a problem domain as well as the proposed solution between domain experts and engineers. Prominent examples of ontologies used for communication are Entity-Relationship diagrams and Object-oriented Modeling languages such as UML. ER-diagrams as well as UML are not only used for communication, they also serve as building plans for data and systems guiding the process of building (engineering) the system. The use of ontologies for the description of information and systems has many benefits. The ontology can be used to identify requirements as well as inconsistencies in a chosen design. Further, it can help to acquire or search for available information. Once a systems component has been implemented, its specification can be used for maintenance and extension purposes. Another very challenging application of ontology-based specification is the reuse of existing software. In this case, the specifying ontology serves as a basis to decide if an existing component matches the requirements of a given task. Depending on the purpose of the specification, ontologies of different formal strength and expressiveness are to be utilized. While the process of communication design decisions and the acquisition of additional information normally benefit from rather informal and expressive ontology representations (often graphical), the directed search for information needs a rather strict specification with a limited vocabulary to limit the computational effort. At

the moment, the support of semiautomatic software reuse seems to be one of the most challenging applications of ontologies, because it requires expressive ontologies with a high level of formal strength.

The previously discussed considerations might provoke the impression that the benefits of ontologies are limited to systems analysis and design. However, an important application area of ontologies is the integration of existing systems. The ability to exchange information at run time, also known as interoperability, is an valid and important topic. The attempt to provide interoperability suffers from problems similar to those associated with the communication amongst different information communities. The important difference being the actors are not people able to perform abstraction and common sense reasoning about the meaning of terms, but machines. In order to enable machines to understand each other, we also have to explicate the context of each system on a much higher level of formality.

Ontologies are often used as Inter-Linguas in order to provide interoperability: They serve as a common format for data interchange. Each system that wants to inter-operate with other systems has to transfer its data information into this common framework. Interoperability is achieved by explicitly considering contextual knowledge in the translation process.

For an appropriate support of an integration of heterogeneous information sources an explicit description of semantics (i. e. an ontology) of each source is required. In principle, there are three ways how ontologies can be applied:

- a centralized approach, where each source is related to one common domain ontology,
- a decentralized approach, where every source is related to its own ontology, or
- a hybrid approach, where every source is related to its own ontology but the vocabulary of these ontologies stem from a common domain ontology

A common domain ontology describes the semantics of the domain in the SIMS mediator (Arens et al., 1996). In the global domain model of these approaches all terms of a domain are arranged in a complex structure. Each information source is related to the terms of the global ontology (e. g. with articulation axioms (Collet et al., 1991)). However, the scalability of such a fixed and static common domain model is low (Mitra et al., 1999), because the kind of information sources which can be integrated in the future is limited. In OBSERVER (Mena et al., 1996) and SKC (Mitra et al., 1999) it is assumed, that a predefined ontology for each information source exists. Consequently, new information sources can easily be added and removed. But the comparison of the heterogeneous ontologies leads to many homonym, synonym, etc. problems, because the ontologies use their own vocabulary. In SKC (Mitra et al., 1999) the ontology of each source is described by graphs. Graph transformation rules are used to transport information from one ontology into another ontology (Mitra et al., 2000). These rules can only solve the schematic heterogeneities between the ontologies. In MESA (Wache et al., 1999) the third hybrid approach is used. Each source is related to its source ontology. In order to make the source ontologies comparable, a common global vocabulary is used, organized in a common domain ontology. This hybrid approach provides the biggest flexibility because new sources can easily be integrated and, in contrast to the decentralized approach, the source ontologies remain comparable. In the next section we will describe how ontologies can help to solve heterogeneity problems.

## 2.2 References

[1]. [Aben, 1993] Aben, M. (1993). Formally specifying reusable knowledge model components. Knowledge Acquisition Journal, 5:119–141.
[2]. [Bergamashi et al., 1999] Bergamashi, Castano, Vincini, and Beneventano (1999). Intelligent techniques for the extraction and integration of heterogeneous information. In Workshop Intelligent Information Integration, IJCAI 99, Stockholm, Sweden.
[3]. [Borgida and Patel-Schneider, 1994] Borgida, A. and Patel-Schneider, P. (1994). A semantics and complete algorithm for subsumption in the classic description logic. JAIR, 1:277–308.
[4]. [Brachman, 1977] Brachman, R. (1977). What's in a concept: Structural foundations for semantic nets. International Journal of Man-Machine Studies, 9:127–152.
[5]. [Brickley and Guha, 2000] Brickley, D. and Guha, R. (2000). Resource description framework (rdf) schema specification 1.0. Technical Report PR-rdf-schema, W3C. http://www.w3.org/TR/2000/CR-rdf-schema- 20000327/.
[6]. [Chawathe et al., 1994] Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y.,
[7]. Ullman, J., and Widom, J. (1994). The TSIMMIS Project: Integration of Heterogeneous Information Sources. In Proceedings of IPSJ Conference, pages 7– 18.
[8]. [Collet et al., 1991] Collet, C., Huhns, M. N., and Shen, W.-M. (1991). Resource integration using a large knowledge base in carnot. IEEE Computer, 24(12):55–62.
[9]. [Genesereth and Fikes, 1992] Genesereth, M. and Fikes, R. (1992). Knowledge interchange format version 3.0 reference manual. Report of the Knowledge Systems Laboratory KSL 91-1, Stanford University.
[10]. [Gruber, 1991] Gruber, T. (1991). Ontolingua: A mechanim to support portable ontologies. KSL Report KSL-91-66, Stanford University.
[11]. [Gruber, 1993] Gruber, T. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2).
[12]. [Guarino, 1998] Guarino, N. (1998). Formal ontology and information systems. In Guarino, N., editor, FOIS 98, Trento, Italy. IOS Press.
[13]. [Guarino and Giaretta, 1995] Guarino, N. and Giaretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarification. In Mars, N., editor, Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pages 25–32. Amsterdam.
[14]. [Horrocks, 1999] Horrocks, I. (1999). FaCT and iFaCT. In (Lambrix et al., 1999), pages 133–135.
[15]. [Jasper and Uschold, 1999] Jasper, R. and Uschold, M. (1999). A framework for understanding and classifying ontoogy applications. In Proceedings of the 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop. University of Calgary/Stanford University.
[16]. [Kim et al., 1995] Kim, W., Choi, I., Gala, S., and Scheevel, M. (1995). Modern Database: The Object Model, Interoperability, and Beyond, chapter On Resolving Schematic Heterogeneity in Multidatabase Systems, pages 521–550. ACM Press / Addison- Wesley Publishing Company.
[17]. [Kim and Seo, 1991] Kim, W. and Seo, J. (1991). Classifying schematic and data heterogeinity in multidatabase systems. IEEE Computer, 24(12):12–18.
[18]. [Lambrix et al., 1999] Lambrix, P., Borgida, A., Lenzerini, M., M¨oller, R., and Patel-Schneider, P., editors (1999). Proceedings of the International Workshop on Description Logics (DL'99).
[19]. [Lenat, 1998] Lenat, D. (1998). The dimensions of context space. Available on the web-site of the Cycorp Corporation. (http://www.cyc.com/publications).
[20]. [Mena et al., 1996] Mena, E., Kashyap, V., Illarramendi, A., and Sheth, A. (1996). Managing multiple information sources through ontologies: Relationship between vocabulary heterogeneity and loss of information. In Baader, F., Buchheit, M., Jeusfeld, M. A., and Nutt, W., editors, Proceedings of the 3rd Workshop Knowledge Representation Meets Databases (KRDB '96).
[21]. [Mitra et al., 1999] Mitra, P., Wiederhold, G., and Jannink, J. (1999). Semi-automatic integration of knowledge sources. In Fusion '99, Sunnyvale CA.
[22]. [Mitra et al., 2000] Mitra, P., Wiederhold, G., and Kersten, M. (2000). A graph-oriented model for articulation of ontology interdependencies. In Proc. Extending DataBase Technologies, EDBT 2000, volume Lecture Notes on Computer Science, Konstanz, Germany. Springer Verlag.
[23]. [Naiman and Ouksel, 1995] Naiman, C. F. and Ouksel, A. M. (1995). A classification of semantic conflicts in heterogeneous database systems. Journal of Organizational Computing, pages 167–193.
[24]. [OMG, 1992] OMG (1992). The common object request broker: Architecture and specification. OMG Document 91.12.1, The Object Management Group. Revision 1.1.92.

[25]. [Papakonstantinou et al., 1996] Papakonstantinou, Y., Garcia-Molina, H., and Ullman, J. (1996). Medmaker: A mediation system based on declarative specifications. In International Conference on Data Engineering, pages 132–141, New Orleans.

[26]. [Schreiber et al., 1994] Schreiber, A., Wielinga, B., Akkermans, H., Velde, W., and Anjewierden, A. (1994). Cml the commonkads conceptual modeling language. In et al., S., editor, A Future of Knowledge Acquisition, Proc. 8th European Knowledge Acquisition Workshop (EKAW 94), number 867 in Lecture Notes in Artificial Intelligence. Springer.

[27]. [Stuckenschmidt and Wache, 2000] Stuckenschmidt, H. and Wache, H. (2000). Context modelling and transformation for semantic interoperability. In Knowledge Representation Meets Databases (KRDB 2000). to appear.

[28]. [van Harmelen and Fensel, 1999] van Harmelen, F. and Fensel, D. (1999). Practical knowledge representation for the web. In Fensel, D., editor, Proceedings of the IJCAI'99 Workshop on Intelligent Information Integration.

[29]. [W3C, 1998] W3C (1998). Extensible markup language (xml) 1.0. W3C Recommendation. [W3C, 1999] W3C (1999). Resource descrition framework (rdf) schema specification. W3C Proposed Recommendation.

[30]. [Wache et al., 1999] Wache, H., Scholz, T., Stieghahn, H., and K¨onig-Ries, B. (1999). An integration method for the specification of rule–oriented mediators. In Kambayashi, Y. and Takakura, H., editors, Proceedings of the International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), pages 109–112, Kyoto, Japan.

[31]. [Wiederhold, 1992] Wiederhold, G. (1992). Mediators in the architecture of future information systems. IEEE Computer, 25(3):38–49. standard reference for mediators. [Wiederhold, 1999] Wiederhold, G. (1999). Mediation to deal with heterogeneous data sources. In Vckovski, A., editor, Interop99, volume 1580 of Lecture Notes in Computer Science, Zurich, Switzerland. Springer.

[32]. [Wiener et al., 1996] Wiener, J., Gupta, H., Labio, W., Zhuge, Y., Garcia-Molina, H., and Widom, J. (1996). Whips: A system prototype for warehouse view maintenance. In Workshop on materialized views, pages 26–33, Montreal, Canada.

# 3 Terms and definitions

This section lists the applicable documents

| Ref | Document Title | Issue/Date |
|---|---|---|
| TA | pSHIELD Technical Annex | 1 |
| M0.1 | Formalized Conceptual Models of the Key pSHIELD Concepts | 1 |
| M0.2 | Proposal for the aggregation of SPD metrics during composition | 1 |
| D2.1.1 | pSHIELD Systems requirements and specifications | 1 |
| D2.2.1 | pSHIELD metrics definition | Draft |

## 3.1 SPD Dictionary

A comprehensive dictionary of the SPD concepts is provided by the project glossary, that is a relevant and essential step in the ontology building process (refer to the section " Building the glossary."

# 4 Semantic Technologies

## 4.1 Ontologies

The World Wide Web represents a huge repository of information which can be retrieved and used. Unfortunately, information is represented with no meaning associated, since the meaning of retrieved information can be (re-)established only in the process of interpreting the information by humans. As a result, information scattered throughout the current (and traditional) version of the web is almost totally useless for software, non-human users (machine agents).

In attempt to respond to this situation, the term "Semantic Web" was coined by Tim Berners-Lee and his colleagues [24] referring to a "web for machines" as opposed to a web to be read by humans. In their understanding, "*The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation*."

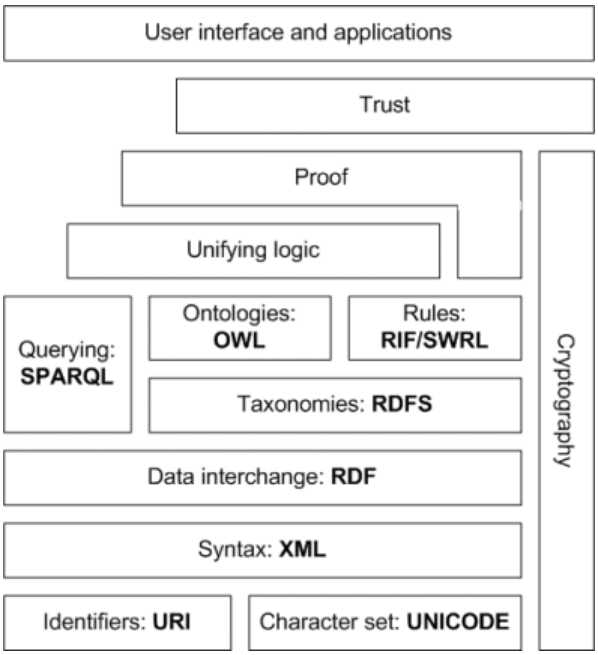


**Figure 4.1 Semantic web layers**

The Semantic Web is the opportunity for providing, finding and processing information via the Internet with the help of (machine) agents which are capable of dealing with the semantics of the information. The idea is to transform information into something meaningful to actors who seek to enhance their knowledge in order to satisfy a specific concern or accomplish a specific task related to their particular context.

The vision of the Semantic Web is based on the employment of semantic technologies that allow the meaning of information and the meaning of associations between information to be known and processed at execution time. To fulfill the promises and enable semantic technologies to work, there must be a *knowledge model* (of some part) of the world that is used to provide meaning to information to be processed within an application. The knowledge model has the form of a semantic model which differs from other kind of models [25]:

- *Connections*: meaning is represented through *connectivity*. The meaning of terms, or concepts, in the model is established by the way they connect to each other.
- *Multiple views*: a semantic model expresses multiple viewpoints and several interconnected models could be used to represent different aspects.

- *Sharing*: semantic models represent knowledge about the world in which systems operate and are *shared* across applications.

- *Reasoning capability*: use of a model is often referred to as “reasoning over the model”. The reasoning can range from a very simple process of graph search to intricate inferencing.

Although the role of a semantic model can be played by a simple taxonomy, nowadays use of semantically richer ontologies (*ontological models*) dominates.

Although most common definition states that “An ontology is a specification of conceptualisation”, a more detailed definition can make things a bit clearer: for our purpose, and a practical point of view, *an ontology is a network of connections defining explicit relationships (named and differentiated) between concepts*.

New knowledge can be *derived* by examining the connections between concepts. Simple ontologies are just networks of connections, richer ontologies include rules and constraints governing these connections. The semantic web is not so much a technology as an infrastructure, enabling the creation of meaning through standards, mark-up languages, and related processing tools. To represent ontologies in a formal way, several languages can be used. The Semantic Web principles are implemented in the layers of Web technologies and standards. The most common ontology languages are briefly described below (all the presented languages are supervised by the World Wide Web Consortium [26].

At the beginning, the idea of the semantic web tried just to enhance the current version of the web. It started out with a document oriented approach. The basic idea was to make web pages identifiable by computers as information resources carrying not only information (readable only by humans) but the meaning of this information as well. The meaning was added by *annotating* these pages with semantic mark-up. Ontologies here define a shared conceptualization of the application domain at hand and provide the basis for defining metadata, that have a precisely defined semantics, and that are therefore machine-processable. The idea of semantically annotated web pages with machine-interpretable description of their content aimed at automated processes of searching and accessing pages enabling human users to better utilise information stored on the web. In addition to human users, the semantic web enables the participation of non-human users as well. These machine agents do not need to deal with whole web pages. Instead of this, they exchange chunks of data with each other. Although they can communicate using different protocols, technology of web services has become a dominant way of communication with and using services of applications in the web environment.

Formerly, the problem of *interoperability* of different agents was tackled by translation technologies, most commonly by field to field mapping. The semantic web enables agents to exchange chunks of data with meaning associated to the data using semantic technologies. Advanced applications can use ontologies to relate the information to a semantic model of a given domain. In this way semantic technologies offer a new way to integrate different applications. Nowadays, the field of semantic interoperability is the most addressed problem connected with the idea of the semantic web.

## 4.2 Ontology representations

A number of possible languages can be used to represent ontologies; many of them evolved from creation of ontology construction methodologies. The Open Knowledge Base Connectivity (OKBC) [1] model and languages like “Knowledge Interchange Format (KIF)” [2] are examples that have become the bases of other ontology languages.

Several languages use frame logic which is basically an object-oriented approach defining frames and attributes (classes and properties). There are also several languages based on description logic, e.g. Loom [3], DAML+OIL [4], or later evolved Web Ontology Language (OWL) [5] standard.

Representation languages can be divided in terms of different abstraction levels used to structure the representation itself:

a) *Extensional level*: the model is formulated by specifying every object from the domain.

b) *Intensional level*: objects are defined by means of (necessary and sufficient) conditions for belonging to the domain.

c) *Meta-level*: concepts from intensional level are abstracted, higher level concepts are specified, and previous concepts are seen as instances of new meta-concepts.

Some issues emerge from analysis of ontology representation, concerning the scope and modality of context expression: these criteria consider the basic formal nature of languages and that various languages deal with the representation of incomplete information in different way

1) We can express:

   a) *Class and relations*: languages aiming at representing objects, classes and relations.

   b) *Actions and processes*: languages that provide specialized representation structures for describing dynamic characteristics of the domain, such as actions, processes, and workflows (they usually can represent static aspects of domain too, but only in elementary level).

   c) *Everything*: languages that may be used for any kind of contexts and applications.

2) The context can be expressed in the following ways:

   a) *Programming languages*: allow representation and manipulation of data in several ways and according to various paradigms, leading to a cleaner separation between data structures and algorithms that handle them. Object oriented paradigm is preferred in recent years. This approach is generally associated with a number of concepts, such as complex objects, object identity, methods, encapsulation, typing and inheritance. Example can be language F-logic [6], logical formalism that tries to capture the features of object-oriented approaches to computation and data representation. F-Logic forms the core of systems such as Ontobroker [7].

   b) *Conceptual and semantic database models:* semantic (or conceptual) models were introduced as schema design tools. Examples of proposed semantic data models are ER and Extended ER data model, FDM (Functional data model), SDM (Semantic Data Model). Semantic models provide more powerful abstractions for the specification of databases.

   c) *Information system / software formalisms*: here belong different formalisms for information system design, especially in object-oriented design. Most widely used formalism is Unified Modelling Language (UML). UML was designed for human-to-human communication of models for building systems in object-oriented programming languages. Over the years its use has been extended to a variety of different aims, including the design of databases schemas, XML document schemas, and knowledge models.

   d) *Logic-based*: very important class of languages is based on logic. Such languages express a domain-ontology in terms of the classes of objects that are of interest in the domain, as well as the relevant relationships holding among such classes. These languages have a formal well-defined semantics. Three different types of logic-based languages exist – languages based on

first-order predicate logic (e.g. KIF [2]), languages based on description logics (e.g. OWL [5]), and process-action specificationlanguages (e.g. PSL [8]).

e) *Frame-based*: frame is a data structure that provides a representation of an object or a class of objects or a general concept or predicate. Some systems define only a single type of frame, other have two or more types, such as class frames and instance frames. The *slots* of a frame describe attributes of represented concept. They may also have other components in addition to the slot name, value and value restrictions, for instance the name of a procedure than can be used to compute the value of the slot – facets. Frames are usually organized into taxonomies. Through taxonomic relations, classes may be described as specializations of more generic classes with inheritance capability. Frame-based ontology languages were often used in many knowledge-based applications, like Ontolingua [9], OCML [10], OKBC [11] or XOL [12].

f) *Graph-based*: formalisms based on various kinds of graph based or graph-oriented notations. Semantic networks [13] and conceptual graphs [14] originated from the Artificial Intelligence community. OML/CKML (Conceptual Knowledge Markup Language) [15] is a framework and markup language for knowledge and ontology representation based on conceptual graphs. Topic Maps [16] are recent proposal originated from the XML community.

g) *XML-related formalisms*: XML [17] is a tag-based language for describing tree structures with a linear syntax and it is a standard language for exchange of information in the Web. Given the popularity of XML in exchange of information, XML-related languages have been considered as suitable for ontology representation. Important languages are based on Resource Description Framework (RDF) [18]. These provide a foundation for processing metadata about documents.

3) We can interpret expression in the following ways:

a) *Single model*: ontology should be interpreted in such a way that only one model of the corresponding logical theory is a good interpretation of the formal description.

b) *Several models*: ontology should be interpreted as specifying what we know about the domain with the reservation that the amount of knowledge we have about the domain can be limited (e.g. first-order logic based languages).

## 4.3 Semantic Web Ontology languages

Description Logics (DLs) are a family of logic-based knowledge representation formalisms designed to represent and reason about the knowledge of an application domain in a structured and well understood way. The basic notions in DLs are *concepts and roles*, which denote sets of objects and binary relations, respectively. Most of today's semantic web ontology languages are DL-based. Also many of them are XML-related, or they possible XML notation. Several ontology languages have been designed for use in the web. Among them, the most important are OIL [19], DAML-ONT [20] and DAML+OIL [21]. More recently, a new language, OWL [5], is being developed by the World Wide Web Consortium (W3C) Web Ontology Working Group, which had to maintain as much compatibility as possible with pre-existing languages and is intended to be proposed as the standard Semantic Web ontology language. The idea of the semantic Web is to annotate web pages with machine-interpretable description of their content. In such a context, ontologies are expected to help automated processes to access information, providing structured vocabularies that explicate the relationships between different terms.

**Extensible Markup Language (XML)** [17] was widely accepted and used as a convenient information representation and exchange format. *XML itself don't carry semantics*, but is serves as the base syntax for the leading ontology languages that we shall survey. Later additions like XML-DTD (Document Type Definition) and XML-Schema, added some syntactic rules like enumerations, cardinality constrains, and data types, but still lacked even simple semantics like inheritance. The purpose of XML Schema is therefore to declare a set of constraints that an XML document has to satisfy in order to be validated. With respect to DTD, however, XML Schema provides a considerable improvement, as the possibility to define much more elaborated constraints on how different part of an XML document fit together, more sophisticated nesting rules, data-typing. Moreover, XML-Schema expresses shared vocabularies and allows machines to carry out rules made by people. Among a large number of other rather complicated features.

**Resource Description Framework (RDF)** [18] is a standard way for defining of simple descriptions. RDF is for semantics - a clear set of rules for providing simple descriptive information. RDF enforces a strict notation for the representation of information, based on resources and relations between them. As referred to in its name, RDF strength is in its descriptive capabilities, but is still lacks some important features required in an ontology language such as inferences for example. However, ontology languages built on top of RDF as a representation and description format. The RDF data model provides three object types: *resources*, *properties*, and *statements*. Resource may be either entire Web page, a part of it, a whole collection of pages or an object that is not directly accessible via the Web, property is a specific aspect, characteristic attribute, or relation used to describe a resource, statement is a triple consisting of two nodes and a connecting edge. These basic elements are all kinds of RDF resources. According to the latter description, a subject is a resource that can be described by some property. The predicate defines the type of property that is being attributed. Finally, the object is the value of the property associated with the subject.

**RDF Schema (RDFS)** [22] enriches the basic RDF model, by providing a vocabulary for RDF, which is assumed to have certain semantics. Predefined properties can be used to model instance of and subclass of relationships as well as domain restrictions and range restrictions of attributes. Indeed, the RDF schema provides modelling primitives that can be used to capture basic semantics in a domain neutral way. That is, RDFS specifies metadata that is applicable to the entities and their properties in all domains. The metadata then serves as a standard model by which RDF tools can operate on specific domain models, since the RDFS meta-model elements will have a fixed semantics in all domain models. RDFS provides simple but powerful modelling primitives for structuring domain knowledge into classes and sub classes, properties and sub properties, and can impose restrictions on the domain and range of properties, and defines the semantics of containers.

**Web Ontology Language (OWL)** The next layer in the Semantic Web architecture is Web Ontology Language (OWL) [5], a language for Web ontologies definition and instantiation. OWL enhances RDF vocabulary for describing properties and classes: relations between classes (e.g. subclasses), cardinality, equality, richer typing of properties, characteristics of properties (e.g. symmetry) and instances. OWL is the W3C recommendation for ontology definition, but other standards also support similar characteristics (DAML+OIL). Several tools support modelling with OWL and DAML+OIL. The OWL language also provides three increasingly expressive sublanguages: *OWL Lite*, *OWL DL*, and *OWL Full*, each offers a different level of expressiveness at the trade-off for simplicity, thus offering a suitable sub language parts available for use according to expressibility needs. There also exists OWL based enhancement to web

services oriented languages, aiming to handle semantic descriptions of such services. OWL-S [23] is framework for containing and sharing ontological description of the capabilities and characteristics of a Web service.

An **OWL-S** specification includes three sub-ontologies that define essential types of knowledge about a service – *service profile* describes the outlining interface and characteristics of the service, a *process profile* defines the control flow of the service and the *service grounding* provides mapping with communication-level protocols. OWL-S has similar characteristics with a number of related protocols. The popularity of OWL-S in the Semantic Web community, as Web services description language, adds to the attractiveness of the language.

## 4.4 References

[1] Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A., “Methodologies, tools and languages for building ontologies. Where is the meeting point?” Data&Knowledge Engineering 46, pp. 41-64, 2003.

[2] Genesereth, M.R., Fikes, R.E., “Knowledge Interchange Format. Version 3.0. Reference Manual.” Stanford University, 1992.

[3] MacGregor, R., Bates, R., “The Loom Knowledge Representation Language”. Technical Report ISIRS-87-188, USC Information Sciences Institute, Marina del Rey, CA, 1987.

[4] Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D., Patel-Schneider, P.F., Stein, L.A., DAML+OIL (March 2001) Reference Description. W3C Note, 2001.

