*I.J. Information Technology and Computer Science*, **2023, 1, 30-41**
Published Online on February 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2023.01.04

# Ontology-driven Intelligent IT Incident Management Model

**Bisrat Betru***
Department of Computer Science, College of Natural and Computational Sciences, Addis Ababa University, Addis Ababa, Ethiopia
E-mail: lebisratm@gmail.com
ORCID iD: https://orcid.org/0000-0003-1860-3881
*Corresponding Author

**Fekade Getahun**
Department of Computer Science, College of Natural and Computational Sciences, Addis Ababa University, Addis Ababa, Ethiopia
E-mail: fekade.getahun@aau.edu.et

Received: 21 March 2022; Revised: 28 November 2022; Accepted: 23 December 2022; Published: 08 February 2023

**Abstract:** A significant number of Information Technology incidents are reported through email. To design and implement an intelligent incident management system, it is significant to automatically classify the reported incident to a given incident category. This requires the extraction of semantic content from the reported email text. In this research work, we have attempted to classify a reported incident to a given category based on its semantic content using ontology. We have developed an Incident Ontology that can serve as a knowledge base for the incident management system. We have also developed an automatic incident classifier that matches the semantical units of the incident report with concepts in the incident ontology. According to our evaluation, ontology-driven incident classification facilitates the process of Information Technology incident management in a better way since the model shows 100% recall, 66% precision, and 79% F1-Score for sample incident reports.

**Indexed Terms:** ITIL, Incident Ontology, Ontology-driven Information Extraction, Semantic Text Classification, Intelligent Systems.

## 1. Introduction

In today's world, Information Technology (IT) plays a significant role in the organization's day-to-day operations and activities. Effective Information Technology Service Management (ITSM) is becoming a fundamental factor in deciding the performance of organizations. IT service delivery involves organizational operations, service management processes, and complex infrastructures along with participating people [1,2]. It is a fact that organizational success is dependent on the high availability, reliability, security, and performance of IT services delivery.

The daily users of IT services have expectations that the technological infrastructure works seamlessly. They demand a continual service from the ITSM team to properly utilize the available service resources for their day-to-day operations. As a result, organizations need to have an effective ITSM to ensure that IT service delivery is working properly [1,3] a nd to tackle and analyze repetitive incidents and their root cause s [4,5]. Effective ITSM teams must work hard in preventing the occurrence of any service-related problems. If an issue occurs, the IT helpdesk teams must record the issue quickly to follow-up and resolve the reported issue. To achieve this, organizations are required to have an efficient incident management system that facilitates the process of logging, recording, escalating, and resolving incidents.

Incident Management is the critical function of ITSM for any IT department in all kinds of organizations. The primary purpose of incident management is to "*restore the service*" to clients as soon as possible usually through a swift workaround on the root causes [6]. The whole processes of incident management are highly dependent on understanding the reported incident. What the client wants to inform about an incident is crucial to log, categorize, prioritize, and resolve the incident. If the Incident Management System (IMS) can automatically capture the meaning of IT incident reports, it shall assign to the concerning person or unit, and the IT professionals shall spend their time on resolving the issue.

In large organizations, most incidents are reported through email. Email messages are usually "*unstructured and not always syntactically well-formed*" [7]. This also applies to incident email reports. As a result, it is a challenge to automatically understand the meaning of incident emails and classify it to a given category. Since the extraction of meaning from an incident report email involves retrieving information from the ITSM domain, a formal and explicit specification of IT incident concepts and their semantical relationships is significant [8-10]. This can be achieved through the construction of incident ontology.

The incident management knowledge domain can be critically defined, and the contextual relationship of IT service terms can be formulated and organized through an Incident Ontology. This is because ontologies are very important in "*achieving the goals of qualitative modeling*" by making knowledge explicit and formal that can be used for automation [11]. Ontologies can provide a vocabulary of terms and their relationships by allowing semantics of such terms to be formally specified [12,13]. This enables a better way of semantic content extraction from the specified terms [14,15]. Since the processes of incident management are interrelated, a common ontology improves the task of meaning extraction from the interrelated processes. Ontologies are also ideal solutions to compare semantic units in a text with concepts defined in a domain since ontology-driven knowledge bases facilitate the semantical matching of "*concepts defined in a concept taxonomy*" [16].

An Ontology-Based Information Extraction System is "*A system that processes unstructured or semi-structured natural language text through a mechanism guided by ontologies to extract certain types of information and presents the output using ontologies*" [17]. While a conceptual schema defines relations on data for traditional systems, an ontology defines terms that represent knowledge that can be used by Ontology-driven systems [18,19].

Our contribution will focus on identifying, logging, prioritizing, escalating, and closing an incident based on the semantic content of the reported incident using Incident Ontology as a data model. Our work also contributes to the ongoing effort of representing the ITSM domain knowledge in a machine-readable manner. The rest of the sections are organized as follows. Section two focuses on the related work. Section three deals with the proposed model. Evaluation of the proposed model is covered in section four. Conclusion and future works are articulated in section four and section five respectively.

## 2. Related Works

Mundie et al. [20] have made research on 'Incident Management Ontology' that aimed to address the lack of "*shared understanding among the IT security community*". The objective of their research is to develop an "*Incident Ontology*" that is used to create a common understanding for defining the "*processes and functions*" associated with CSIRTs (Computer Security Incident Response Team) [20]. They have used classes and relationships to ensure the cybersecurity knowledge is represented correctly and completely. Moreover, they used Web Ontology Language (OWL) annotation to capture definitions that made their ontology usable as an interactive dictionary. The "*Incident Ontology*" they have developed only focuses on security incidents whereas our ontology comprehensively considers the overall ITSM incidents.

Freitas et al. [21] have proposed "*An Ontology for IT Services*" in their research work. Their main objective is to create a shared and consistent understanding of the concepts involved in the ITSM sphere. They have presented an ontology that provides an abstract view of IT Services in a very formal approach. They were concerned with a "*more generic and technology-independent*" view of IT services "*that can be applied for every IT service*" [21]. The perspective of their study doesn't consider any best practices of ITSM including ITIL, unlike our study that focuses on ITIL-based incident management.

Czarnecki and Orlows ki [22] have made research titled "*Application of ontology in the ITIL domain*" to determine the benefit of formal, explicit, and machine-readable representation of processes in the ITIL domain. They have contributed in formulating "*competency questions*" for the ITSM domain that can be applied to evaluate other related ITSM ontologies. However, they did not present an ontology for any of ITIL publications or core processes except demonstrating how an IT service concept can be represented using ontology by taking "*a basic process*" of ITIL service strategy as a case study.

