DEPONTO: A Reusable Dependability Domain Ontology

Teodora Sanislav

Technical University of Cluj-Napoca, Department of Automation, 26-28 G. Barițiu, 400027 Cluj-Napoca, Romania

***Abstract* – This paper proposes a dependability reusable ontology for knowledge representation. The fundamental knowledge related to dependability follows its taxonomy. Thus, this paper gives an analysis of what is the dependability domain ontology andof its components.The dependability domain ontology plays an important role in ensuring the dependability of information systems by providing support for their diagnosis in case of faults, errors and failures.The proposed ontology is used as a dependability framework in two case study Cyber-Physical Systemswhich demonstrate its reusability within this category of systems.**

***Keywords* – Ontology, Dependability, Knowledge representation, Cyber-Physical Systems.**

1. Introduction

According to the Oxford dictionary, ontology is "the branch of metaphysics dealing with the nature of being". This term has been adopted by the computer science society in the '90s in order to be used in artificial intelligence domain as a specification of a conceptualization [1], [2]. Over the years many definitions for ontology have been stated.Thus,ontology is a representation vocabulary specialized to a domain [3]. A more complex definition is given in paper [4]: "ontologies are formal artefacts that are designed to represent the knowledge related to a specific or generic domain in terms of the relevant concepts, relationships between these concepts and the instances of these concepts".Recently, ontologies have become important in many domains, and some of them, e.g. Semantic Web, rely on ontology evolution, which implies the continued development of new tools, techniques and methodsfor knowledge acquisition, knowledge representation, ontology validation, andontology-based reasoning.

The current generation of information systems, Cyber-Physical Systems (CPSs) [5], [6], must fulfil the dependability feature, which represents"the ability to deliver service that can justifiably be trusted" [7]. This type of systems has to deal with the following research challenges regarding dependability: new methods for threat identification, new fault-tolerant mechanisms, new test methods, and certification and validation metrics.In order to ensure the dependability of CPSs, an ontological analysis,which clarifies the structure of this feature's knowledge,is required.

The goal of this paper is to address the above mentioned CPSs research challenges by providing complete and detailed dependability domain ontology.The contributions are twofold. First, dependability domain ontology, named DEPONTO, is presentedand discussed, highlighting the main concepts, the relationships between them and their instances.Second, the proposed ontology is validated on two case study CPSsthrough queries, which demonstrate its reusability and flexibility.

The paper is structured as follows. Section 2 gives a brief description of the related previous work.Section 3 introduces the DEPONTO ontology, while Section 4 shows the reusability of the proposed ontology. Finally, the conclusions are listed in Section 5.

2. Related work

In the last two decades the scientific community has developed ontologieswhich can be classified according to the expressivity and formality of the languages used, or according to the scope of the ontologies. The first category includes information ontologies, linguistic/terminological ontologies, software ontologies, and formal ontologies. The second category comprises local/application ontologies, domain ontologies, core reference ontologies, general ontologies,and foundational/top level/upper level ontologies [8]. [8] providesa brief description of each type of ontology, the languages that describe them, and details somesuggestive examples. The list of the examples can be completed with the ontologies presented in [9], [10], [11] and many others.

In the context of CPSs several ontologies have been developed. Paper [12] introduces an upper level context ontology for modelling context in Cyber-Physical-Social Systems (CPSSs) which can be applied in the domain of self-organising resource network.The research work presented in [13] addresses the dependability concept by using a semantic ontology for the detection of errors in a CPS for water distribution network. Paper [14] focuses,also,on the challenges of CPSs modelling, discusses the technologies that address this challenge, and proposes the use of domain-specific ontologies to enhance the modularity of these systems. Therefore, the development of ontologiesas CPSs componentsrepresents an important task in order to model some aspects (e.g. modularity, heterogeneity, dependability)of this type of systems.

