

CONTEXT-AWARE ONTOLOGY AND WEB SERVICES DISCOVERY FOR DISTRIBUTED EMBEDDED REAL-TIME SYSTEMS

Furkh Zeshan^{1}, Radziah Mohamad², Mohammad Nazir Ahmad³, Marini Bte Othman⁴, Arafat Abdulgader Mohammed Elhag⁵, Syed Asad Hussain⁶, Adnan Ahmad⁷, Amjad Ali⁸, Mehmood Ashraf⁹, Imran Babar¹⁰*

^{1,6,7,8}Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Pakistan

^{2,10}Department of Software Engineering, Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia

³Institute of Visual Informatics, Universiti Kebangsaan Malaysia

⁴College of Computer Science and Information Technology, Universiti Tenaga Nasional

⁵Department of Information System, Bisha Community College, University of Bisha, Saudi Arabia

⁹Federal Urdu University of Arts, Science & Technology, Islamabad

Email: farrukh05@hotmail.com^{1*} (corresponding author); radziahm@utm.my²; mnazir@ukm.edu.my³; Marini@uniten.edu.my⁴; arofei@hotmail.com⁵; asadhussain@cuilahore.edu.pk⁶; adnanahmad@cuilahore.edu.pk⁷; amjad.ali@cuilahore.edu.pk⁸; mahmood313@gmail.com⁹

DOI: <https://doi.org/10.22452/mjcs.vol32no3.2>

ABSTRACT

Distributed Embedded Real-Time Systems (DERTS) consists of hundreds of interconnected devices, typically small and wirelessly-connected, which are designed to work for a long period of time. The massive interconnection of devices and the usage of heterogeneous languages, operating platforms and data standards make DERTS a competitive and complex system. In addition, the DERTS program is setting a trend of moving away from centralized, high-cost products towards lower cost and high volume products. In this regard, there is nothing more natural than considering the use of Service Oriented Architecture (SOA) to assist in the development of DERTS. This is because SOA enables different devices to exchange data regardless of issues of complexity. Moreover, context-awareness, which is widely studied for DERTS, also plays an important role for effective communication among devices. Thus, to build service-based DERTS while managing the complexity, context-aware ontologies are the best solution. In this paper, we developed a context-aware ontology for DERTS which is known as ConOntDERTS. To evaluate ConOntDERTS, we used two methods. In the first method, a criteria-based ontology evaluation was used; while in the second method, a survey was conducted to show that the results produced by ConOntDERTS were almost the same as human perception. Results of the evaluation show the consistency and feasibility of our ontology and the statistical test results show that ConOntDERTS ontology can produce consistent results with human perception.

Keywords: *Context-Aware Ontology; Distributed Embedded Real-Time Systems; Context Knowledge Management; Semantic Web Services Discovery.*

1.0 INTRODUCTION

Distributed Embedded Real-Time Systems (DERTS) are typically small and wirelessly connected devices designed to work for a long period of time [1, 45, 46, 47, 48, 49, 50, 51, 62]. The DERTS system is quite different from traditional distributed systems as they interconnect hundreds of devices. With the advancement of the electronic world, particularly in the field of microprocessor technology, DERTS are becoming popular and have recently been as widely used in industry as mobile phones. Over the past decade, two major trends have been observed in the embedded devices world: firstly, hardware is becoming smaller, cheaper and more powerful; and secondly, the software industry is moving towards service-oriented integration technologies [2]. According to the Internet of Things (IoT) vision [3], the majority of devices will soon have communication and computation capabilities. These will be used to connect, interact and cooperate with their surrounding environment. The massive interconnection of devices and the usage of different languages, data standards and operating platforms make DERTS both competitive and complex. In this regard, there is nothing more natural than considering the use of Service Oriented Architecture (SOA) for the development of DERTS. SOA, which is an advanced methodology, can be used for the development of loosely-coupled, flexible, distributed and cost-effective applications for the dynamic environments of DERTS. SOA enables different applications to exchange data and participate in different processes regardless of the complexity of the applications. SOA provides a standard approach for the development of an application in which

heterogeneously-programmed and operating platform components can work together through exposing their functionalities as services.

Ontologies have been determined as an important tool for context modeling in relation to integration, sharing and reuse of context knowledge from distributed and heterogeneous sources of information [5, 6]. Moreover, context-aware ontologies provide a communal model of information-sharing as well as a language for defining constraints on the shared information; hence, reducing both cost and the possibility of redundancy in context-sensing. However, to facilitate the automatic reasoning, dynamic nature of devices should be considered. Therefore, in this research work, we developed a DERTS context ontology which defines the formal specification (axioms) of the concepts to enable flawless exchange of information and knowledge discovery.