A semantics definition of the ITSM model using OWL and Semantic Web Rule Language (SWRL) presented in a work by Valiente et al. [23] th at targeted "*how the knowledge of ITIL can be captured and formally described*" using Domain Specific Language (DSL). They have contributed in defining the semantics and constraints associated with the ITSM domain. Their research work also helps organizations in adapting the ITIL framework in a well-defined manner. Their research work focused to semantically enrich the ITIL service management practices whereas our work deals with a specific process called incident management found in the ITIL service management publications.

The research work entitled "*Incident and Problem Management Using a Semantic Wiki-enabled ITSM Platform*" by Kleiner and Abecker [24] focused on developing a semantic incident and problem analyzer. The researchers developed the semantic incident and problem analyzer that is built on a semantic wiki-based configuration management system; by which "*storing all IT-relevant information*" is possible. They have used existing technologies like wikis to enable users to contribute to the process of incident and problem management. They extend non-semantic wikis by adding means to express structured data while keeping the wiki's flexibility and collaborative working style. They have achieved this using semantic annotation of the wiki pages and by creating relations between wiki pages. They have also used ontology to construct the underlying knowledge model of the wiki in addition to "*deductive reasoning for the additional derivation of implicit information from the wiki*" [24]. The study contributed significantly to prevent an error from reoccurring and to tackle-down the cause of an incident, while our study attempts to address how incidents are automatically managed after they occur.

## 3. The Proposed Model

### *3.1. System Architecture*

As it is shown in Fig.1, the proposed Incident Management System (IMS) consists of Ontology Developer, OWLExporter, Text Preprocessor, Screenshot Text Extractor, Automatic Semantic Annotator, and Incident Classifier. The Incident Ontology Developer component deals with activities related to building and populating the ontology and mapping the ontology to a relational database. The Text Preprocessor manages the extraction of relevant information from the incident email using tokenization, stop-word removal, and stemming. The Screenshot Text Extractor is responsible to turn screenshot error images found in the incident report into text. The Automatic Semantic Annotator focuses on annotating the incident report sent to the IMS. The Automatic Incident Classifier categorizes the incident to a given incident category with the consultation of the incident ontology using semantic similarity.

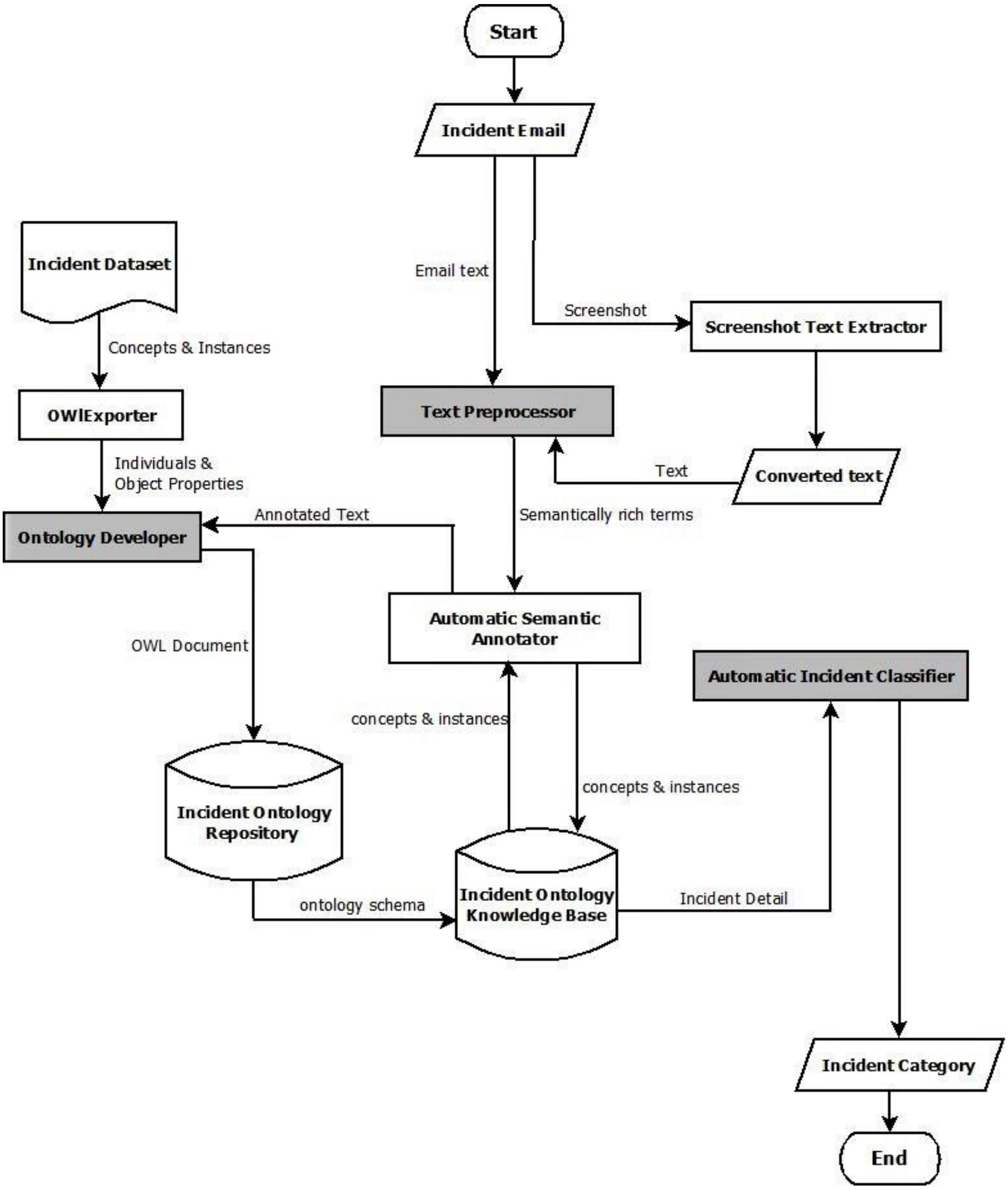


Fig.1. Architecture of the proposed Model.