3. DEPONTO – Dependability domain ontology

The ontology proposed in this paper, named DEPONTO,aims to model CPSs from the dependability point of view. DEPONTO is a domain ontology. It describes the knowledge of the dependability domain using a declarative formalism. The set of dependability concepts, their relations and constraints are reflected in the representation of knowledge.The three main concepts of dependability refer to its taxonomy: (1) *Attributes* that define it (Availability,Reliability,Maintainability,Safety,Integrity, Confidentiality); (2) *Means* by which it is achieved (Fault Removal, Fault Tolerance,Fault Forecasting,Fault Prevention); (3) *Threats* it has to deal with (Fault,Error,Failure) [7], [15]. The *Faults* and *Failures*concepts play an important role in the dependability domain ontology. A fault is the adjudged or hypothesized cause of an error, while a failure is an instance in time when a system displays the behaviour that is contrary to its specification or the way in which a system can fail. Other important concepts of the ontology consist in: *Locations* of the system where the faults and the failures could appear; *Failure Effects*upon the system; and *Failure Modes* (Figure 1). *Faults* and *Failures* concepts could be supplemented with more subcategories according to [7]. As Figure 1 shows, the relations between the ontology concepts are subordination ones,except the relation between the *Faults* and *Failures* concepts which is causal. The *Faults* concept is defined through four constraints: fault cause (the changing of a system state), fault result (the new system state considering the propagation of faults), system's location where the fault occurred, and fault occurrence time [16].The *Failures* concept is defined through seven constraints: failure mode, effect of the failure over the entire system, cause of the failure, severity of the failure, occurrence of the failure within some time interval, possibility to detect the failure, and failure occurrence time [16].These constraints are defined based on aqualitative dependability analysis technique – Failure Mode and Effects Analysis (FMEA) [17]. FMEA identifies all the possible failure modes of a system, quantifies the risk level associated with each failure mode through values of severity, occurrence and detection, and identifies the necessary corrective actions.

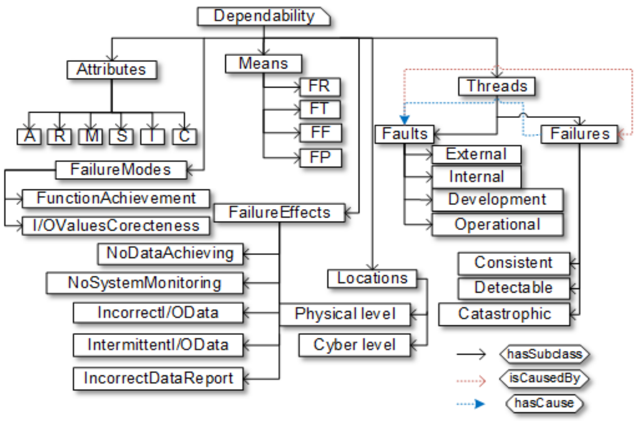


Figure 1.DEPONTO ontology main concepts

DEPONTO is also a formal ontology. A formal ontology requires formal logic to define the concepts and uses a formal language like Description Logics (DL), Web Ontology Language (OWL), First Order Logic (FOL), and Conceptual Graphs (CG). DEPONTO uses OWL to define the concepts of the ontology, their relations (classes and properties), and the instances (individuals with properties).OWL is also used for reasoning about the classes and individuals. DEPONTO is implemented in Protégé[18] and can be considered a knowledge base within CPSs. DEPONTO contains 49 classes and 10 object and data properties' definitions (Figure 2) for reasoning and not for retrieval and storage of data. The reasoning is based on axioms, which are combinations of concepts and semantic relations. Protégé allows the definition of DEPONTO classes instances.The next section of thepaper presents the instances of two CPSs and shows DEPONTO's reusability and validation.

