

Ontology Driven Testing Strategies for IoT Applications

Muhammad Raza Naqvi¹, Muhammad Waseem Iqbal², Muhammad Usman Ashraf³, Shafiq Ahmad⁴,
Ahmed T. Soliman⁴, Shahzada Khurram⁵, Muhammad Shafiq^{6,*} and Jin-Ghoo Choi⁶

¹INP-ENIT, University of Toulouse, Tarbes, 65000, France

²Department of Software Engineering, Superior University, Lahore, 54000, Pakistan

³University of Management and Technology, Sialkot Campus, Pakistan

⁴Industrial Engineering Department, College of Engineering, King Saud University, P.O. Box 800, Riyadh, 11421, Saudi Arabia

⁵Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan

⁶Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, Korea

*Corresponding Author: Muhammad Shafiq. Email: shafiq@ynu.ac.kr

Received: 05 April 2021; Accepted: 29 July 2021

Abstract: Internet-of-Things (IoT) has attained a major share in embedded software development. The new era of specialized intelligent systems requires adaptation of customized software engineering approaches. Currently, software engineering has merged the development phases with the technologies provided by industrial automation. The improvements are still required in testing phase for the software developed to IoT solutions. This research aims to assist in developing the testing strategies for IoT applications, therein ontology has been adopted as a knowledge representation technique to different software engineering processes. The proposed ontological model renders 101 methodology by using Protégé. After completion, the ontology was evaluated in three-dimensional view by the domain experts of software testing, IoT and ontology engineering. Satisfied results of the research are showed in interest of the specialists regarding proposed ontology development and suggestions for improvements. The Proposed reasoning-based ontological model for development of testing strategies in IoT application contributes to increase the general understanding of tests in addition to assisting for the development of testing strategies for different IoT devices.

Keywords: Ontology; software testing; internet of things; automation

1 Introduction

Internet of Things (IoT) is a concept that allows billions of tiny machines, such as sensors, to be connected to the internet. Software testing is the crucial process of software quality assurance because it represents the final revision of specification, design patterns and code generation [1,2]. The main goal of conducting software testing is to reduce the likelihood of errors occurring when the system is in production [3]. It is verified through this research that testers face problems and difficulties which refer to the complexity between tests in IoT applications [4]. The internet and



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

its things are only limited for mere moments in time, so the test is problematic. Furthermore, one issue analyzers will confront is the manner by which to characterize a test architecture with a related environment that is adequate and exhaustive [5]. Likewise, the absence of legitimate information about IoT testing open new research challenges including several quality attributes including interoperability, user-friendly in context of end user satisfaction, and reliability [6]. The research problem is the fact that, in addition to the evidence of problems and difficulties mentioned above, the study conducted in this study did not identify patterns, models or rules regarding tests on IoT devices. The IoT applications of the consulted jobs use convenient ways for different products [7]. The objective of this work is to assist in the development of testing strategies in IoT applications. Thus, the contribution of this work is to provide help in ontology development of a testing strategy focused on the IoT. To maintain good monitoring and performance in IoT it is necessary to improve the quality of processes and one of the most important steps to maintain them, is the testing phase. Still, software testing is essential to the proper functioning of the IoT. After ontology development, it is evaluated by domain experts of IoT, software testing and ontology engineering. The consistency and coherency of developed ontology checked by using “Pellet Reasoner” which aims to determine the satisfaction. It shows interest for specialists in current research. It also provided the reflection of suggestions for improvements in ontology development as well as future work.

To ensure IoT application quality, this research proposed the ontological modelling technique for test strategies development and consequent execution. The proposed testing methodology has expressed the potentials for domain specific attributes incorporation in the testing process planning. We concluded our study with the proven Potentials of Ontology Driven Information System Development Lifecycle being utilized for IoT Application Testing. Hence, the work adds to build the overall comprehension of Tests in IoT, as well as aiding the Development of Testing Strategies for various IoT gadgets, applications, devices.

The rest of the paper is structured as follows. Section 2 describes summary of the literature review. Section 3, elaborates the problem statement. Section 4 explains ontology engineering. Section 5 defines ontology development. The validation is done in Section 6 while data analysis and discussion are stated in Section 7. Section 8 is about the conclusion of our research and provides horizons towards the future.

2 Literature Review

Ontology development, specifically for the domain of software testing in IoT, still has research gap. However, the description of some ontological based IoT approaches is discussed in this section.

To accomplish consistent software development, researchers like Meng et al. [8], Pinheiro et al. [9], Wang et al. [10], Gyrard, et al. [11], Kim et al. [12], Bali et al. [13], Uskov et al. [14] have shown that certain model families typically have some features which are considered better than others. Bermudez-edo et al. proposed an ontology for the network of semantic sensors that describe the main concepts of IoT data in heterogeneous platforms [15]. According to Voas et al. there are three factors for software failure including, a failure is performed, a faulty internal data state is created and the faulty state propagates and so resulting incorrect output [16]. These failure factors can be avoided by applying an appropriate testing methodology for the product or application. According to Geiger et al. automated testing frameworks are effective because IoT-enabled devices have the ability to be self-diagnosed. It is important to note that automated testing is a powerful in an IoT application, as it is applicable in all layers [17]. Furthermore,

in the research by Murad et al. emphasize that there are several types of tests that can be used in IoT. There has been a huge trend to shift data analytics from OLAP to Big Data Architectures [18].