The components are designed in consideration of the core functionality of the proposed solution called Incident Classification. This is because, if the reported incident is classified into its corresponding category, further incident management operations can be done efficiently. The non-shaded components in Fig.1 refer to the opensource off-the-shelf (OTS) components integrated into the proposed system.

### *3.2. The Incident Ontology*

Incident Ontology needs to be developed to semantically match the error message (incident report) to its corresponding category. We have used Ushold's and King's [25] method as an ontology development methodology. We constructed all the incident management concepts based on the three main notions of ITSM called the asset, process, and people.

#### *A. Scope of the Incident Ontology*

We have used "*Competency Questions*" suggested by [26,27] to define the scope of the incident ontology. We later validate the scope of the ontology through the competency questions and by consulting domain experts who are participated in the research work. The competency questions are:

- What are the core concepts in the ITIL framework of incident management?
- Which incident properties should we consider when we record an IT service incident?
- What are the possible IT service incident categories?
- What are the key factors that decide the priority of an IT service incident?
- What are the key characteristics of an incident used for automatic incident classification?
- What kind of semantical relationships existed among assets, processes, and people involving in IT services?

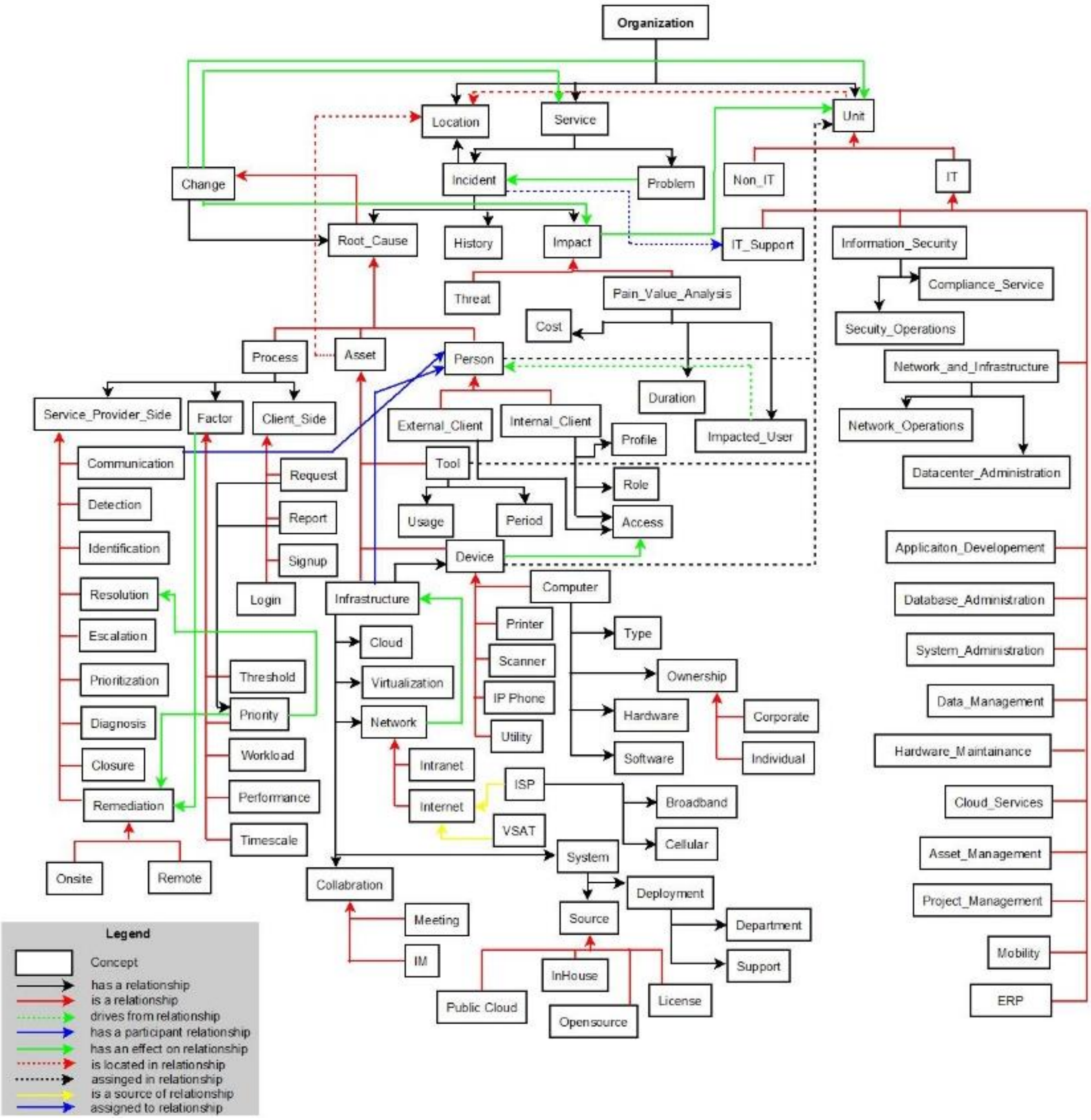


Fig.2. The Incident Concept Taxonomy.