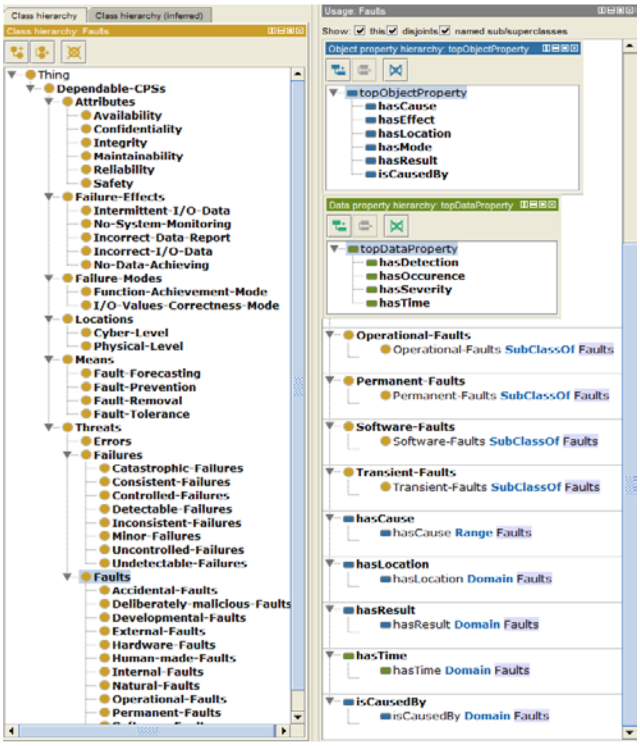


Figure 2. DEPONTO classes in Protégé

4. DEPONTO validation

CPSs are applied in many application domains, such as critical infrastructure control (energy, water), safe and efficient transport, environmental monitoring, telemedicine, manufacturing and agriculture, due to their features: cyber capabilities in physical components, high degrees of automation, dynamic reconfiguration and reorganization, large scale networking, and dependable and certified operation. The proposed ontology was validated on two case study CPSs from the above mentioned domains. The first one is a CPS which monitors and controls a water treatment plant [15].The second CPSmonitors the environment [16], [19]. Papers [15], [16], [19] provide a brief description of the CPSs architectures and present in detail the proposed dependability domain ontology in both cases. The instances of the ontology differ in the two cases because they are in close liaison with the CPSs components and with their FMEAs.

4.1. CPS for water treatment plant monitoring and control

DEPONTO drives the aspects of the water treatment plant CPS related to system diagnosis, and represents a comprehensive framework for ensuring the dependability of thissystem. DEPONTO works together with a multi-agent society composed of intelligent agents with distinctive tasks and implemented under the JADE platform.The entire agents' organization is presented in [20]. The Diagnostic Agent tests and diagnoses the system based on proactive and reactive decision algorithms, and using DEPONTO. The Intelligent Control Agent uses DEPONTO as a knowledge base in order to provide decision algorithms for the CPS control, self-reconfiguration and self-adaptation.

DEPONTO has more than 20 instances (individuals) as shown in Figure 3. There is an instance for each line of the FMEA table. The list of instances is completed with those of the *Failure-Effects*, *Failure-Modes* and *Faults*subclasses. Figure 3 presents the properties of one of these instances: *PH-Sensor-Out-of-Work-1*, which is an instance of the *Detectable-Failures* subclass, with the seven proprieties that characterize a failure.

The validation of the proposed dependability domain ontology, in the case of the CPS for water treatment plant monitoring and control, was performed through DL (Description Logics – a formal knowledge representation language) queries in Protégé, which are used by the Diagnostic Agent. The queries allow the listing of the classes with some properties, and of the classes' instances. Such examples are presented in Table 1.