For the last two decades, researchers have considered the concept of context-awareness in relation to enhancing human-computer interaction by providing applications with context information [7, 8, 9]; while in recent years, context-aware ontologies have been playing an important role in DERTS computing. Context-aware ontologies have emerged as an important tool for context modeling by which to integrate, share and reuse context knowledge from distributed and heterogeneous sources of information [5, 15]. Building context-aware DERTS for context reasoning and knowledge-sharing is difficult and costly without the proper support of context-aware ontologies. Context-aware ontologies play an important role in real-time computing and in real world live data, where context changes take place rapidly. In such situations, context can enable systems to be adapted and provide accurate results to the user in a precise way. However, context information of the object may have different representations [4], which make it difficult to interpret. In such situations, context-aware ontologies are the best solutions for representing domain knowledge in an application-independent way. In order to use context information effectively, the context information should be presented in an unambiguous way and the dynamic nature of the embedded real-time systems should be considered so that other systems may process the information and perform reasoning automatically. In this regard, the developed ConOntDERTS ontology eliminates the terminology heterogeneity and enables the use of reasoning tools for knowledge discovery. In this paper, ConOntDERTS is used to facilitate the effective semantic matching of resource requests and advertisements in the dynamic environment of DERTS. To check the effectiveness of ConOntDERTS, we developed a questionnaire and submitted it to the domain experts. We then compared the results produced through ConOntDERTS with the results obtained through the questionnaire. The comparative evaluation shows the effectiveness of ConOntDERTS, while the rest of the detail is provided in Section 5.

The remaining paper is organized as follows: Section 2 presents the background and related works. In Section 3, DERTS context requirements and ontology development methodologies are discussed. Section 4 provides the detail of ontology construction while Sections 5 and 6 give ontology evaluation details. Finally, the overall conclusions and suggestions for future work were presented in Section 7.

2.0 BACKGROUND AND RELATED WORK

In computer science, an ontology is a software artifact [10] which allows the introduction of mechanisms for automatic reasoning, aiming at reducing conceptual and terminological confusion so as to improve communication, sharing, interoperability and the degree of possible reuse. The major advantages of ontologies are described in [11]: Ontologies can help a designer to increase the quality of conceptual analysis, the use and maintainability of the system, and the reuse and sharing of domain knowledge by using a common vocabulary across heterogeneous software platforms [11]. Ontology development is an iterative process. An ontology can be built either from scratch (through the re-engineering of other existing ontologies) or by a process of ontology merging. At the end of each iteration, the ontology is evaluated using either an application or problem-solving methods or by discussing it with the domain experts and then revised if needed.

Context-awareness, which is closely related to embedded systems [24], deals with the parameters of objects like location, time, role, state, user identity and device capabilities [60 - 62]. According to [16], context can be interpreted as follows: *“Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”* Context information is very broad and highly interrelated which makes it difficult to interpret and reuse. Context information plays an important role in the delivery of correct service and it can be considered a specific kind of knowledge which can be modeled with the help of ontology.

Context-aware ontology, by providing formal semantics to context knowledge, supports the sharing and integration of context information.

A formal context of languages for representing and reasoning the context knowledge based on the relationships among facts has remained a hot topic of research since the last decade. During that time, different semantic matching approaches have been proposed. These approaches have addressed many limitations of traditional techniques where an overview of these techniques is presented below.