[5] McGuinness, D., van Harmelen, F., “OWL Web Ontology Language Overview”. W3C Recommendation, 2004.

[6] Kifer, M., Lausen, G., F-Logic: “A Higher-Order language for Reasoning about Objects, Inheritance, and Scheme”, in Proc. of SIGMOD Conference 1989, pp. 134-146, 1989.

[7] Fensel, D., Decker, S., Erdmann, M., Studer, R., Ontobroker: “The Very High Idea”, in Proceedings of the 11th International Flairs Conference (FLAIRS-98), Florida, 1998.

[8] Knutilla, A., et. al., “Process Specification Language: Analysis of Existing Representations”, NISTIR 6133, National Institute of Standards and Technology, Gaithersburg, MD, 1998.

[9] Gruber, T.R., A Translation Approach to Portable Ontology Specifications, in Knowledge Acquisition, 5(2), 1993.

[10] Motta E., Reusable Components for Knowledge Modelling. IOS Press, Amsterdam, Netherlands, 1999

[11] Chaudhri, V., Farquhar, A., Fikes, R., Karp, P., Rice, J. (1998), OKBC: A programming foundation for knowledge base interoperability, in Proceedings of AAAI’98, pp. 600-607, 1998.

[12] Karp, R., Chaudhri, V., Thomere, J., XOL: An XML-Based Ontology Exchange Language, Technical report, 1999.

[13] Sowa, J.F., Semantic networks, available online http://www.jfsowa.com/pubs/semnet.htm, 2002.

[14] Sowa, J.F., Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[15] R.E. Kent., Conceptual Knowledge Markup Language: The Central Core’, in: Twelfth Workshop on Knowledge Acquisition, Modeling and Management, 1999.

[16] TopicMaps.Org Authoring Group, XML Topic Maps (XTM) 1.0 TopicMaps.Org specification, 2001

[17] Yergeau, F., Bray, T., Paoli, J., Sperberg-McQueen, J.M., Maler, E., Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation, 2004.

[18] Manola, F., Miller, E., McBride, B., RDF Primer. W3C Recommendation, 2004.

[19] Fensel, D., Horrocks, I., van Harmelen, F., McGuiness, D., Patel-Schneider, P.F., OIL: Ontology Infrastructure to Enable the Semantic Web. IEEE Intelligent Systems, 16 (2), 2001.

[20] Stein, L.A., Connolly, D., McGuinness, D.,Annotated DAML Ontology Markup. Initial draft, http://www.daml.org/2000/10/daml-walkthru, 2000.

[21] Connolly, D., van Harmelen, F., Horrocks, I., McGuinness, D., Patel-Schneider, P.F., Stein, L.A., DAML+OIL (March 2001) Reference Description. W3C Note, 2001.

[22] Brickley, D., Guha, R.V., McBride, B., RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 2004.

[23] Martin, D., et al., Bringing Semantics to Web Services: The OWL-S Approach, in Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), July 6-9, 2004, San Diego, California, USA.

[24] Berners-Lee T., Hendler J., Lassila O.: "The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities". Scientific American, 284 (5), pp. 34-43, May 2001. http://www.sciam.com/

[25] Semantic Technology. TopQuadrant Technology Briefing, March 2004, http://www.topquadrant.com/tq_white_papers.htm

[26] The World Wide Web Consortium. http://www.w3.org/

# 5 Methodology for the design of pSHIELD Ontology

## 5.1 Overview

A sound framework for an effective exploitation of ontologies in ES should provide a formal methodology to build, verify and maintain an ontology.

As a matter of fact, in order to make available large-scale, high quality domain ontologies, effective and usable methodologies are needed to facilitate the process of ontology building. Ontology building is a task that pertains to the ontology engineers, that we classify as knowledge engineers (KE) and domain experts (DE). Even though automatic ontology learning methods (such as text mining) significantly support ontology engineers, speeding up their task, there is still the need of a significant manual effort, in the integration and validation of the automatically generated ontology.

Existing ontology building methods only partly are built capitalizing the large experience that can be drawn from widely used standards in other areas, like software engineering and knowledge representation. In this project, we shall embrace a methodology for ontology building derived from a well-established and widely used software engineering process, the Unified Software Development Process

This is a novel approach to large-scale ontology building that takes advantage of the Unified Process (UP) and the Unified Modeling Language (UML). This choice makes ontology building an easier task for modellers familiar with these techniques: each phase of the method fits in the UP, providing a number of consolidated steps that guide the process of ontology development. UML has been already shown to be suitable to this end, confirming its nature of rich and extensible language. What distinguishes the UP and the ontology building methodology from the other methodologies, respectively for software and ontology engineering, is their use-case driven, iterative and incremental nature.

The metholdology is use-case driven since it does not aims at building generic domain ontologies, but its goal is the production of ontologies that serve its users, both humans and automated systems (e.g. semantic web services, intelligent agents, etc.), in a well defined application area. Use cases are the first diagrams that drive the exploration of the application area, at the beginning of the ontology building process.

The nature of the process is iterative since each iteration allows the designer to concentrate on part of the ontology being developed, but also incremental, since at each cycle the ontology is further detailed and extended.

Following the UP, in the metholdology we have cycles, phases, iterations and workflows. Each cycle consists of four phases (inception, elaboration, construction and transition) and results in the release of a new version of the ontology. Each phase is further subdivided into iterations. During each iteration, five workflows (described in the next subsections) take place: requirements, analysis, design, implementation and test. Workflows and phases are *orthogonal* in that the contribution of each workflow to an iteration of a phase can be more or less significant: early phases are mostly concerned with establishing the requirements (identifying the domain, scoping the ontology, etc.), whereas later iterations result in additive increments that eventually bring to the final release of the ontology (Figure 5.1). Notice that, as illustrated in the figure, more than one iteration may be required to complete each of the four phases. This scheme follows faithfully the Unified Process.

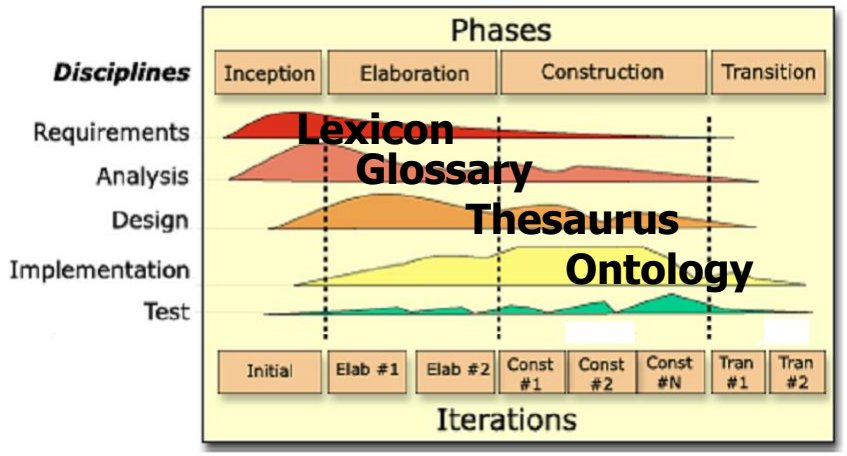


**Figure 5.1 Mapping onto workflows and phases of UP**

The first iterations (inception phase) are mostly concerned with capturing requirements and partly performing some conceptual analysis. Neither implementation nor test is performed. During subsequent iterations (belonging to the elaboration phase) analysis is performed and the fundamental concepts are identified and loosely structured. This may require some design effort and it is also possible that the modellers provide a preliminary implementation in order to have a small skeletal blueprint of the ontology, but most of the design and implementation workflows pervade iterations in the construction phase. Here some additional analysis could be still required aiming at identifying concepts to be further added to the ontology. During the final iterations (transition phase), testing is heavily performed and the ontology is eventually released. In parallel, the material necessary to start the new cycle, that will produce the next version of the ontology, is collected. As shown in Figure 5.1, and detailed in the next sections, at each iteration different workflows come into play and a richer and more complete version of the target ontology is produced. The incremental nature of the methodology requires first the identification of the relevant terms in the domain, gathered in a lexicon; then the latter is progressively enriched with definitions, yielding a glossary; adding to it the basic ontological relationships allows a thesaurus to be produced, until further enrichments and a final formalization produces the sought reference ontology.

In the following subsections each ontology building workflow is described in detail.

## 5.2 The Requirements Workflow

Requirements capture is the process of specifying the semantic needs and the knowledge to be encoded in the ontology. The essential purpose of this workflow is to reach an agreement between the modellers, the knowledge engineers, and the final users, represented by the domain experts. During the first meetings, knowledge engineers and domain experts establish the guidelines for building the ontology. The first goal is the identification of the objectives of the ontology users. To this end, it is necessary to:

- determining the domain of interest and the scope, and
- defining the purpose.

These objectives are achieved by:

- writing one or more storyboards
- creating an application lexicon
- identifying the competency questions, and
- the related use cases.

### 5.2.1 Determining the domain of interest and the scope.

Delimiting the domain of interest is a fundamental step to be performed, aiming at focusing on the appropriate fragment of reality to be modelled. If the domain is large, one or more sub-domains may also be determined. Defining the scope of the ontology consists in the identification of the most important concepts to be represented, their characteristics and granularity. For this purpose, a set of ontological commitments are required, bringing some part of the domain into focus at the (required and expected) expense of other parts. These ontological commitments are not incidental: they provide a guidance in deciding what aspects of the domain are relevant and what to ignore. The ontological commitment can be seen as “a mapping between a language and something which can be called an ontology”. This allows one to preliminarily identify terms as representatives of ontology concepts. Usually at this stage modellers have only a vague idea of the role each concept will play, i.e., their semantic interconnections, within the ontology. If necessary, they can informally annotate these ideas for further development during subsequent iterations.

### 5.2.2 Defining the purpose (or motivating scenario)

The reason for having an ontology, its intended uses, and the kinds of users must be established. In the pSHIELD, the goal of the ontology is to provide a support for semantic interoperability between entities at different layers.In particular, we envisage three basic uses of the developed ontology:

- Ontology-based search and retrieval of services (discovery);
- Ontology-based reconciliation of data messages exchanged between entities
- Reasoning about configuration and SPD metrics, as defined in the ontology in terms of composition rules, to (dinamically) find new configuratione of the system at run time or design time.

### 5.2.3 Writing a storyboard.

In this step domain experts are asked to write a panel or series of panels outlining the sequence of the activities that take place in a particular scenario. Storyboards model contexts and situations is a narrative way that can be used in the next steps.

### 5.2.4 Creating the application lexicon.

The storyboard can be also used to extract the terminology of domain experts, building a preliminary version of the application lexicon. This task can be supported by using some automatic tools to extract knowledge from documents, such as OntoLearn or other methodologies. An application lexicon is more specific than a domain lexicon, but both are necessary to accomplish an effective ontology.

In pSHIELD, the application lexicon has been mainly extracted by the applicable documents listed in section 3.

### 5.2.5 Identifying the competency questions.

Competency questions are questions an ontology must be able to answer. They are identified through interviews with domain experts, brainstorming, an analysis of the document base concerning the domain, etc. The questions do not generate ontological commitments, but are used during the *test workflow* to evaluate the ontological commitments that have been made. The competency questions are more significant when the use of the ontology will be mainly for querying and discovering rather than for reconciliation.

### 5.2.6 Use-case identification and prioritization.

In this methodology. competency questions are taken into account through use-case models. A use-case model, that contains a number of use case diagrams, serves as a basis to reach an agreement between the users (i.e., who require the ontology) and the modellers. In the context of ontologies, use cases correspond to *knowledge paths* through the ontology to be followed for answering one or more competency questions. Although they are to be specified during the analysis and design workflows, it is necessary to prioritize and package (i.e. group) them during requirements. The result will help dictate which use cases the team should focus on during early iterations, and which ones can be postponed.

## 5.3 The Analysis Workflow

The conceptual analysis consists of the refinement and structuring of the ontology requirements identified in previous section. The ontological commitments derived from the definition of scope are extended, by reusing existing resources and through concept refinement. The application lexicon will be enriched through the definition of a more general domain lexicon, then definitions will be added to produce the *Reference Glossary*.

### 5.3.1 Considering reuse of existing resources: identification of the domain lexicon

The domain lexicon is defined as the terminology used in the domain of interest, extracted by analyzing a corpus of existing resources. The analysis is mainly based on external resources, such as documents, standards, glossaries, thesauri, legacy computational lexicons and available ontologies. This task, like in the case of the application lexicon, can be supported by automatic tools. The description of this activity adheres to the view of linguistic ontology in which concepts, at least the lower and intermediate levels, are anchored to texts, i.e. they have a counterpart in natural language.

In order to build the Embedded Systems domain lexicon, in pSHIELD we have mainly considered as resources: the applicable documents listed in section 3.

A statistical analysis shall be done in a corpus of documents of reference to identify frequently used terms to be included in the domain lexicon. The domain experts shall decide to include, in this lexicon, all the terms present in, for instance, at least two standards. Some other terms, present in only one resource, can be included after approval from a wider panel of experts. After this activity, the domain lexicon shall contain a number of terms (including synonyms).

### 5.3.2 Modelling the application scenario using UML diagrams.

The goal of this activity is to model the application scenario and better specify the Use Case Diagrams, drawn in the requirement workflow, with the aid of Activity and Class Diagrams. UML diagrams represent a model of the application and will be used for the validation of the ontology. In principle, all the classes, actors, and activities modeled in UML must have a corresponding concept in the ontology.

### 5.3.3 Building the glossary.

A first version of a glossary of the domain of interest has to be built merging the application lexicon and the domain lexicon. During the merge of the two lexicons we can organize all the concepts in two major areas: the intersection area and the disjoint area (Figure 5.2). Then we use the following “inclusion policy”: the glossary should include all the concepts coming from the intersection area and, after the domain experts approval, some concept belonging to the disjoint area. The output is a reference lexicon that will grow into a glossary by associating one or more definitions to each term. The definitions should be selected from knowledgeable sources and agreed among domain experts.

Refer to “Annex 1 – pSHIELD Glossary” for the pSHIELD glossary built during this phase.

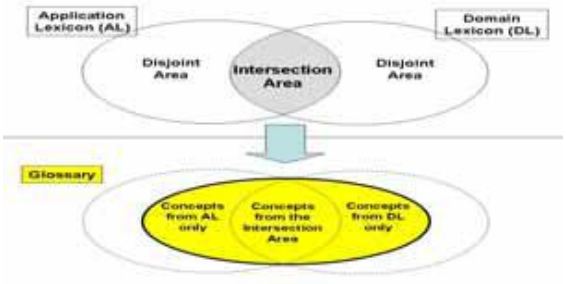


**Figure 5.2 Glossary building**

## 5.4 The Design Workflow

The main goal of this workflow is to give an ontological structure to the set of terms gathered in the Glossary. The refinement of entities, actors and processes identified in the analysis workflow, as well as the identification of their relationships, is performed during the design workflow.

### 5.4.1 Categorising the concepts.

Each concept is categorized by associating a “kind” to it. Such *kinds* should include the major ontological categories, according to proposals of *upper ontologies*, or *meta-ontologies*,

A partial outcome of this phase is provided by Table 1

| Concept | Relation | Related concept |
|---|---|---|
| Audit | Specializes | SPD Functionality |
| Authentication | Specializes | SPD Functionality |
| Availability | Specializes | SPD Attribute |
| Confidentiality | Specializes | SPD Attribute |
| Cryptographic | Specializes | SPD Functionality |
| Dependability | Specializes | SPD Concept |
| Error | Specializes | Threat |
| Failure | Specializes | Threat |
| Fault | Specializes | Threat |
| Forecasting | Specializes | Mean |
| Identification | Specializes | SPD Functionality |
| Integrity | Specializes | SPD Attribute |
| Maintainability | Specializes | SPD Attribute |
| Mean | Generalizes | Prevention |
| Mean | Generalizes | Tolerance |
| Mean | Generalizes | Forecasting |
| Mean | Generalizes | Removal |
| Prevention | Specializes | Mean |
| Reliability | Specializes | SPD Attribute |
| Removal | Specializes | Mean |
| Safety | Specializes | SPD Attribute |
| Security | Specializes | SPD Concept |
| SPD Attribute | Generalizes | Availability |
| SPD Attribute | Generalizes | Reliability |
| SPD Attribute | Generalizes | Integrity |
| SPD Attribute | Generalizes | Safety |
| SPD Attribute | Generalizes | Confidentiality |
| SPD Attribute | Generalizes | Maintainability |
| SPD Concept | Generalizes | Dependability |
| SPD Concept | Generalizes | Security |
| SPD Functionality | Generalizes | Identification |
| SPD Functionality | Generalizes | Authentication |
| SPD Functionality | Generalizes | Audit |
| SPD Functionality | Generalizes | Cryptographic |
| Threat | Generalizes | Fault |
| Threat | Generalizes | Error |
| Threat | Generalizes | Failure |
| Tolerance | Specializes | Mean |

**Table 1 pSHIELD concept categorization**

### 5.4.2 Refining the concepts and their relations.

At this stage, concepts are organised by introducing formal relations among them. between sets of synonyms identified in the previous phase. A first step consists in organizing the concepts in a taxonomic hierarchy through the generalization (i.e., kind-of or is-a) relation. To this end, three main approaches are known in the literature:

- top-down (from general to particular)
- bottom-up (from particular to general)
- middle-out (or combined), which consists in finding the salient concepts and then generalizing and specializing them. This approach is considered to be the most effective because concepts “in the middle” tend to be more informative about the domain.

The resulting taxonomy can be extended with other relations, i.e., part-of and association. The outcome of this step is Thesaurus, structured according the UML class diagram relations: generalization (IsA), aggregation (Part-Of) and association. In parallel, the actual UML diagrams can be built.

The detailed outcome of this phase can be found in the pSHIELD ontology (refer to section “Annex 2 – pSHIELD OWL Models”)

## 5.5 The Implementation Workflow

The purpose of this workflow is to perform the final building step, by formalize defining the actual ontology in a formal language. The structure of the ontology will be the one given in the enriched Thesaurus, but here the different elements will be formally represented. To this end, the Ontology Web Language (OWL) proposed by the W3C shall be adopted.

The outcome of this workflow is the implementation model, i.e., a reference ontology encoded in OWL.

The detailed outcome of this phase can be found in the pSHIELD ontology (refer to section “Annex 2 – pSHIELD OWL Models”)

## 5.6 TheTest Workflow

The test workflow allows to verify that the ontology correctly implements the requirements produced in the first workflow. We envisage two kinds of test.

The first concerns the coverage of the ontology with respect to the application domain. In particular, the domain experts are asked to semantically annotate the UML diagrams, representing the application scenario, with the ontology concepts. (This test is particularly relevant for ontologies used in ontology-based reconciliation of messages).

The second kind of test concerns the competency questions and the possibility to answer them by using concepts in the ontology. Such questions will trigger a traversal of the ontology that will produce proper concepts. Competency questions represent a good test for ontologies to be used in search and discovery of resources

This phase is considered out of the scope of pilot SHIELD project

# 6 pSHIELD Semantic Models

## 6.1 Introduction

The described methodology is the most valuable chain to produce ontology and meta-models for a specific scenario (in this case the context of Embedded Systems).
The problem is: given a clear procedure on how to build ontology, what are we supposed to describe in it?
Starting from that, we can affirm that the context is the one of Embedded Systems; in particular the more specific contest are the SPD functionalities provided by their interaction/composition.
The main objective of our approach with semantic models are:

1. the *abstraction* of the *real word* from a technology-dependent perspective into a technology-independent representation.

2. the *representation* of *functional properties* by means of ontology as well

3. the *identification* of the *relations* between real/structural and functional world.

So, as depicted in Figure 6.1, the problem of modeling SPD in the context of ES is reduced to the formulation of three different meta-models describing: i) structure, ii) functions, iii) relations between structure and functions.

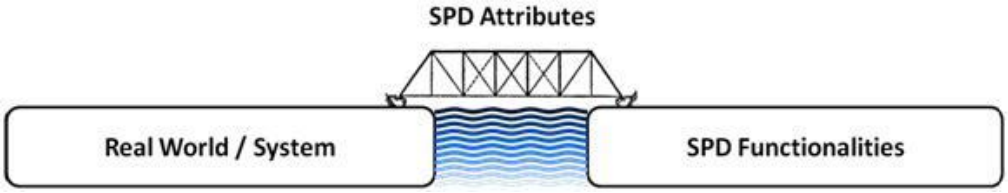


**Figure 6.1 Proposed approach to model SPD for ES**

The bridge has been built thank to the introduction of a third metamodel taking into account the atomic attributes that are impacted in this context and to map them in these two worlds, thus creating relations.

For each of the three models, a justification and a detailed description is provided in the prosecution of the document, while the global pSHIELD meta-model is depicted in Figure 6.2. All the relevant concepts (classes) are highlighted: they are the basic and exhaustive building blocks by which it is possible to reach the pSHIELD objectives. In particular:

- For the structural ontology these classes have been selected: `System`, `Element`, `Hardware`, `SPD Component`,
- For the functional ontology these classes have been selected: `SPDFunctionality`, `GeneralFunctionality`, `Connector`, `SPDCompositionSpecification`
- For the attribute ontology these classes have been selected: `SPDConcept`, `SPDAttribute`, `SPDThreat`, `SPDMean`


**Figure 6.2 - pSHIELD meta-model**

## 6.2 Structural System meta-model

pSHIELD is a framework composed by the interaction of dozens of interconnected Embedded Systems: they constitute the “physical world”. This world is made by hardware and software components, mapped on three layers: Node (the devices), Network (the interaction between devices) and Middleware (the software services that make the devices run). The first semantic model captures this concept.


**Figure 6.3 Structural Ontology**

### 6.2.1 pSHIELD Node Model

The structural ontology is the easier to model, because it is a simple description of the Embedded System component. It contains the hardware components and basic functionalities provided by the individual element and the related attribute, all in an SPD relevant environment. For example a node, in a first simplification, is composed by a memory a CPU, a battery and a transmission antenna; furthermore the CPU is characterized by the frequency and bit length and, in SPD relevant context, the possibility of performing hardware cryptography. By doing so, all the components constituting a complex system can be represented. The hardware element tha tconstitute a node are represented in Figure 6.5.


**Figure 6.4 pSHIELD Node Model**

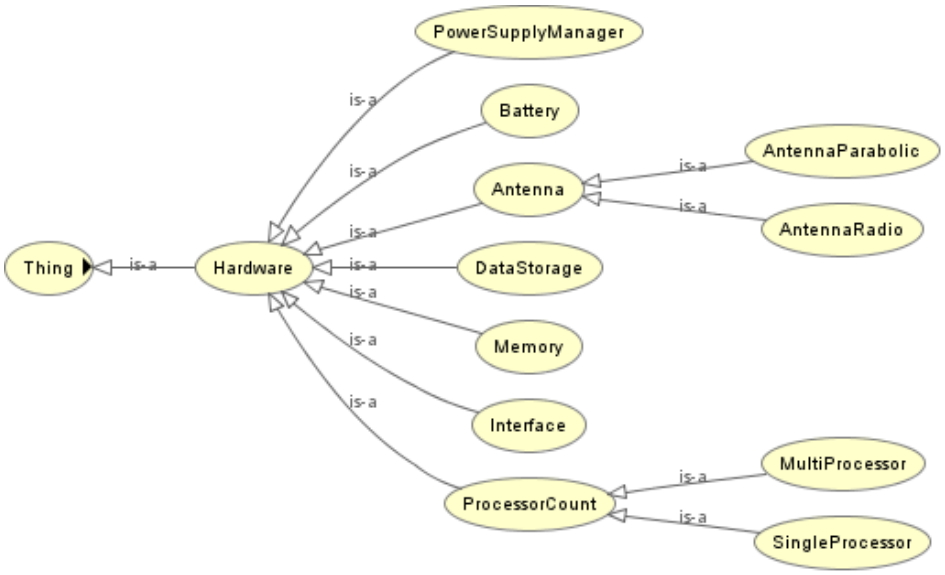


**Figure 6.5 Node hardware ontology**

### 6.2.2 pSHIELD Network Model

The network model is trivial, since the only relevant information are about the topology, the transmission medium and the typology.

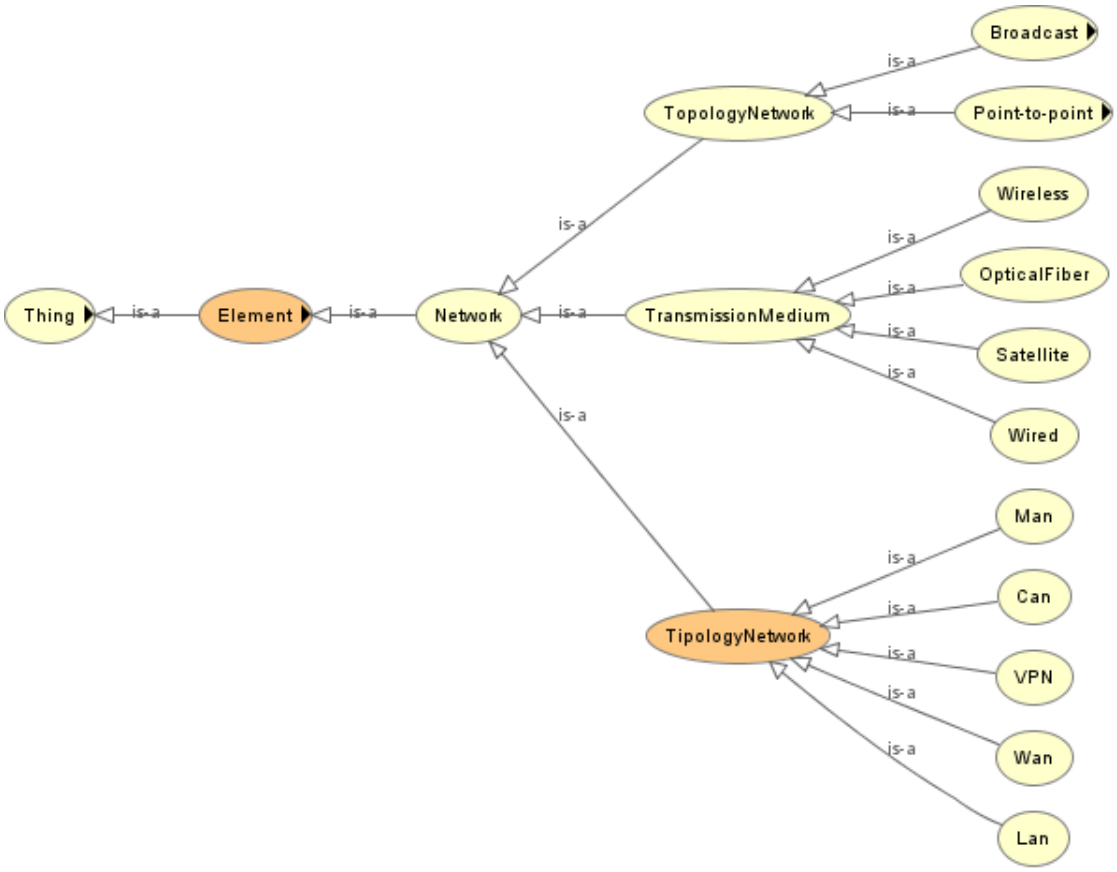


**Figure 6.6 pSHIELD Network Model**

### 6.2.3 pSHIELD Middleware and Overlay Model

The model for the Middleware services is the standard OWL-S to describe services. This choice has been done to maximize interoperability.