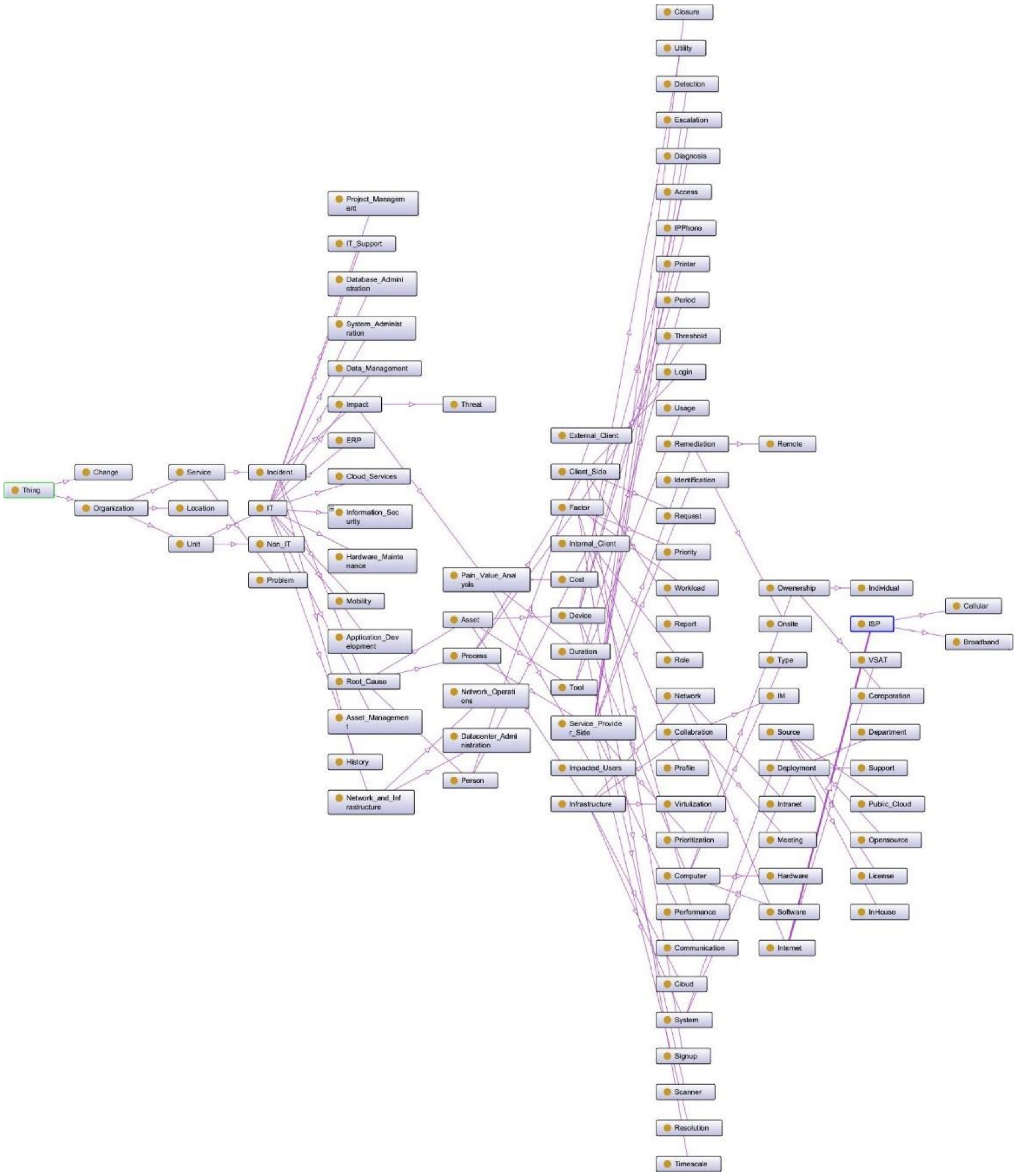


Fig.3. The Incident Ontology.

### *B. Building the Incident Ontology*

The construction of the term glossary is an important task in ontology engineering since it helps to identify key concepts and their descriptio ns [10,28,29]. Thus, we have constructed a glossary of terms and later validated it by domain experts. A concept taxonomy that shows concepts and their relationships is a "*backbone of ontology*" [10]. Researches have shown that concept taxonomies play a significant role in searching and classifying information from a t ext [30]. Concept taxonomies also carry semantics unambiguously that makes them "*an essential component of knowledge modeling*" [16]. As a result, we have identified the relevant terms and their relationships based on the term glossary and organized them as the concept taxonomy depicted in Fig.2. After the concept taxonomy is developed, the corresponding relationships among concepts are identified. In addition to the traditional "*is_a*" and "*is_part_of*" relationships, other domain-specific relationships are defined including "*drives_from*", "*has_an_effect_on*", "*is_a_source_of*", "*is_located_in*", and "*assigned_to*" relationships.

*C. Coding the Incident Ontology*

Once the concepts and their relationship are identified and modeled in a concept hierarchy, the ontology is created using protégé as it is shown in Fig.3. The OWL language is used to author the Incident Ontology.

*D. Populating the Incident Ontology*

To use the ontology as a knowledge base for further semantic content extraction, the Incident Ontology has to be populated with instances or individuals. OwlExp orter [31] is used to populate the incident ontology with individuals along with their datatype and object properties. We choose OwlExporter because it is relatively easy to implement and well-suited for incidents reported in English. It also allows us to "*easily map existing NLP analysis pipelines to OWL ontologies*" [31]. We have used two datasets to populate the incident ontology. The OwlExporter used the first dataset that is available online for free by "*Center for Machine Learning and Intelligent Systems*" and contains more than twenty-four thousand actual incidents reported to a commercial ITSM platform called ServiceNow. We have also consolidated our mini dataset that contains more than a hundred concept relationships, individuals, datatype properties, and values based on the concept taxonomy we developed. We use the second dataset to create a relationship between individuals. For this purpose, we have developed the Instance Relationship Maker Algorithm shown in Algorithm 1.

*Algorithm 1. Instance Relationship Maker Algorithm*

**Instance Relationship Maker Algorithm**

```
Input:
        Dataset: incident_Dataset
        Ontology: incident_Ontology
        first_Instance=NULL
        second_Instance=NULL
        first_Property=NULL
Output:
        incident_Ontology ← new_Instance_Relationship
        // new_Instance_Relationship among the participating concept individuals shall be added.

BEGIN:
FOR a File F IN incident_Dataset
  DO
   FOR EACH line L in F
     FOR EACH instance_Relationship R in L
        READ first_Instance,second_Instance, first_property FROM R
          FOR EACH concept C IN incident_ontology
             READ first_instance.concept_Type, second_instance.concept_Type, FROM incident_ontology
             Incident_ontolgoy.INSERT(first_instance, property, second_instance)
          END FOR
     END FOR
   END FOR
  WHILE (NOT EndOfFile)
 END FOR
RETURN incident_Ontology
END:
```

*3.3. The Incident Database*

The OWL incident ontology shown in Fig. 3 is mapped onto the relational MYSQL database. We have used a plugin called OWL2ToRDB developed by research wor k in [32] that implements the mapping scheme suggested in [33] to convert the incident ontology to a relational database. We used this plugin because it directly converts the incident ontology concepts and their relationships, object and datatype properties, and instances to relational database tables.

*3.4. The Screenshot Text Extractor*

There shall be a scenario where clients may send an incident report with a screenshot image as it is shown in Fig.5. In such a scenario, the Extractor component converts a screenshot of the incident report to editable text for further text preprocessing. We use an open-source OTS component called '*pytesseract 0.3.4*' suggested in [34] to convert a screenshot of an error image to text. We have selected this component because it supports different image types including jpeg, jpg, png, and gif and it provides an API to adapt and use its features.