The approach proposed by Chen [13] is based on a standard ontology known as SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications), which is used for knowledge-sharing and context reasoning to achieve interoperability among agents in the Context Broker Architecture (CoBrA). The author has used FOAF, DAMLtime and spatial ontology along with the OpenGIS vocabularies to make it a part of ontology and has focused on ontology extension as well. Strimpakou [14] presents a generic ontology called COMANTO (Context Management ONTOlogy) for the support of sharing and synchronization of the context knowledge. This ontology classifies context into smaller dimensions including time, service, person, preference and place. The ontology (hybrid context modelling approach to handle context objects and context knowledge) is used to collect a structured semantic representation relating to generic context information. COMANTO provides an expressive formal model, however its drawback is that it cannot delete useless contexts. The Aspect-Scale-Context (ASC) model [6] is an ontology-based context model which is implemented through Context Ontology Language (CoOL) and is based on three different ontology languages (OWL, DAML+OIL, and F-Logic). This model provides a general way to view context where each category of the context can be seen as a scale of one dimension or measurement unit, and the context as the exact value (scales such as a kilometre scale or mile scale to express some context information like 20) is missing. Mapping functions of the ontology are used to convert context information from one scale to another. CONON is a context model proposed by [17] which is based on the idea of ASC/CoOL approach. It is a two-layered ontology where at the upper layer of the ontology, high level contexts such as location, person and activity have been presented while at the lower layer, a domain-dependent abstraction has been presented. The CAS-mine framework [18] efficiently extracts generalized association rules from user context data and the requested service allows service providers to personalize their services. Tan et al. [19] have proposed a general context ontology called SOCOM (multi Sensor Oriented Context Model). SOCOM provides a detailed view of the relationships between the sensor and the context by exploring the characteristics of different sensors being used in a context-aware computing environment. SOCOM discusses the general context knowledge of sensors and how to use this knowledge in the middleware. CoDaMoS [20, 21] is a two-layered context model based on general ontology which seeks to describe contexts in terms of fine-grained facts and higher level situations which denote logical conditions. The notion of mobile services seems to be beyond the scope of this context model. This model does not offer any explicit ways by which to limit the number of expressible contexts resulting in service discovery limitations. Moreover, multidimensional context space is not represented in a proper way resulting in difficulty relating the relevant data to the specific context. In CoBrA [22], the context is represented as a Context Knowledge Base [23], while the presence of a Context Broker makes this approach perfectly suited for context-sharing and context reasoning. During the application of CoBrA for information tailoring, the authors felt that ontologies should be enriched in order to make a Context Knowledge Base more effective. They aim at defining a set of context reasoning rules for describing and combining the context of various components. Moreover, CoBrA has not focused very much on services along with their related aspects (user interfaces and mobile devices on which these services are to be deployed). Due to those reasons, it was difficult to model the multiple contexts for multidimensional applications.

Although the above-discussed context-aware ontologies address many limitations, still, a number of unaddressed issues also exist which restrict effective communication and interaction among devices. Based on these issues, it is important that service discovery approaches of a dynamic environment should consider issues of role, resources, schedules and timeliness. These requirements, if considered, can help to deal with heterogeneity and complexity in addition to increasing the possibility of effective communication.

3.0 DERTS CONTEXT REQUIREMENTS AND ONTOLOGY DEVELOPMENT METHODOLOGY

Service composition is increasingly gaining in popularity since the composite service presents features that an individual service cannot, hence, the selection of a service for composition from many context similar services is a significant challenge. In this regard, quality of service (QoS) can be used to resolve the issue of selection of the “best service” among many context-similar services. However, context is not easily defined and modeled, since it is a very broad and diverse concept. Context information also plays an important role in the delivery of correct service and it can be considered as a specific kind of knowledge which can be modelled with the help of ontology. Context-aware ontology, by providing formal semantics to context knowledge, supports the sharing and integration of context information.

The development of successful DERT systems depends on the provision of the correct requirements specification. This section focuses on some of the general important requirements of DERTS related to the context-awareness [24, 25] that have been considered in the development of context-aware ontology for DERTS. Details are as shown in Table 1 below.

Table 1: DERTS requirements

Location	DERTS are used to automatically identify and track the location of objects or people in real time, usually within a building or other contained area. Location awareness refers to devices that can passively or actively determine their location. Location information usually does not include speed, direction, or spatial orientation. Location services can be used to address both common and critical activities of everyday life [26].
Activity and schedule	Two tasks, periodic (having hard deadlines dedicated for control loops) and aperiodic (having soft deadlines and dedicated for user interaction) are commonly used by DERTS. Periodic tasks are activated with specific intervals of time while aperiodic tasks run at unpredictable intervals. However, in the presence of constraints (minimum interval time between the execution of two consecutive tasks), the tasks are called sporadic. Tasks may also have an offset that denotes the time before any instance may be activated.
Response time	The response time is the time taken by the real-time system between the start and the end of the task. Response time is different from worst-case execution time (the maximum length of time taken by the task) and the deadline (the time during which the output of the task remains valid in the context of the specific system).
Timeliness / deadline	Timeliness refers to the ability of the system to respond to external requests within a finite and specified period of time. To achieve timeliness in DERTS, the constraints should be specified on the activation patterns of aperiodic tasks.
Power consumption	Battery power of any DERTS device is one of the most important considerations for communication, since DERTS devices use energy from batteries for their operations. In the case of DERTS devices, battery capacity is severely restricted due to constraints on size and weight of the device which means that such devices should be energy efficient in proportion to their usability.
Memory allocation	Memory as a recording media is used to retain digital data. It is a core function and fundamental component of embedded devices. It must be considered before communication and interaction.
Role	Role defines temporal relations with the other objects. For example, a person in one role may be an actor while in another role he/she may be a director. Roles can be beneficial in separating the inherent characteristics of individuals and can serve as an ontological reuse unit.