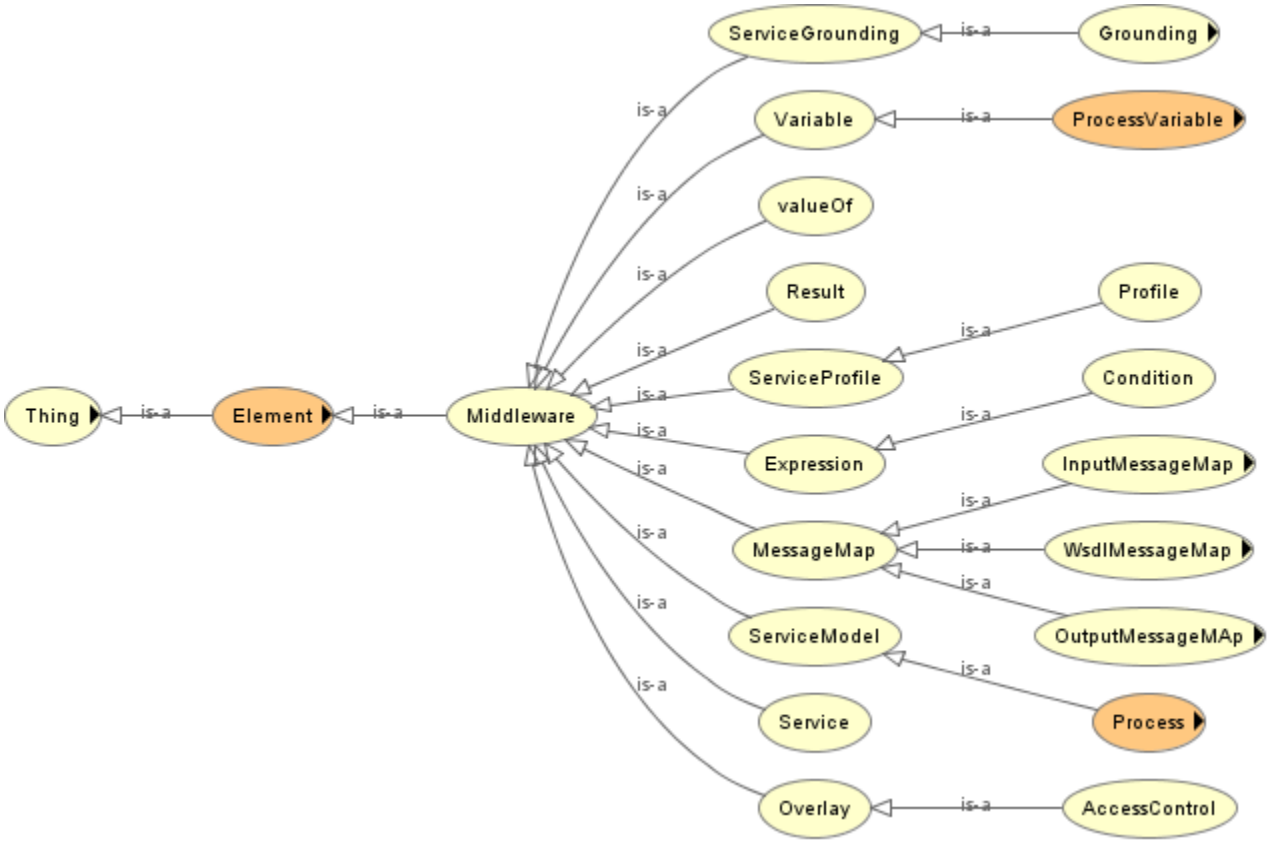


**Figure 6.7 pSHIELD Middleware Model**

## 6.3 SPD Functionalities meta-model

The SPD functionalities are modeled according to documents M0.1 and D2.2.1 in section Terms and definitions.

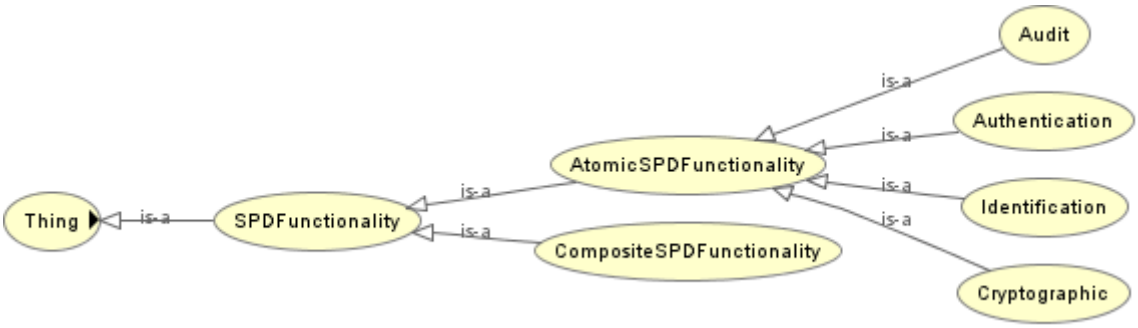


**Figure 6.8 SPD Functionality**

A `SPD Functionality` can be an `Atomic SPD Functionality` or a `Composite SPD Functionality`: the latter embodies a composite pattern (from a functional point of view it acts as a `SPD Functionality`) and represents an aggregation of other `SPD Functionality` (possibly composite) by means of a `Connector`.

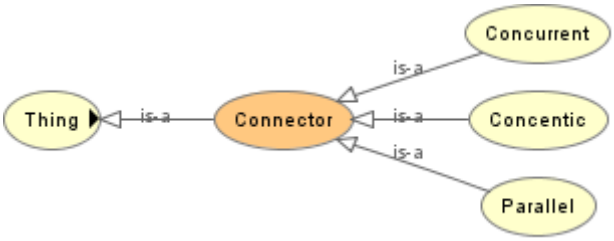


**Figure 6.9 Connector**

The `Connector` stands for a mean of aggregation, and specifies:

- one of the identified patterns: A connector provides a specification of the structure of the composition, by means of basic “control constructs” (whose names are reminiscent of control structures in programming languages). Specific instances of connector implement the conceived aggregations of pSHIELD SPD Metrics
- a `SPDCompositionSpecification`: a connector offers an analytical specification of the algorithm that composes the SPD status values of contributing SPD functionalities into the overall SPD status value

Composition is modeled after analogous Composite Processes in OWL for services (OWL-S). We can draw the following analogies:

| SPD Composition | Description | Analogous OWL-S Control Construct |
|---|---|---|
| Concentric | The SPD functionalities are composed serially | Sequence |
| Concurrent | The SPD functionalities are composed separately on different assets | Split |
| Parallel | The SPD functionalities protect at the same time the same assets | Choice |

**Table 2 SPD Composition modeling**

A `SPD Functionality` can be measured by a `SPDStatus`, that ultimately assesses the overall functionality of the ES from the viewpoint of SPD.

Figure 6.10 shows a snapshot of the concepts above-mentioned.

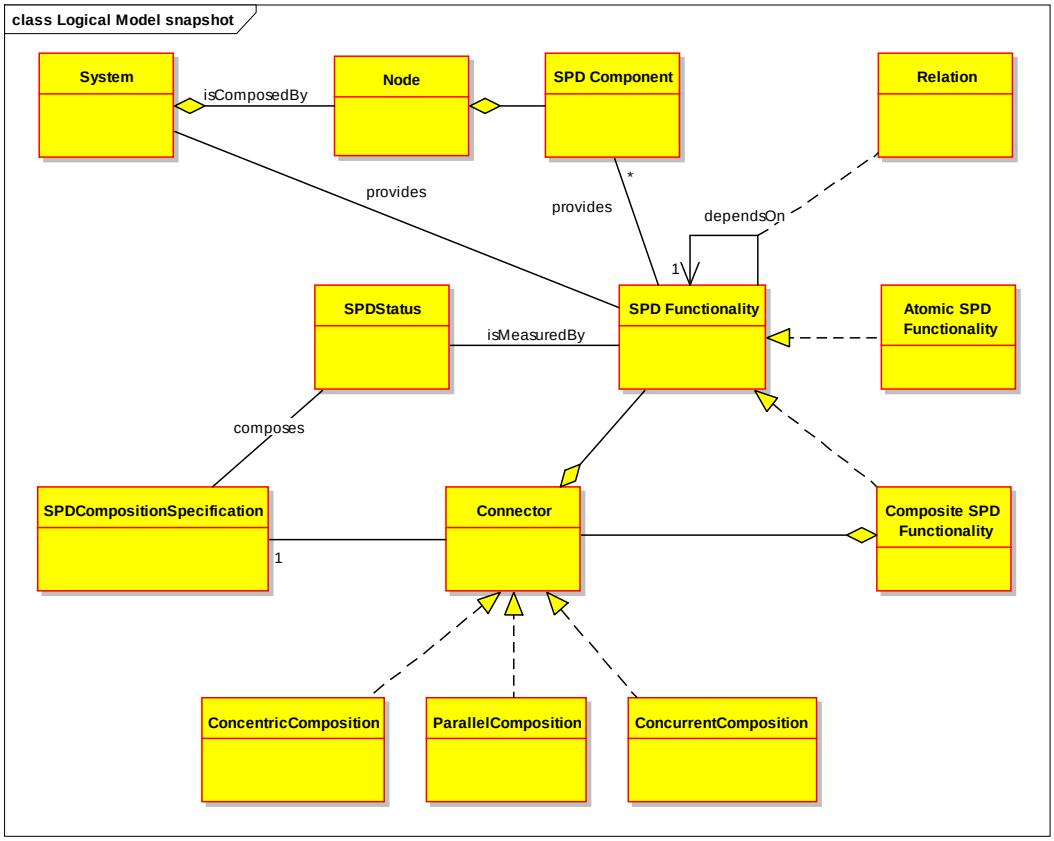


**Figure 6.10 snaphot of pSHIELD SPD functionality model**

## 6.4 SPD Attributes

SPD concepts, in terms of attributes (that assess them), threat (that affect them) and Mean (that increase them), are modeled according to documents D2.1.1 in documents in section Terms and definitions (see Figure 6.11).

Some SPD functionalities (shown in Figure 6.13) have been modeled after functionality taxonomy found in D2.2.1. these functionalities represent abstract definitions, that must be realized by means of application specific technologies.

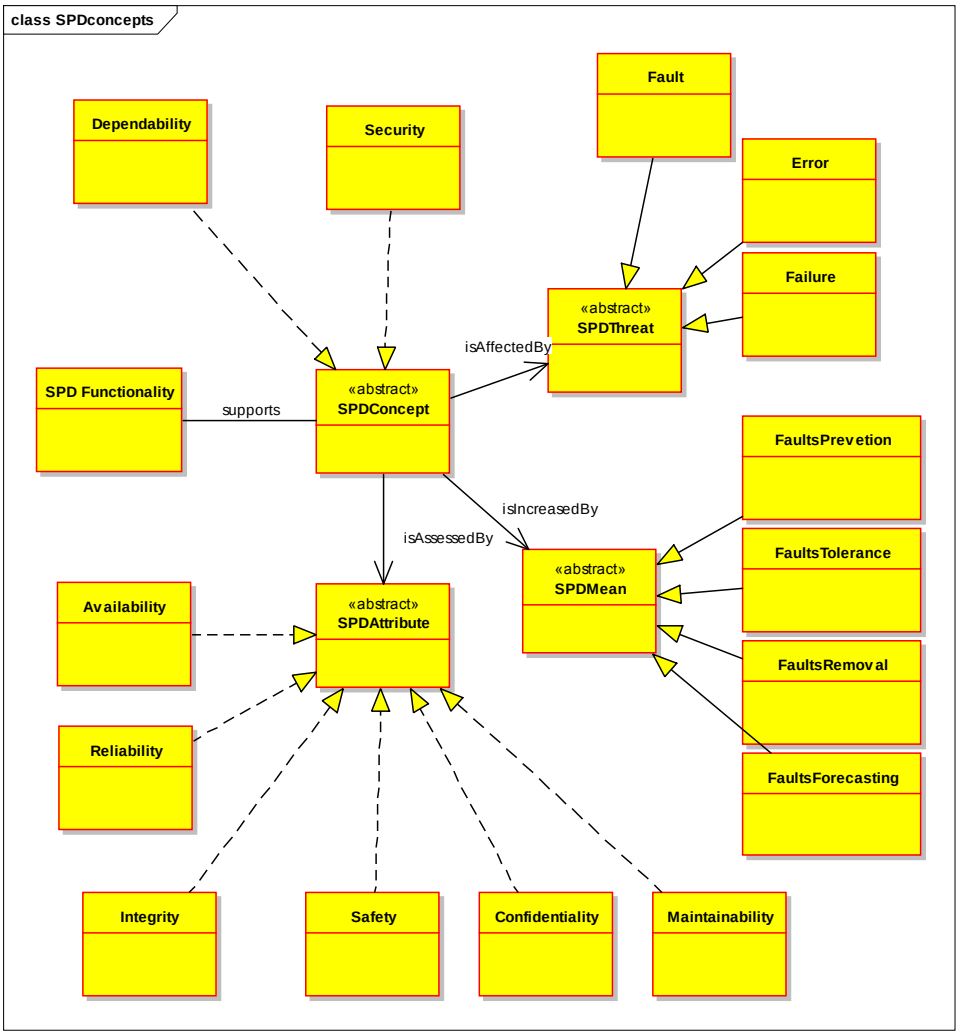


**Figure 6.11 Snapshot of SPD concepts model**

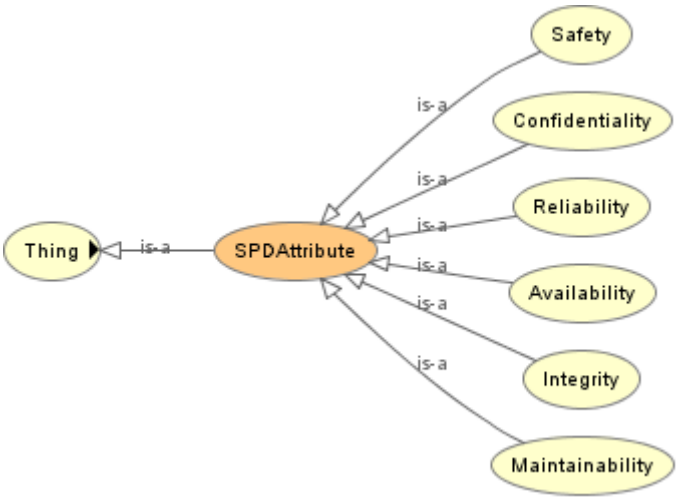


**Figure 6.12 SPD Attributes**

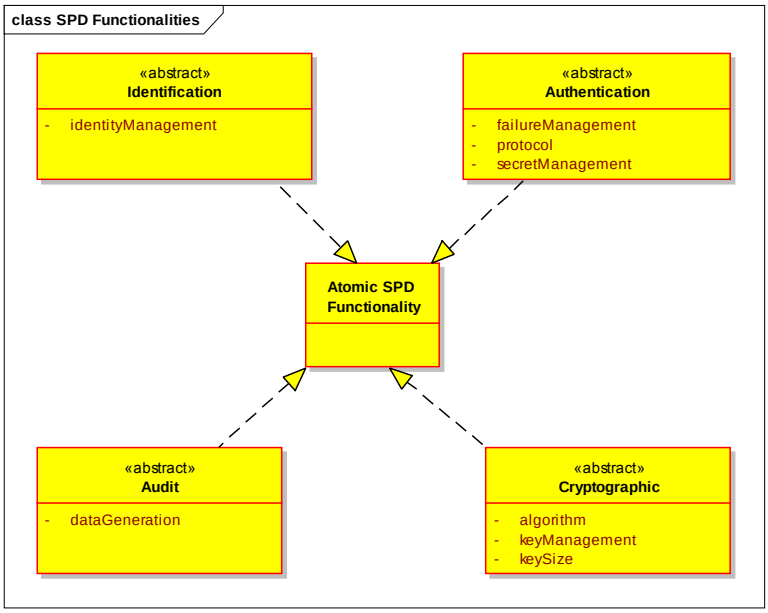


**Figure 6.13 Snapshot of classes of SPD functionalities**

Following the Common Criteria approach, the SPD attributes are:

- Menaced by SPD Threats (error, failure, …)
- Improved by SPD Means (fault forecasting, tolerance, ecc.)

This is depicted in the following figure.


**Figure 6.14 SPD Threat**

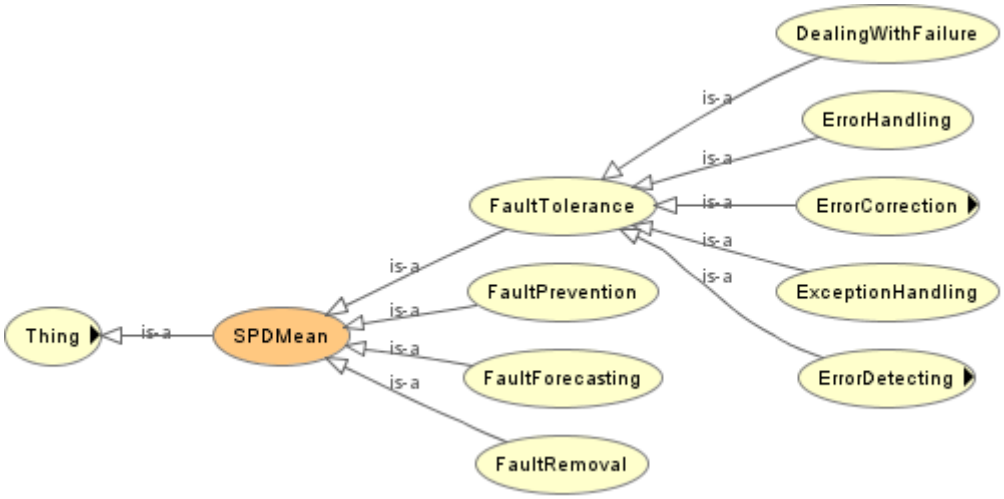


**Figure 6.15 SPD Mean**

To sum UP, all the presented metamodel are linked by relations that allow a complete knowledge of the environment.

1) The pSHIELD System is composed by Node, Network and Middleware elements

2) These elements are made by real hardware components and realizes some functionalities

3) Some of these components can be considered as SPD Component as well as some of these functionalities realise SPD Functionalities

4) SPD Functionalities can be composed

5) SPD Functionalities impact SPD Attributes

6) SPD Attributes are affected by SPD Threats

7) SPD Threats can be improved by SPD Means

At run time the user asks for a specific value of SPD. This is translated into a value for the SPD Attributes. The attributes allow to identify the threats and means for the current sytem configuration and, via SPD functionalities and components, it is possible to understand which physical element of the system is related to the specific menace or mean and consequently act on it.

# 7 Ontology Implementations

The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with formal semantics. OWL has three increasingly expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

The basic reasons for decision to use of OWL for modelling in pSHIELD are:

1) OWL extends all other languages like XML, RDF, and RDF-S. Actually, OWL has been developed on top of the existing XML and RDF standards, which did not appear adequate for achieving efficient semantic interoperability.

    a) E.g. in XML and XML Schema same term may be used with different meaning in different contexts, and different terms may be used for items that have the same meaning.

    b) E.g. RDF and RDF-S address some problem by allowing simple semantics to be associated with identifiers. With RDFS, one can define classes that may have multiple subclasses and superclasses, and can define properties, which may have subproperties, domains, and ranges. However, in order to achieve interoperation between numerous, autonomously developed and managed schemas, richer semantics are needed, like disjoints and cardinality of relations.

2) OWL adds more vocabulary for describing properties and classes, relations between classes, cardinality, equality, richer typing of properties, characteristics of properties and enumerated classes, and all available in three increasingly expressive and increasingly complex sublanguages (Lite, DL, Full) designed for use by specific communities of implementers and users.

3) OWL is well-known widely used open W3C standard with very good support and promising potential and real usage in several industry applications.

4) OWL has wide support of modelling tools, platforms, and reasoners.

5) Previous languages could express (in most cases) the same things, but for some of them OWL provide direct solution by a predefined type of predicates.

6) There are several well-known mechanisms for expressing OWL-Lite and OWL-DL ontologies to stay on decidable level, where Description Logic (DL) could be used correctly.

7) OWL language has proved its potential to use for modelling of semantic interoperability in several middleware-based applications and domains.

In pSHIELD the same OWL-based framework can be used for representation of context, device descriptions (capabilities), descriptions of middleware components, services, security aspects, with several specific goals such as:

1. Semantic reasoning based on ontology model may carry out a reconciliation of heterogeneous formats of parameters exchanged between different layers (also suitable for interaction with legacy agents).

2. The semantic characterization of the behavioral aspect of components makes it suitable for an agent to determine “what the service does”.

3. The semantic characterization of the composition of functionalities and of the relations among them makes it suitable for an agent to reason about SPD metrics of the current configuration and – if needed - to carry out reconfigurations of the system at run-time, by means of rule-based combination / composition of components and SPD technologies, in order to achieve the new intended values for SPD metrics.

The ontology has many merits, of which the most notable are the excellent extensibility, and high expression power. Many systems in the “ubiquitous” and embedded environments are developed using

DL-based ontologies and used with DL-based reasoning. Usually, ontologies are used for modelling context that the systems should collect and analyze. A pure DL-based approach, however, has certain limitations in a context environment. OWL and other ontology languages based on Description Logic cannot properly handle rules expressed in Horn-Logic. Hence, to ensure syntactic and semantic interoperability on device level (e.g. “low-level” ontologies), *SWRL* (Semantic Web Rule Language) or *Jena* can be used for expressing rules.

The detailed OWL representation of the pSHIELD ontology can be found in section “Annex 2 – pSHIELD OWL Models”

## 7.1 Semantic composition

Semantic (inferential) engines, enabled by SPD ontologies, may be profitably exploited to carry out a number of basic functionalities supporting composability based on SPD metric in the pSHIELD framework. In this novel approach to the determination of semantic enabled SPD composition, an automatic reasoner can infer the overall level of SPD metrics resting upon model axioms and declarative rules.

First of all, the pSHIELD framework can leverage the semantic support to composability during the two following stages:

- Analysis: at run time (online), the semantic engine oversees the current value of the overall SPD level as the state of the system evolves in time
- Synthesis:at design time (offline) the semantic engine helps along the configuration of a system architecture, by discovering proper combinatons of SPD modules, according to the corresponding semantic model of modules and composability rules picked out from an offline repository (catalogue); at run time (online), changes in the state of the system trigger the semantic engine to devise new compositions, based on knowledge of modules that at the moment are active in the system (possibly discovered at run time), in order to guarantee the prearranged overall SPD level.

With regards to the operations that have been identified in the proposal for the aggregation of SPD metrics, and to the requirements of ontological SPD modeling, a number of suitable mixes of rules and ontology axioms can be used to develop the aggregation features.

**MIN operation**: can be modeled after the concept of ontological functional features (behaviour of the SPD module) and ontological composition features (all the SPD modules are present at the same time, with weak correlation between the two). Antecedent and consequent of the declarative rule look like the following, given that we are assessing the composite SPD level “L” of a component “X” which exposes 2 distinct SPD functionalities S1 and S2:

$$hasSPDFunct(X,S1) \wedge hasSPDFunct(X,S2) \wedge hasSPDLevel(\mathrm{S1},\mathrm{L1}) \wedge hasSPDLevel(\mathrm{S2},\mathrm{L2}) \wedge MINOp(L1,L2,L) \Rightarrow hasCompositeSPDLevel(X,L)$$

**OR and MEAN operation**: can be modeled in a similar fashion, but the ontological composition feature conveys a stronger relation between the SPD modules; as a matter of fact, every SPD function combine to bring a higher value of overall SPD level. The relation between several SPD functionalities is captured at different extent of complexity by an ontological composition feature, in order to render OR operation (one SPD function “includes” a second one) or MEAN operation (SPD functions act as a sort of alternative)

$$hasSPDFunct(X,S1) \wedge hasSPDFunct(X,S2) \wedge hasSPDLevel(\mathrm{S1},\mathrm{L1}) \wedge hasSPDLevel(\mathrm{S2},\mathrm{L2}) \wedge includesSPD(S1,S2) \wedge OROp(L1,L2,L) \Rightarrow hasCompositeSPDLevel(X,L)$$

$$hasSPDFunct(X,S1) \wedge hasSPDFunct(X,S2) \wedge hasSPDLevel(\mathrm{S1},\mathrm{L1}) \wedge hasSPDLevel(\mathrm{S2},\mathrm{L2}) \wedge alternativeSPD(S1,S2) \wedge MEANOp(L1,L2,L) \Rightarrow hasCompositeSPDLevel(X,L)$$

**Redundancy**: can be modeled by means of ontological axioms of subclassing or (application) ontological features of equivalence. In our system model, we can state that a functionality S2 is the equivalent of another S1, or that a functionality S2 is a specialization of S1 (concept of inheritance), and consequently that S2 can act as S1. Of course, a composite value of SPD metric can be linked to the redundancy as it’s recognized by the reasoner

$$hasSPDFunct(X, S1) \wedge hasSPDFunct(X, S2) \wedge hasSPDFunctionalEquivalence(S1, S2)...$$
$$\Rightarrow hasSPDRedundancy(X, S1) \wedge hasCompositeSPDLevel(X, L)$$

MINOp, MEANOp, OROp are examples of function objects (functors), that are customizable operators, possibly partially based on operators provided as built-in by the inferential engine. We can think of them as application ready-for-use functions, automatically made available by meta-relations between SPD functions inside the model, further extensible to fit new instances of meta-relations.