- Usability Testing: It is important to check the usability of system in IoT testing because the perception varies from one user to another.
- Reliability and Scalability Testing: IoT testing can provide reliability and scalability in an environment that involves simulating sensors using virtualization tools and technologies.
- Compatibility Testing: The IoT compatibility testing may provide configurations that can be connected with various devices.
- Security Testing: It is important to control the users' access in IoT environment, by defining the user authentication and data privacy control.

Another study by Bernardino et al. proposed an ontology in performance testing domain. The ontology was represented in the OWL (Web Ontology Language), developed in the editor Protégé and following the methodology of Noy and McGuinness, being directed to assist testers with test planning [19]. It was found, through the literature study, that testers faced problems and difficulties which refer to the complexity between tests in IoT applications. The motivation of this research is to help the professionals to carry out their testing activities in the domain of IoT. It also concentrates that testers can adopt testing strategies driven by ontological model. It is necessary to improve the quality of the processes, their steps to maintain good monitoring and performance in IoT. Further, the software testing is essential for the correct functioning of IoT and must be very well modeled and standardized.

The IoT applications of the consulted jobs use convenient ways for different products. The problems are still existing in the development of testing strategies to carry out software testing in IoT. The study carried out in this work did not identify standards, models or rules with regard to tests on IoT applications. Thus, the objective of this work is to assist, through an ontology, in the development of test strategies in IoT for various connecting devices. In addition, the tests can be applied in a more standardized way by defining phases, strategies and context.

3 Ontology Engineering

Ontologies in software testing process deliver the common shared knowledge for understanding of concerned entities and their relationships with a rather simple and easy structure, maintenance and updates [20]. This research has exploratory theme and our ontology is a domain type, that developed in semi-formal languages. Ontology development 101 is the well-known ontology engineering methodology. It consists of seven objective and simple steps of building an ontology process [21]. Uschold and Gruninger emphasize on the importance of defining the scope of ontology [22]. They also provide the list of significant competency questions to define the domain scope of ontology.

These standardized questions also incorporate to enrich our research.

- What is the domain of ontology?
- What is the purpose of building this ontology?
- What kind of questions, ontology should answer?
- Who will use and maintain the ontology?

Tab. 1 presents the scope of determination for our ontology and view the answers to previous questions. After defining ontology scope, we consider the method of ontology reuse that aims to

identify whether it required or not. In assisting the construction of ontology, two ontologies have been reused.

Table 1: Scope determination

Questions	Answers
What is the domain of ontology?	The ontology covers software testing applied in IoT.
What is the purpose of building this ontology?	Provide a clear and more holistic view of testing processes in IoT applications.
What kind of questions should the ontology's information answers?	Ontology answers when and how to test the IoT.
Who will use and maintain ontology?	The ontology's users include software testers and researchers. However, it is maintained by the author of this research.

The most important step in ontology construction is terms enumeration, the concepts of terms can be defined in primitive classes and defined classes. Classes serve to describe basic concepts of a domain and represent the broadest concepts in a hierarchy. Next step is defining properties that present the relationships between concepts as attributes. A property must have domain and scope, so properties connect individuals from a domain to individuals of a scope [23]. Finally, defining constraints that demonstrate a class of individuals according to the relationships of class members.

4 Ontology Development

For the development of ontology, research was carried out in order to seek strategies and techniques for testing in IoT applications. The construction of the proposed ontology is carried out based on seven stages of methodology 101, justified and conceptualized in the previous topics. Ontology is represented in English to increase its scope and follow the pattern adopted in reuse ontologies. For the development, the Protégé tool was used and it is an open-source free software with tools to build models and applications based on the areas of knowledge using ontologies.

4.1 Ontology Reuse Consideration

Fig. 1 presents proposed ontology by reusing two ontologies (Onto test and software test ontology).

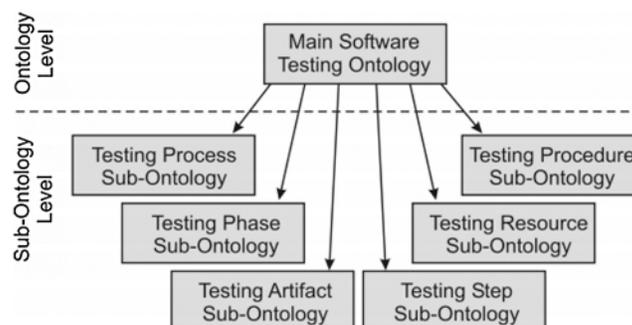


Figure 1: Structure of the 1st reuse ontology

Firstly, OntoTest ontology that explores the different aspects involved in software testing activity, defines a common vocabulary and assists in establishing reference architectures [24]. OntoTest is a software testing ontology that assists in the acquisition, organization, reuse and sharing of knowledge about the area. Secondly, In Fig. 2 structure of second reuse ontology the software testing ontology [25] is similar to OntoTest, but represents concepts: tester, context, method, artifact, activity and environment.