3.1 Ontology Development Methodology

Research methodology is academia’s established regularity framework for the collection and evaluation of the existing knowledge for the purpose of arriving at, and validating, new knowledge. Research methodology not only frames the study but can also identify the tools, strategies and criteria for the success of the research.

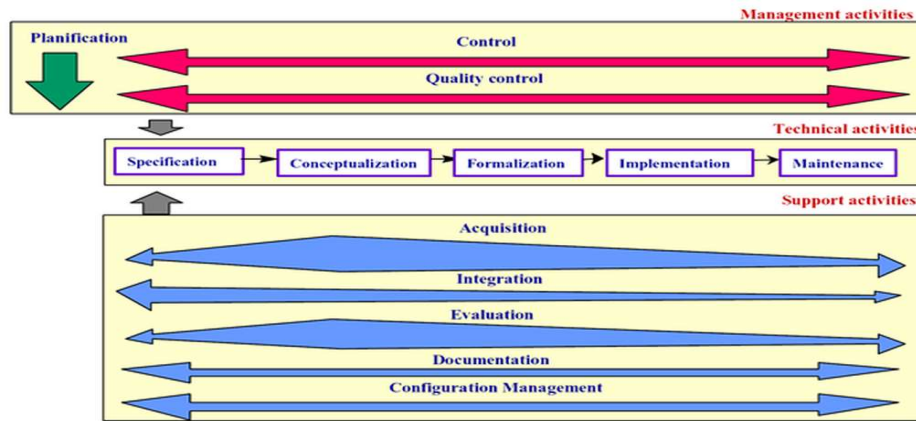


Fig. 1: METHONTOLOGY [29] ontology development process life cycle.

A number of different ontology development methodologies have been proposed in literature [27-34]. In this paper, we employed the METHONTOLOGY [29] methodology guidelines for the development of context-aware ontology because the guidelines of this methodology are based on sound principles of the software engineering which helps to improve the ontology applicability. Ontology development lifecycle of the methodology is based on stages and a set of activities through which ontology passes during its development. These tasks and activities are part of the incremental development of the ontology which makes it easier to validate the ontology during its earlier stages and increases the ontology complexity step by step. Details of these steps are represented in Figure 1 while a short introduction of each phase is discussed in the next subsection.

3.2 Specification, Conceptualisation & Formalization

The specification phase discusses the reasons for building the ontology and determines who will be the end users. In our case, DERTS designers are the end users while the objective of the context ontology is discussed in the introduction section.

The activities of the conceptualisation & formalization phases are shown in Figure 2 while the details are discussed in the next section.

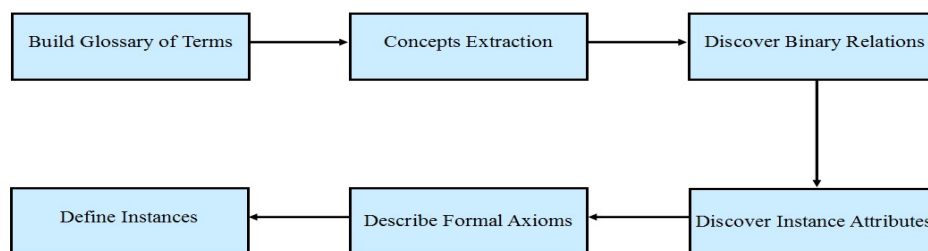


Fig. 2: METHONTOLOGY [29] ontology conceptualisation tasks

3.3 Implementation and Maintenance

These phases deal with the usage of ontology computing models and ontology corrective activities. In this paper, we used protégé as an ontology implementation tool and OWL [35] as an ontology language. Details are discussed in Section 6.

4.0 CONSTRUCTION OF CONTEXT-AWARE ONTOLOGY

Embedded real-time systems are often part of a well-specified system and operate in the context of a larger engineering system. Sometimes such systems are designed for a dedicated platform and application. Usually, real-time systems interact with other sub-systems and respond to a request from a user in the physical world within a specified period of time. These systems have time constraints regarding responses to a request. This ontology was

constructed while keeping in mind the important requirements as well as the limitations of the existing work as discussed in Sections 2 and 3 in order to achieve the goal of effective communication within the constraints. This is because the ontology can never be complete and always requires a design to be extended with new concepts without changing the underlying model and without affecting the semantic annotations. Thus, this is a first step towards the development of a context-aware ontology for communication among devices. The details of ontology construction steps are described below.