The clauses inside the rules may be rendered by means of automatic expansion of predicates, based on the constraints that are stated in the model.

At last, if the variables in the clauses are bound, the reasoner works as analyzer, and the compoiste level appear in the consequent clauses since it's unknown. If the variables are unbound, the reasoner acts as synthesizer: in this case, the composite level is a target (known) value, so it must appear in the antecedent clauses, whereas composition of modules and functionalities are the unknown data to find out.

# 8 Conclusions

In order to address SPD composability in embedded systems, we think that we can benefit the ability of semantic technologies to process explicit knowledge of a domain in an effective way. SPD composability can be reformulated in a problem of interoperability, and for this purpose we leverage the features of validation, analysis and harmonization provided by ontologies and semantic reasoning

We argue that interoperability issues can be divided onto three levels of integration, the syntactic, structural, and semantic level. And we try out a holistic approach in which all the three levels of integration are working together by means of a unifying ontology
The framework we propose is able to address the first two issues by means of reconciliation, and the latter by means of the semantic annotation of the behavioral aspect of components in an ES
Further work involves several areas, among which:

- Automatic translation of constraints of the model in reasoner's rules (clauses)
- support to the customization of generic SPD metrics
- Investigation on completeness issues in reasoner's solution
- Assessment of performance of inference process

# Annex 1 – pSHIELD Glossary

| Concept | Description | Source |
|---|---|---|
| Application Processor (AP) | Main processing unit | 5.1 pSHIELD Node –M01 |
| Atomic pSHIELD SPD Component | Is a generic atomic SPD Functionality (innovative or legacy) provided by a pSHIELD device at node, network or middleware level | 4.4.2.2 pSHIELD SPD components composition |
| Audit | Involves recognizing, recording, storing, and analyzing information related to SPD relevant activities. The resulting audit records can be examined to determine which SPD relevant activities took place | 3.2 - The pSHIELD System: Application-Oriented Definitions- D 2.1.1 |
| Automatic Web Service Composition and Interoperation | This task involves the automatic selection, composition, and interoperation of Web services to perform some complex task, given a high-level description of an objective. | http://www.w3.org/Submission/OWL-S/ |
| Automatic Web Service Discovery | Is an automated process for location of Web services that can provide a particular class of service capabilities, while adhering to some client-specified constraints | http://www.w3.org/Submission/OWL-S/ |
| Automatic Web Service Invocation | Is the automatic invocation of an Web service by a computer program or agent, given only a declarative description of that service, as opposed to when the agent has been pre-programmed to be able to call that particular service. | http://www.w3.org/Submission/OWL-S/ |
| Availability | Refers to a system's readiness to provide correct service, whilst reliability refers to continuity of correct service, but these two attributes can considered as one because both guarantee the correct service with an error $e(t) < \varepsilon$. | 4.6.2 Formalized conceptual model -M01- |
| Awareness | Capability of the Cognitive Radio to understand, learn, and predict what is happening in the radio spectrum, e.g., to identify the transmitted waveform, to localize the radio sources, etc. | 5.3.2 Formal conceptual Model –M01- |
| Bridge | Is a network device that is at the link layer of the ISO / OSI model and translates from one physical media to another within the same local network. | - |
| Cognitive Radio | Is an intelligent wireless communication system that is aware of its surrounding environment (i.e., outside world), and uses the methodology of understanding-by-building to learn from the environment and to adapt its internal states to statistical variations in the incoming Radio-Frequency (RF) stimuli by making corresponding changes in certain operating parameters (e.g., transmit-power, carrier-frequency, and modulation strategy) in real-time, with two primary objectives in mind: (i) highly Reliable and Dependable communications whenever and wherever needed and (ii) efficient utilization of the radio spectrum. | 5.3.2 Formal conceptual model- M01- |

| Concept | Description | Source |
|---|---|---|
| Common Criteria Approach | Is approach based in three fondamental part:<br>• Assets to protect and in particular SPD attribute of these assets definition<br>• Threats identifications (Fault Errors Failures); in our approach faults are grouped in HMF (FUA, NFUA) and NHMF.<br>SPD functionalities (whose purpose is to mitigate threats) are implemented to meet pSHIELD SPD objectives. | 5.7 Common Criteria Approach –M01- |
| Composability | Is the possibility to compose different (possibly heterogeneous) SPD functionalities (also referred to as SPD components) aiming at achieving in the considered system of Embedded System Devices a target SPD level which satisfies the requirements of the considered scenario. | 4.4 Composability -M01- |
| Confidentiality | Refers to the property that information or data are not available to unauthorized persons or processes, or that unauthorized access to a system's output will be blocked by the system's filter.<br>Confidentiality faults are mainly caused by access control problems originating in cryptographic faults, security policy faults, hardware faults, and software faults. | 4.6.2 Formalized conceptual modelM01- |
| Contiki Operating System | Contiki is also an open source, highly portable, multi-tasking operating system for memory-efficient networked ESDs and WSNs | 5.1.3.2 Nano, Micro and Personal Node operating systems -M01- |
| Control Algorithms | Retrieves the aggregated information on the current SPD status of the subsystem, as well as of the other interconnected subsystems, by the pS-CA interface connected to the Semantic Knowledge Representation; such retrieved information is used as input for the Control Algorithms. The outputs of the Control Algorithms consist in decisions to be enforced in the various ESDs included in the pSHIELD subsystem controlled by the Security Agent in question; these decisions are sent back via the pS-MS interface, as well as communicated to the other Security Agents on the Overlay, through the pS-OS interface. | 3.1.1 pShield Functional Architecture –M01- |

| Concept | Description | Source |
|---|---|---|
| Core SPD Services | The core SPD services are a set of mandatory basic SPD functionalities provided by a pSHIELD Middleware Adapter in terms of pSHIELD enabling middleware services. The core SPD services aim to provide a SPD middleware environment to actuate the decisions taken by the pSHIELD Overlay and to monitor the Node, Network and Middleware SPD functionalities of the Embedded System Devices under the pSHIELD Middleware Adapter control. | 5.5 -M01-Core SPD services |
| Cryptographic Algorithms | Algorithms to hiding the information, to provide security and information protection against different forms of attacks | 5.2 Cryptographic algorithms –M01- |
| Dependability | Is a composite concept that encompasses the following attributes: Availability, Reliability, Safety Integrity, Maintainability | 5 Reference SPD Taxonomy -pShield System requirement Specification-D 2.1.1- |
| Discovery | Provide to the pSHIELD Middleware Adapter the information, raw data, description of available hardware resources and services in order to allow the system composability | 5.1.1.3 pSHIELD Node SPD components –M01- |
| Discovery Engine | it is in charge to handle the queries to search for available pSHIELD components sent by the Composition service. | 5.5.2 formalized conceptual model -M01- |
| Discovery Protocol | it is in charge to securely discover all the available SPD components description stored in the Service Registry, using a specific protocol | 5.5.2 formalized conceptual model -M01- |
| Error | Is defined as the part of a system's total state that may lead to a failure. | 5.2-Fault Errors Failure - System Requirement Specification D 2.1.1- |
| Failure | Occurs when an error causes the delivered service to deviate from correct service. | 5.2-Fault Errors Failure - System Requirement Specification D 2.1.1- |
| Fault | Is defined as a cause of an error | 5.2-Fault Errors Failure - System Requirement Specification D 2.1.1- |
| Fault injection | This block has the responsibility to inject a fault into Demodulator | 2 SPD Node Internal Demonstrator Structural Description SPDDemosv7-EB |
| Faults with Unauthorized Access | The class of Faults with unauthorized access (FUA) attempts to cover traditional security issues caused by malicious attempt faults. Malicious attempt fault has the objective of damaging a system. A fault is produced when this attempt is combined with other system faults. | 3- Term and definition-M0.2: Proposal for the aggregation of SPD metrics during composition |
| Filter Engine | it is in charge to semantically match the query with the descriptions of the discovered SPD components | 5.5.2 formalized conceptual model -M01- |
| Flash Memory | It stores the bit-stream and system status information | 2 SPD Node Internal Demonstrator Structural Description SPDDemosv7-EB |
| Forecasting (Fault) | Mechanism that predicts faults so that they can be removed or their effects can be circumvented | - |

| Concept | Description | Source |
|---|---|---|
| Functional Component | Describes a functional entity that, in general, does not have necessarily a physical counterpart (e.g. a software functionality, a middleware service, an abstract object, etc.). | 3.1.1 pSHIELD functional architecture –M01- |
| Gateway | Is a network device that operates at the network layer and above the ISO / OSI model. Its main purpose is to transmit network packets outside a local area network (LAN).Functional Component | - |
| Grounding | Provides details on how to interoperate with a service, via messages. | http://www.w3.org/Submission/OWL-S/ |
| Health Status Monitoring (HSM) | Monitoring for checking the status of each individual component. | 5.1 Pshield Node –M01 |
| Hub | Is a concentrator, a network device that acts as a routing node of a data communications network | |
| Human-Made Faults | Human-made faults result from human actions. They include absence of actions when actions should be performed (i.e., omission faults). Performing wrong actions leads to commission faults. HMF are categorized into two basic classes: faults with unauthorized access (FUA), and other faults (NFUA). | 3- Term and definition-M0.2: Proposal for the aggregation of SPD metrics during composition |
| Hybrid Automata | Is composed by automaton formulation hybrid formulation that Permit to choose the most suitable configuration rules for components that must be composed on the basis of the Overlay control algorithms. | 5.6 Hybrid Automata –M01- |
| HYDRA Middleware | Middleware for Heterogeneous Physical Devices | 5.1.3.2.3 Hydra Middleware -M01- |
| I/O Interface | (I/O) to connect to any peripheral and to the rest of the pSHIELD embedded functionalities. | 5.1 Pshield Node –M01 |
| Innovative SPD Functionalities | Reside in the pSHIELD Middleware, Network and Node Adapters. They are constituted by a variety of pSHIELD-specific components. Each pSHIELD-specific component. represents an innovative SPD functionality ad hoc developed for the pSHIELD project which is included in the pSHIELD Node, Network or Middleware Adapter. | 4.5 Innovative SPD functionalities –M01- |
| Integrity | Refers to the absence of improper alteration of information. An integrity problem can arise if, for instance, internal data are tampered with and the produced output relies on the correctness of the data. | 4.6.2 Formalized conceptual model -M01- |
| Legacy Device Component | i.e. the SPD functionalities already present in the legacy devices which can be accessed through the pSHIELD Adapters; they can be classified in Node, Network and Middleware Component according to whether they are included in a legacy Node/Network/Middleware which can be accessed through the corresponding pSHIELD Adapter. | 4.4.2 Formalized Conceptual model -M01- |

| Concept | Description | Source |
|---|---|---|
| Legacy Embedded System Device (L-ESD) | It represents an individual, atomic physical Embedded System device characterized by legacy Node, Network and Middleware functionalities. | 3.1.1 pShield Functional Architecture –M01- |
| Legacy Functionalities of L-ESD | Is a functionality partitioned into three subsets:<br>- Node layer functionalities: hardware functionalities such as processors, memory, battery, I/O interfaces, peripherals, etc.<br>- Network layer functionalities: communication functionalities such as connectivity, protocols stack, etc.<br>- Middleware layer functionalities: firmware and software functionalities such as services, functionalities, interfaces, protocols, etc. | 3.1.1 pShield Functional Architecture –M01- |
| Legacy Middleware Services | Includes all the legacy middleware services (i.e. messaging, remote procedure calls, objects/content requests, etc.) provided by the Legacy Embedded System Device which are not pSHIELD-compliant. | Par 3.1 pShield functional architecture -M01- |
| Legacy Network Services | Includes all the legacy network services (protocol stacks, routing, scheduling, Quality of Service, admission control, traffic shaping, etc.) provided by the Legacy Embedded System Device which are not pSHIELD-compliant. | Par 3.1 pShield functional architecture – M01- |
| Legacy Node Capabilities | Component includes all the legacy node capabilities (i.e. battery, CPU, memory, I/O ports, IRQ, etc.) provided by the Legacy Embedded System Device which are not pSHIELD compliant. | 3.1.1 pShield Functional Architecture –M01- |
| Maintainability: | Ability to undergo modifications and repairs. | 4.6.2 Formalized conceptual model -M01- |
| Mean | All the mechanisms that break the chains of errors and thereby increase the dependability of a system | - |
| Memory | Memory RAM, SRAM, DRAM, | 5.1 Pshield Node –M01 |
| Metadata | Information that describe set of data | - |
| Micro/Personal nodes | Are richer of nanonode in terms of hardware and software resources, network access capabilities, mobility, interfaces, sensing capabilities, etc. | 5.1 Pshield Node –M01 |
| Middleware Layer | Includes the software functionalities that enable the discovery, composition and execution of the basic services necessary to guarantee SPD as well as to perform the tasks assigned to the system (for example, in the railway scenario, the monitoring functionality) | 3.1 The pSHIELD System: General Definitions - System Requirement Specification D 2.1.1- |
| Nano Nodes | Are typically small ESD with limited hardware and software resources, such as wireless sensors. | 5.1 Pshield Node –M01- |
| Net Device | Components used to connect computers or other electronic devices | - |
| Network CAN | Control Area Network | - |
| Network LAN | Local Area Network | - |

| Concept | Description | Source |
|---|---|---|
| Network Layer | Includes the communication technologies (specific for the rail transportation scenarios) that allow the data exchange among pSHIELD components, as well as the external world. These communication technologies, as well as the networks to which pSHIELD is interconnected can be (and usually are) heterogeneous. | 3.1 The pSHIELD System: General Definitions - System Requirement Specification D 2.1.1- |
| Network MAN | Metropolitan Area Network | - |
| Network VPN | Virtual Private Network | - |
| Network WAN | Wide Area Network | - |
| Node Layer | Includes the hardware components that constitute the physical part of the system. | 3.1 The pSHIELD System: General Definitions - System Requirement Specification D 2.1.1- |
| Node Metrics / Health Status | It receives periodic health messages and metrics from the other blocks. This block contains the information about the full node configuration, metrics and health status. If e.g. the “Assertions” block detects some erroneous output, “Node Metrics / Health Status” block receives this information and must act accordingly. | 2 SPD Node Internal Demonstrator Structural Description SPDDemosv7-EB |
| NonHuman-Made Faults | NHMF refers to faults caused by natural phenomena without human participation. These are physical faults caused by a system’s internal natural processes (e.g., physical deterioration of cables or circuitry), or by external natural processes. They can also be caused by natural phenomena | 3- Term and definition-M0.2: Proposal for the aggregation of SPD metrics during composition |
| Non-Volatile Memory (NVM) | Memory ROM, EEPROM, FLASH, Hard Disk | 5.1 Pshield Node –M01 |
| Not Faults with Unauthorized Access | Human-made faults that do not belong to FUA. Most of such faults are introduced by error, such as configuration problems, incompetence issues, accidents, and so on. | 3- Term and definition- M02--M0.2: Proposal for the aggregation of SPD metrics during composition |
| Overlay | Consists of a set of SPD Security Agents, each one controlling a given pSHIELD subsystem. | 4.2 Overlay –M01- |
| Overlay Layer | The “embedded intelligence” that drives the composition of the pSHIELD components in order to meet the desired level of SPD. This is a software layer as well. | 3.1 The pSHIELD System: General Definitions - System Requirement Specification D 2.1.1- |
| Physical Component | Describes an entity that can be mapped into a physical object (e.g. a hardware component). | 3.1.1 pSHIELD functional architecture –M01- |
| Power Management (PM) | Module for managing power sources, monitoring power consumption, etc. | 5.1 Pshield Node –M01 |
| Power nodes | Is a node that Offer high performance computing in one self-contained board offering data storage, networking, memory and (multi-)processing. | 5.1 Pshield Node –M01 |
| Prevention (Fault) | Mechanism that deals with preventing faults incorporated into a system | |
| Privacy | It is a information that must be accessed only by authorized users, for confidentiality reasons. | - |

| Concept | Description | Source |
|---|---|---|
| pSHIELD Adapter | Is a technology dependent component in charge of interfacing with the legacy Node, Network and Middleware functionalities (through the MS, NS and NC interfaces). The legacy functionalities can be enhanced by the pSHIELD Adapter in order to make them pSHIELD-compliant, i.e. they become SDP legacy device components. In addition, the pSHIELD Adapter includes Innovative SPD functionalities which are SPD pSHIELD-specific components, which can be composed with other SPD components. The pSHIELD Adapter exposes the technology independent pSHIELD Middleware layer functionalities that are used by the Security Agent component. | 3.1.1 pShield Functional Architecture –M01- |
| pSHIELD Communication | This block interfaces SPD Node to pShield Network. It is composed by: Ethernet interface: it allows a communication data interface based on a TCP/IP protocol Message encoding/decoding: it receives the commands from pShield network, decodes them, and acts accordingly. It also sends messages to the network. | 2 SPD Node Internal Demonstrator Structural Description SPDDemosv7-EB |
| pSHIELD Demonstrator | Demonstrator for the project that Have the task<br>To monitor on-carriage environment;<br>To integrate the sensors at the wagon with the M2M platform;<br>To allow secure interoperability of sensor information (between railway infrastructure and third<br>party service provider). | 5.8 pShield Demonstrator -M01- |
| pSHIELD Embedded System Device (pS-ESD) | It is a L-ESD equipped at least with the minimal set of<br>pSHIELD functionalities at Middleware Layer. The<br>pS-ESD exposes the same functionalities as the L-ESD plus an additional interface: the pSHIELD<br>Middleware layer services. | 3.1.1 pShield Functional Architecture –M01- |
| pSHIELD Middleware Adapter | Is a component partitioned in the Core SPD services and in the Innovative SPD functionalities. These two components are linked through the pS-MS interface.<br>The pSHIELD Middleware Adapter should also carry into operation the decisions taken by the Overlay and communicated via the pS-MS interface by actually composing the discovered SPD functionalities. The pSHIELD Middleware Adapter includes a set of Innovative SPD functionalities interoperating with the legacy ESD middleware services (through the MS interface) in order to make them discoverable and composable SPD functionalities. | 3.1.1 pShield Functional Architecture –M01- |

| Concept | Description | Source |
|---|---|---|
| pSHIELD Multi-Layered Approach | The pSHIELD multi-layered approach considers the partition of a given Embedded System into three technology-dependent horizontal layers: the node layer (meaning the hardware functionalities), the network layer (meaning the communication functionalities) and the middleware layer (meaning the software functionalities). | 4.1 pSHIELD multi-layered approach –M01- |
| pSHIELD Network Adapter | Includes a set of Innovative SPD functionalities interoperating with the legacy ESD network services (through the NS interface) and the pSHIELD Node Adapter (through the pS-NC interface) in order to enhance them with the pSHIELD Network layer SPD enabling technologies (such as Smart Transmission). This adapter is also in charge to provide (through the pS-NS interface) all the needed information to the pSHIELD Middleware adapter to enable the SPD composability of the Network layer legacy and Network pSHIELD-specific functionalities. Moreover, the pSHIELD Network Adapter translates the technology independent commands, configurations and decisions coming from the pS-NS interface into technology dependent ones and enforce them also to the legacy Network functionalities through the NS interface. | 3.1.1 pShield Functional Architecture –M01- |
| pSHIELD Node | Is an Embedded System Device (ESD) equipped with several legacy Node Capabilities and with a pSHIELD Node Adapter. A pSHIELD Node is deployed as a hardware/software platform, encompassing intrinsic, Innovative SPD functionalities, providing proper services to the other pSHIELD Network and Middleware Adapters to enable the pSHIELD Composability and consequently the desired system SPD | 5.1 pshied Node –M01- |

| Concept | Description | Source |
|---|---|---|
| pSHIELD Node Adapter | Includes a set of Innovative SPD functionalities interoperating with the legacy ESD node capabilities (using the NC interface) in order to enhance them with the pSHIELD Node layer SPD enabling technologies (such as FPGA Firmware and Lightweight Asymmetric Cryptography). This adapter is in charge to provide (through the pS-NC interface) all the needed information to the pSHIELD Middleware adapter to enable the SPD composability of the Node layer legacy and Node pSHIELD-specific functionalities. Moreover, the pSHIELD Node Adapter translates the technology independent commands, configurations and decisions coming from the pS-NC interface into technology dependent ones and enforce them also to the legacy Node functionalities through the NC interface. | 3.1.1 pShield Functional Architecture –M01- |
| pSHIELD Proxy (pS-P) | Is a technology dependent component of a pS-SPD-ESD that, interacting with the available legacy Node, Network and Middleware capabilities and functionalities (through the NC, NS and MS interfaces, respectively), provides all the needed pSHIELD enhanced SPD functionalities. | 3.1.1 pShield Functional Architecture –M01- |
| pSHIELD SPD Embedded System Device (pS-SPD-ESD): | It is a pS-ESD equipped at least with the minimal set of pSHIELD Overlay functionalities. The pS-SPD-ESD exposes the same functionalities as the pS-ESD plus an additional interface: the pSHIELD Overlay layer SPD services provided by a so-called Service Agent operating in that ESD. | 3.1.1 pShield Functional Architecture –M01- |
| pSHIELD Subsystem (pS-S) | Is an architecture of a set of Embedded System Devices including several L-ESD, connected to several pS-ESD and one and only one pS-SPD-ESD. Connections between two generic ESDs (L-ESD, pS-ESD or pS-SPD-ESD) can be performed, by means of legacy functionalities at Node, Network and/or Middleware layer, through the so-called NC, NS and MS functionalities, respectively. | 3.1.1 pShield Functional Architecture –M01- |

| Concept | Description | Source |
|---|---|---|
| pSHIELD-Specific Components | It is i.e. the innovative SPD functionalities ad hoc developed for the pSHIELD project which are included in the pSHIELD Adapters. They can be classified in Node Network and Middleware pSHIELD-specific components according to whether they are included in the pSHIELD Node, Network or Middleware Adapter. They can be directly accessed by pSHIELD Middleware Core SPD Services through the pSNC, pS-NS and pS-MS interfaces. | 4.4.2 Formalized Conceptual model -M01- |
| Query Preprocessor | it is in charge to enrich the query sent by the Composition service with semantic information related to the peculiar context. | 5.5.2 formalized conceptual model -M01- |
| Reconfigurability | Provide self-configuration of some internal parameters according to the observed radio spectrum. | 5.3.2 Formal Conceptual Model -M01- |
| Reconfiguration / Recovery | This block runs at the PPC static core. It must receive periodically health status information, otherwise it restarts the system | 2 SPD Node Internal Demonstrator Structural Description SPDDemosv7-EB |
| Reconfiguration/Recovery Controller | This is a hard processor or microcontroller, responsible for either reconfiguring the node or recovering in case of an error. It | 5.1.1.2 –Detailded Module description –M01- |
| Recovery Watchdog Timer (RWDT) | Timer for restarting recovery if no activity is detected from the SHSM. | 5.1 Pshield Node –M01 |
| Reliability | Continuity of correct service. | 4.6.2 Formalized conceptual model-M01- |
| Removal (Fault) | mechanism that permits to the system to record failures and remove them via a maintenance cycle | |
| Repeater | A digital device that amplifies, reshapes, retimes, or performs a combination of any of these functions on a digital input signal for retransmission | - |
| Router | Is a device that forwards data packets between telecommunications networks, creating an overlay internetwork. A router is connected to two or more data lines from different networks. | - |
| Rules for Discovery, Configuration and Composition of the SPD Components. | Design and implementation of the Control Algorithms which, on the basis of the sensed metadata (i.e) on the basis of the ontological description (possibly semantically enriched) of the SPD Components provide Rules for discovery, configuration and composition of the SPD components. | 4.4 Composability |

| Concept | Description | Source |
|---|---|---|
| Safety | Refers to absence of catastrophic consequence on System users end environment. A safety fault can arise if, for instance, an unauthorized system access can cause the possibility of human lives being endangered. | 4.6.2 Formalized conceptual modelM01- |
| SDP Network | Is a Network implementable and interoperable with standard networks to comply the main business cases of the application scenarios. | 5.1.3 Nano, Micro and Personal nodes –M01- |
| Seamless Approach | Common approach which leaves out of consideration the specificity of the underlying technologies providing enriched SPD functionalities to heterogeneous Embedded Systems | 4.3 Seamless Approach -M01- |
| Secure Service Discovery | Allows any pSHIELD Middleware Adapter to discover in a secure manner the available SPD functionalities and services over heterogeneous environment, networks and technologies that are achievable by the pSHIELD Embedded System Device (pS-ESD) where it is running. | 5.5 Core SPD Service –M01- |
| Secure/Privacy (SP) | Module to perform security and privacy actions, such as encryption, decryption, key generation, etc. | 5.1 Pshield Node –M01 |
| Security | Is a composite of the attributes of confidentiality, integrity (in the security context, “improper” means “unauthorized"), and availability (for authorized actions only), | 5 Reference SPD Taxonomy -pShield System requirement Specification-D 2.1.1- |
| Security Agent | Is a technology-independent component in charge of aggregating the information coming from the pSHIELD Middleware Services provided by the internal pSHIELD Adapter or by other pSHIELD Proxies located in the same subsystem. The Security Agent is also in charge of gathering the information coming from other Security Agents connected on the same Overlay (through the pS-OS interface). The Security Agent includes proper control algorithms working on the basis of the available information; the decisions taken by these Control Algorithms are enforced through the pS-MS and the pS--OS interfaces. | 3.1.1 pShield Functional Architecture –M01- |
| Semantic Database | It holds any semantic information related to the pSHIELD components (interface, contract, SPD status, context, etc.). | 5.5.2 formalized conceptual model -M01- |
| Semantic Engine (Reasoner) | Enable interoperability within Middleware Layer and rule based discovery and composition within Overlay Agents. | 5.4 Semantic Model –M01- |