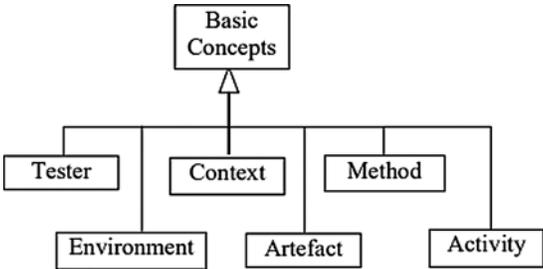


Figure 2: Structure of the 2nd reuse ontology

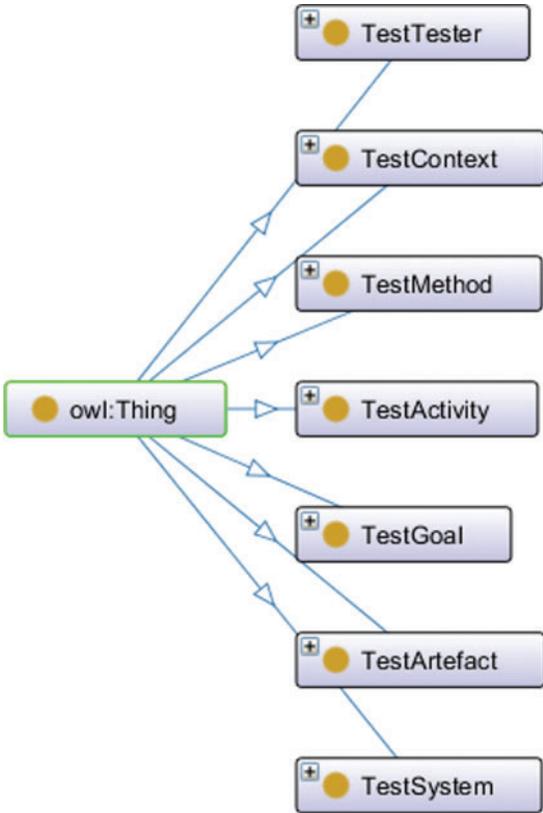


Figure 3: Main classes

4.2 List Classes (Concepts)

The terms of the ontologies start their construction, remembering that the terms presented below and defined from the reuse ontologies. In the following Fig. 3 it is possible to see the terms defined in the Protégé and Fig. 4 defining the test context class.

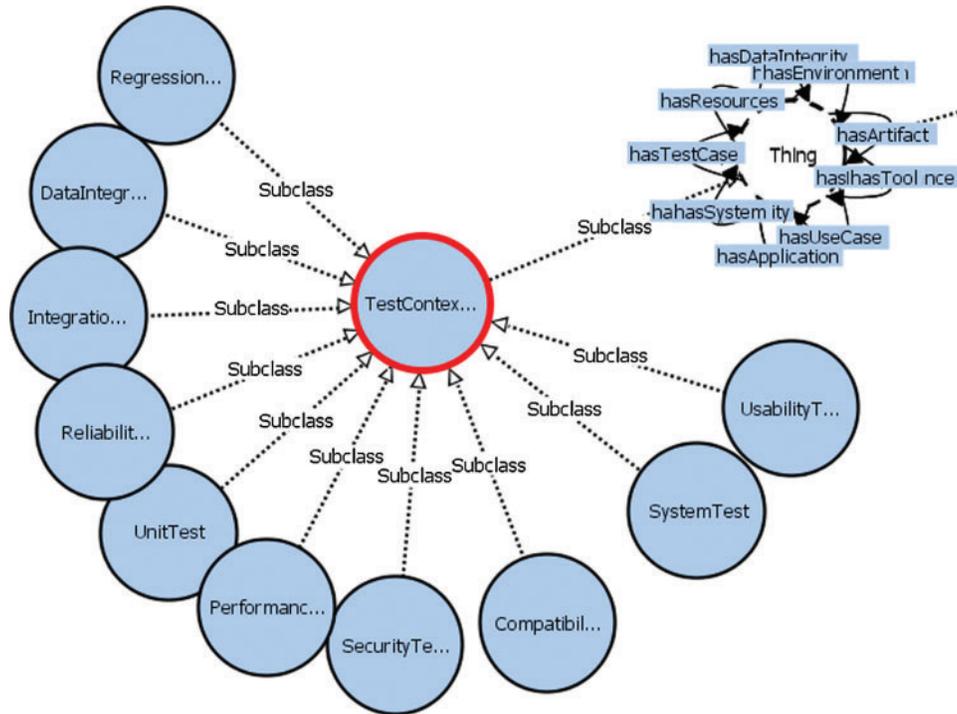


Figure 4: Text context class visualization

In Tab. 2, the definition of each class has been defined.

Table 2: Definition of concepts of ontology terms

Concept	Definition
TestTester	Represents a tester, who is responsible for performing the test activities (TestActivity).
TestContext	Represents the test step that will be applied to the product.
TestMethod	It represents methods applicable in testing activities, which may vary as to the techniques and approaches used.

(Continued)

Table 2: Continued

Concept	Definition
TestActivity	Represents a test activity.
TestGoal	It represents a reason or objective that justifies the execution of a test step (TestContext).
TestArtifact	It represents a test artifact or deliverable, that is, a work product generated from the execution of a test activity (TestActivity).
TestSystem	It represents the target of a software test, that is, the system under test.