*3.5. The Text Preprocessor*

Several studies [35,36,37] have revealed the significant impact of text preprocessing on different phases of text classification systems including feature extraction, feature selection, and classification. The Text Preprocessor combines the incident report and the converted text retrieved from the screenshot in a single file. This component is also responsible for tokenization, stop-word removal, and stemming tasks on the incident report. The tokenization is the first preprocessing task done to extract the words from the incident report. The next text preprocessing task is to remove stop-words since they are semantically irrelevant to the proposed solution. The last text preprocessing task on the incident report is stemming by which the words in the incident report are converted to their corresponding stem word. We have used the NLTK 3.5 Python library [38] for all text preprocessing tasks including stop-word removal and stemming.

### *3.6. The Automatic Semantic Annotator*

We have used the automatic semantic annotation tool called Ontea [39] that is capable to identify objects and properties (semantically rich terms) found in the incident report by applying a regular expression pattern on the report and by consulting the knowledge base. We choose Ontea because of its simplicity for prototype development (as it is based on regular expressions) and because it can be easily adapted to a domain like IT incident management. Ontea also works well to search relevant instances in the incident knowledge base and even to create new instances if no instance is found in the incident ontology.

### *3.7. The Automatic Incident Classifier*

The Automatic Incident Classifier is the component that classifies a reported incident to a given category based on the semantic similarity between the semantic units in the incident report and concepts in the Incident Ontology. The incident classifier component queries the IMS knowledge base in consideration of the instances and their relationship (the semantic units) in the report. To achieve this, the classifier uses Algorithm 2.

*Algorithm 2. Incident Classifier*

| Step | Incident Classifier Algorithm |
|---|---|
| 1<br>2<br>3<br>4 | Input:<br>Concept_List: Set of semantically rich concepts<br>Ontology:incident_Ontology<br>Incident_Category=NULL<br>Semantic_Similarity=0 //No semantic similarity exists between two unknown concepts in a list. |
| 5 | Output:<br>incident_Class ← incident_Category<br>// category of an incident used for further incident management processes. |
| 6<br>7<br>8<br>9<br>10<br>11<br>12<br>13<br>14<br>15 | BEGIN:<br>FOR EACH concept $C_1$ in concept_List C<br>FOR EACH concept $C_2$ in incident_Ontology<br>COMPUTE semantic_Similarity($C_1$, $C_2$)<br>READ first_Instance,second_Instance, first_property FROM R<br>IF semantic ($C_1$)==semantic ($C_2$)<br>incident_Category=superConcept($C_2$)<br>ELSE<br>Incident_Category=$C_1$<br>END FOR<br>END FOR |
| 16 | RETURN incident_Ontology<br>END: |

### *3.8. A System Scenario*

In this section, we present a simple scenario to demonstrate how the proposed model works. Let's consider the Incident instance found in the incident ontology that is shown in Fig.4 and a sample incident report shown in Fig.5. The '*Incident*' concept is related to the '*Service*', '*Person*', and '*IT_Unit*' concepts. The semantics is, an incident reported by a person that involves a given IT service must be assigned to a unit in IT. If we consider the sample incident report, we will have the following output after we applied the OwlExporter.

*Incident1={"Incident": "Unable to access", "Service": "OneDrive Cloud Storage", "Device": "Personal Device"}*

As it is shown in the system architecture, the automatic annotator Ontea adds semantically significant notes on the above data by consulting the knowledge base. So, information about who reports the incident (Client) and which IT unit is responsible for the incident (Unable to access OneDrive Cloud Storage) will be added and the output will be enhanced as follows.

*Incident1={"Incident": "Unable to access", "Service": "OneDrive Cloud Storage", "Device": "Personal Device", "Client": "John Smith", "IT_Unit": "Cloud Services"}*

The interpretation is John Smith (a client) has reported an 'unable to access' issue (Incident) from his personal device (Device) involving OneDrive cloud storage (Service) which has to be assigned to Cloud Services Unit (IT_Unit).

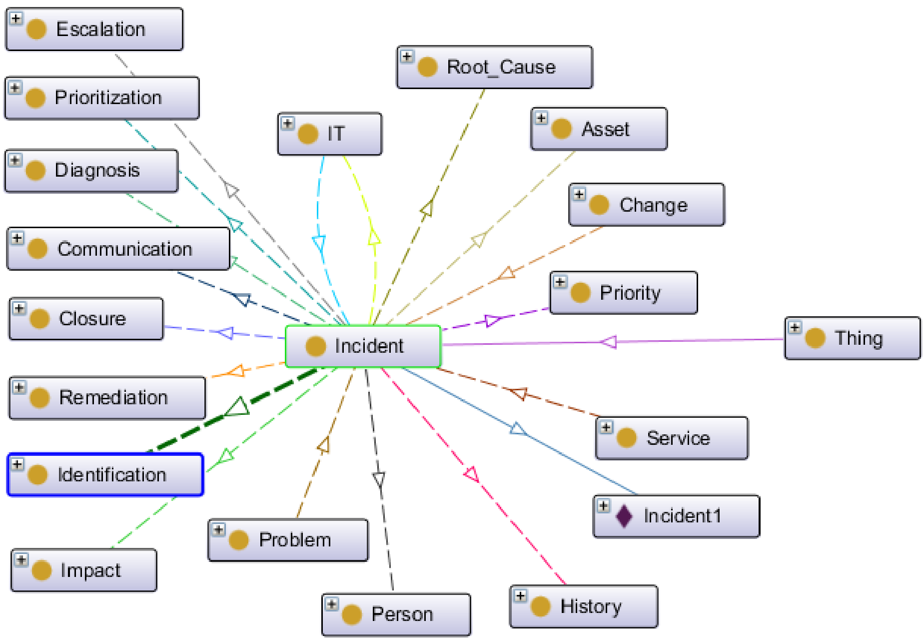


Fig.4. The 'Incident' Concept.