| Concept | Description | Source |
|---|---|---|
| Semantic Knowledge Repository | Is (i.e. a database) that storing the dynamic, semantic, enriched, ontological aggregated representation of the SPD functionalities of the pSHIELD subsystem controlled by the SPD Security Agent; | 4.2 Overlay -M01- |
| Semantic Knowledge Representation | Is in charge of bi-directionally exchanging technology independent (and semantic enriched) information from the pS-MS and the pS-OS interfaces. It is also in charge to provide such information via the pS-SKR interface to the Control Algorithms component. | 3.1.1 pShield Functional Architecture –M01- |
| Semantic Model | It is a conceptual model in which semantic information is included. | - |
| Sensor/Actuator | Are represented by the Core SPD Services lying at the pSHIELD Middleware layer. | 4.2.2 Formalized concept Model –M01- |
| Sensors/Actuators | Represent the Core SPD Services lying at the pSHIELD Middleware layer. | 4.2.2 Formalized conceptual Model-M01- |
| Service Composition | Is in charge to select atomic SPD services that, once composed, provide a complex and integrated SPD functionality that is essential to guarantee the required SPD level. The service composition is a pSHIELD Middleware Adapter functionality that cooperates with the pSHIELD Overlay in order to apply the configuration strategy decided by the Control Algorithms residing in the pSHIELD Security Agent. | 5.5 Core SPD Services –M01- |
| Service Grounding | Specifies the details of how an agent can access a service-details having mainly to do with protocol and message formats, serialization, transport, and addressing | http://www.w3.org/Submission/OWL-S/ |
| Service Orchestration | Deploy, execute and monitoring SPD services. | 5.5 Core SPD Services –M01- |
| Service Profile | Tells "what the service does", in a way that is suitable for a service-seeking agent (or matchmaking agent acting on behalf of a service-seeking agent) to determine whether the service meets its needs. | http://www.w3.org/Submission/OWL-S/ |
| Services Model | Tells a client how to use the service, by detailing the semantic content of requests, the conditions under which particular outcomes will occur, and, where necessary, the step by step processes leading to those outcomes. | http://www.w3.org/Submission/OWL-S/ |
| Services Registry | It acts as a database to store the service entries | 5.5.2 formalized conceptual model -M01- |
| Smart SPD Driven Transmission | New advances signal processing techniques. | 5.3 Smart SPD Driven Transmission –M01- |
| SOA | Service-oriented architecture | - |
| Software Agent | Permit a computer-interpretable description of the service. | http://www.w3.org/Submission/OWL-S/ |
| Software Defined Radio (SDR) | Software programmable Components | 5.3 Smart SPD driven transmission-M01- |

| Concept | Description | Source |
|---|---|---|
| Software Wireless Sensor Networks (WSN) | Software part that can be layered into three levels: sensor software, node software and sensor network software. | 5.1.1.3 Specific SPD Considerations for Wireless Sensor Networks |
| SPD | Security Privacy Dependability | |
| SPD Component | Is defined as a functionality which (i) offers a given SPD service through an interface which can be semantically described according to the provided SPD Metrics, (ii) can be accessed through the pSHIELD Middleware Core SPD Services for being configured (if necessary) and activated (or deactivated). | 4.4 Composability -M01- |
| SPD Metrics | Is the possibility to identify and quantify the SPD properties of each component, as well as the SPD properties of the overall system. SPD Metrics allow (i) users to define in an univocal way the requirements for the specific application, (ii) to describe the SPD level provided by the components, and (iii) to compute the SPD level achieved by the system through the Composability mechanism. | 4.6 SPD Metrics-M01 |
| SPD Node | It is composed by the following sub- blocks: FM Signal Acquisition: this blocks principally handles the receiving of data samples from “FM Signal Generator” and pre-processes the data to feed to the “Demodulation Processing” block. This block provides also periodic status & metrics information to the “Node Metrics/Health Status” block. Demodulation Processing: it is responsible for the demodulation processing of the data coming from the “FM Signal Acquisition“ | 2 SPD Node Internal Demonstrator Structural Description SPDDemosv7-EB |
| SPD Security Agent | Consists of two key elements: (i) the Semantic Knowledge Repository (i.e. a database) storing the dynamic, semantic, enriched, ontological aggregated representation of the SPD functionalities of the pSHIELD subsystem (ii) the Control Algorithms generating, on the grounds of the above representation, key SPD-relevant decisions (consisting, as far as the Composability feature is concerned, in a set of discovery, configuration and composition rules). | |
| SPD Status | It represents the current SPD level associated to the function. | 4.4.2 Formalized Conceptual Model –M01- |
| Special Purpose Processor | (SPP) module for any pre- or post-processing, such as compression/decompression, conversion, etc. | 5.1 Pshield Node –M01 |

| Concept | Description | Source |
|---|---|---|
| Stable Storage | Used for storing the status of the system, a bit stream to program an FPGA, and/or the software for system start-up, operating system and application. it receives from each block check-pointing data. It is able to perform a stable write with this data (write on a circular buffer on flash memory, and then validate the just written data). On system restart, this module is able to recover the last valid data. | 5.1 Pshield Node –M01 |
| Switch | Is a computer networking device that connects network segments. | - |
| System Health Status Monitoring (SHSM) | Monitoring for checking the status of the whole system. | 5.1 Pshield Node –M01 |
| The Security/Privacy controller | Consists of one or more modules able to perform different security-related functionalities, such as Data Encryption, Data Decryption, Generation of Cryptographic Keys, etc | 5.1.1.2 –Detailded Module description –M01- |
| Threat | Include faults, errors and failures, as well as their causes, consequences and characteristics. | 5.2-Fault Errors Failure - System Requirement Specification D 2.1.1- |
| TinyOS | This operating system (OS) is a free and open source operating system and platform that is designed for WSNs. | 5.1.3.2 Nano, Micro and Personal Node operating systems -M01- |
| Tolerance (Fault) | System architecture that deals with putting mechanisms in place that will allow a system to still deliver the required service in the presence of faults, although that service may be at a degraded level | - |
| WSDL | Web Services Description Language | - |

**Table 3 pSHIELD Glossary**

# Annex 2 – pSHIELD OWL Models

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY xsp "http://www.owl-ontologies.com/2005/08/07/xsp.owl#" >
    <!ENTITY Ontology1300273978 "http://www.owl-ontologies.com/Ontology1300273978.owl#" >
    <!ENTITY TCP "http://www.owl-ontologies.com/Ontology1300273978.owl#TCP/" >
]>
<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1300273978.owl#"
     xml:base="http://www.owl-ontologies.com/Ontology1300273978.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
     xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
     xmlns:Ontology1300273978="http://www.owl-ontologies.com/Ontology1300273978.owl#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:TCP="&Ontology1300273978;TCP/">
    <owl:Ontology rdf:about="">
        <owl:imports rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
    </owl:Ontology>

    <!--
    ////////////////////////////////////////////////////////////////////////////////////
    //
    // Object Properties
    //
    ////////////////////////////////////////////////////////////////////////////////////
     -->

    <!-- http://www.owl-ontologies.com/2005/08/07/xsp.owl#minExclusive -->
    <owl:ObjectProperty rdf:about="&xsp;minExclusive">
        <rdfs:domain rdf:resource="&rdfs;Datatype"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#HasAutorization -->
    <owl:ObjectProperty rdf:about="#HasAutorization"/>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#affects -->
    <owl:ObjectProperty rdf:about="#affects">
        <rdfs:range rdf:resource="#SPDComponent"/>
        <rdfs:domain rdf:resource="#SPDThreat"/>
        <owl:inverseOf rdf:resource="#isAffectedBy"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#assesses -->
    <owl:ObjectProperty rdf:about="#assesses">
        <rdfs:domain rdf:resource="#SPDAttribute"/>
        <rdfs:range rdf:resource="#SPDConcept"/>
        <owl:inverseOf rdf:resource="#isAssessedBy"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#describedBy -->
    <owl:ObjectProperty rdf:about="#describedBy">
        <rdfs:domain rdf:resource="#Service"/>
        <rdfs:range rdf:resource="#ServiceModel"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#describes -->
    <owl:ObjectProperty rdf:about="#describes">
        <rdfs:range rdf:resource="#Service"/>
        <rdfs:domain rdf:resource="#ServiceModel"/>
        <owl:inverseOf rdf:resource="#describedBy"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasAntenna -->
    <owl:ObjectProperty rdf:about="#hasAntenna">
        <rdfs:range rdf:resource="#Antenna"/>
        <rdfs:domain rdf:resource="#NetDevice"/>
        <owl:inverseOf rdf:resource="#isAntennaOf"/>
```

```
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasAtomicSPDFunctionality -->
    <owl:ObjectProperty rdf:about="#hasAtomicSPDFunctionality">
        <rdfs:domain rdf:resource="#CompositeSPDFunctionality"/>
        <rdfs:range>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasAtomicSPDFunctionality"/>
                <owl:onClass rdf:resource="#Functionality"/>
                <owl:minQualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minQualifiedCardinality>
            </owl:Restriction>
        </rdfs:range>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasAutentication -->
    <owl:ObjectProperty rdf:about="#hasAutentication"/>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasBattery -->
    <owl:ObjectProperty rdf:about="#hasBattery">
        <rdfs:range rdf:resource="#Battery"/>
        <rdfs:domain rdf:resource="#Node"/>
        <owl:inverseOf rdf:resource="#isBatteryOf"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasCPU -->
    <owl:ObjectProperty rdf:about="#hasCPU">
        <rdfs:range rdf:resource="#ProcessorCount"/>
        <rdfs:domain rdf:resource="#Sensor"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasClient -->
    <owl:ObjectProperty rdf:about="#hasClient">
        <rdfs:subPropertyOf rdf:resource="#hasParticipiant"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasCryptography -->
    <owl:ObjectProperty rdf:about="#hasCryptography">
        <owl:inverseOf rdf:resource="#inverse_of_hasCryptography"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasExistential -->
    <owl:ObjectProperty rdf:about="#hasExistential">
        <rdfs:subPropertyOf rdf:resource="#hasVar"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasFunctionality -->
    <owl:ObjectProperty rdf:about="#hasFunctionality">
        <rdfs:domain rdf:resource="#Element"/>
        <rdfs:range rdf:resource="#Functionality"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasInput -->
    <owl:ObjectProperty rdf:about="#hasInput">
        <rdfs:range rdf:resource="#Input"/>
        <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasLocal -->
    <owl:ObjectProperty rdf:about="#hasLocal">
        <rdfs:subPropertyOf rdf:resource="#hasVar"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasMemory -->
    <owl:ObjectProperty rdf:about="#hasMemory">
        <rdfs:range rdf:resource="#Memory"/>
        <rdfs:domain rdf:resource="#Sensor"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasNet -->
    <owl:ObjectProperty rdf:about="#hasNet">
        <rdfs:range rdf:resource="#Network"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasNetDev -->
    <owl:ObjectProperty rdf:about="#hasNetDev">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
```

```
        <rdfs:range rdf:resource="#NetDevice"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasOutput -->
    <owl:ObjectProperty rdf:about="#hasOutput">
        <rdfs:range rdf:resource="#Output"/>
        <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasParameter -->
    <owl:ObjectProperty rdf:about="#hasParameter">
        <rdfs:range rdf:resource="#Parameter"/>
        <rdfs:domain rdf:resource="#Process"/>
        <rdfs:subPropertyOf rdf:resource="#hasVar"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasParticipiant -->
    <owl:ObjectProperty rdf:about="#hasParticipiant">
        <rdfs:range rdf:resource="#Partecipant"/>
        <rdfs:subPropertyOf rdf:resource="#hasVar"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasProcess -->
    <owl:ObjectProperty rdf:about="#hasProcess">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:range rdf:resource="#Process"/>
        <rdfs:domain rdf:resource="#Profile"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasProfile -->
    <owl:ObjectProperty rdf:about="#hasProfile"/>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasQoS -->
    <owl:ObjectProperty rdf:about="#hasQoS">
        <rdfs:range rdf:resource="#QoS"/>
        <owl:inverseOf rdf:resource="#inverse_of_hasQoS"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasResult -->
    <owl:ObjectProperty rdf:about="#hasResult">
        <rdfs:domain rdf:resource="#Process"/>
        <rdfs:range rdf:resource="#Result"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasResultVar -->
    <owl:ObjectProperty rdf:about="#hasResultVar">
        <rdfs:domain rdf:resource="#Result"/>
        <rdfs:range rdf:resource="#ResultVar"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasRouting -->
    <owl:ObjectProperty rdf:about="#hasRouting">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:range rdf:resource="#Routing"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasSO -->
    <owl:ObjectProperty rdf:about="#hasSO">
        <rdfs:range rdf:resource="#SystemOperative"/>
        <owl:inverseOf rdf:resource="#inverse_of_hasSO"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasSPDLevel -->
    <owl:ObjectProperty rdf:about="#hasSPDLevel">
        <rdfs:domain rdf:resource="#SPDComponent"/>
        <rdfs:range>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#SPDLevel"/>
                <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
                <owl:onDataRange rdf:resource="&xsd;int"/>
            </owl:Restriction>
        </rdfs:range>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasSPDStatus -->
    <owl:ObjectProperty rdf:about="#hasSPDStatus">
        <rdfs:domain rdf:resource="#SPDFunctionality"/>
```

```
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasServiceModel -->
<owl:ObjectProperty rdf:about="#hasServiceModel">
    <rdfs:domain rdf:resource="#Node"/>
    <rdfs:range rdf:resource="#ServiceModel"/>
    <owl:inverseOf rdf:resource="#inverse_of_hasServiceModel"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasSoftware -->
<owl:ObjectProperty rdf:about="#hasSoftware">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
    <rdfs:domain rdf:resource="#Node"/>
    <rdfs:range rdf:resource="#Software"/>
    <owl:inverseOf rdf:resource="#isSoftwareOf"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasTransmissionMedium -->
<owl:ObjectProperty rdf:about="#hasTransmissionMedium">
    <rdfs:range rdf:resource="#TransmissionMedium"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasTransport -->
<owl:ObjectProperty rdf:about="#hasTransport">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasVar -->
<owl:ObjectProperty rdf:about="#hasVar">
    <rdfs:domain rdf:resource="#Process"/>
    <rdfs:range rdf:resource="#ProcessVariable"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#iisMemoryOf -->
<owl:ObjectProperty rdf:about="#iisMemoryOf">
    <rdfs:domain rdf:resource="#Memory"/>
    <rdfs:range rdf:resource="#Sensor"/>
    <owl:inverseOf rdf:resource="#hasMemory"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#increases -->
<owl:ObjectProperty rdf:about="#increases">
    <rdfs:range rdf:resource="#SPDConcept"/>
    <rdfs:domain rdf:resource="#SPDMean"/>
    <owl:inverseOf rdf:resource="#isIncreasedBy"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#inverse_of_hasCryptography -->
<owl:ObjectProperty rdf:about="#inverse_of_hasCryptography"/>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#inverse_of_hasNetDev -->
<owl:ObjectProperty rdf:about="#inverse_of_hasNetDev">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#NetDevice"/>
    <owl:inverseOf rdf:resource="#hasNetDev"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#inverse_of_hasQoS -->
<owl:ObjectProperty rdf:about="#inverse_of_hasQoS">
    <rdfs:domain rdf:resource="#QoS"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#inverse_of_hasRouting -->
<owl:ObjectProperty rdf:about="#inverse_of_hasRouting">
    <rdfs:domain rdf:resource="#Routing"/>
    <owl:inverseOf rdf:resource="#hasRouting"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#inverse_of_hasSO -->
<owl:ObjectProperty rdf:about="#inverse_of_hasSO">
    <rdfs:domain rdf:resource="#SystemOperative"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#inverse_of_hasServiceModel -->
<owl:ObjectProperty rdf:about="#inverse_of_hasServiceModel">
    <rdfs:range rdf:resource="#Node"/>
```

```
        <rdfs:domain rdf:resource="#ServiceModel"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#inverse_of_hasTransmissionMedium
-->
    <owl:ObjectProperty rdf:about="#inverse_of_hasTransmissionMedium">
        <rdfs:domain rdf:resource="#TransmissionMedium"/>
        <owl:inverseOf rdf:resource="#hasTransmissionMedium"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#inverse_of_hasTransport -->
    <owl:ObjectProperty rdf:about="#inverse_of_hasTransport">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <owl:inverseOf rdf:resource="#hasTransport"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isAffectedBy -->
    <owl:ObjectProperty rdf:about="#isAffectedBy">
        <rdfs:domain rdf:resource="#SPDConcept"/>
        <rdfs:range rdf:resource="#SPDThreat"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isAntennaOf -->
    <owl:ObjectProperty rdf:about="#isAntennaOf">
        <rdfs:domain rdf:resource="#Antenna"/>
        <rdfs:range rdf:resource="#NetDevice"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isAssessedBy -->
    <owl:ObjectProperty rdf:about="#isAssessedBy">
        <rdfs:range rdf:resource="#SPDAttribute"/>
        <rdfs:domain rdf:resource="#SPDConcept"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isBatteryOf -->
    <owl:ObjectProperty rdf:about="#isBatteryOf">
        <rdfs:domain rdf:resource="#Battery"/>
        <rdfs:range rdf:resource="#Node"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isCPUof -->
    <owl:ObjectProperty rdf:about="#isCPUof">
        <rdfs:domain rdf:resource="#ProcessorCount"/>
        <rdfs:range rdf:resource="#Sensor"/>
        <owl:inverseOf rdf:resource="#hasCPU"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isComposedBy -->
    <owl:ObjectProperty rdf:about="#isComposedBy">
        <rdfs:range rdf:resource="#Element"/>
        <rdfs:domain rdf:resource="#System"/>
        <owl:inverseOf rdf:resource="#isPartOf"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isComposedByFunctionality -->
    <owl:ObjectProperty rdf:about="#isComposedByFunctionality">
        <rdfs:domain rdf:resource="#CompositeSPDFunctionality"/>
        <rdfs:range rdf:resource="#SPDFunctionality"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isFunctionalityOf -->
    <owl:ObjectProperty rdf:about="#isFunctionalityOf">
        <rdfs:range rdf:resource="#Element"/>
        <rdfs:domain rdf:resource="#Functionality"/>
        <owl:inverseOf rdf:resource="#hasFunctionality"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isIncreasedBy -->
    <owl:ObjectProperty rdf:about="#isIncreasedBy">
        <rdfs:domain rdf:resource="#SPDConcept"/>
        <rdfs:range rdf:resource="#SPDMean"/>
    </owl:ObjectProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isNetOf -->
    <owl:ObjectProperty rdf:about="#isNetOf">
        <rdfs:domain rdf:resource="#Network"/>
        <owl:inverseOf rdf:resource="#hasNet"/>
    </owl:ObjectProperty>
```

```
<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isPartOf -->
<owl:ObjectProperty rdf:about="#isPartOf">
    <rdfs:domain rdf:resource="#Element"/>
    <rdfs:range rdf:resource="#System"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isPartofSPDFunctionality -->
<owl:ObjectProperty rdf:about="#isPartofSPDFunctionality">
    <rdfs:range rdf:resource="#CompositeSPDFunctionality"/>
    <rdfs:domain rdf:resource="#SPDFunctionality"/>
    <owl:inverseOf rdf:resource="#isComposedByFunctionality"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#isSoftwareOf -->
<owl:ObjectProperty rdf:about="#isSoftwareOf">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
    <rdfs:range rdf:resource="#Node"/>
    <rdfs:domain rdf:resource="#Software"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#owlsParameter -->
<owl:ObjectProperty rdf:about="#owlsParameter">
    <rdfs:domain rdf:resource="#MessageMap"/>
    <rdfs:range rdf:resource="#Parameter"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#performedBy -->
<owl:ObjectProperty rdf:about="#performedBy">
    <rdfs:subPropertyOf rdf:resource="#hasParticipiant"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#presentedBy -->
<owl:ObjectProperty rdf:about="#presentedBy">
    <rdfs:range rdf:resource="#Service"/>
    <rdfs:domain rdf:resource="#ServiceProfile"/>
    <owl:inverseOf rdf:resource="#presents"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#presents -->
<owl:ObjectProperty rdf:about="#presents">
    <rdfs:domain rdf:resource="#Service"/>
    <rdfs:range rdf:resource="#ServiceProfile"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#provides -->
<owl:ObjectProperty rdf:about="#provides">
    <rdfs:domain rdf:resource="#SPDComponent"/>
    <rdfs:range rdf:resource="#SPDFunctionality"/>
    <rdfs:domain rdf:resource="#System"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#realizes -->
<owl:ObjectProperty rdf:about="#realizes">
    <rdfs:domain rdf:resource="#AtomicProcess"/>
    <rdfs:range rdf:resource="#SimpleProcess"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#relizedBy -->
<owl:ObjectProperty rdf:about="#relizedBy">
    <rdfs:range rdf:resource="#AtomicProcess"/>
    <rdfs:domain rdf:resource="#SimpleProcess"/>
    <owl:inverseOf rdf:resource="#realizes"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#supportedBy -->
<owl:ObjectProperty rdf:about="#supportedBy">
    <rdfs:range rdf:resource="#Service"/>
    <rdfs:domain rdf:resource="#ServiceGrounding"/>
    <owl:inverseOf rdf:resource="#supports"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#supports -->
<owl:ObjectProperty rdf:about="#supports">
    <rdfs:domain rdf:resource="#Service"/>
    <rdfs:range rdf:resource="#ServiceGrounding"/>
</owl:ObjectProperty>
```

```
<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#theParam -->
<owl:ObjectProperty rdf:about="#theParam">
    <rdfs:range rdf:resource="#Parameter"/>
    <rdfs:domain rdf:resource="#valueOf"/>
</owl:ObjectProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#useConnectorTopology -->
<owl:ObjectProperty rdf:about="#useConnectorTopology">
    <rdfs:domain rdf:resource="#CompositeSPDFunctionality"/>
    <rdfs:range rdf:resource="#Connector"/>
</owl:ObjectProperty>

<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Data properties
//
///////////////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ByteEncryptionTime -->
<owl:DatatypeProperty rdf:about="#ByteEncryptionTime">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:range rdf:resource="#genid155"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CapacityLeveMedium -->
<owl:DatatypeProperty rdf:about="#CapacityLeveMedium">
    <rdfs:subPropertyOf rdf:resource="#CapacityLevel"/>
    <rdfs:range rdf:resource="#genid137"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CapacityLevel -->
<owl:DatatypeProperty rdf:about="#CapacityLevel">
    <rdfs:range rdf:resource="#genid148"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CapacityLevelHigh -->
<owl:DatatypeProperty rdf:about="#CapacityLevelHigh">
    <rdfs:subPropertyOf rdf:resource="#CapacityLevel"/>
    <rdfs:range rdf:resource="&xsd;float"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CapacityLevelLow -->
<owl:DatatypeProperty rdf:about="#CapacityLevelLow">
    <rdfs:subPropertyOf rdf:resource="#CapacityLevel"/>
    <rdfs:range rdf:resource="#genid156"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FirmwareProgrammable -->
<owl:DatatypeProperty rdf:about="#FirmwareProgrammable">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Firmware"/>
    <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IDNetwork -->
<owl:DatatypeProperty rdf:about="#IDNetwork">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Network"/>
    <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IDnode -->
<owl:DatatypeProperty rdf:about="#IDnode">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Node"/>
    <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#KeyLenght -->
<owl:DatatypeProperty rdf:about="#KeyLenght">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:range rdf:resource="#genid147"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Microchip -->
```

```
    <owl:DatatypeProperty rdf:about="#Microchip">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#ProcessorCount"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NodeBackup -->
    <owl:DatatypeProperty rdf:about="#NodeBackup">
        <rdfs:domain rdf:resource="#Node"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#OperativLevel -->
    <owl:DatatypeProperty rdf:about="#OperativLevel">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:range rdf:resource="#genid3"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ParameterValue -->
    <owl:DatatypeProperty rdf:about="#ParameterValue">
        <rdfs:domain rdf:resource="#ProcessVariable"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PowerGenerator -->
    <owl:DatatypeProperty rdf:about="#PowerGenerator">
        <rdfs:domain rdf:resource="#Battery"/>
        <rdfs:range>
            <rdf:Description>
                <rdf:type rdf:resource="&owl;DataRange"/>
                <owl:oneOf>
                    <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first rdf:datatype="&xsd;string">eolic</rdf:first>
                        <rdf:rest>
                            <rdf:Description>
                                <rdf:type rdf:resource="&rdf;List"/>
                                <rdf:first rdf:datatype="&xsd;string">piezoeletric</rdf:first>
                                <rdf:rest>
                                    <rdf:Description>
                                        <rdf:type rdf:resource="&rdf;List"/>
                                        <rdf:first
rdf:datatype="&xsd;string">solar</rdf:first>
                                        <rdf:rest rdf:resource="&rdf;nil"/>
                                    </rdf:Description>
                                </rdf:rest>
                            </rdf:Description>
                        </rdf:rest>
                    </rdf:Description>
                </owl:oneOf>
            </rdf:Description>
        </rdfs:range>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProtocolDiscovery -->
    <owl:DatatypeProperty rdf:about="#ProtocolDiscovery">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SPDLevel -->
    <owl:DatatypeProperty rdf:about="#SPDLevel">
        <rdfs:range rdf:resource="&xsd;int"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SPDStatus -->
    <owl:DatatypeProperty rdf:about="#SPDStatus">
        <rdfs:domain rdf:resource="#SPDFunctionality"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ServiceName -->
    <owl:DatatypeProperty rdf:about="#ServiceName">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#Functionality"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SizeMemory -->
```