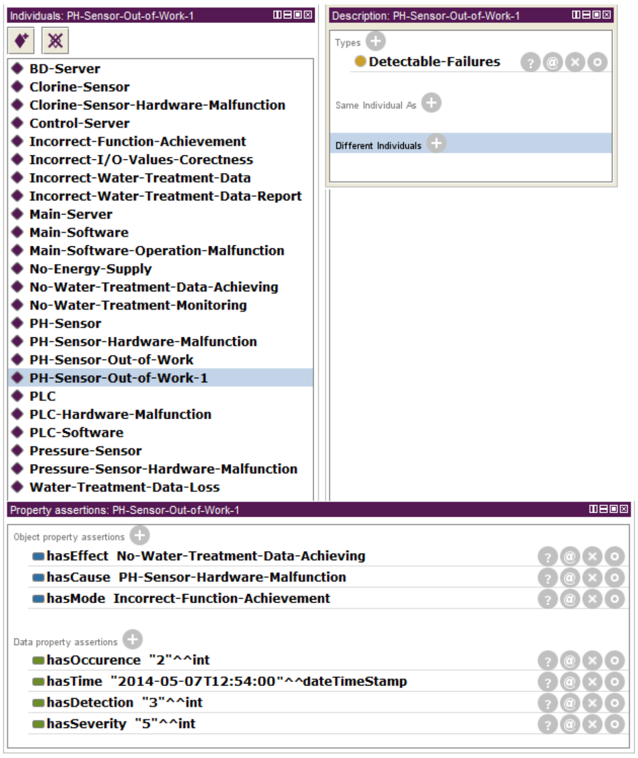
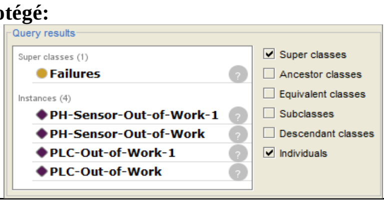


Figure 3. DEPONTO instances of the CPS for water treatment plant monitoring and control

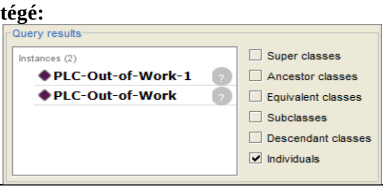
The DL queries are processed after a reasoner is started. Protégé works with HetmiT [21]. This reasoner classifies theDEPONTO ontology, determines its consistency, and identifies the relationship between classes.If the ontology is properly classified any queries can be performed.

Table 1.DL queries of the DEPONTO

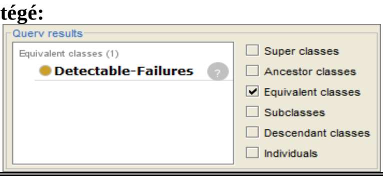
Query 1: Some detectable failures instances
Detectable-Failures
Result in Protégé:



Query 2: All instances with Severity=6 and Occurrence=2 and Detection=3
hasSeverity value 6 and hasOccurence value 2 and hasDetection value 3
Result in Protégé:



Query 3: Class equivalent with Failures and with Detection ≥ 3
Failures and hasDetection min 3
Result in Protégé:



The entire CPS solution based on a multi-agent society and DEPONTO as a knowledge base increases the *A(t)* (availability– dependability attribute) of the hardware and software components. Table 2 presents the *A(t)* values before and after theaddition of the proposed dependability domain ontology.

Table 2.A(t) values of the case study CPS hardware and software components

Source of failures	*A(t) initial*	*A(t) final*
Hardware	*0.98913*	*0.99911*
Software	*0.98378÷0.99544*	*0.99288÷0.99801*

4.2. CPS for environmental monitoring

DEPONTO representsa dependability assurance model (DAM) of the CPS for environmental monitoring having a role in system diagnosis. It works in conjunction with two software modules: the Diagnostic software module, and the Ontology Management software module [19]. The first software module diagnoses the CPS based on the ontology axioms, while the second interrogates and updates the ontology accordingly.

DEPONTO has 19 instances as can be seen in Figure 4.These instances match the FMEA lines of the Wi-Fi sensor nodes. The number of instances increases if the FMEA of the CPS includes the other physical and cyber components of the system. Figure 4 presents one of these instances: *Wi-Fi-Sensor-Battery-Depleted*, which is an instance of the *Hardware-Faults* subclass, with the four proprieties that characterize a fault.