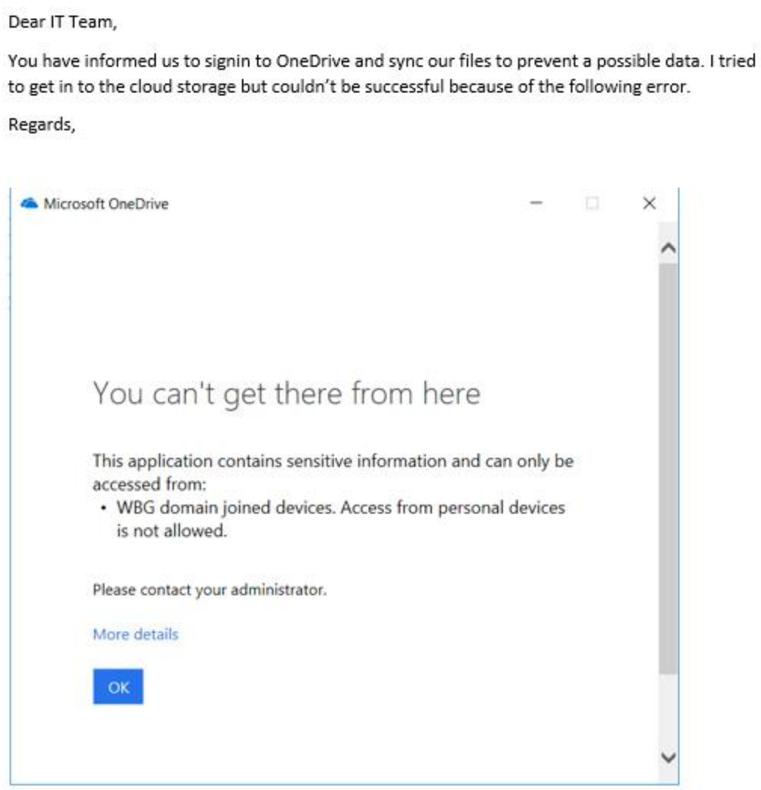


Fig.5. Sample Screenshot of Incident Report.

## 4. Evaluation of the Model

### *4.1. Overview*

We have designed a system architecture that can be followed to develop intelligent incident management systems.

We have developed a simple web application (Fig.6) based on the architecture of the system to validate the proposed system's performance. The prototype is developed using the opensource Django Web Framework 2.2 with Python 3.7 using PyCharm 2019.3.1 (Student Edition) IDE. The prototype considers the basic incident management activities including sending an incident report, recording a reported incident, escalating an incident, resolving and closing the incident ticket.

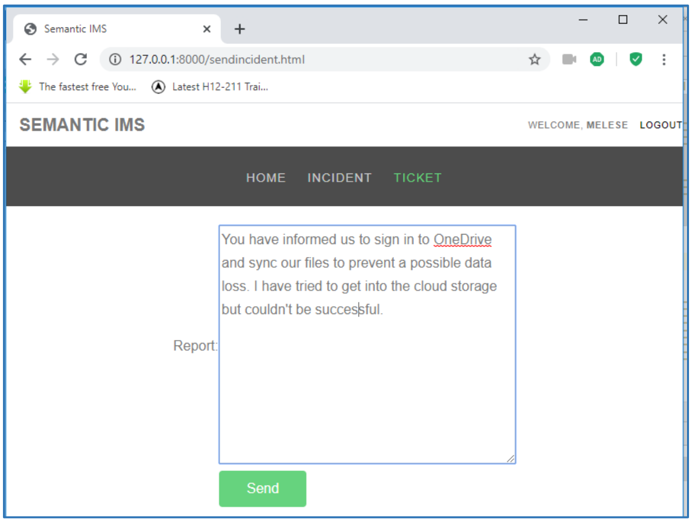


Fig.6. Incident Reporting Scenario.

### *4.2. Evaluation Dataset*

We have used a dataset to populate ontology and later mapped it to the relational MYSQL database. The dataset is compiled and enriched by the "*Center for Machine Learning and Intelligent Systems*" [40]. The dataset contains extracted incident logs and the data was modified to make it anonymous and to handle privacy issues [40]. We choose this dataset because it consists of relevant concepts and a list of individuals along with their properties that is well-suit for a prototype. The dataset contains 24,918 actual incidents and thirty-six attributes including *Incident_number, incident_state, location*, and *priority* [40].

### *4.3. Evaluation Model*

We have implemented the Semantic Measures Library & ToolKit package to evaluate six sample incident reports. The model used the graph-based semantic similarity measure. We prefer to use this toolkit since its implementation is well tested and documentation is provided. The graph-based similarity measurement model which perceives the ontology as a semantic graph is widely used to evaluate concepts and instances defined in domain ontology [16,41]. The notion of semantic similarity measurement is by itself a separate research topic and beyond this research work.

### *4.4. Sample Incident Reports*

Table 1. Sample Incident Reports

| Incident ID | Incident Report |
|---|---|
| IN001 | You have informed us to sign in to OneDrive and sync our files to prevent a possible data loss. I have tried to get into the cloud storage but couldn't be successful because of the following error. |
| IN002 | Scanner login problem |
| IN003 | Access to Abebe Kebede Alemu |
| IN004 | Hi Team,<br>My email keeps asking me for a password. I have tried many times but nothing happens. |
| IN005 | Dear IT Team,<br>I am a consultant and I want to request a payment. Would you please assist me? |
| IN006 | Hi Team,<br>Did you check the 3$^{rd}$-floor color printer? I removed the paper jam but still showing error. |

### *4.5. Evaluation Metrics*

We have considered six actual incident reports to evaluate whether they are correctly classified and assigned to a relevant IT unit. As it is depicted in Table 2, we have considered three classes of an incident which are labeled as 'Device', 'Infrastructure', and 'Request'.

Table 2. Evaluation Metrics

| Label | Device | Infrastructure | Request | Device | Infrastructure | Request |
|---|---|---|---|---|---|---|
| Classification | Device | Infrastructure | Device | Device | Infrastructure | Device |

We present the evaluation result in Table 3 using a confusion matrix.