```
    <owl:DatatypeProperty rdf:about="#SizeMemory">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#Memory"/>
        <rdfs:range rdf:resource="&xsd;float"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Standard -->
    <owl:DatatypeProperty rdf:about="#Standard">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#StateNode -->
    <owl:DatatypeProperty rdf:about="#StateNode">
        <rdfs:domain rdf:resource="#Node"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TipologyAntenna -->
    <owl:DatatypeProperty rdf:about="#TipologyAntenna">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#Antenna"/>
        <rdfs:range>
            <rdf:Description>
                <rdf:type rdf:resource="&owl;DataRange"/>
                <owl:oneOf>
                    <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first rdf:datatype="&xsd;string">Directional</rdf:first>
                        <rdf:rest>
                            <rdf:Description>
                                <rdf:type rdf:resource="&rdf;List"/>
                                <rdf:first
rdf:datatype="&xsd;string">Omnidirectional</rdf:first>
                                <rdf:rest rdf:resource="&rdf;nil"/>
                            </rdf:Description>
                        </rdf:rest>
                    </rdf:Description>
                </owl:oneOf>
            </rdf:Description>
        </rdfs:range>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Transceiver -->
    <owl:DatatypeProperty rdf:about="#Transceiver">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#Antenna"/>
        <rdfs:range>
            <rdf:Description>
                <rdf:type rdf:resource="&owl;DataRange"/>
                <owl:oneOf>
                    <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first rdf:datatype="&xsd;string">Receiver</rdf:first>
                        <rdf:rest>
                            <rdf:Description>
                                <rdf:type rdf:resource="&rdf;List"/>
                                <rdf:first rdf:datatype="&xsd;string">Sender</rdf:first>
                                <rdf:rest>
                                    <rdf:Description>
                                        <rdf:type rdf:resource="&rdf;List"/>
                                        <rdf:first rdf:datatype="&xsd;string"
                                            >Sender | Receiver</rdf:first>
                                        <rdf:rest rdf:resource="&rdf;nil"/>
                                    </rdf:Description>
                                </rdf:rest>
                            </rdf:Description>
                        </rdf:rest>
                    </rdf:Description>
                </owl:oneOf>
            </rdf:Description>
        </rdfs:range>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TymeComposition -->
    <owl:DatatypeProperty rdf:about="#TymeComposition">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:range rdf:resource="&xsd;time"/>
```

```
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TymeDiscovering -->
    <owl:DatatypeProperty rdf:about="#TymeDiscovering">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:range rdf:resource="&xsd;time"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TypeService -->
    <owl:DatatypeProperty rdf:about="#TypeService">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#Functionality"/>
        <rdfs:range>
            <rdf:Description>
                <rdf:type rdf:resource="&owl;DataRange"/>
                <owl:oneOf>
                    <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first rdf:datatype="&xsd;string">HW</rdf:first>
                        <rdf:rest>
                            <rdf:Description>
                                <rdf:type rdf:resource="&rdf;List"/>
                                <rdf:first rdf:datatype="&xsd;string">SW</rdf:first>
                                <rdf:rest rdf:resource="&rdf;nil"/>
                            </rdf:Description>
                        </rdf:rest>
                    </rdf:Description>
                </owl:oneOf>
            </rdf:Description>
        </rdfs:range>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#VersionFirmware -->
    <owl:DatatypeProperty rdf:about="#VersionFirmware">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#Firmware"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#connectorType -->
    <owl:DatatypeProperty rdf:about="#connectorType">
        <rdfs:range>
            <rdf:Description>
                <rdf:type rdf:resource="&owl;DataRange"/>
                <owl:oneOf>
                    <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first rdf:datatype="&xsd;string">CONCENTRIC</rdf:first>
                        <rdf:rest>
                            <rdf:Description>
                                <rdf:type rdf:resource="&rdf;List"/>
                                <rdf:first rdf:datatype="&xsd;string">CONCURRENT</rdf:first>
                                <rdf:rest>
                                    <rdf:Description>
                                        <rdf:type rdf:resource="&rdf;List"/>
                                        <rdf:first
rdf:datatype="&xsd;string">PARALLEL</rdf:first>
                                        <rdf:rest rdf:resource="&rdf;nil"/>
                                    </rdf:Description>
                                </rdf:rest>
                            </rdf:Description>
                        </rdf:rest>
                    </rdf:Description>
                </owl:oneOf>
            </rdf:Description>
        </rdfs:range>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasBandwidth -->
    <owl:DatatypeProperty rdf:about="#hasBandwidth">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="#Network"/>
        <rdfs:range rdf:resource="&xsd;float"/>
    </owl:DatatypeProperty>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasKey -->
    <owl:DatatypeProperty rdf:about="#hasKey">
        <rdfs:range rdf:resource="&xsd;float"/>
```

```
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasThroughput -->
<owl:DatatypeProperty rdf:about="#hasThroughput">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#QoS"/>
    <rdfs:range rdf:resource="&xsd;float"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#hasTransitDelay -->
<owl:DatatypeProperty rdf:about="#hasTransitDelay">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Transit_Delay"/>
    <rdfs:range rdf:resource="&xsd;time"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#initialValue -->
<owl:DatatypeProperty rdf:about="#initialValue">
    <rdfs:domain rdf:resource="#Loc"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#invocable -->
<owl:DatatypeProperty rdf:about="#invocable">
    <rdfs:domain rdf:resource="#CompositeProcess"/>
    <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#name -->
<owl:DatatypeProperty rdf:about="#name">
    <rdfs:domain rdf:resource="#Process"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#numProcessor -->
<owl:DatatypeProperty rdf:about="#numProcessor">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="#ProcessorCount"/>
    <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#parameterType -->
<owl:DatatypeProperty rdf:about="#parameterType">
    <rdfs:domain rdf:resource="#ProcessVariable"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#serviceName -->
<owl:DatatypeProperty rdf:about="#serviceName">
    <rdfs:domain rdf:resource="#Profile"/>
</owl:DatatypeProperty>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#textDescription -->
<owl:DatatypeProperty rdf:about="#textDescription">
    <rdfs:domain rdf:resource="#Profile"/>
</owl:DatatypeProperty>

<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ARQ -->
<owl:Class rdf:about="#ARQ">
    <rdfs:subClassOf rdf:resource="#ErrorCorrection"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ASCII -->
<owl:Class rdf:about="#ASCII">
    <rdfs:subClassOf rdf:resource="#CharactersStandards"/>
    <owl:disjointWith rdf:resource="#EBCDIC"/>
    <owl:disjointWith rdf:resource="#UNICODE"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ATM -->
<owl:Class rdf:about="#ATM">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer2"/>
    <owl:disjointWith rdf:resource="#Ethernet"/>
```

```
    <owl:disjointWith rdf:resource="#FrameRelay"/>
    <owl:disjointWith rdf:resource="#HDLC"/>
    <owl:disjointWith rdf:resource="#IEEE_802.11"/>
    <owl:disjointWith rdf:resource="#PPP"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#AccessControl -->
<owl:Class rdf:about="#AccessControl">
    <rdfs:subClassOf rdf:resource="#Overlay"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasAutentication"/>
            <owl:someValuesFrom rdf:resource="#Autentication"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#HasAutorization"/>
            <owl:someValuesFrom rdf:resource="#Autorization"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Accuracy -->
<owl:Class rdf:about="#Accuracy">
    <rdfs:subClassOf rdf:resource="#Size"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ActiveAttacks -->
<owl:Class rdf:about="#ActiveAttacks">
    <rdfs:subClassOf rdf:resource="#Attacks"/>
    <owl:disjointWith rdf:resource="#PassiveAttacks"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Actuator -->
<owl:Class rdf:about="#Actuator">
    <rdfs:subClassOf rdf:resource="#Node"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasNetDev"/>
            <owl:someValuesFrom rdf:resource="#NetDevice"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasTransmissionMedium"/>
            <owl:someValuesFrom rdf:resource="#TransmissionMedium"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#NetDevice"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Addressing -->
<owl:Class rdf:about="#Addressing">
    <rdfs:subClassOf rdf:resource="#NetworkLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#AlgoCompression -->
<owl:Class rdf:about="#AlgoCompression">
    <rdfs:subClassOf rdf:resource="#Compression"/>
    <owl:disjointWith rdf:resource="#AudioCompression"/>
    <owl:disjointWith rdf:resource="#Encoding_and_Decoding_Algorithms"/>
    <owl:disjointWith rdf:resource="#ImageCompression"/>
    <owl:disjointWith rdf:resource="#VideoCompression"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Antenna -->
<owl:Class rdf:about="#Antenna">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Satellite"/>
                <rdf:Description rdf:about="#Wireless"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#Hardware"/>
    <rdfs:subClassOf>
```

```
            <owl:Restriction>
                <owl:onProperty rdf:resource="#TipologyAntenna"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#Standard"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#Transceiver"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <owl:disjointWith rdf:resource="#Battery"/>
        <owl:disjointWith rdf:resource="#Interface"/>
        <owl:disjointWith rdf:resource="#Memory"/>
        <owl:disjointWith rdf:resource="#ProcessorCount"/>
        <owl:disjointWith rdf:resource="#TransmissionMedium"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#AntennaParabolic -->
    <owl:Class rdf:about="#AntennaParabolic">
        <rdfs:subClassOf rdf:resource="#Antenna"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#AntennaRadio -->
    <owl:Class rdf:about="#AntennaRadio">
        <rdfs:subClassOf rdf:resource="#Antenna"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ApplicationLayer -->
    <owl:Class rdf:about="#ApplicationLayer">
        <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#AtomicProcess -->
    <owl:Class rdf:about="#AtomicProcess">
        <rdfs:subClassOf rdf:resource="#Process"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Attacks -->
    <owl:Class rdf:about="#Attacks">
        <rdfs:subClassOf rdf:resource="#MechanismSecurity"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Audio -->
    <owl:Class rdf:about="#Audio">
        <rdfs:subClassOf rdf:resource="#Sensor"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#AudioCompression -->
    <owl:Class rdf:about="#AudioCompression">
        <rdfs:subClassOf rdf:resource="#Compression"/>
        <owl:disjointWith rdf:resource="#Encoding_and_Decoding_Algorithms"/>
        <owl:disjointWith rdf:resource="#ImageCompression"/>
        <owl:disjointWith rdf:resource="#VideoCompression"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Autentication -->
    <owl:Class rdf:about="#Autentication">
        <rdfs:subClassOf rdf:resource="#Functionality"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasTransport"/>
                <owl:someValuesFrom rdf:resource="#ProtocolLayer5"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#AutenticationProtocol -->
    <owl:Class rdf:about="#AutenticationProtocol">
        <rdfs:subClassOf rdf:resource="#Autentication"/>
    </owl:Class>
```

```
    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Autorization -->
    <owl:Class rdf:about="#Autorization">
        <rdfs:subClassOf rdf:resource="#MechanismSecurity"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Availability -->
    <owl:Class rdf:about="#Availability">
        <rdfs:subClassOf rdf:resource="#SPDAttribute"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#BGP -->
    <owl:Class rdf:about="#BGP">
        <rdfs:subClassOf rdf:resource="#EGP"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Bandwidth -->
    <owl:Class rdf:about="#Bandwidth">
        <rdfs:subClassOf rdf:resource="#NetworkManagement"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasBandwidth"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Battery -->
    <owl:Class rdf:about="#Battery">
        <rdfs:subClassOf rdf:resource="#Hardware"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#PowerGenerator"/>
                <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">2</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <owl:disjointWith rdf:resource="#Interface"/>
        <owl:disjointWith rdf:resource="#Memory"/>
        <owl:disjointWith rdf:resource="#ProcessorCount"/>
        <owl:disjointWith rdf:resource="#TransmissionMedium"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#BinaryData -->
    <owl:Class rdf:about="#BinaryData">
        <rdfs:subClassOf rdf:resource="#InformationRapresentation"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#BiometricsDevice -->
    <owl:Class rdf:about="#BiometricsDevice">
        <rdfs:subClassOf rdf:resource="#Autentication"/>
        <owl:disjointWith rdf:resource="#OTPTokens"/>
        <owl:disjointWith rdf:resource="#Passwords"/>
        <owl:disjointWith rdf:resource="#SmartCards"/>
        <owl:disjointWith rdf:resource="#USBTokens"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Bits -->
    <owl:Class rdf:about="#Bits">
        <rdfs:subClassOf rdf:resource="#StorageUnits"/>
        <owl:disjointWith rdf:resource="#Bytes"/>
        <owl:disjointWith rdf:resource="#Words"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Bluetooth -->
    <owl:Class rdf:about="#Bluetooth">
        <rdfs:subClassOf rdf:resource="#ProtocolLayer1"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Bridge -->
    <owl:Class rdf:about="#Bridge">
        <rdfs:subClassOf rdf:resource="#NetDevice"/>
        <owl:disjointWith rdf:resource="#Gateway"/>
        <owl:disjointWith rdf:resource="#Hub"/>
        <owl:disjointWith rdf:resource="#Repeater"/>
        <owl:disjointWith rdf:resource="#Router"/>
        <owl:disjointWith rdf:resource="#Switch"/>
    </owl:Class>
```

```
<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Broadcast -->
<owl:Class rdf:about="#Broadcast">
    <rdfs:subClassOf rdf:resource="#TopologyNetwork"/>
    <owl:disjointWith rdf:resource="#Point-to-point"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#BufferOverFlowAttacks -->
<owl:Class rdf:about="#BufferOverFlowAttacks">
    <rdfs:subClassOf rdf:resource="#ActiveAttacks"/>
    <owl:disjointWith rdf:resource="#DenialOfService"/>
    <owl:disjointWith rdf:resource="#ForcedEntry"/>
    <owl:disjointWith rdf:resource="#Malware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#BufferTuning -->
<owl:Class rdf:about="#BufferTuning">
    <rdfs:subClassOf rdf:resource="#CongestionAvoidance"/>
    <owl:disjointWith rdf:resource="#WRED"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Bus -->
<owl:Class rdf:about="#Bus">
    <rdfs:subClassOf rdf:resource="#Broadcast"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Bytes -->
<owl:Class rdf:about="#Bytes">
    <rdfs:subClassOf rdf:resource="#StorageUnits"/>
    <owl:disjointWith rdf:resource="#Words"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CRC -->
<owl:Class rdf:about="#CRC">
    <rdfs:subClassOf rdf:resource="#ErrorDetecting"/>
    <rdfs:subClassOf rdf:resource="#FEC"/>
    <owl:disjointWith rdf:resource="#FEC"/>
    <owl:disjointWith rdf:resource="#HammingCode"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Can -->
<owl:Class rdf:about="#Can">
    <rdfs:subClassOf rdf:resource="#TipologyNetwork"/>
    <owl:disjointWith rdf:resource="#Lan"/>
    <owl:disjointWith rdf:resource="#Man"/>
    <owl:disjointWith rdf:resource="#VPN"/>
    <owl:disjointWith rdf:resource="#Wan"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Character -->
<owl:Class rdf:about="#Character">
    <rdfs:subClassOf rdf:resource="#InformationRapresentation"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CharactersStandards -->
<owl:Class rdf:about="#CharactersStandards">
    <rdfs:subClassOf rdf:resource="#Character"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Checksums -->
<owl:Class rdf:about="#Checksums">
    <rdfs:subClassOf rdf:resource="#IntegrityChecking"/>
    <owl:disjointWith rdf:resource="#CryptographicHashing"/>
    <owl:disjointWith rdf:resource="#ErrorCorrectingCodes"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CircuitSwitching -->
<owl:Class rdf:about="#CircuitSwitching">
    <rdfs:subClassOf rdf:resource="#SwitchingModes"/>
    <owl:disjointWith rdf:resource="#PacketSwitching"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Class_70 -->
<owl:Class rdf:about="#Class_70">
    <rdfs:subClassOf rdf:resource="#Autorization"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CollisionAttcaks -->
<owl:Class rdf:about="#CollisionAttcaks">
```

```
        <rdfs:subClassOf rdf:resource="#ForcedEntry"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CompiledCode -->
    <owl:Class rdf:about="#CompiledCode">
        <rdfs:subClassOf rdf:resource="#BinaryData"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CompositeProcess -->
    <owl:Class rdf:about="#CompositeProcess">
        <rdfs:subClassOf rdf:resource="#Process"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#invocable"/>
                <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CompositeSPDFunctionality -->
    <owl:Class rdf:about="#CompositeSPDFunctionality">
        <rdfs:subClassOf rdf:resource="#SPDFunctionality"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#useConnectorTopology"/>
                <owl:onClass rdf:resource="#Connector"/>
                <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#isComposedByFunctionality"/>
                <owl:someValuesFrom rdf:resource="#SPDFunctionality"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Compressed -->
    <owl:Class rdf:about="#Compressed">
        <rdfs:subClassOf rdf:resource="#BinaryData"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Compression -->
    <owl:Class rdf:about="#Compression">
        <rdfs:subClassOf rdf:resource="#EncodingStandard"/>
        <rdfs:subClassOf rdf:resource="#PhysicalLayer"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ComunicationAndNetworking -->
    <owl:Class rdf:about="#ComunicationAndNetworking">
        <rdfs:subClassOf rdf:resource="#Functionality"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Condition -->
    <owl:Class rdf:about="#Condition">
        <rdfs:subClassOf rdf:resource="#Expression"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Confidentiality -->
    <owl:Class rdf:about="#Confidentiality">
        <rdfs:subClassOf rdf:resource="#SPDAttribute"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CongestionAvoidance -->
    <owl:Class rdf:about="#CongestionAvoidance">
        <rdfs:subClassOf rdf:resource="#CongestionManagement"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CongestionManagement -->
    <owl:Class rdf:about="#CongestionManagement">
        <rdfs:subClassOf rdf:resource="#QoS"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ConnectionManagement -->
    <owl:Class rdf:about="#ConnectionManagement">
        <rdfs:subClassOf rdf:resource="#TransportLayer"/>
```

```
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Connector -->
        <owl:Class rdf:about="#Connector">
            <rdfs:subClassOf rdf:resource="&owl;Thing"/>
            <rdfs:subClassOf>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#connectorType"/>
                    <owl:qualifiedCardinality
    rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
                    <owl:onDataRange rdf:resource="&xsd;string"/>
                </owl:Restriction>
            </rdfs:subClassOf>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CryptographicFault -->
        <owl:Class rdf:about="#CryptographicFault">
            <rdfs:subClassOf rdf:resource="#Fault"/>
            <owl:disjointWith rdf:resource="#HardwareFault"/>
            <owl:disjointWith rdf:resource="#MaliciousFault"/>
            <owl:disjointWith rdf:resource="#MistakeFault"/>
            <owl:disjointWith rdf:resource="#NaturalFault"/>
            <owl:disjointWith rdf:resource="#NetworkingProtocolFault"/>
            <owl:disjointWith rdf:resource="#SecurityPolicyFault"/>
            <owl:disjointWith rdf:resource="#SoftwareFault"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#CryptographicHashing -->
        <owl:Class rdf:about="#CryptographicHashing">
            <rdfs:subClassOf rdf:resource="#Autentication"/>
            <rdfs:subClassOf rdf:resource="#IntegrityChecking"/>
            <owl:disjointWith rdf:resource="#ErrorCorrectingCodes"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DNS -->
        <owl:Class rdf:about="#DNS">
            <rdfs:subClassOf rdf:resource="#ProtocolLayer5"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DWRR -->
        <owl:Class rdf:about="#DWRR">
            <rdfs:subClassOf rdf:resource="#SchedulingAlgorithms"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DataBuffer -->
        <owl:Class rdf:about="#DataBuffer">
            <rdfs:subClassOf rdf:resource="#FlowControl"/>
            <owl:disjointWith rdf:resource="#NetworkCongestion"/>
            <owl:disjointWith rdf:resource="#WindowingFlowControl"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DataGrouping -->
        <owl:Class rdf:about="#DataGrouping">
            <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Datagrams -->
        <owl:Class rdf:about="#Datagrams">
            <rdfs:subClassOf rdf:resource="#DataGrouping"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DatalinkLayer -->
        <owl:Class rdf:about="#DatalinkLayer">
            <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DealingWithFailure -->
        <owl:Class rdf:about="#DealingWithFailure">
            <rdfs:subClassOf rdf:resource="#FaultTolerance"/>
        </owl:Class>

        <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DenialOfService -->
        <owl:Class rdf:about="#DenialOfService">
            <rdfs:subClassOf rdf:resource="#ActiveAttacks"/>
            <owl:disjointWith rdf:resource="#ForcedEntry"/>
            <owl:disjointWith rdf:resource="#Malware"/>
        </owl:Class>
```

```
    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Dependability -->
    <owl:Class rdf:about="#Dependability">
        <rdfs:subClassOf rdf:resource="#SPDConcept"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Detection -->
    <owl:Class rdf:about="#Detection">
        <rdfs:subClassOf rdf:resource="#Intrusion"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DictionaryAttacks -->
    <owl:Class rdf:about="#DictionaryAttacks">
        <rdfs:subClassOf rdf:resource="#ForcedEntry"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Dinamic -->
    <owl:Class rdf:about="#Dinamic">
        <rdfs:subClassOf rdf:resource="#Routing"/>
        <owl:disjointWith rdf:resource="#Static"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DirectInputMessageMap -->
    <owl:Class rdf:about="#DirectInputMessageMap">
        <rdfs:subClassOf rdf:resource="#WsdlInputMessageMap"/>
        <owl:disjointWith rdf:resource="#XSLTInputMessageMap"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DirectOutputMessageMap -->
    <owl:Class rdf:about="#DirectOutputMessageMap">
        <rdfs:subClassOf rdf:resource="#WsdlOutputlMessageMap"/>
        <owl:disjointWith rdf:resource="#XSLTOutputMessageMap"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#DistantVector -->
    <owl:Class rdf:about="#DistantVector">
        <rdfs:subClassOf rdf:resource="#IGP"/>
        <owl:disjointWith rdf:resource="#Hybrid"/>
        <owl:disjointWith rdf:resource="#LinkState"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Djikstra -->
    <owl:Class rdf:about="#Djikstra">
        <rdfs:subClassOf rdf:resource="#Static"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#EBCDIC -->
    <owl:Class rdf:about="#EBCDIC">
        <rdfs:subClassOf rdf:resource="#CharactersStandards"/>
        <owl:disjointWith rdf:resource="#UNICODE"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#EGP -->
    <owl:Class rdf:about="#EGP">
        <rdfs:subClassOf rdf:resource="#Dinamic"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#EIGRP -->
    <owl:Class rdf:about="#EIGRP">
        <rdfs:subClassOf rdf:resource="#Hybrid"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Element -->
    <owl:Class rdf:about="#Element">
        <owl:equivalentClass>
            <owl:Class>
                <owl:intersectionOf rdf:parseType="Collection">
                    <owl:Class>
                        <owl:unionOf rdf:parseType="Collection">
                            <rdf:Description rdf:about="#Middleware"/>
                            <rdf:Description rdf:about="#Network"/>
                            <rdf:Description rdf:about="#Node"/>
                        </owl:unionOf>
                    </owl:Class>
                    <owl:Restriction>
                        <owl:onProperty rdf:resource="#OperativLevel"/>
                        <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
                    </owl:Restriction>
```

```
                </owl:intersectionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#isPartOf"/>
                <owl:onClass rdf:resource="#System"/>
                <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#EncodingAndModulation -->
    <owl:Class rdf:about="#EncodingAndModulation">
        <rdfs:subClassOf rdf:resource="#PhysicalLayer"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#EncodingStandard -->
    <owl:Class rdf:about="#EncodingStandard">
        <rdfs:subClassOf rdf:resource="#RapresentationData"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Encoding_and_Decoding_Algorithms
-->
    <owl:Class rdf:about="#Encoding_and_Decoding_Algorithms">
        <rdfs:subClassOf rdf:resource="#Compression"/>
        <owl:disjointWith rdf:resource="#ImageCompression"/>
        <owl:disjointWith rdf:resource="#VideoCompression"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Encrypted -->
    <owl:Class rdf:about="#Encrypted">
        <rdfs:subClassOf rdf:resource="#BinaryData"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Error -->
    <owl:Class rdf:about="#Error">
        <rdfs:subClassOf rdf:resource="#SPDThreat"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ErrorControl -->
    <owl:Class rdf:about="#ErrorControl">
        <rdfs:subClassOf rdf:resource="#Functionality"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ErrorCorrectingCodes -->
    <owl:Class rdf:about="#ErrorCorrectingCodes">
        <rdfs:subClassOf rdf:resource="#IntegrityChecking"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ErrorCorrection -->
    <owl:Class rdf:about="#ErrorCorrection">
        <rdfs:subClassOf rdf:resource="#FaultTolerance"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ErrorDetecting -->
    <owl:Class rdf:about="#ErrorDetecting">
        <rdfs:subClassOf rdf:resource="#FaultTolerance"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ErrorHandling -->
    <owl:Class rdf:about="#ErrorHandling">
        <rdfs:subClassOf rdf:resource="#FaultTolerance"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Ethernet -->
    <owl:Class rdf:about="#Ethernet">
        <rdfs:subClassOf rdf:resource="#ProtocolLayer2"/>
        <owl:disjointWith rdf:resource="#FrameRelay"/>
        <owl:disjointWith rdf:resource="#HDLC"/>
        <owl:disjointWith rdf:resource="#IEEE_802.11"/>
        <owl:disjointWith rdf:resource="#PPP"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ExceptionHandling -->
    <owl:Class rdf:about="#ExceptionHandling">
        <rdfs:subClassOf rdf:resource="#FaultTolerance"/>
```