4.1 Building a glossary of terms and entity extraction

Ontology of any domain explains the structure of the knowledge, hence, plays the role of heart for that domain. During the development of the ontology, a number of papers and manuals were reviewed [52-59] for the purpose of information extraction (names, characteristics of the entities and the relationship among these terms). We started by recording the structured English sentences that were used in later stages for identifying the concept and the relationships among that concept. For example, from the sentence “each device consumes resources”, it can be concluded that “device and resources” are the concepts while “consume” refers to the relationship among these concepts.

Domain experts can use these concepts and the relationships to check the context. Details of the extracted concepts are given in Table 2.

Table 2: Glossary of terms

Concept name	Description
Activity	An activity is a process that is executed in order to achieve a goal.
Bandwidth	This is the rate of data transfer over the network; measured in bits per second.
CPU	A central processing unit (CPU) is the hardware within the computer systems where most of the calculations take place.
Device	A device is an object for which the mechanical or electrical workings are controlled by a microprocessor.
Embedded Device	An embedded device is a special purpose computer system. It is a combination of hardware and software which facilitates a variety of applications.
Real-Time Device	A real-time device is one that must respond within strict time constraints.
Laptop	This is a personal computer for mobile use.
Mobile Phone	This is an electronic telecommunications device that can provide voice communications and Internet services while moving around a wide geographic area.
Date Time	This possesses digital date and time information attached to a certain event/activity which has occurred, and usually gives the date and time of day.
Execution Time	This is a number of event/activity/process runs denoted in seconds.
Start Time	This is the start time of an event/activity/process to execute.
End Time	This is an end time of an event/activity/process to execute.
Service	The software system has been designed to support interaction between two machines over the network.
QoS Metric	Quality of Service metrics measure the quality of service provided. QoS metric is an important concept of context-aware ontology because it can play a vital role in services-ranking as an additional set of criteria which enables the requester to select the most suitable service among context similar services [62].
Unit	Measurement unit of the QoS metric.
Activity Schedule	An activity schedule is an execution sequence of activities.
Remaining Bandwidth	This refers to remaining data communication resources expressed usually in bits per second.
Remaining Memory	This has a current amount of available memory.
Resource Usage Level	This is an activity designed to see the level of resources used such as CPU, memory, battery, bandwidth etc.

Hard Disk Drive	This is a permanent data storage device used for storing and retrieving digital information.
Memory	This comprises a total amount of memory of a system.
Tablet Personal Computer	This acts as both a mobile computer and a kind of embedded device.
Device Status	Status is a scale of standing; it informs other devices whether or not the device is available for interaction.
Battery	This comprises a total amount of battery power of a system.
Human/Agent	This is an autonomous entity which observes through sensors and acts upon an environment using actuators.
Operating System	This possesses sufficient software to manage computer resources.
Real Time Operating System	A real-time operating system has been designed to serve real-time application requests without buffering delays.
Resources	A resource is a hardware or software component with limited availability within a computer system.
Network	A network is an electronic medium through which different devices can share resources and information.
Context	This deals with situation recognition of moving entities.
Context Provider	Context providers are devices which supply a context of the object / entities.
Location	Location awareness refers to devices that can passively or actively determine their location.
Longitude / Latitude	Every location has a global address consisting of numbers called coordinates. The two numbers denote a location's latitude and longitude numbers respectively.
Actuator	An actuator is the machine (a type of motor) by which an agent acts upon an environment.
Sensor	A sensor is a device capable of detecting change in temperature, pressure, speed, etc.
Remaining Battery	This refers to the current amount of available battery.
Role	A role is a pattern of behavior that provides a meaning of identifying and placing an object in the environment.
General Purpose OS	General purpose operating systems are non-real-time operating systems that are often quite non-deterministic.
Mode	Mode is a way of expressing important information about an object.
Replay	This is an action that is repeated several times.
Invoke	This activates an action.

For information extraction, in order to develop the proposed ontology, we reviewed a number of papers and manuals. In the selection stage of the concepts, a term-weighting technique [36] was used. According to this technique, during the term's extraction process, each entity was given one point and on its repetition the associated score was increased by one. To select the concepts among a number of identified concepts during the entity extraction process, the following formula (1) was used.

$$\text{AvgConceptScore} = \frac{\sum \text{ConceptScore}}{\sum \text{Concepts}} \quad (1)$$

This technique ranks the concepts according to their relative importance (weighted based on frequency of occurrence). Table 1 provides the list of these concepts.

4.2 Taxonomy formation

Taxonomy is a classification scheme used to categorize the identified domain concepts. Taxonomy arranges agreed domain concepts around a particular theme. Typically, taxonomies use a hierarchical structure (like a tree) for classifying the domain concepts. This is an important method of classification of concepts for managing complexity with an increasing number of concepts. Figure 3 presents the taxonomy for the DERTS context ontology.