Table 3. Confusion Matrix

| | | Correctly Classified | |
|---|---|---|---|
| | | Negative | Positive |
| Actual | Negative | True Negative: 0 | False Positive: 2 |
| | Positive | False Negative: 0 | True Positive: 4 |

We have used recall, precision, and F1 score as evaluation metrics of the proposed system. The Recall is defined in Equation (1), Precision is defined in Equation (2), and F1 Score is defined in Equation (3).

$$recall = \frac{truePositive}{truePositive + falsePositive} = 1.0 \tag{1}$$

$$\mathrm{Pr}ecision = \frac{truePositive}{truePositive + FalsePositive} = 0.66 \tag{2}$$

$$F1 = 2 * \frac{p\mathrm{r}ecision * recall}{p\mathrm{r}ecision + recall} = 0.79 \tag{3}$$

## 5. Conclusion

In this research work, we developed a comprehensive Incident Ontology that can serve as a knowledge base for intelligent IMS. We have also proposed a framework that can serve as a guidance to develop ontology-driven intelligent IMS. The proposed intelligent IMS receives incident from the user through email. If the email consists screenshot of an error, the Screenshot Text Extractor converts the screenshot into text using OCR technology. Otherwise, the Text Preprocessor fetch semantically rich terms from the email using text preprocessing techniques and forward the terms to Automatic Semantic Annotator component. The Semantic Annotator annotates the terms in consultation with the Incident Ontology and repopulate the knowledge base with concepts and instances that remodels the reported incident in a machine understandable manner. Later, the Automatic Incident Classifier, classifies the reported incident based on the semantic similarity between the incident details from the knowledge base and concepts in the Incident Ontology. Finally, the remediation, escalation, prioritization, and closing operation of the incident continues accordingly once the incident category is automatically identified. We have developed a prototype to evaluate the performance of the proposed ontology-driven model, and observed that ontology-driven incident classification facilitates the process of IT incident management in a better way since the model shows 100% recall, 66% precision, and 79% F1-Score for sample incident reports.

## 6. Future Works

This research work contributes in two directions. Our first contribution is the framework we proposed which can be used to design and implement intelligent helpdesk systems. Secondly, we constructed a complete ITIL-based incident ontology that can be integrated and used by other researchers and ontology engineers. This contribution will significantly reduce the time and effort that will be spent to develop incident ontology. Despite our effort to design and implement an ontology-driven IT incident management model, there are still some gaps that have to be filled to produce fully-fledged intelligent incident management systems. We have summarized the future directions in this topic as follows:

- In the production environment, clients often send incident reports that include only a screenshot of an error. Besides, some of the screenshots may contain vague texts or numeric values like memory addresses. The interpretation and classification of these kinds of incident reports require further study.
- The ITIL distinct IT service management processes as Incident Management, Event Management, and Problem

Management. These three processes are interrelated [42,43] and designing an intelligent helpdesk system has to consider integrating the concepts found in these processes in a single ITSM ontology.
- The evaluation of the model is done only using six sample incidents. An extensive evaluation has to be conducted for an improved model with better performance.

## References


[1] Galup, S.D., Dattero, R., Quan, J.J., and Conger, S.: 'An overview of IT service management', Communications of the ACM, 2009, 52, (5), pp. 124-127.
[2] Pollard, C., and Cater-Steel, A.: 'Justifications, strategies, and critical success factors in successful ITIL implementations in US and Australian companies: an exploratory study', Information systems management, 26, (2), pp. 164-175, 2009. DOI: 10.1080/10580530902797540
[3] Kneller, M.: 'Executive briefing: the benefits of ITIL®', The Stationery Office, UK, 2010.
[4] Van Bon, J., De Jong, A., Kolthof, A., Pieper, M., Tjassing, R., Van der Veen, A., and Verheijen, T.: 'Fundamentos de ITIL®', Van Haren, 2008.
[5] Brewster, E., Griffiths, R., Lawes, A., and Sansbury, J.: 'IT service management: a guide for ITIL foundation exam candidates', BCS, The Chartered Institute for IT, 2012.
[6] Cusick, J.J., and Ma, G.: 'Creating an ITIL inspired Incident Management approach: Roots, response, and results', IEEE, (Ed.): 'Book Creating an ITIL inspired Incident Management approach: Roots, response, and results', pp. 142-148, 2010. DOI: 10.1109/NOMSW.2010.5486589
[7] Berrios, D.C., and Keller, R.M.: 'Semantic Analysis of Email Using Domain Ontologies and WordNet', 2005
[8] Wimalasuriya, D.C., Dou, D., Farley, A., and Young, M.: 'Ontology-Based Information Extraction', University of Oregon, 2009.
[9] Sheth, A.P., and Ramakrishnan, C.: 'Semantic (Web) technology in action: Ontology driven information systems for search, integration, and analysis', IEEE Data Engineering Bulletin, vol 26, (4), pp. 40, 2003.
[10] Staab, S., and Studer, R.: 'Handbook on ontologies, Springer Science & Business Media, 2010.
[11] Van Harmelen, F., Lifschitz, V., and Porter, B.: 'Handbook of knowledge representation', Elsevier, 2008.
[12] Miller, S.J.: 'Introduction to Ontology Concepts and Terminology', in Editor (Ed.)^(Eds.): 'Book Introduction to Ontology Concepts and Terminology' 2013.
[13] Bench-Capon, T.J.: 'Knowledge representation: An approach to artificial intelligence' , Elsevier, 2014.
[14] Domingue, J., Fensel, D., and Hendler, J.A.: 'Handbook of semantic web technologies, Springer Science & Business Media, 2011.
[15] Kaur, N., and Aggarwal, H.: 'Evaluation of information retrieval based ontology development editors for semantic web', International Journal of Modern Education and Computer Science, vol 9, (7), pp. 63, 2017.
[16] Harispe, S., Ranwez, S., Janaqi, S., and Montmain, J.: 'Semantic similarity from natural language and ontology analysis', Synthesis Lectures on Human Language Technologies, vol 8, (1), pp. 1-254, 2015. DOI: 10.2200/S00639ED1V01Y201504HLT027
[17] Karkaletsis, V., Fragkou, P., Petasis, G., and Iosif, E.: 'Ontology based information extraction from text': 'Knowledge-driven multimedia information extraction and ontology evolution', Springer, pp. 89-109, 2011.
[18] Embley, D.W.: 'Toward semantic understanding: an approach based on information extraction ontologies', in Editor (Ed.)^(Eds.): 'Book Toward semantic understanding: an approach based on information extraction ontologies', pp. 3-12, 2004.
[19] Li, Y., and Bontcheva, K.: 'Hierarchical, perceptron-like learning for ontology-based information extraction', in Editor (Ed.)^(Eds.): 'Book Hierarchical, perceptron-like learning for ontology-based information extraction', pp. 777-786, 2007.
[20] Mundie, D.A., Ruefle, R., Dorofee, A.J., Perl, S.J., McCloud, J., and Collins, M.: 'An Incident Management Ontology', in Editor (Ed.)^(Eds.): 'Book An Incident Management Ontology', pp. 62-71, 2014.
[21] Freitas, J.M., Correia, A., and e Abreu, F.B.: 'An Ontology for IT Services', in Editor (Ed.)^(Eds.): 'Book An Ontology for IT Services', pp. 367-372, 2008.
[22] Czarnecki, A., and Orłowski, C.: 'Application of Ontology in the ITIL Domain', 2011.
[23] Valiente, M.C., Rodríguez, D., and Vicente Chicote, C.: 'Defining the semantics of IT service management models using OWL and SWRL', 2010.
[24] Kleiner, F., Abecker, A., and Mauritczat, M.: 'Incident and Problem Management using a Semantic Wiki-enabled ITSM Platform', in Editor (Ed.)^(Eds.): 'Book Incident and Problem Management using a Semantic Wiki-enabled ITSM Platform', pp. 363-372, 2012.
[25] Uschold, M., and King, M.: 'Towards a methodology for building ontologies' Citeseer, 1995.
[26] Mizoguchi, R.: 'Tutorial on ontological engineering Part 2: Ontology development, tools and languages', New Generation Computing, vol 22, (1), pp. 61-96, 2004.
[27] Slimani, T.: 'Ontology development: A comparing study on tools, languages and formalisms', Indian Journal of Science and Technology, vol 8, (24), pp. 1-12, 2015. DOI: 10.17485/ijst/2015/v8i34/54249
[28] Suárez-Figueroa, M.C., de Cea, G.A., and Gómez-Pérez, A.: 'Lights and shadows in creating a glossary about ontology engineering', Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication, vol 19, (2), pp. 202-236, 2013. DOI: 10.1075/term.19.2.03sua
[29] Novalija, I., Mladenić, D., and Bradeško, L.: 'OntoPlus: Text-driven ontology extension using ontology content, structure and co-occurrence information', Knowledge-Based Systems, vol 24, (8), pp. 1261-1276, 2011. DOI: 10.1016/j.knosys.2011.06.002
[30] Meijer, K., Frasincar, F., and Hogenboom, F.: 'A semantic approach for extracting domain taxonomies from text', Decision Support Systems, vol 62, pp. 78-93, 2014. DOI: 10.1016/j.dss.2014.03.006
[31] Witte, R., Khamis, N., and Rilling, J.: 'Flexible Ontology Population from Text: The OwlExporter', in Editor (Ed.)^(Eds.): 'Book Flexible Ontology Population from Text: The OwlExporter', pp. 3845-3850, 2010.
[32] Vysniauskas, E., Nemuraite, L., and Paradauskas, B.: 'Hybrid method for storing and querying ontologies in databases', Elektronika ir Elektrotechnika, vol 115, (9), pp. 67-72, 2011. DOI: 10.5755/j01.eee.115.9.752