```
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Existential -->
<owl:Class rdf:about="#Existential">
    <rdfs:subClassOf rdf:resource="#ProcessVariable"/>
    <owl:disjointWith rdf:resource="#Local"/>
    <owl:disjointWith rdf:resource="#Parameter"/>
    <owl:disjointWith rdf:resource="#Partecipant"/>
    <owl:disjointWith rdf:resource="#ResultVar"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Expression -->
<owl:Class rdf:about="#Expression">
    <rdfs:subClassOf rdf:resource="#Middleware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FEC -->
<owl:Class rdf:about="#FEC">
    <rdfs:subClassOf rdf:resource="#ErrorControl"/>
    <rdfs:subClassOf rdf:resource="#ErrorCorrection"/>
    <rdfs:subClassOf rdf:resource="#PhysicalLayer"/>
    <owl:disjointWith rdf:resource="#HammingCode"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FTP -->
<owl:Class rdf:about="#FTP">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer5"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Failure -->
<owl:Class rdf:about="#Failure">
    <rdfs:subClassOf rdf:resource="#SPDThreat"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Fault -->
<owl:Class rdf:about="#Fault">
    <rdfs:subClassOf rdf:resource="#SPDThreat"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FaultForecasting -->
<owl:Class rdf:about="#FaultForecasting">
    <rdfs:subClassOf rdf:resource="#SPDMean"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FaultPrevention -->
<owl:Class rdf:about="#FaultPrevention">
    <rdfs:subClassOf rdf:resource="#SPDMean"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FaultRemoval -->
<owl:Class rdf:about="#FaultRemoval">
    <rdfs:subClassOf rdf:resource="#SPDMean"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FaultTolerance -->
<owl:Class rdf:about="#FaultTolerance">
    <rdfs:subClassOf rdf:resource="#SPDMean"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Firewalls -->
<owl:Class rdf:about="#Firewalls">
    <rdfs:subClassOf rdf:resource="#PerimeterDefenses"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Firmware -->
<owl:Class rdf:about="#Firmware">
    <rdfs:subClassOf rdf:resource="#Software"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#FirmwareProgrammable"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#VersionFirmware"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
```

```
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#SDE"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Flooding -->
<owl:Class rdf:about="#Flooding">
    <rdfs:subClassOf rdf:resource="#Static"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FlowControl -->
<owl:Class rdf:about="#FlowControl">
    <rdfs:subClassOf rdf:resource="#TransportLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ForcedEntry -->
<owl:Class rdf:about="#ForcedEntry">
    <rdfs:subClassOf rdf:resource="#ActiveAttacks"/>
    <owl:disjointWith rdf:resource="#Malware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FrameRelay -->
<owl:Class rdf:about="#FrameRelay">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer2"/>
    <owl:disjointWith rdf:resource="#HDLC"/>
    <owl:disjointWith rdf:resource="#IEEE_802.11"/>
    <owl:disjointWith rdf:resource="#PPP"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Framing -->
<owl:Class rdf:about="#Framing">
    <rdfs:subClassOf rdf:resource="#MediumAccessControl"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Functionality -->
<owl:Class rdf:about="#Functionality">
    <rdfs:subClassOf rdf:resource="#SPDFunctionality"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#TypeService"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#isFunctionalityOf"/>
            <owl:someValuesFrom rdf:resource="#Element"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#ServiceName"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Gateway -->
<owl:Class rdf:about="#Gateway">
    <rdfs:subClassOf rdf:resource="#NetDevice"/>
    <owl:disjointWith rdf:resource="#Hub"/>
    <owl:disjointWith rdf:resource="#Repeater"/>
    <owl:disjointWith rdf:resource="#Router"/>
    <owl:disjointWith rdf:resource="#Switch"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Grounding -->
<owl:Class rdf:about="#Grounding">
    <rdfs:subClassOf rdf:resource="#ServiceGrounding"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#HDLC -->
<owl:Class rdf:about="#HDLC">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer2"/>
    <owl:disjointWith rdf:resource="#IEEE_802.11"/>
    <owl:disjointWith rdf:resource="#PPP"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#HFSC -->
```

```
<owl:Class rdf:about="#HFSC">
    <rdfs:subClassOf rdf:resource="#SchedulingAlgorithms"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#HTTP -->
<owl:Class rdf:about="#HTTP">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer5"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#HWFailure -->
<owl:Class rdf:about="#HWFailure">
    <rdfs:subClassOf rdf:resource="#Failure"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#HammingCode -->
<owl:Class rdf:about="#HammingCode">
    <rdfs:subClassOf rdf:resource="#ErrorControl"/>
    <rdfs:subClassOf rdf:resource="#ErrorCorrection"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Hardware -->
<owl:Class rdf:about="#Hardware">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Antenna"/>
                <rdf:Description rdf:about="#Battery"/>
                <rdf:Description rdf:about="#Interface"/>
                <rdf:Description rdf:about="#Memory"/>
                <rdf:Description rdf:about="#ProcessorCount"/>
                <rdf:Description rdf:about="#TransmissionMedium"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#OperativLevel"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#HardwareFault -->
<owl:Class rdf:about="#HardwareFault">
    <rdfs:subClassOf rdf:resource="#Fault"/>
    <owl:disjointWith rdf:resource="#MaliciousFault"/>
    <owl:disjointWith rdf:resource="#MistakeFault"/>
    <owl:disjointWith rdf:resource="#NaturalFault"/>
    <owl:disjointWith rdf:resource="#NetworkingProtocolFault"/>
    <owl:disjointWith rdf:resource="#SecurityPolicyFault"/>
    <owl:disjointWith rdf:resource="#SoftwareFault"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Hub -->
<owl:Class rdf:about="#Hub">
    <rdfs:subClassOf rdf:resource="#NetDevice"/>
    <owl:disjointWith rdf:resource="#Repeater"/>
    <owl:disjointWith rdf:resource="#Router"/>
    <owl:disjointWith rdf:resource="#Switch"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#HuffmanCoding -->
<owl:Class rdf:about="#HuffmanCoding">
    <rdfs:subClassOf rdf:resource="#AlgoCompression"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Human -->
<owl:Class rdf:about="#Human">
    <rdfs:subClassOf rdf:resource="#Actuator"/>
    <owl:disjointWith rdf:resource="#Robot"/>
    <owl:disjointWith rdf:resource="#Server"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Humidity -->
<owl:Class rdf:about="#Humidity">
    <rdfs:subClassOf rdf:resource="#Sensor"/>
</owl:Class>
```

```
<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Hybrid -->
<owl:Class rdf:about="#Hybrid">
    <rdfs:subClassOf rdf:resource="#IGP"/>
    <owl:disjointWith rdf:resource="#LinkState"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ICMP -->
<owl:Class rdf:about="#ICMP">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer3"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IEEE_802.11 -->
<owl:Class rdf:about="#IEEE_802.11">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer2"/>
    <owl:disjointWith rdf:resource="#PPP"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IEEE_802.x -->
<owl:Class rdf:about="#IEEE_802.x">
    <rdfs:subClassOf rdf:resource="#NetworkStandards"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IGP -->
<owl:Class rdf:about="#IGP">
    <rdfs:subClassOf rdf:resource="#Dinamic"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IGRP -->
<owl:Class rdf:about="#IGRP">
    <rdfs:subClassOf rdf:resource="#LinkState"/>
    <owl:disjointWith rdf:resource="#RIP"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IP -->
<owl:Class rdf:about="#IP">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer3"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasTransport"/>
            <owl:allValuesFrom rdf:resource="#TCP"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasNetDev"/>
            <owl:allValuesFrom>
                <owl:Class>
                    <owl:unionOf rdf:parseType="Collection">
                        <rdf:Description rdf:about="#Gateway"/>
                        <rdf:Description rdf:about="#Router"/>
                    </owl:unionOf>
                </owl:Class>
            </owl:allValuesFrom>
        </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#X.25"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IPMobilitySupport -->
<owl:Class rdf:about="#IPMobilitySupport">
    <rdfs:subClassOf rdf:resource="#Roaming"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IPsec -->
<owl:Class rdf:about="#IPsec">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer3"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IPv4 -->
<owl:Class rdf:about="#IPv4">
    <rdfs:subClassOf rdf:resource="#IP"/>
    <owl:disjointWith rdf:resource="#IPv6"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IPv6 -->
<owl:Class rdf:about="#IPv6">
    <rdfs:subClassOf rdf:resource="#IP"/>
```

```
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ISDN -->
<owl:Class rdf:about="#ISDN">
    <rdfs:subClassOf rdf:resource="#CircuitSwitching"/>
    <owl:disjointWith rdf:resource="#PBX"/>
    <owl:disjointWith rdf:resource="#POTS"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ISO_Model -->
<owl:Class rdf:about="#ISO_Model">
    <rdfs:subClassOf rdf:resource="#NetworkStandards"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ImageCompression -->
<owl:Class rdf:about="#ImageCompression">
    <rdfs:subClassOf rdf:resource="#Compression"/>
    <owl:disjointWith rdf:resource="#VideoCompression"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#InformationRapresentation -->
<owl:Class rdf:about="#InformationRapresentation">
    <rdfs:subClassOf rdf:resource="#RapresentationData"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Input -->
<owl:Class rdf:about="#Input">
    <rdfs:subClassOf rdf:resource="#Parameter"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#InputMessageMap -->
<owl:Class rdf:about="#InputMessageMap">
    <rdfs:subClassOf rdf:resource="#MessageMap"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#owlsParameter"/>
            <owl:allValuesFrom rdf:resource="#Input"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Integer -->
<owl:Class rdf:about="#Integer">
    <rdfs:subClassOf rdf:resource="#Numeric"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Integrity -->
<owl:Class rdf:about="#Integrity">
    <rdfs:subClassOf rdf:resource="#SPDAttribute"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#IntegrityChecking -->
<owl:Class rdf:about="#IntegrityChecking">
    <rdfs:subClassOf rdf:resource="#MechanismSecurity"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Interface -->
<owl:Class rdf:about="#Interface">
    <rdfs:subClassOf rdf:resource="#Hardware"/>
    <owl:disjointWith rdf:resource="#Memory"/>
    <owl:disjointWith rdf:resource="#ProcessorCount"/>
    <owl:disjointWith rdf:resource="#TransmissionMedium"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Intrusion -->
<owl:Class rdf:about="#Intrusion">
    <rdfs:subClassOf rdf:resource="#MechanismSecurity"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Jitter -->
<owl:Class rdf:about="#Jitter">
    <rdfs:subClassOf rdf:resource="#QoS"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Kerberos -->
<owl:Class rdf:about="#Kerberos">
    <rdfs:subClassOf rdf:resource="#AutenticationProtocol"/>
</owl:Class>
```

```
<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Key -->
<owl:Class rdf:about="#Key">
    <rdfs:subClassOf rdf:resource="#MechanismSecurity"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasKey"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#KeyFormat -->
<owl:Class rdf:about="#KeyFormat">
    <rdfs:subClassOf rdf:resource="#Key"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#KeyInformationProtocol -->
<owl:Class rdf:about="#KeyInformationProtocol">
    <rdfs:subClassOf rdf:resource="#KeyProtocol"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#KeyProtocol -->
<owl:Class rdf:about="#KeyProtocol">
    <rdfs:subClassOf rdf:resource="#Key"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#KeyRegistrationProtocol -->
<owl:Class rdf:about="#KeyRegistrationProtocol">
    <rdfs:subClassOf rdf:resource="#KeyProtocol"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Lan -->
<owl:Class rdf:about="#Lan">
    <rdfs:subClassOf rdf:resource="#TipologyNetwork"/>
    <owl:disjointWith rdf:resource="#Man"/>
    <owl:disjointWith rdf:resource="#VPN"/>
    <owl:disjointWith rdf:resource="#Wan"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Latency -->
<owl:Class rdf:about="#Latency">
    <rdfs:subClassOf rdf:resource="#QoS"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Layer2Switching -->
<owl:Class rdf:about="#Layer2Switching">
    <rdfs:subClassOf rdf:resource="#DatalinkLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#LeakyBucket -->
<owl:Class rdf:about="#LeakyBucket">
    <rdfs:subClassOf rdf:resource="#TrafficShaping"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Link -->
<owl:Class rdf:about="#Link">
    <rdfs:subClassOf rdf:resource="#Local"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#LinkEfficency -->
<owl:Class rdf:about="#LinkEfficency">
    <rdfs:subClassOf rdf:resource="#QoS"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#LinkState -->
<owl:Class rdf:about="#LinkState">
    <rdfs:subClassOf rdf:resource="#IGP"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Loc -->
<owl:Class rdf:about="#Loc">
    <rdfs:subClassOf rdf:resource="#Local"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Local -->
<owl:Class rdf:about="#Local">
    <owl:equivalentClass>
```

```
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="#Link"/>
                    <rdf:Description rdf:about="#Loc"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf rdf:resource="#ProcessVariable"/>
        <owl:disjointWith rdf:resource="#Parameter"/>
        <owl:disjointWith rdf:resource="#Partecipant"/>
        <owl:disjointWith rdf:resource="#ResultVar"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MAC -->
    <owl:Class rdf:about="#MAC">
        <rdfs:subClassOf rdf:resource="#CryptographicHashing"/>
        <owl:disjointWith rdf:resource="#MD5"/>
        <owl:disjointWith rdf:resource="#SHA-1"/>
        <owl:disjointWith rdf:resource="#SHA-2"/>
        <owl:disjointWith rdf:resource="#SHA-3"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MACLayerAddressing -->
    <owl:Class rdf:about="#MACLayerAddressing">
        <rdfs:subClassOf rdf:resource="#MediumAccessControl"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MD5 -->
    <owl:Class rdf:about="#MD5">
        <rdfs:subClassOf rdf:resource="#CryptographicHashing"/>
        <owl:disjointWith rdf:resource="#SHA-1"/>
        <owl:disjointWith rdf:resource="#SHA-2"/>
        <owl:disjointWith rdf:resource="#SHA-3"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Maintainability -->
    <owl:Class rdf:about="#Maintainability">
        <rdfs:subClassOf rdf:resource="#SPDAttribute"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MaliciousFault -->
    <owl:Class rdf:about="#MaliciousFault">
        <rdfs:subClassOf rdf:resource="#Fault"/>
        <owl:disjointWith rdf:resource="#MistakeFault"/>
        <owl:disjointWith rdf:resource="#NaturalFault"/>
        <owl:disjointWith rdf:resource="#NetworkingProtocolFault"/>
        <owl:disjointWith rdf:resource="#SecurityPolicyFault"/>
        <owl:disjointWith rdf:resource="#SoftwareFault"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Malware -->
    <owl:Class rdf:about="#Malware">
        <rdfs:subClassOf rdf:resource="#ActiveAttacks"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Man -->
    <owl:Class rdf:about="#Man">
        <rdfs:subClassOf rdf:resource="#TipologyNetwork"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasTransmissionMedium"/>
                <owl:someValuesFrom>
                    <owl:Class>
                        <owl:unionOf rdf:parseType="Collection">
                            <rdf:Description rdf:about="#OpticalFiber"/>
                            <rdf:Description rdf:about="#Wired"/>
                        </owl:unionOf>
                    </owl:Class>
                </owl:someValuesFrom>
            </owl:Restriction>
        </rdfs:subClassOf>
        <owl:disjointWith rdf:resource="#VPN"/>
        <owl:disjointWith rdf:resource="#Wan"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ManInTheMiddle -->
    <owl:Class rdf:about="#ManInTheMiddle">
        <rdfs:subClassOf rdf:resource="#PassiveAttacks"/>
```

```
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MechanismSecurity -->
<owl:Class rdf:about="#MechanismSecurity">
    <rdfs:subClassOf rdf:resource="#Functionality"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MediumAccessControl -->
<owl:Class rdf:about="#MediumAccessControl">
    <rdfs:subClassOf rdf:resource="#DatalinkLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MediumInterfacing -->
<owl:Class rdf:about="#MediumInterfacing">
    <rdfs:subClassOf rdf:resource="#MediumAccessControl"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Memory -->
<owl:Class rdf:about="#Memory">
    <rdfs:subClassOf rdf:resource="#Hardware"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#SizeMemory"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#ProcessorCount"/>
    <owl:disjointWith rdf:resource="#TransmissionMedium"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Mesh -->
<owl:Class rdf:about="#Mesh">
    <rdfs:subClassOf rdf:resource="#Point-to-point"/>
    <owl:disjointWith rdf:resource="#Ring"/>
    <owl:disjointWith rdf:resource="#Star"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MessageMap -->
<owl:Class rdf:about="#MessageMap">
    <rdfs:subClassOf rdf:resource="#Middleware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Middleware -->
<owl:Class rdf:about="#Middleware">
    <rdfs:subClassOf rdf:resource="#Element"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MistakeFault -->
<owl:Class rdf:about="#MistakeFault">
    <rdfs:subClassOf rdf:resource="#Fault"/>
    <owl:disjointWith rdf:resource="#NaturalFault"/>
    <owl:disjointWith rdf:resource="#NetworkingProtocolFault"/>
    <owl:disjointWith rdf:resource="#SecurityPolicyFault"/>
    <owl:disjointWith rdf:resource="#SoftwareFault"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#MultiProcessor -->
<owl:Class rdf:about="#MultiProcessor">
    <rdfs:subClassOf rdf:resource="#ProcessorCount"/>
    <owl:disjointWith rdf:resource="#SingleProcessor"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Multiplexing -->
<owl:Class rdf:about="#Multiplexing">
    <rdfs:subClassOf rdf:resource="#MediumAccessControl"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NaturalFault -->
<owl:Class rdf:about="#NaturalFault">
    <rdfs:subClassOf rdf:resource="#Fault"/>
    <owl:disjointWith rdf:resource="#NetworkingProtocolFault"/>
    <owl:disjointWith rdf:resource="#SecurityPolicyFault"/>
    <owl:disjointWith rdf:resource="#SoftwareFault"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NetDevice -->
<owl:Class rdf:about="#NetDevice">
    <rdfs:subClassOf rdf:resource="#Node"/>
```

```
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasAntenna"/>
                <owl:someValuesFrom rdf:resource="#Antenna"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <owl:disjointWith rdf:resource="#Sensor"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Network -->
    <owl:Class rdf:about="#Network">
        <rdfs:subClassOf rdf:resource="#Element"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasFunctionality"/>
                <owl:someValuesFrom rdf:resource="#Functionality"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#isNetOf"/>
                <owl:onClass rdf:resource="#Node"/>
                <owl:minQualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">2</owl:minQualifiedCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#IDNetwork"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NetworkCongestion -->
    <owl:Class rdf:about="#NetworkCongestion">
        <rdfs:subClassOf rdf:resource="#FlowControl"/>
        <owl:disjointWith rdf:resource="#WindowingFlowControl"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NetworkLayer -->
    <owl:Class rdf:about="#NetworkLayer">
        <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasRouting"/>
                <owl:allValuesFrom rdf:resource="#Routing"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NetworkManagement -->
    <owl:Class rdf:about="#NetworkManagement">
        <rdfs:subClassOf rdf:resource="#Functionality"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NetworkStandards -->
    <owl:Class rdf:about="#NetworkStandards">
        <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NetworkingProtocolFault -->
    <owl:Class rdf:about="#NetworkingProtocolFault">
        <rdfs:subClassOf rdf:resource="#Fault"/>
        <owl:disjointWith rdf:resource="#SecurityPolicyFault"/>
        <owl:disjointWith rdf:resource="#SoftwareFault"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Node -->
    <owl:Class rdf:about="#Node">
        <rdfs:subClassOf rdf:resource="#Element"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasServiceModel"/>
                <owl:someValuesFrom rdf:resource="#ServiceModel"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
```

```
            <owl:Restriction>
                <owl:onProperty rdf:resource="#StateNode"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasBattery"/>
                <owl:someValuesFrom rdf:resource="#Battery"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasSoftware"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasNet"/>
                <owl:someValuesFrom rdf:resource="#Network"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasFunctionality"/>
                <owl:someValuesFrom rdf:resource="#Functionality"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#NodeBackup"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#IDnode"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NonInteger -->
    <owl:Class rdf:about="#NonInteger">
        <rdfs:subClassOf rdf:resource="#Numeric"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#NotProgrammable -->
    <owl:Class rdf:about="#NotProgrammable">
        <rdfs:subClassOf rdf:resource="#Firmware"/>
        <owl:disjointWith rdf:resource="#Programmable"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Numeric -->
    <owl:Class rdf:about="#Numeric">
        <rdfs:subClassOf rdf:resource="#InformationRapresentation"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#OSPF -->
    <owl:Class rdf:about="#OSPF">
        <rdfs:subClassOf rdf:resource="#DistantVector"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#OTPTokens -->
    <owl:Class rdf:about="#OTPTokens">
        <rdfs:subClassOf rdf:resource="#Autentication"/>
        <owl:disjointWith rdf:resource="#Passwords"/>
        <owl:disjointWith rdf:resource="#SmartCards"/>
        <owl:disjointWith rdf:resource="#USBTokens"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#OpticalFiber -->
    <owl:Class rdf:about="#OpticalFiber">
        <rdfs:subClassOf rdf:resource="#TransmissionMedium"/>
        <owl:disjointWith rdf:resource="#Satellite"/>
        <owl:disjointWith rdf:resource="#Wired"/>
        <owl:disjointWith rdf:resource="#Wireless"/>
```

```
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Output -->
<owl:Class rdf:about="#Output">
    <rdfs:subClassOf rdf:resource="#Parameter"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#OutputMessageMAp -->
<owl:Class rdf:about="#OutputMessageMAp">
    <rdfs:subClassOf rdf:resource="#MessageMap"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#owlsParameter"/>
            <owl:allValuesFrom rdf:resource="#Output"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Overlay -->
<owl:Class rdf:about="#Overlay">
    <rdfs:subClassOf rdf:resource="#Middleware"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasSO"/>
            <owl:someValuesFrom rdf:resource="#SystemOperative"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasMemory"/>
            <owl:someValuesFrom rdf:resource="#Memory"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasCPU"/>
            <owl:someValuesFrom rdf:resource="#ProcessorCount"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PBX -->
<owl:Class rdf:about="#PBX">
    <rdfs:subClassOf rdf:resource="#CircuitSwitching"/>
    <owl:disjointWith rdf:resource="#POTS"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#POTS -->
<owl:Class rdf:about="#POTS">
    <rdfs:subClassOf rdf:resource="#CircuitSwitching"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PPP -->
<owl:Class rdf:about="#PPP">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer2"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PacketFiltering -->
<owl:Class rdf:about="#PacketFiltering">
    <rdfs:subClassOf rdf:resource="#Firewalls"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PacketSwitching -->
<owl:Class rdf:about="#PacketSwitching">
    <rdfs:subClassOf rdf:resource="#SwitchingModes"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Parameter -->
<owl:Class rdf:about="#Parameter">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Input"/>
                <rdf:Description rdf:about="#Output"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ProcessVariable"/>
```

```
    <owl:disjointWith rdf:resource="#Partecipant"/>
    <owl:disjointWith rdf:resource="#ResultVar"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Parity -->
<owl:Class rdf:about="#Parity">
    <rdfs:subClassOf rdf:resource="#ErrorControl"/>
    <rdfs:subClassOf rdf:resource="#ErrorDetecting"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Partecipant -->
<owl:Class rdf:about="#Partecipant">
    <rdfs:subClassOf rdf:resource="#ProcessVariable"/>
    <owl:disjointWith rdf:resource="#ResultVar"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PassiveAttacks -->
<owl:Class rdf:about="#PassiveAttacks">
    <rdfs:subClassOf rdf:resource="#Attacks"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PasswordGuessing -->
<owl:Class rdf:about="#PasswordGuessing">
    <rdfs:subClassOf rdf:resource="#ForcedEntry"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Passwords -->
<owl:Class rdf:about="#Passwords">
    <rdfs:subClassOf rdf:resource="#Autentication"/>
    <owl:disjointWith rdf:resource="#SmartCards"/>
    <owl:disjointWith rdf:resource="#USBTokens"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PerimeterDefenses -->
<owl:Class rdf:about="#PerimeterDefenses">
    <rdfs:subClassOf rdf:resource="#MechanismSecurity"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PersonalFirewall -->
<owl:Class rdf:about="#PersonalFirewall">
    <rdfs:subClassOf rdf:resource="#Firewalls"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PhysicalLayer -->
<owl:Class rdf:about="#PhysicalLayer">
    <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Point-to-point -->
<owl:Class rdf:about="#Point-to-point">
    <rdfs:subClassOf rdf:resource="#TopologyNetwork"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#PointerToAnAddress -->
<owl:Class rdf:about="#PointerToAnAddress">
    <rdfs:subClassOf rdf:resource="#InformationRapresentation"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Precision -->
<owl:Class rdf:about="#Precision">
    <rdfs:subClassOf rdf:resource="#Size"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Pressure -->
<owl:Class rdf:about="#Pressure">
    <rdfs:subClassOf rdf:resource="#Sensor"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Prevention -->
<owl:Class rdf:about="#Prevention">
    <rdfs:subClassOf rdf:resource="#Intrusion"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Privacy -->
<owl:Class rdf:about="#Privacy">
    <rdfs:subClassOf rdf:resource="#SPDConcept"/>
</owl:Class>
```