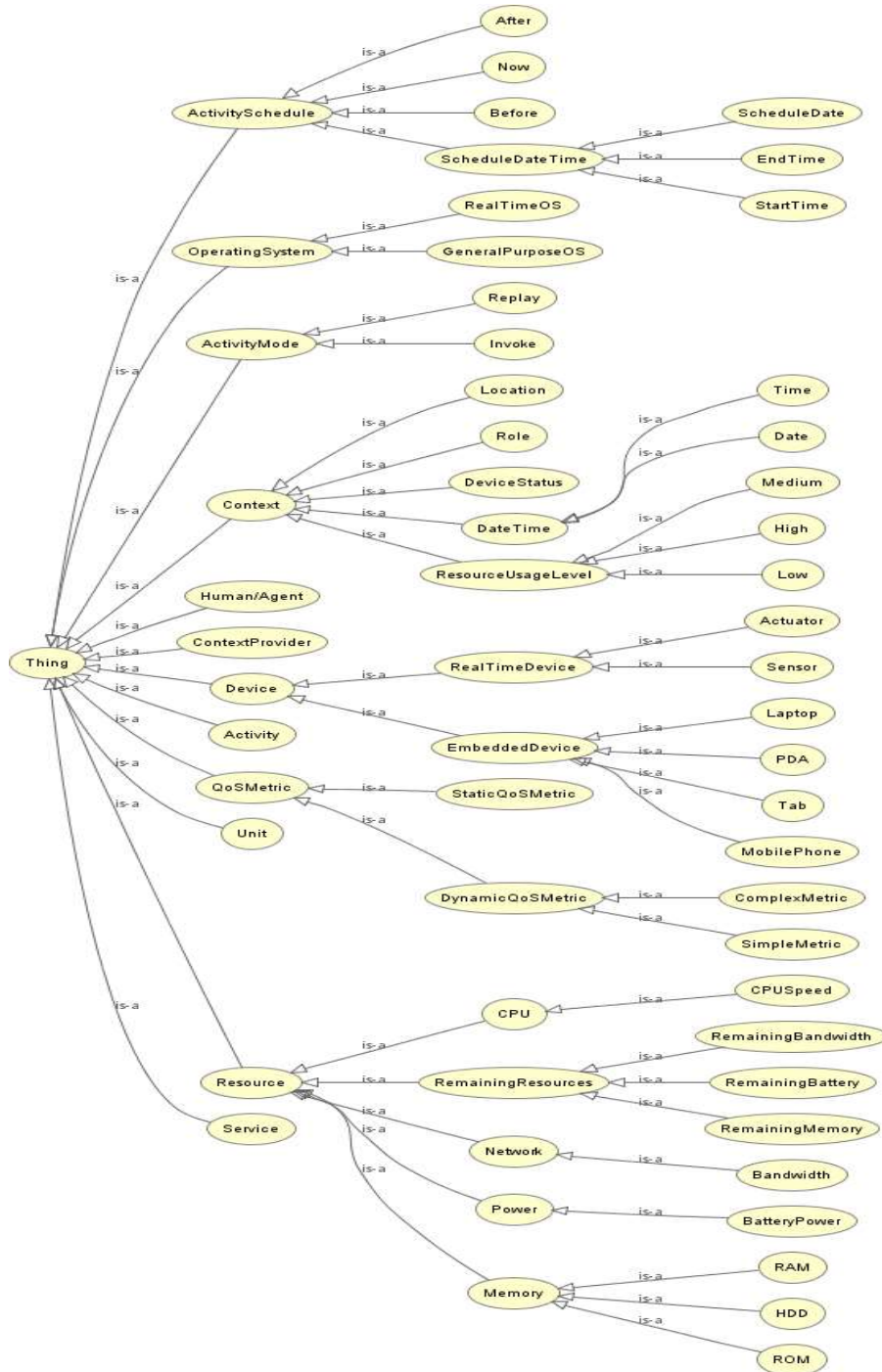


Fig. 3: Context-aware ontology for DERTS

4.3 Relationships in attributes table

Relationship binds two or more ontological concepts into a meaningful sentence. Relationship terms can be categorized into different groups like universal relationships (is-a and part-of) as well as domain specific relationships (“consume” and “is consumed by”). Some of the extracted direct and inverse relationships are shown in Table 3 .