[33] Zhang, H., Wang, Z., Gao, Z., and Li, W.: ‘Design and implementation of mapping rules from OWL to relational database’, in Editor (Ed.)^(Eds.): ‘Book Design and implementation of mapping rules from OWL to relational database’, IEEE, pp. 71-75, 2009. DOI: 10.1109/CSIE.2009.290
[34] Smith, R.: ‘Tesseract ocr engine’, Lecture. Google Code. Google Inc, 2007.
[35] Uysal, A.K., Gunal, S., Ergin, S., and Gunal, E.S.: ‘A novel framework for SMS spam filtering’, in Editor (Ed.)^(Eds.): ‘Book A novel framework for SMS spam filtering’, IEEE, pp. 1-4, 2012. DOI: 10.1109/INISTA.2012.6246947
[36] Zareapoor, M., and Seeja, K.: ‘Feature extraction or feature selection for text classification: A case study on phishing email detection’, International Journal of Information Engineering and Electronic Business, vol 7, (2), pp. 60, 2015.
[37] Tan, S., Wang, Y., and Wu, G.: ‘Adapting centroid classifier for document categorization’, Expert Systems with Applications, vol 38, (8), pp. 10264-10273, 2011. DOI: 10.1016/j.eswa.2011.02.114
[38] OpenNLP, A.: ‘Natural language toolkit’
[39] Laclavik, M., Šeleng, M., Ciglan, M., and Hluchý, L.: ‘Ontea: Platform for pattern based automated semantic annotation’, Computing and informatics, vol 28, (4), pp. 555–579, 2012.
[40] Amaral, C.A., Fantinato, M., Reijers, H.A., and Peres, S.M.: ‘Enhancing Completion Time Prediction Through Attribute Selection’: ‘Information Technology for Management: Emerging Research and Applications’, Springer, pp. 3-23, 2018. DOI: 10.1007/978-3-030-15154-6_1
[41] Lord, P.W., Stevens, R.D., Brass, A., and Goble, C.A.: ‘Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation’, Bioinformatics, vol 19, (10), pp. 1275-1283, 2003. DOI: 10.1093/bioinformatics/btg153
[42] ItSMF, U.: ‘ITIL Foundation Handbook’, The Stationery Office, 2012.
[43] Ray, S., and Chandra, N., "Domain based ontology and automated text categorization based on improved term frequency-Inverse document frequency", International Journal of Modern Education and Computer Science(IJMECS), vol 4, No.1, pp. 28-35, 2012. DOI: 10.5815/ijmecs.2012.04.04

## Authors’ Profiles

**Bisrat Betru** has received his Master of Computer Science from Addis Ababa University at Addis Ababa, Ethiopia. He has also received another Master of Science in Applied Computer and Information Technology from Oslo Metropolitan University at Oslo, Norway. His research interest includes intelligent systems, semantic web, HCI, Interaction Design and Learning Technologies, Universal Design, AI, Deep Learning, Knowledge Graphs, and Ontology-driven Information Extraction.

**Fekade Getahun** has received his Ph. D. degree in Computer Science from University of Bourgogne at Dijon, France. He is serving as an Assistant Professor in Department of Computer Science at Addis Ababa University. His major research interest includes Sematic Web, Knowledge Management, AI, Cloud Computing, Content-based Retrieval, Web-based Similarity, Multimedia Database, E-Service Computing, Ontology-driven Systems, Emotion recognition, and Computer Vision.

**How to cite this paper:** Bisrat Betru, Fekade Getahun, "Ontology-driven Intelligent IT Incident Management Model", International Journal of Information Technology and Computer Science(IJITCS), Vol.15, No.1, pp.30-41, 2023. DOI:10.5815/ijitcs.2023.01.04