The validation of the proposed dependability domain ontology, in the case of the CPS for environmental monitoring, was performed through SPARQL (SPARQL Protocol and RDF Query Language) [22] queries in Protégé, which are used by the Diagnostic software module. Such examples are presented in Table 3. All the queries use the prefixes listed at the beginning of Table 3.

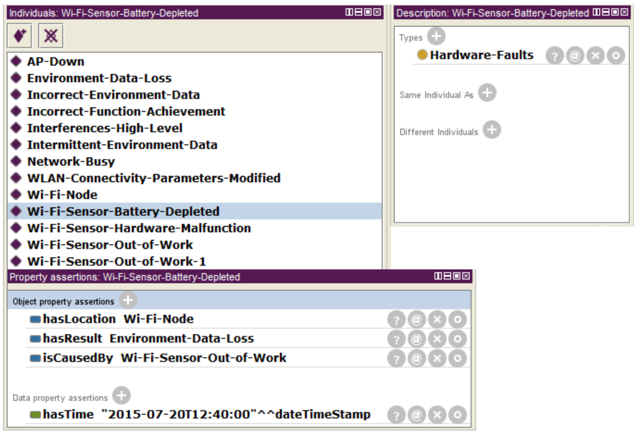


Figure 4. DEPONTO instancesofthe CPS for environmental monitoring

Table 3.SPARQL queries of the DEPONTO

Prefixes used by the SPARQL queries
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX ont:
<http://www.semanticweb.org/teo/ontologies/2014/2/untitled-ontology-13#>

Query 1: All hardware faults instances defined
SELECT ?DEPONTO_instances WHERE
{
?DEPONTO_instances rdf:type ?TYPE .
FILTER (regex(str(?TYPE),"Hardware-Faults"))
}
Result in Protégé:

DEPONTO_instances	TYPE
Wi-Fi-Sensor-Battery-Depleted	Hardware-Faults
Wi-Fi-Sensor-Hardware-Malfunction	Hardware-Faults
Wi-Fi-Sensor-Transducers-Malfunction	Hardware-Faults

Query 2: All instances with "Intermittent-Environment-Data" effect
SELECT ?DEPONTO_instances ?EFFECT ?CAUSE ?SEV WHERE{
?DEPONTO_instances ont:hasEffect ?EFFECT .
?DEPONTO_instances ont:hasSeverity ?SEV .
?DEPONTO_instances ont:hasCause ?CAUSE .
FILTER (regex(str(?EFFECT),"Intermittent-Environment-Data"))
}

Result in Protégé:

DEPONTO_instances	EFFECT	CAUSE	SEV
Wi-Fi-Sensor-Out-of-Work-2	Intermittent-Environment-Data	Network-Busy	"3"
Wi-Fi-Sensor-Out-of-Work-6	Intermittent-Environment-Data	Interferences-High-Level	"3"

Query 3: All the instances of the Wi-Fi sensor node in alphabetical order

```
SELECT ?DEPONTO_instances ?CAUSE ?SEV ?OCC ?DET
WHERE
{
 ?DEPONTO_instances ont:hasCause ?CAUSE .
 ?DEPONTO_instances ont:hasSeverity ?SEV .
 ?DEPONTO_instances ont:hasOccurence ?OCC .
 ?DEPONTO_instances ont:hasDetection ?DET .
}
ORDER BY ASC(?DEPONTO_instances)
```

Result in Protégé:

DEPONTO_instances	CAUSE	SEV	OCC	DET
Wi-Fi-Sensor-Out-of-Work	Wi-Fi-Sensor-Battery-Depleted	"5"	"3"	"2"
Wi-Fi-Sensor-Out-of-Work-1	Wi-Fi-Sensor-Transducers-Malfunction	"3"	"2"	"3"
Wi-Fi-Sensor-Out-of-Work-2	Network-Busy	"3"	"4"	"2"
Wi-Fi-Sensor-Out-of-Work-3	Wi-Fi-Sensor-Hardware-Malfunction	"5"	"2"	"2"
Wi-Fi-Sensor-Out-of-Work-4	AP-Down	"5"	"2"	"2"
Wi-Fi-Sensor-Out-of-Work-5	WLAN-Connectivity-Parameters-Modified	"5"	"1"	"2"
Wi-Fi-Sensor-Out-of-Work-6	Interferences-High-Level	"3"	"4"	"2"

Query 4: The total RPN (Risk Priority Number from FMEA) of the Wi-Fi sensor node

```
SELECT (SUM(?SEV*?DET*?OCC) AS ?totalRPN) WHERE
{
  SELECT DISTINCT * WHERE
   {
   ?DEPONTO_instances ont:hasSeverity ?SEV .
   ?DEPONTO_instances ont:hasOccurence ?OCC .
   ?DEPONTO_instances ont:hasDetection ?DET .
  }
}
```

Result in Protégé:

TOTAL_RPN_WIFI_COMPONENT
"146"^^<http://www.w3.org/2001/XMLSchema#integer>

Query 3 highlights the severity, occurrence and detection values for each possible cause of the Wi-Fi sensor failures. The results of the multiplication of these values($RPN_i = SEV_i*OCC_i*DET_i$) for each cause i are added and the final $RPN_{Wi\text{-}Fi\_Sensor}$ is equal with 146, as Query 4 shows.

The Ontology Management software module of the CPS works with SPARQL queries in order to update the ontology every time when new values of the severity, occurrence and detection are suggested by the Diagnosis software module. Unlike the first case study CPS, this solution doesn't use a multi-agent society, but only simple software modules implemented in C# language with the aid of the OwlDotNetApi [23] library.The two CPS modules, which work in conjunction with the ontology, are under development.

5. Conclusion

This paper proposes the representation of the dependability concept in a form that can be used to extract knowledge in order to ensure a support for CPSs diagnosis in case of threats. In this direction,a dependability domain ontology, DEPONTO, is proposed. The ontology centres on two concepts: *Faults* and *Failures* which aredefined based on a qualitative dependability analysis technique – FMEA. The proposed ontology uses OWL to outline its concepts and their relations (classes and properties), and its instances (individuals with properties). Protégé is the environment which has allowed the definition of 49 classes and 10 object and data properties of DEPONTO.

Two case study CPSs validate the ontology and prove its reusability:

- CPS for water treatment plant monitoring and control – the ontology acts as a knowledge base for a multi-agent society to increase the availability of the CPS hardware and software components;
- CPS for environment monitoring – the ontology acts as a dependability assurance model working together with dedicated software modules.

In both cases, specific queries (DL, respectively SPARQL) have been performed to classify the ontology and to demonstrate its consistency. Both approachesfacilitate the modelling of the CPSs behaviour at run-time in an evolving and adaptable context.

In conclusion,it can be seen that the dependability domain ontology is suitable for any CPSs implementation modality chosen by the developers due to knowledge formalization through dedicated languages.

Acknowledgements

This paper was supported by the Post-Doctoral Programme POSDRU/159/1.5/S/137516-PARTING, project co-funded from European Social Fund through the Human Resources Sectorial Operational Program 2007-2013.

References

[1]. Gruber, T.R. (1993). A translation approach to portable ontology specification. *Knowledge Acquisition.*

[2]. Gruber, T.R. (1993). Toward principles for the design of ontologies used for knowledge sharing. *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation.* Padova, Italy.

[3]. Chandrasekaran, B., Josephson, J. R., &Benjamins, V. R. (1999). What Are Ontologies, and Why Do We Need Them?,*IEEE Intelligent Systems and their Applications*, 14(1), 20–26.