4.3 Terms and Set Classes/Subclasses Enumeration

For each created term, their respective classes were defined according to the IoT testing techniques and strategies. In this Fig. 5 the body of the ontology is created where the test actions are defined.

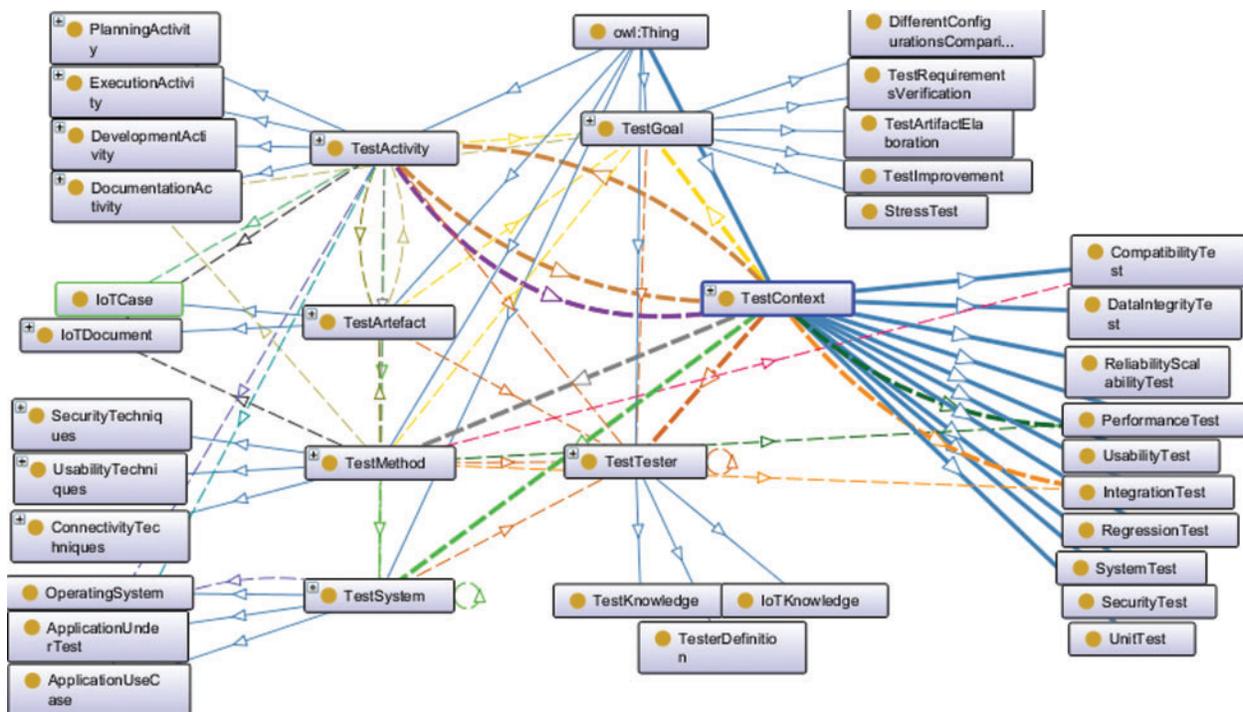


Figure 5: Ontology structure (classes & subclasses)

4.4 Properties Setting and Instances Creation

Fig. 6 presents the properties and individuals defined in Protégé. In the case of this ontology, the object property type was constructed. Consequently, in each of the properties, individuals from a domain connected to individuals of a scope.

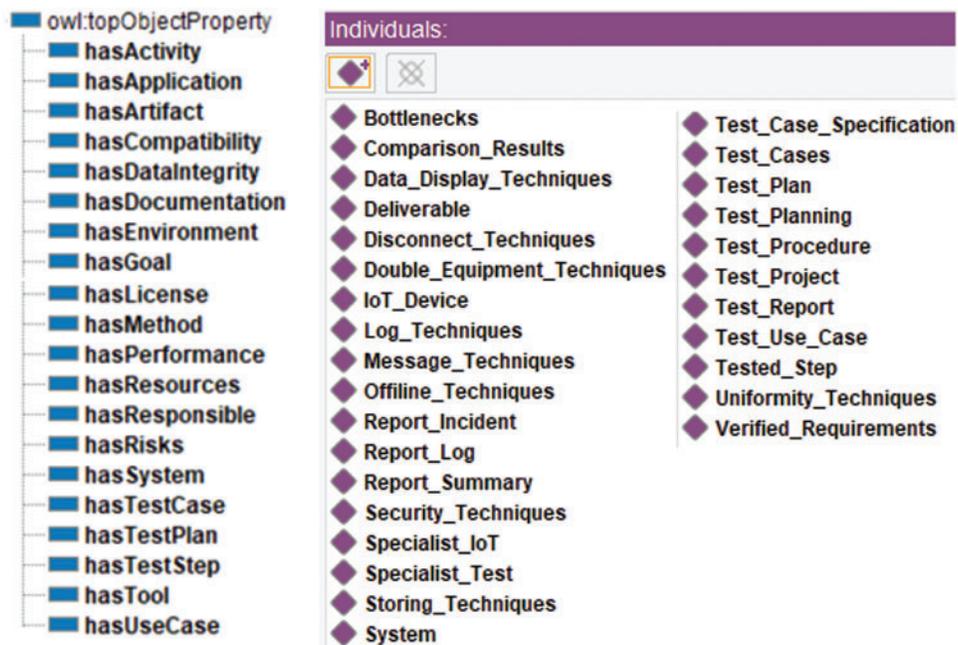


Figure 6: Ontology object properties & ontology instances

Being an instance, the realization of a class, were then defined in each class of the proposed ontology. For example, the TestActivity class aims to represent a test activity, with the result that test cases (TestCases), test plans (TestPlan), test planning (TestPlanning) and test reports (TestReport) must be generated to represent the instances.