```
    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Process -->
    <owl:Class rdf:about="#Process">
        <owl:equivalentClass>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="#AtomicProcess"/>
                    <rdf:Description rdf:about="#CompositeProcess"/>
                    <rdf:Description rdf:about="#SimpleProcess"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf rdf:resource="#ServiceModel"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasResult"/>
                <owl:someValuesFrom rdf:resource="#Result"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#name"/>
                <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasParameter"/>
                <owl:someValuesFrom rdf:resource="#Parameter"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProcessVariable -->
    <owl:Class rdf:about="#ProcessVariable">
        <owl:equivalentClass>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="#Existential"/>
                    <rdf:Description rdf:about="#Local"/>
                    <rdf:Description rdf:about="#Parameter"/>
                    <rdf:Description rdf:about="#Partecipant"/>
                    <rdf:Description rdf:about="#ResultVar"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf rdf:resource="#Variable"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#parameterType"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProcessorCount -->
    <owl:Class rdf:about="#ProcessorCount">
        <rdfs:subClassOf rdf:resource="#Hardware"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#numProcessor"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#Microchip"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <owl:disjointWith rdf:resource="#TransmissionMedium"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Profile -->
    <owl:Class rdf:about="#Profile">
        <rdfs:subClassOf rdf:resource="#ServiceProfile"/>
        <rdfs:subClassOf>
            <owl:Restriction>
```

```
            <owl:onProperty rdf:resource="#textDescription"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#serviceName"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Programmable -->
<owl:Class rdf:about="#Programmable">
    <rdfs:subClassOf rdf:resource="#Firmware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Programs -->
<owl:Class rdf:about="#Programs">
    <rdfs:subClassOf rdf:resource="#InformationRapresentation"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProtocolLayer1 -->
<owl:Class rdf:about="#ProtocolLayer1">
    <rdfs:subClassOf rdf:resource="#PhysicalLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProtocolLayer2 -->
<owl:Class rdf:about="#ProtocolLayer2">
    <rdfs:subClassOf rdf:resource="#DatalinkLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProtocolLayer3 -->
<owl:Class rdf:about="#ProtocolLayer3">
    <rdfs:subClassOf rdf:resource="#NetworkLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProtocolLayer4 -->
<owl:Class rdf:about="#ProtocolLayer4">
    <rdfs:subClassOf rdf:resource="#TransportLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProtocolLayer5 -->
<owl:Class rdf:about="#ProtocolLayer5">
    <rdfs:subClassOf rdf:resource="#ApplicationLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProtocolNetManagement -->
<owl:Class rdf:about="#ProtocolNetManagement">
    <rdfs:subClassOf rdf:resource="#NetworkManagement"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ProtocolTunneling -->
<owl:Class rdf:about="#ProtocolTunneling">
    <rdfs:subClassOf rdf:resource="#VirtualPrivateNetworks"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Proxy -->
<owl:Class rdf:about="#Proxy">
    <rdfs:subClassOf rdf:resource="#Firewalls"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#QoS -->
<owl:Class rdf:about="#QoS">
    <rdfs:subClassOf rdf:resource="#Functionality"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#QueueManagement -->
<owl:Class rdf:about="#QueueManagement">
    <rdfs:subClassOf rdf:resource="#QoS"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#RFID -->
<owl:Class rdf:about="#RFID">
    <rdfs:subClassOf rdf:resource="#Sensor"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#RIP -->
```

```
<owl:Class rdf:about="#RIP">
    <rdfs:subClassOf rdf:resource="#LinkState"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#RMON -->
<owl:Class rdf:about="#RMON">
    <rdfs:subClassOf rdf:resource="#ProtocolNetManagement"/>
    <owl:disjointWith rdf:resource="#SNMP"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Range -->
<owl:Class rdf:about="#Range">
    <rdfs:subClassOf rdf:resource="#Size"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#RapresentationData -->
<owl:Class rdf:about="#RapresentationData">
    <rdfs:subClassOf rdf:resource="#Functionality"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Reliability -->
<owl:Class rdf:about="#Reliability">
    <rdfs:subClassOf rdf:resource="#SPDAttribute"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Repeater -->
<owl:Class rdf:about="#Repeater">
    <rdfs:subClassOf rdf:resource="#NetDevice"/>
    <owl:disjointWith rdf:resource="#Router"/>
    <owl:disjointWith rdf:resource="#Switch"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Result -->
<owl:Class rdf:about="#Result">
    <rdfs:subClassOf rdf:resource="#Middleware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ResultVar -->
<owl:Class rdf:about="#ResultVar">
    <rdfs:subClassOf rdf:resource="#ProcessVariable"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Ring -->
<owl:Class rdf:about="#Ring">
    <rdfs:subClassOf rdf:resource="#Point-to-point"/>
    <owl:disjointWith rdf:resource="#Star"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Roaming -->
<owl:Class rdf:about="#Roaming">
    <rdfs:subClassOf rdf:resource="#NetworkLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Robot -->
<owl:Class rdf:about="#Robot">
    <rdfs:subClassOf rdf:resource="#Actuator"/>
    <owl:disjointWith rdf:resource="#Server"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Router -->
<owl:Class rdf:about="#Router">
    <rdfs:subClassOf rdf:resource="#NetDevice"/>
    <owl:disjointWith rdf:resource="#Switch"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Routing -->
<owl:Class rdf:about="#Routing">
    <rdfs:subClassOf rdf:resource="#NetworkLayer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SDE -->
<owl:Class rdf:about="#SDE">
    <rdfs:subClassOf rdf:resource="#Software"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SHA-1 -->
<owl:Class rdf:about="#SHA-1">
    <rdfs:subClassOf rdf:resource="#CryptographicHashing"/>
```

```
        <owl:disjointWith rdf:resource="#SHA-2"/>
        <owl:disjointWith rdf:resource="#SHA-3"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SHA-2 -->
    <owl:Class rdf:about="#SHA-2">
        <rdfs:subClassOf rdf:resource="#CryptographicHashing"/>
        <owl:disjointWith rdf:resource="#SHA-3"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SHA-3 -->
    <owl:Class rdf:about="#SHA-3">
        <rdfs:subClassOf rdf:resource="#CryptographicHashing"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SMTP -->
    <owl:Class rdf:about="#SMTP">
        <rdfs:subClassOf rdf:resource="#ProtocolLayer5"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SNMP -->
    <owl:Class rdf:about="#SNMP">
        <rdfs:subClassOf rdf:resource="#ProtocolNetManagement"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SPDAttribute -->
    <owl:Class rdf:about="#SPDAttribute">
        <owl:equivalentClass>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="#Confidentiality"/>
                    <rdf:Description rdf:about="#Integrity"/>
                    <rdf:Description rdf:about="#Maintainability"/>
                    <rdf:Description rdf:about="#Reliability"/>
                    <rdf:Description rdf:about="#Safety"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#assesses"/>
                <owl:someValuesFrom rdf:resource="#SPDConcept"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SPDComponent -->
    <owl:Class rdf:about="#SPDComponent">
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#provides"/>
                <owl:onClass rdf:resource="#SPDFunctionality"/>
                <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SPDConcept -->
    <owl:Class rdf:about="#SPDConcept">
        <owl:equivalentClass>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="#Dependability"/>
                    <rdf:Description rdf:about="#Privacy"/>
                    <rdf:Description rdf:about="#Security"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#isAssessedBy"/>
                <owl:someValuesFrom rdf:resource="#SPDAttribute"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#isIncreasedBy"/>
```

```
                <owl:someValuesFrom rdf:resource="#SPDMean"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#isAffectedBy"/>
                <owl:someValuesFrom rdf:resource="#SPDThreat"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SPDFunctionality -->
    <owl:Class rdf:about="#SPDFunctionality">
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#SPDStatus"/>
                <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
                <owl:onDataRange rdf:resource="&xsd;int"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#isPartofSPDFunctionality"/>
                <owl:someValuesFrom rdf:resource="#CompositeSPDFunctionality"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SPDMean -->
    <owl:Class rdf:about="#SPDMean">
        <owl:equivalentClass>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="#FaultForecasting"/>
                    <rdf:Description rdf:about="#FaultPrevention"/>
                    <rdf:Description rdf:about="#FaultRemoval"/>
                    <rdf:Description rdf:about="#FaultTolerance"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#increases"/>
                <owl:someValuesFrom rdf:resource="#SPDConcept"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SPDThreat -->
    <owl:Class rdf:about="#SPDThreat">
        <owl:equivalentClass>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="#Error"/>
                    <rdf:Description rdf:about="#Failure"/>
                    <rdf:Description rdf:about="#Fault"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#affects"/>
                <owl:someValuesFrom rdf:resource="#SPDConcept"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SWFailure -->
    <owl:Class rdf:about="#SWFailure">
        <rdfs:subClassOf rdf:resource="#Failure"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Safety -->
    <owl:Class rdf:about="#Safety">
        <rdfs:subClassOf rdf:resource="#SPDAttribute"/>
    </owl:Class>
```

```
    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Satellite -->
    <owl:Class rdf:about="#Satellite">
        <rdfs:subClassOf rdf:resource="#TransmissionMedium"/>
        <owl:disjointWith rdf:resource="#Wired"/>
        <owl:disjointWith rdf:resource="#Wireless"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SchedulingAlgorithms -->
    <owl:Class rdf:about="#SchedulingAlgorithms">
        <rdfs:subClassOf rdf:resource="#Bandwidth"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Security -->
    <owl:Class rdf:about="#Security">
        <rdfs:subClassOf rdf:resource="#SPDConcept"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SecurityPolicyFault -->
    <owl:Class rdf:about="#SecurityPolicyFault">
        <rdfs:subClassOf rdf:resource="#Fault"/>
        <owl:disjointWith rdf:resource="#SoftwareFault"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Sensor -->
    <owl:Class rdf:about="#Sensor">
        <rdfs:subClassOf rdf:resource="#Node"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasCPU"/>
                <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">2</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasMemory"/>
                <owl:someValuesFrom rdf:resource="#Memory"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasNetDev"/>
                <owl:someValuesFrom rdf:resource="#NetDevice"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasTransmissionMedium"/>
                <owl:someValuesFrom rdf:resource="#TransmissionMedium"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SensorGrounding -->
    <owl:Class rdf:about="#SensorGrounding">
        <rdfs:subClassOf rdf:resource="#Grounding"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Server -->
    <owl:Class rdf:about="#Server">
        <rdfs:subClassOf rdf:resource="#Actuator"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Service -->
    <owl:Class rdf:about="#Service">
        <rdfs:subClassOf rdf:resource="#Middleware"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#describedBy"/>
                <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ServiceGrounding -->
    <owl:Class rdf:about="#ServiceGrounding">
        <rdfs:subClassOf rdf:resource="#Middleware"/>
```

```
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#supportedBy"/>
                <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ServiceModel -->
    <owl:Class rdf:about="#ServiceModel">
        <rdfs:subClassOf rdf:resource="#Middleware"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#ServiceProfile -->
    <owl:Class rdf:about="#ServiceProfile">
        <rdfs:subClassOf rdf:resource="#Middleware"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasProfile"/>
                <owl:someValuesFrom rdf:resource="#Node"/>
            </owl:Restriction>
        </rdfs:subClassOf>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Service_Availability -->
    <owl:Class rdf:about="#Service_Availability">
        <rdfs:subClassOf rdf:resource="#QoS"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SimpleProcess -->
    <owl:Class rdf:about="#SimpleProcess">
        <rdfs:subClassOf rdf:resource="#Process"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SingleProcessor -->
    <owl:Class rdf:about="#SingleProcessor">
        <rdfs:subClassOf rdf:resource="#ProcessorCount"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Size -->
    <owl:Class rdf:about="#Size">
        <rdfs:subClassOf rdf:resource="#InformationRapresentation"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SmartCards -->
    <owl:Class rdf:about="#SmartCards">
        <rdfs:subClassOf rdf:resource="#Autentication"/>
        <owl:disjointWith rdf:resource="#USBTokens"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Sniffing -->
    <owl:Class rdf:about="#Sniffing">
        <rdfs:subClassOf rdf:resource="#PassiveAttacks"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Software -->
    <owl:Class rdf:about="#Software">
        <owl:equivalentClass>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#describedBy"/>
                <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
            </owl:Restriction>
        </owl:equivalentClass>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SoftwareFault -->
    <owl:Class rdf:about="#SoftwareFault">
        <rdfs:subClassOf rdf:resource="#Fault"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SourceRouting -->
    <owl:Class rdf:about="#SourceRouting">
        <rdfs:subClassOf rdf:resource="#Layer2Switching"/>
        <owl:disjointWith rdf:resource="#SpanningTree"/>
    </owl:Class>

    <!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SpanningTree -->
```

```
<owl:Class rdf:about="#SpanningTree">
    <rdfs:subClassOf rdf:resource="#Layer2Switching"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Spyware -->
<owl:Class rdf:about="#Spyware">
    <rdfs:subClassOf rdf:resource="#Malware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Star -->
<owl:Class rdf:about="#Star">
    <rdfs:subClassOf rdf:resource="#Point-to-point"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Static -->
<owl:Class rdf:about="#Static">
    <rdfs:subClassOf rdf:resource="#Routing"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#StorageUnits -->
<owl:Class rdf:about="#StorageUnits">
    <rdfs:subClassOf rdf:resource="#RapresentationData"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Streams -->
<owl:Class rdf:about="#Streams">
    <rdfs:subClassOf rdf:resource="#DataGrouping"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Switch -->
<owl:Class rdf:about="#Switch">
    <rdfs:subClassOf rdf:resource="#NetDevice"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SwitchingModes -->
<owl:Class rdf:about="#SwitchingModes">
    <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#System -->
<owl:Class rdf:about="#System">
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#provides"/>
            <owl:someValuesFrom rdf:resource="#SPDFunctionality"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#isComposedBy"/>
            <owl:someValuesFrom rdf:resource="#Element"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#SystemOperative -->
<owl:Class rdf:about="#SystemOperative">
    <rdfs:subClassOf rdf:resource="#SDE"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TCP -->
<owl:Class rdf:about="#TCP">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer4"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TCP/IP_Model -->
<owl:Class rdf:about="#TCP/IP_Model">
    <rdfs:subClassOf rdf:resource="#NetworkStandards"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TLS-SSL -->
<owl:Class rdf:about="#TLS-SSL">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer5"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Telnet -->
<owl:Class rdf:about="#Telnet">
```

```
    <rdfs:subClassOf rdf:resource="#ProtocolLayer5"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Temperature -->
<owl:Class rdf:about="#Temperature">
    <rdfs:subClassOf rdf:resource="#Sensor"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TheClient -->
<owl:Class rdf:about="#TheClient">
    <rdfs:subClassOf rdf:resource="#Partecipant"/>
    <owl:disjointWith rdf:resource="#TheServer"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TheServer -->
<owl:Class rdf:about="#TheServer">
    <rdfs:subClassOf rdf:resource="#Partecipant"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TipologyNetwork -->
<owl:Class rdf:about="#TipologyNetwork">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Can"/>
                <rdf:Description rdf:about="#Lan"/>
                <rdf:Description rdf:about="#Man"/>
                <rdf:Description rdf:about="#VPN"/>
                <rdf:Description rdf:about="#Wan"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#Network"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasNetDev"/>
            <owl:someValuesFrom rdf:resource="#NetDevice"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#TopologyNetwork"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TokenBucket -->
<owl:Class rdf:about="#TokenBucket">
    <rdfs:subClassOf rdf:resource="#TrafficShaping"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TopologyNetwork -->
<owl:Class rdf:about="#TopologyNetwork">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Broadcast"/>
                <rdf:Description rdf:about="#Point-to-point"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#Network"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TrafficShaping -->
<owl:Class rdf:about="#TrafficShaping">
    <rdfs:subClassOf rdf:resource="#Bandwidth"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Transit_Delay -->
<owl:Class rdf:about="#Transit_Delay">
    <rdfs:subClassOf rdf:resource="#QoS"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Transmission -->
<owl:Class rdf:about="#Transmission">
    <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TransmissionFailure -->
<owl:Class rdf:about="#TransmissionFailure">
    <rdfs:subClassOf rdf:resource="#Failure"/>
```

```
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TransmissionMedium -->
<owl:Class rdf:about="#TransmissionMedium">
    <rdfs:subClassOf rdf:resource="#Hardware"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#inverse_of_hasTransmissionMedium"/>
            <owl:someValuesFrom>
                <owl:Class>
                    <owl:unionOf rdf:parseType="Collection">
                        <rdf:Description rdf:about="#Network"/>
                        <rdf:Description rdf:about="#Node"/>
                    </owl:unionOf>
                </owl:Class>
            </owl:someValuesFrom>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TransmitEncryption -->
<owl:Class rdf:about="#TransmitEncryption">
    <rdfs:subClassOf rdf:resource="#Transmission"/>
    <owl:disjointWith rdf:resource="#TransmitPlainText"/>
    <owl:disjointWith rdf:resource="#TransmitSign"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TransmitPlainText -->
<owl:Class rdf:about="#TransmitPlainText">
    <rdfs:subClassOf rdf:resource="#Transmission"/>
    <owl:disjointWith rdf:resource="#TransmitSign"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TransmitSign -->
<owl:Class rdf:about="#TransmitSign">
    <rdfs:subClassOf rdf:resource="#Transmission"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#TransportLayer -->
<owl:Class rdf:about="#TransportLayer">
    <rdfs:subClassOf rdf:resource="#ComunicationAndNetworking"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Trojans -->
<owl:Class rdf:about="#Trojans">
    <rdfs:subClassOf rdf:resource="#Malware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#UDP -->
<owl:Class rdf:about="#UDP">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer4"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#UNICODE -->
<owl:Class rdf:about="#UNICODE">
    <rdfs:subClassOf rdf:resource="#CharactersStandards"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#USBTokens -->
<owl:Class rdf:about="#USBTokens">
    <rdfs:subClassOf rdf:resource="#Autentication"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#VPN -->
<owl:Class rdf:about="#VPN">
    <rdfs:subClassOf rdf:resource="#TipologyNetwork"/>
    <owl:disjointWith rdf:resource="#Wan"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Variable -->
<owl:Class rdf:about="#Variable">
    <rdfs:subClassOf rdf:resource="#Middleware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Vibration -->
<owl:Class rdf:about="#Vibration">
    <rdfs:subClassOf rdf:resource="#Sensor"/>
</owl:Class>
```

```
<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Video -->
<owl:Class rdf:about="#Video">
    <rdfs:subClassOf rdf:resource="#Sensor"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#VideoCompression -->
<owl:Class rdf:about="#VideoCompression">
    <rdfs:subClassOf rdf:resource="#Compression"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#VirtualPrivateNetworks -->
<owl:Class rdf:about="#VirtualPrivateNetworks">
    <rdfs:subClassOf rdf:resource="#MechanismSecurity"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Viruses -->
<owl:Class rdf:about="#Viruses">
    <rdfs:subClassOf rdf:resource="#Malware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WFQ -->
<owl:Class rdf:about="#WFQ">
    <rdfs:subClassOf rdf:resource="#SchedulingAlgorithms"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WRED -->
<owl:Class rdf:about="#WRED">
    <rdfs:subClassOf rdf:resource="#CongestionAvoidance"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WRR -->
<owl:Class rdf:about="#WRR">
    <rdfs:subClassOf rdf:resource="#SchedulingAlgorithms"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WSN -->
<owl:Class rdf:about="#WSN">
    <rdfs:subClassOf rdf:resource="#Sensor"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Wan -->
<owl:Class rdf:about="#Wan">
    <rdfs:subClassOf rdf:resource="#TipologyNetwork"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WindowingFlowControl -->
<owl:Class rdf:about="#WindowingFlowControl">
    <rdfs:subClassOf rdf:resource="#FlowControl"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Wired -->
<owl:Class rdf:about="#Wired">
    <rdfs:subClassOf rdf:resource="#TransmissionMedium"/>
    <owl:disjointWith rdf:resource="#Wireless"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Wireless -->
<owl:Class rdf:about="#Wireless">
    <rdfs:subClassOf rdf:resource="#TransmissionMedium"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Words -->
<owl:Class rdf:about="#Words">
    <rdfs:subClassOf rdf:resource="#StorageUnits"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Worms -->
<owl:Class rdf:about="#Worms">
    <rdfs:subClassOf rdf:resource="#Malware"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WsdlGounding -->
<owl:Class rdf:about="#WsdlGounding">
    <rdfs:subClassOf rdf:resource="#Grounding"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WsdlInputMessageMap -->
```

```
<owl:Class rdf:about="#WsdlInputMessageMap">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#DirectInputMessageMap"/>
                <rdf:Description rdf:about="#XSLTInputMessageMap"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#InputMessageMap"/>
    <rdfs:subClassOf rdf:resource="#WsdlMessageMap"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WsdlMessageMap -->
<owl:Class rdf:about="#WsdlMessageMap">
    <rdfs:subClassOf rdf:resource="#MessageMap"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#WsdlOutputlMessageMap -->
<owl:Class rdf:about="#WsdlOutputlMessageMap">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#DirectOutputMessageMap"/>
                <rdf:Description rdf:about="#XSLTOutputMessageMap"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#OutputMessageMAp"/>
    <rdfs:subClassOf rdf:resource="#WsdlMessageMap"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#X.25 -->
<owl:Class rdf:about="#X.25">
    <rdfs:subClassOf rdf:resource="#ProtocolLayer3"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#XSLTInputMessageMap -->
<owl:Class rdf:about="#XSLTInputMessageMap">
    <rdfs:subClassOf rdf:resource="#WsdlInputMessageMap"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#XSLTOutputMessageMap -->
<owl:Class rdf:about="#XSLTOutputMessageMap">
    <rdfs:subClassOf rdf:resource="#WsdlOutputlMessageMap"/>
</owl:Class>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#valueOf -->
<owl:Class rdf:about="#valueOf">
    <rdfs:subClassOf rdf:resource="#Middleware"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#theParam"/>
            <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.w3.org/2000/01/rdf-schema#Datatype -->
<owl:Class rdf:about="&rdfs;Datatype"/>

<!-- http://www.w3.org/2002/07/owl#Thing -->
<owl:Class rdf:about="&owl;Thing"/>

<!--
////////////////////////////////////////////////////////////////////////////////////
//
// Individuals
//
////////////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#Connector_1 -->
<owl:Thing rdf:about="#Connector_1">
    <connectorType rdf:datatype="&xsd;string">PARALLEL</connectorType>
</owl:Thing>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#FComposite_1 -->
```

```
<CompositeSPDFunctionality rdf:about="#FComposite_1">
    <rdf:type rdf:resource="&owl;Thing"/>
    <SPDStatus rdf:datatype="&xsd;int">30</SPDStatus>
    <useConnectorTopology rdf:resource="#Connector_1"/>
</CompositeSPDFunctionality>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_A_1 -->
<owl:Thing rdf:about="#F_A_1">
    <rdf:type rdf:resource="#Autorization"/>
    <SPDStatus rdf:datatype="&xsd;int">2</SPDStatus>
</owl:Thing>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_A_2 -->
<Autorization rdf:about="#F_A_2">
    <rdf:type rdf:resource="&owl;Thing"/>
    <SPDStatus rdf:datatype="&xsd;int">5</SPDStatus>
</Autorization>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_A_3 -->
<Autorization rdf:about="#F_A_3">
    <rdf:type rdf:resource="&owl;Thing"/>
    <SPDStatus rdf:datatype="&xsd;int">8</SPDStatus>
</Autorization>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_C_1 -->
<owl:Thing rdf:about="#F_C_1">
    <SPDStatus rdf:datatype="&xsd;int">6</SPDStatus>
</owl:Thing>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_C_2 -->
<owl:Thing rdf:about="#F_C_2">
    <SPDStatus rdf:datatype="&xsd;int">1</SPDStatus>
</owl:Thing>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_C_3 -->
<owl:Thing rdf:about="#F_C_3">
    <SPDStatus rdf:datatype="&xsd;int">3</SPDStatus>
</owl:Thing>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_I_1 -->
<Autentication rdf:about="#F_I_1">
    <rdf:type rdf:resource="&owl;Thing"/>
    <SPDStatus rdf:datatype="&xsd;int">1</SPDStatus>
</Autentication>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_I_2 -->
<owl:Thing rdf:about="#F_I_2">
    <rdf:type rdf:resource="#Autentication"/>
    <SPDStatus rdf:datatype="&xsd;int">1</SPDStatus>
</owl:Thing>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#F_I_3 -->
<Autentication rdf:about="#F_I_3">
    <rdf:type rdf:resource="&owl;Thing"/>
    <SPDStatus rdf:datatype="&xsd;int">8</SPDStatus>
</Autentication>

<!-- http://www.owl-ontologies.com/Ontology1300273978.owl#S -->
<System rdf:about="#S">
    <rdf:type rdf:resource="&owl;Thing"/>
    <provides rdf:resource="#FComposite_1"/>
</System>

<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// General axioms
//
///////////////////////////////////////////////////////////////////////////////////////
 -->
<rdf:Description>
    <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
    <owl:members rdf:parseType="Collection">
        <rdf:Description rdf:about="#FaultForecasting"/>
        <rdf:Description rdf:about="#FaultPrevention"/>
        <rdf:Description rdf:about="#FaultRemoval"/>
        <rdf:Description rdf:about="#FaultTolerance"/>
    </owl:members>
```

```
        </rdf:Description>
        <rdf:Description>
            <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
            <owl:members rdf:parseType="Collection">
                <rdf:Description rdf:about="#Error"/>
                <rdf:Description rdf:about="#Failure"/>
                <rdf:Description rdf:about="#Fault"/>
            </owl:members>
        </rdf:Description>
        <rdf:Description>
            <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
            <owl:members rdf:parseType="Collection">
                <rdf:Description rdf:about="#Availability"/>
                <rdf:Description rdf:about="#Confidentiality"/>
                <rdf:Description rdf:about="#Integrity"/>
                <rdf:Description rdf:about="#Maintainability"/>
                <rdf:Description rdf:about="#Reliability"/>
                <rdf:Description rdf:about="#Safety"/>
            </owl:members>
        </rdf:Description>
    </rdf:RDF>
    <!-- Generated by the OWL API (version 2.2.1.1138) http://owlapi.sourceforge.net -->
```