Table 3: Binary relations table

Relation name	Source concept	Source cardinality	Target concept	Target cardinality	Inverse relation
Execute	Device	1	Activity	1	Is executed by
Consume	Device	1	Resource	N	Is consumed by
Has status	Device	1	Status	1	Is a status of
Recognize	Device	1	Context	N	Is recognized by
Offers	Device	1	Service	N	Is offered by
Has usage level	Device	1	Resource usage level	N	Is a usage level of
Has schedule	Activity	1	Activity Schedule	1	Is a schedule of
Has role	Human/Agent	1	Role	N	Is a role of
Has start time	Activity	1	Start time	1	Is a start of
Has end time	Activity	1	End time	1	Is a end of

Attributes are the properties, features, characteristics, or parameters that objects can have and share. Attributes are notably different from relations as the range of the attributes is a datatype-like number and string etc. while the range of relations is a concept. Table 4 consists of the attributes of the DERTS context-aware ontology. In Figure 4, a UML model of the most relevant classes of ConOntDERTS ontology is presented.

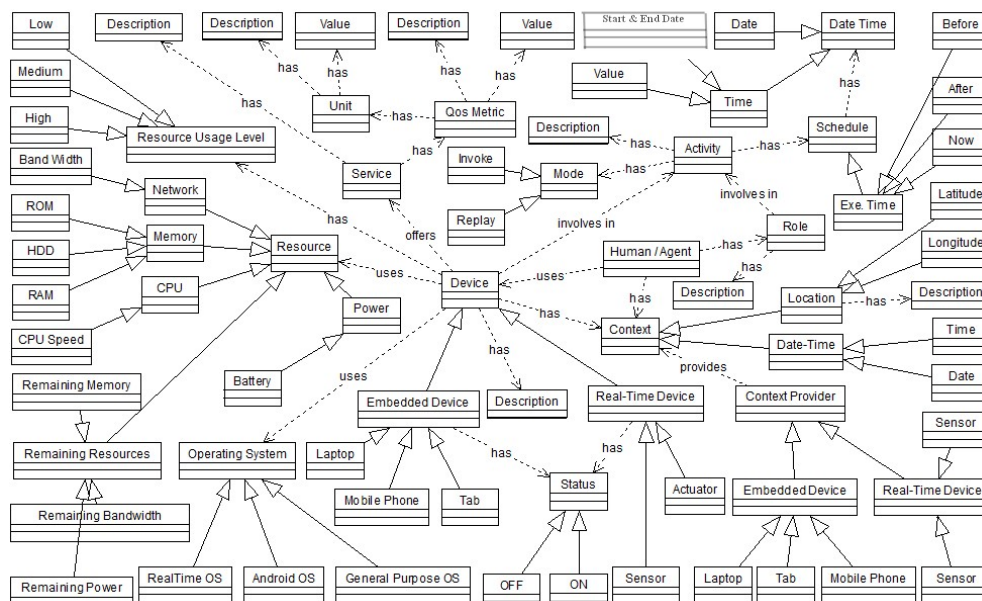


Fig. 4: UML diagram of context ontology

4.4 Axioms

Axioms are used to facilitate mutual understanding among machines during machine to machine communication by removing terminological and conceptual ambiguities. For ontological reasoning, axioms are the best way to achieve the insertion of logical expressions to ontology. Basically, an ontology consists of a set of axioms and each axiom makes a statement that is always true. Moreover, these axioms say something about classes, properties, and individuals respectively. For example, consider an axiom given below, where a Sensor is a RealTimeDevice which would satisfy the axiom, whereas when we have a Sensor that is not a RealTimeDevice, it would not satisfy the specified axiom. Axioms for the context ontology are presented in the following Table 4.

$$\text{Sensor} \sqsubseteq \text{RealTimeDevice} \sqcap \exists \text{hasOperatingSystem. RealTime-OperatingSystem}$$