4.5 Define Properties

In the [Tab. 3](#), it is possible to view the properties defined in Protégé, as well as the definition of each one. In the case of this ontology, the object property type was built.

Table 3: Properties definitions

Property	Definitions
hasLicense	Maps the tool licenses used in the tests.
hasTestCase	Represents the test cases.
hasUseCase	Represents the use cases to be tested.
hasTestStep	It represents the test step being performed in each application.
hasResponsible	It contains the information of those responsible for the execution of each test activity.
hasArtifact	It is used to specify which deliverables each test activity can generate.

(Continued)

Table 3: Continued

Property	Definitions
hasActivity	Contains information about the test activities to be performed in the IoT application.
hasGoal	It contains information about the objectives that a software test should have.
hasSystem	Defines the operating system where the application under test is hosted.
hasTool	Indicates the test tools chosen for each type of software test.
hasApplication	Contains information on which IoT application is being tested.
hasMethod	Defines the methods (techniques and approaches) used to carry out the testing activities.
hasDataIntegrity	It represents data integrity, that is, maintenance, guarantee of data accuracy and consistency.
hasPerformance	Maps the number of connected devices, the volume of data propagation between devices.
hasCompatibility	Represents software and hardware that is compatible with the devices that are connected.
hasRisks	Determines what risks can occur at each test step.
hasResources	Ensures that there are resources to proceed with the test.
hasEnvironment	Determines the environment in which the test will be performed.
hasDocumentation	Represents the documentation that the testing process must contain.
hasTestPlan	It contains the information of the test plan created.

4.6 Ontology Restrictions

It is necessary to define their respective restrictions whereas the object properties describe the relationships between individuals. This is done through a class of individuals created by the relationships in which they are supposed to participate.

The restrictions covered here are of the “Quantifier Restriction” type, with two available quantifiers being: i) the existential quantifier (some) (\exists), which define classes of individuals who participate in one relationship (at least) through a specific feature with member individuals of a certain class, ii) the universal quantifier (only) (\forall), which define classes of individuals for a particular property, only have relationships by that property member individuals of that class. The definition of restrictions can made in three parts:

- The quantifier, which can be an “some” existential quantifier (\exists) or a universal “only” (\forall).
- A property, on which the constraint acts.
- A complement to the constraint (filler), which correspond to name class whose individuals meet the constraint.

We have shown the constraints between classes in [Fig. 7](#). For example, the constraint property on the TestActivity class: hasArtifact; the type of constraint: “some” (existential constraint); the mandatory complement of the restriction (Filler).