[4]. Zablith, F., Antoniou, G., D'aquin, M., Flouris, G., Kondylakis, H., Motta, E., Plexousakis, D., &Sabou, M. (2013). Ontology evolution: a process-centric survey, *The Knowledge Engineering Review*, 30(1), 45–75.

[5]. Sanislav, T., &Miclea, L. (2012). Cyber-Physical Systems - Concept, Challenges and Research Areas. *Journal of Control Engineering and Applied Informatics*, 14(2), 28–33.

[6]. Wan, J., Chen, M., Xia, F., Li, D., & Zhou, K., (2013). From machine-to-machine communications towards cyber-physical systems, Computer Science and Information Systems, 10(3), 1105–1128.

[7]. Avizienis, A., Laprie, J. C., et. al. (2004). Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions On Dependable And Secure Computing*, 1, 11–33.

[8]. Roussey, C., Pinet, F., Ah Kang, M., &Corcho, O., (2011). An Introduction to Ontologies and Ontology Engineering. *Ontologies in Urban Development Projects*, Advanced Information and Knowledge Processing, 9–38, Springer-Verlag London.

[9]. Noy N.F., & Rubin, D. L., (2008). Translating the Foundational Model of Anatomy into OWL. *Web semantics (Online)*, 6(2), 133–136.

[10]. Gil, Y., & Blythe, J., (2000). PLANET: A Shareable and Reusable Ontology for Representing Plans. *Proceedings of the AAAI 2000 workshop on Representational Issues for Real-world Planning Systems.*

[11]. Wongthongtham, P., Chang, E., Dillon, T., &Sommerville, I., (2009). Development of a Software Engineering Ontology for Multi-site Software Development. *IEEE Transactions on Knowledge and Data Engineering*, 21(8), 1205–1217.

[12]. Smirnov, A., Levashova, T., Shilov, N., &Sandkuhl, K., (2014). Ontology for cyber-physical-social systems self-organisation. *Proceedings of the 2014 16th Conference of Open Innovations Association (FRUCT16)*, 101–107.

[13]. Lin, J., Sedigh, S., & Miller, A., (2011). A semantic agent framework for cyber-physical systems. *Semantic Agent Systems*, Studies in Computational Intelligence, 189-213, Springer Berlin Heidelberg.

[14]. Derler, P., Lee, E.A., &Vincentelli, A.S., (2012). Modeling Cyber–Physical Systems. *Proceedings of the IEEE*, 100(1), 13–28.

[15]. Sanislav, T., &Miclea, L., (2013). An Ontology-Driven Dependable Water Treatment Plant CPS. *Journal of Computer Science and Control Systems*, 6(1), 99–104.

[16]. Sanislav, T., Mois, G., &Miclea, L., (2015). A New Approach towards Increasing Cyber-Physical Systems Dependability. *Proceedings of the 2015 16th International Carpathian Control Conference (ICCC)*, 443–447.

[17]. *Analysis Techniques for System Reliability - Procedure for Failure Mode and Effects Analysis (FMEA).*(2006). International Standard, vol. IEC 60812, 2nd edition.

[18]. *Protégé documentation.*(2015). available at http://protege.stanford.edu/

[19]. Sanislav, T., &Miclea, L., (2015). A Dependability Modeling Approach for Cyber-Physical Systems. *Journal of Computer Science and Control Systems*, 8(1), 23–28.

[20]. Sanislav, T., Miclea, L., &Prinetto, P., (2012). Improving the Dependability of a Water Supply System via a Multi-Agent based CPS. *Proceedings of IEEE East-West Design & Test Symposium - EWDTS 2012*, 425-431.

[21]. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., & Wang, Z., (2014). HermiT: An OWL 2 Reasoner.*Journal of Automated Reasoning*, 53(3), 245-269.

[22]. World Wide Web Consortium (W3C), (2013). *SPARQL 1.1 Federated Query*, available at http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[23]. *OwlDotNetApi Project Documentation*, (2010). available at https://code.google.com/p/owldotnetapi.