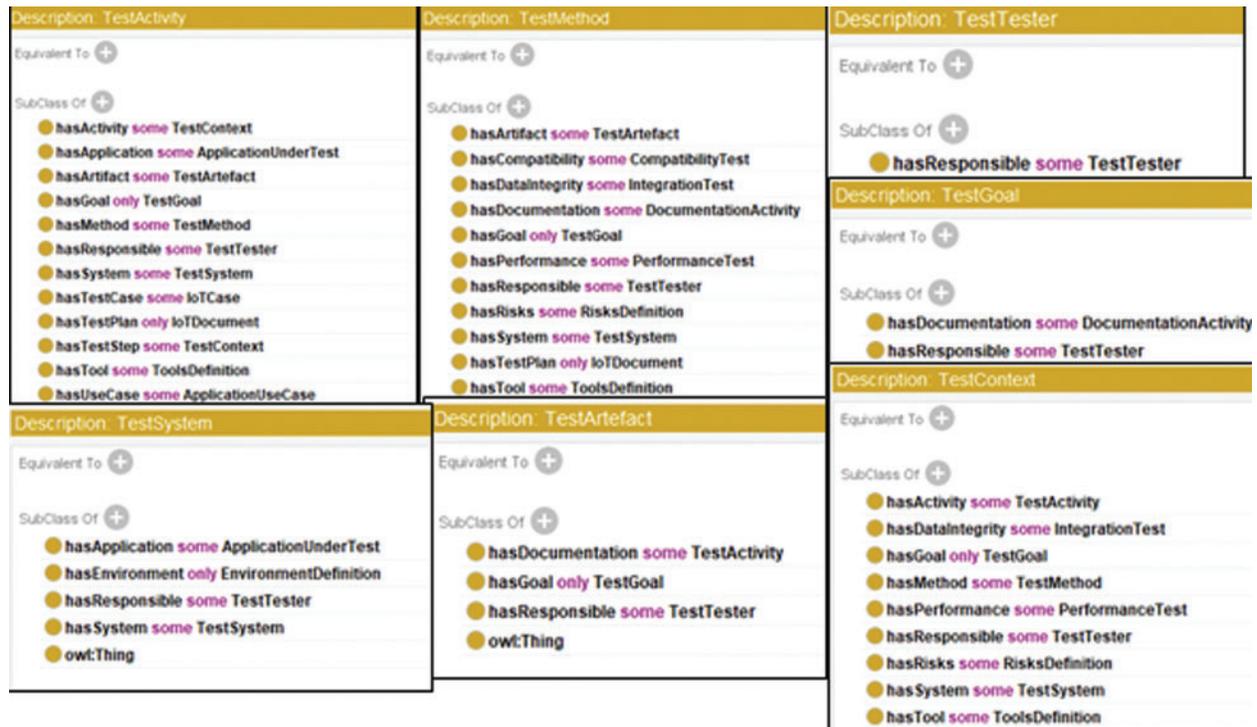


Figure 7: Visualizes the constraints between the classes

The TestArtifact class; then there is hasArtifact property has some (TestArtifact) The informed constraint is specifying that the test activity (TestActivity) has at least one test artifact or deliverable TestArtifact, through the teamArtifact includes hasArtifact property.

Another example, the constraint property on the TestActivity class: hasGoal; the type of restriction: “only” (universal restriction); the mandatory complement of the restriction (Filler).

The TestGoal class; Then there is: hasGoal only (TestGoal). The informed constraint is specifying that a test activity (TestActivity) has only one objective TestGoal through the object property hasGoal.

5 Validation

Protégé provides third-party Reasoner like HerMiT and FaCT++ and others to validate the ontologies [26]. Reasoner detects and finds inconsistencies or contradictions of ontologies structure and data based on mathematical models [27]. It provides logical deductions based on inference rules defined and specified in description logic and uses forward chaining or backward chaining to perform inference [28]. Pellet Reasoner is used to validate our ontology model. Previously researchers and ontology engineers use reasoner to validate their work [29–31]. It can be used in unification with Jena and OWL APIs. Fig. 8 shows the validation of our ontology.

6 Data Analysis and Discussion

The technique for data analysis of this work was performed to determine the satisfaction of ontology, whether it meets the necessary requirements referring to software testing, IoT and ontology engineering. The proposed ontology was evaluated by specialists from the domain.

For data collection, the experts were sent a questionnaire containing 15 questions to cover the aspects of the ontological model with regard to the satisfaction of ontology. These are semi-open questions with alternatives “Satisfies”, “Does not satisfy” and “other Option”, when choosing the last option, the professional will have the opportunity to express themselves reporting questions or suggestions. Along with the questionnaire, the ontology documentation is sent, so that it can be evaluated. 15 questions were answered by experts, 8 of them obtained 87% satisfaction. In Tab. 4 below aims to demonstrate the opinion of the experts regarding the satisfaction of ontology.

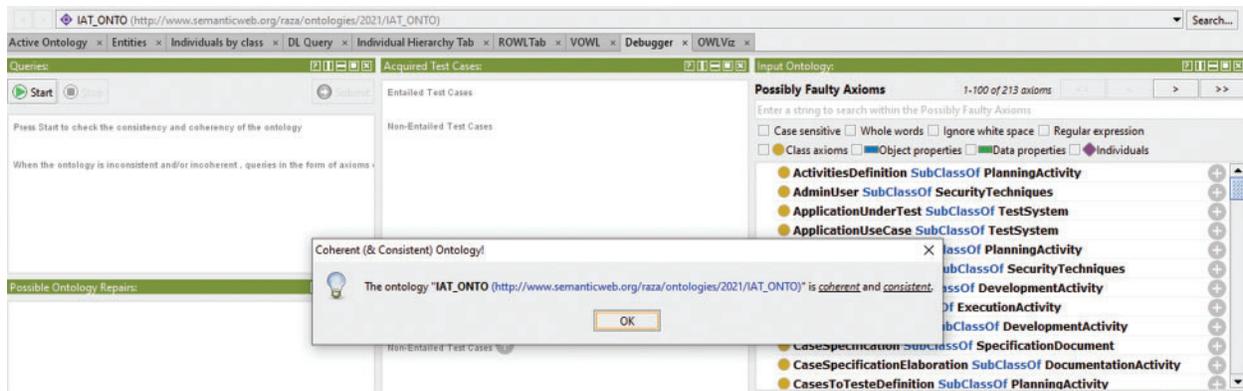


Figure 8: Ontology validation using “Pellet” reasoner

Table 4: Ontology satisfaction

Subject	Satisfies	"Other option"
The types of test and test activities are consistent with what the professional performs.	100%	N/A
The IoT testing techniques are consistent with the experience of the specialist.	83.3%	Other methods that can be applied in IoT projects that are not included in ontology were mentioned
Ontology represents the correct approach to Software Testing.	83.3%	The specialist is not sure, as he needs a practical example
Ontology represents the correct approach to the Internet of Things.	100%	N/A

(Continued)

Table 4: Continued

Subject	Satisfies	"Other option"
The way in which IoT tests are applied, are in accordance with the knowledge of experts in the fields.	83.3%	Again emphasized the need to apply test ontology in IoT equipment and products.
Ontology can be useful for testing IoT applications.	100%	N/A
Ontology may be useful for future work.	83.3%	A work in which ontology is applied in practical examples.
Understanding the structure of ontology.	83.3%	Within Usability Techniques, the Double Equipment technique could be better described, the text is confusing.
The structure of ontology is in accordance with the Ontology Engineering.	83.3%	Professional does not have the necessary knowledge to answer this question.
The instances, constraints, and properties of ontology are in accordance with the expert's knowledge of the domains.	100%	N/A
The understanding of the proposed ontology is clear.	100%	N/A
Open question for general positions on ontology.	N/A	Technical suggestion to verify the extent of the connectivity surface; Property suggestion that informs you if the system is upgradeable; Cost concern in test teams; Praise for the ontology proposal; Practical application interests of ontology in an IoT software.

According to the percentage of ontology satisfaction and the comments of the experts mentioned in [Tab. 4](#), it can be said that ontology can help in the development of a testing strategy focused on the IoT. The choice of professionals participating in the research was due to the experience in at least 5 years working with IoT. The same criterion of choice was used for software testing and ontology engineering professionals.

7 Conclusion

IoT is a specialized domain of computer applications including embedded systems, communication, cloud, fog, edge services and data stores. These specialized information system architecture, process flow and functionality are engineered and tested in their own way. The study here identified the research gap related to domain specific software engineering and more precisely the testing phase for the IoT applications.

To ensure IoT application quality, this research proposed the ontological modelling technique for test strategies development and consequent execution. The proposed testing methodology has expressed the potentials for domain specific attributes incorporation in the testing process

planning. The testing model's consistency is verified through the reasoning software that can highlight existing anomalies in test plans. The challenging part of the research is formal representation of testing processes and the constraints among them. The identified prerequisite, dependencies, overlapping tasks and synchronous behavior of complete test plan are the examples of the constraints in the studied testing models. Furthermore, the proposed model is reviewed through the domain experts with a high satisfaction rate (87%) observed via conducted survey. The study concluded with the proven potentials of ontology driven information system development lifecycle being used for IoT application testing. Thus, the work contributes to increase the general understanding of tests in IoT, in addition to assisting in the development of testing strategies for different IoT devices.

The presented testing model, can be generalized for any domain specific application testing through domain concept definition and respective knowledgebase linking. The future work in this research directions specific to IoT includes, the variety of tests plan development for individual unit/component and integrated testing with the required span of knowledge sharing in the global software engineering perspectives. In the case of possible studies and future studies in which the research treated in this article can advance we can mention: Measure the cost with testers and the impact of this cost with later support of a project in which the ontology proposal is applied and to one that was not used ontology; Practical application of ontology methods to verify how robust it is to test on IoT equipment and products; Improvement ontology in terms of techniques, strategies and processes.

Acknowledgement: The authors extend their appreciation to King Saud University for funding this work through Researchers Supporting Project number (RSP-2021/387), King Saud University, Riyadh, Saudi Arabia.

Funding Statement: This work was supported by the King Saud University (in Riyadh, Saudi Arabia) through the Researcher Support Project Number (RSP-2021/387).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] A. K. Dwivedi and S. M. Satapathy, "Ontology-based modelling of iot design patterns," *Journal of Information & Knowledge Management*, vol. 20, no. 1, pp. 1–25, 2021.
- [2] K. Sahu and R. K. Srivastava, "Soft computing approach for prediction of software reliability," *ICIC Express Letter*, vol. 12, no. 12, pp. 1213–1222, 2018.
- [3] J. A. P. Lima and S. R. Vergilio, "Test case prioritization in continuous integration environments: A systematic mapping study," *Information and Software Technology*, vol. 121, pp. 1–16, 2020.
- [4] K. Sahu and R. K. Srivastava, "Revisiting software reliability in data management, analytics and innovation," *Singapore: Analytics and Innovation Springer*, vol. 801, pp. 221–235, 2020.
- [5] J. D. Hagar, "Software test architectures and advanced support environments for IoT," in *IEEE Int. Conf. on Software Testing, Verification and Validation Workshops (ICSTW)*, Sweden, pp. 252–256, 2018.
- [6] J. H. Andrews, L. C. Briand and Y. Labiche, "Is mutation an appropriate tool for testing experiments?," in *Proc. of the 27th Int. Conf. on Software Engineering*, St. Louis, USA, pp. 402–411, 2005.
- [7] A. Gyrard, A. Zimmermann and A. Sheth, "Building IoT-based applications for smart cities: How can ontology catalogs help," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3978–3990, 2018.

- [8] M. Ma, P. Wang and C. H. Chu, "Ontology-based semantic modeling and evaluation for internet of things applications," in *IEEE Int. Conf. on Internet of Things (IThings)*, New York, USA, pp. 24–30, IEEE, 2014.
- [9] V. Pinheiro, N. Neumann, M. Endler and F. Silva Deklaer, "An ontology-driven framework for generating iot applications using contextnet," in *IEEE Symp. on Computers and Communications (ISCC)*, Natal, Brazil. vol. 25, pp. 608–614, 2018.
- [10] M. Tao, K. Ota and M. Dong, "Ontology-based data semantic management and application in IoT-and cloud-enabled smart homes," *Future Generation Computer Systems*, vol. 76, no. 1, pp. 528–39, 2017.
- [11] W. Wang, S. De, R. Reetz and K. Moessner, "A comprehensive ontology for knowledge representation in the internet of things," in *IEEE 11th Int. Conf. on Trust, Security and Privacy in Computing and Communications*, Liverpool, United Kingdom, vol. 1, no. 1, pp. 1793–1798, 2012.
- [12] A. Gyrard, S. K. Datta and C. Bonnet, "A survey and analysis of ontology-based software tools for semantic interoperability in IoT and WoT landscapes," in *IEEE 4th World Forum on Internet of Things (WF-IoT)*, Singapore, vol. 1, no. 1, pp. 86–91, 2018.
- [13] H. Kim, A. Ahmad, J. Hwang, H. Baqa and F. L. Gall, "Iot-TaaS: Towards a prospective IoT testing framework," *IEEE Access*, vol.6, pp. 15480–15493, 2018.
- [14] A. Bali, M. Al-Osta and G. Abdelouahed, "An ontology-based approach for IoT data processing using semantic rules," in *International SDL Forum*, vol. 1, pp. 61–79, 2017.
- [15] V. Uskov, A. Pandey, J. P. Bakken and V. S. Margapuri, "Smart engineering education: The ontology of internet-of-things applications," in *IEEE Global Engineering Education Conf. (EDUCON)*, Abu Dhabi, UAE, vol. 1, no. 1, pp. 476–481, 2016.
- [16] M. Bermudez, T. Elsaleh, P. Barnaghi and K. Taylor, "Iot-lite: A lightweight semantic model for the internet of things and its use with dynamic semantics," *Personal and Ubiquitous Computing*, vol. 21, no. 3, pp. 475–487, 2017.
- [17] K. Androutopoulos, D. Clark, H. Dan, R. M. Hierons and M. Harman, "An analysis of the relationship between conditional entropy and failed error propagation in software testing," in *Proc. of the 36th Int. Conf. on Software Engineering*, Hyderabad, India. vol. 1, no. 1, pp. 573–583, 2014.
- [18] C. Geiger. and W. J. Headrick, "The thingosity of automated test systems in the internet of things," in *Proc IEEE AUTOTESTCON*, Illinois, USA, vol. 1, no. 1, pp. 1–4, 2017.
- [19] G. Murad, A. Badarneh, A. Qusef. and F. Almasalha, "Software testing techniques in IoT," in *8th Int. Conf. on Computer Science and Information Technology (CSIT)*, Amman, Jordan vol. 1, no. 1, pp. 17–21, 2018.
- [20] M. Bernardino, A. F. Zorzo and E. M. Rodrigues, "Canopus: A domain-specific language for modeling performance testing," in *IEEE Int. Conf. on Software Testing, Verification and Validation (ICST)*, Illinois, USA, vol. 1, no. 1, pp. 157–167, 2013.
- [21] K. R. Felizardo, E. F. Souza, R. A. Falbo and N. L. Vijaykumar, "Defining protocols of systematic literature reviews in software engineering: a survey," in *43rd Euromicro Conf. on Software Engineering and Advanced Applications (SEAA)*, Vienna, Austria vol. 1, no. 1, pp. 202–209, 2017.
- [22] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," Technical Report-University of Edinburgh Artificial Intelligence Applications Institute Aiai Tr, 1996.
- [23] E. F. Barbosa, E. Y. Nakagawa and J. C. Maldonado, "Towards the establishment of an ontology of software testing," in *Seke*, vol. 6, pp. 522–525, 2006.
- [24] H. Zhu and H. Qingning, "Developing software testing ontology in UML for a software growth environment of web-based applications," in *The Software Evolution with UML and XML*, 1st ed., Leicester, UK: Idea Group Publishing, Chapter No. 09, pp. 263–295, 2005.
- [25] H. Naseer and A. Rauf, "Validation of ontology-based test case generation for graphical user interface," in *15th Int. Multitopic Conf.*, Islamabad, Pakistan, vol. 1, no. 1, pp. 465–469, 2012.
- [26] H. Zhao, S. Zhang and J. Zhao, "Research of using protégé to build ontology," in *IEEE/ACIS 11th Int. Conf. on Computing and Information Science*, Shanghai, China. vol. 1, no. 1, pp. 697–700, 2012.

- [27] S. Abburu, “A survey on ontology reasoner and comparison,” *International Journal of Computing and Applications*, vol. 57, no. 17, pp. 33–39, 2012.
- [28] T. Naz, M. Akhtar, S. K. Shahzad, M. Fasli, M. W. Iqbal *et al.*, “Ontology-driven advanced drug-drug interaction,” *Computers & Electrical Engineering*, vol. 86, pp. 66–95, 2020.
- [29] K. Rabbani, M. Lissandrini and K. Hose, “Optimizing SPARQL queries using shape statistics,” in *24th Int. Conf. on Extending Database Technology (EDBT)*, Nicosia, Cyprus, 2021.
- [30] N. C. Nicholson, F. Giusti, M. Bettio, R. N. Carvalho, N. Dimitrova *et al.*, “An ontology-based approach for developing a harmonised data-validation tool for european cancer registration,” *Journal of Biomedical Semantics*, vol. 12, no. 1, pp. 1–15, 2021.
- [31] S. Liang, K. Stockinger, M. Anisimova and M. Gil, “Querying knowledge graphs in natural language,” *Journal of Big Data*, vol. 8, no. 1, pp. 1–23, 2021.