Table 4: DERTS Logical axioms

Concept name	Axiom description	Logical expression
Device	A machine controlled by a microprocessor.	Device \sqsubseteq Machine \sqcap \exists hasProcessor.MicroProcessor \sqcap \exists belongsTo.EmbeddedDevice \sqcup \exists belongsTo.RealTimeDevice
Embedded Device	A device that does not use real-time operating system.	EmbeddedDevice \sqsubseteq Device \sqcap \exists hasProcessor.MicroProcessor \sqcap \exists belongsTo.EmbeddedDevice
Real-Time Device	A device that uses real-time operating system.	RealTimeDevice \sqsubseteq Device \sqcap \exists hasProcessor.MicroProcessor \sqcap \exists belongsTo.RealTimeDevice
Laptop	An embedded device which uses the general purpose operating system.	Laptop \sqsubseteq EmbeddedDevice \sqcap \exists hasOperatingSystem.General-PurposeSystem
Activity	An activity is a process that is executed in order to achieve a goal.	Activity \sqsubseteq Task \sqcap \exists performed By.Device \sqcap \exists performedBy.Agent
Activity Schedule	An activity schedule is the execution of a sequence of activities.	ActivitySchedule \sqsubseteq Activity \sqcap \exists hasStartTime.Time \sqcap \exists hasEndTime.Time \sqcap \exists hasStartDate.Date
CPU	A central processing unit (CPU) is the hardware within the computer systems where all calculations take place.	CPU \sqsubseteq ProcessingUnit \sqcap \exists uses.Resoruces \sqcap \exists belongs-To.Device \sqcap \exists usedFor.InformationProcessing
Mobile Phone	This is a device that can provide voice communications and Internet services while moving around a wide geographic area.	MobilePhone \sqsubseteq EmbeddedDevice \sqcap \exists hasOperatingSystem.Mobile-OperatingSystem
Service	Software system designed to support interaction between two machines over the network.	Service \sqsubseteq Functionality \sqcap \exists isOfferedBy.Device
Resources	A resource is a hardware or software component with limited availability within a computer system.	Resources \sqsubseteq ComponentWithLimited-Availability \sqcap \exists consumedBy.Device \sqcup \exists consumedBy.Agent
Bandwidth	This refers to remaining data communication resources expressed usually in bits per second.	Bandwidth \sqsubseteq CommunicationMode \sqcap \exists belongsTo.Resoruces \sqcap \exists usedBy.Device
Memory	This refers to the current amount of available memory.	Memory \sqsubseteq TemporaryData-StorageUnit \sqcap \exists belongsTo.Resoruces \sqcap \exists usedBy.Device
Battery	The total amount of battery power of a system.	Battery \sqsubseteq PowerStorageUnit \sqcap \exists belongsTo.Resoruces \sqcap \exists usedBy.Device
Operating System	Software to manage computer resources.	OperatingSystem \sqsubseteq DeviceControlSoftware \sqcap \exists usedBy.Device \sqcap \exists hasCategory.GeneralOS \sqcup \exists hasCategory.MobileOS \sqcup \exists hasCategory.RealTimeOS
Real Time Operating System	A real-time operating system is an operating system used by real-time devices.	RealTimeOperatingSystem \sqsubseteq OperatingSystem \sqcap \exists usedBy.RealTimeDevice
Actuator	An actuator is the machine (a type of motor) by which an agent acts upon an environment.	Actuator \sqsubseteq Device \sqcap \exists controlledWith.Motor \sqcap \exists hasOperatingSystem.RealTime-OperatingSystem

Sensor	A device capable of detecting changes in temperature, pressure, speed etc.	$\text{Sensor} \sqsubseteq \text{Device} \sqcap$ $\exists \text{hasOperatingSystem. RealTime-OperatingSystem}$
General Purpose OS	General purpose operating systems are non-real-time operating systems that are often quite non-deterministic.	$\text{GeneralPurposeOperatingSystem} \sqsubseteq \text{OperatingSystem} \sqcap \exists \text{usedBy. EmbeddedDevice}$
Mode	The mode is a way of expressing important information about an object.	$\text{Mode} \sqsubseteq \text{ActivityMode} \sqcap \exists \text{hasMode. Replay}$

5.0 ONTOLOGY EVALUATION

With the increased use of ontologies in different domains along with other software applications, the need for correct ontologies is now augmented. Accordingly, in order to verify the correctness of the developed ontology, we used two approaches: the first is a criteria-based ontology evaluation (Section 5.3) and the second is a human evaluation approach (Section 5.4). However, before applying these approaches, ontology consistency checking was performed where details are provided in the following section.

5.1 Ontology consistency checking

Consistent ontologies play a vital role in the success of any application since they identify any instances of duplication. Thus, ontology consistency checking is an important task which can be performed with the help of ontology reasoners. Reasoners compute the class hierarchies and identify any instances of duplication which could reduce the effectiveness of the ontology.

For DERTS context ontology development, the Protégé [37] editor (as shown in Figure 5) is used while consistency checking and reasoning is performed by utilizing a Description Logic Reasoner with Protégé editor. As reasoning tools can help in assessing the overall consistency of the ontology, we used Pellet reasoner [38] instead of other reasoners because it was found that they provide no support for incremental classification and consistency checking.

Besides than that, Pellet reasoner can point out the axioms that may cause ontology inconsistency. Pellet reasoner in Protégé via the Pellet Reasoner Plug-in starts checking hierarchies, domains, ranges, and conflicting disjoint assertions to find inconsistencies. In our case, by invoking a Pellet reasoner, no inconsistencies were found.

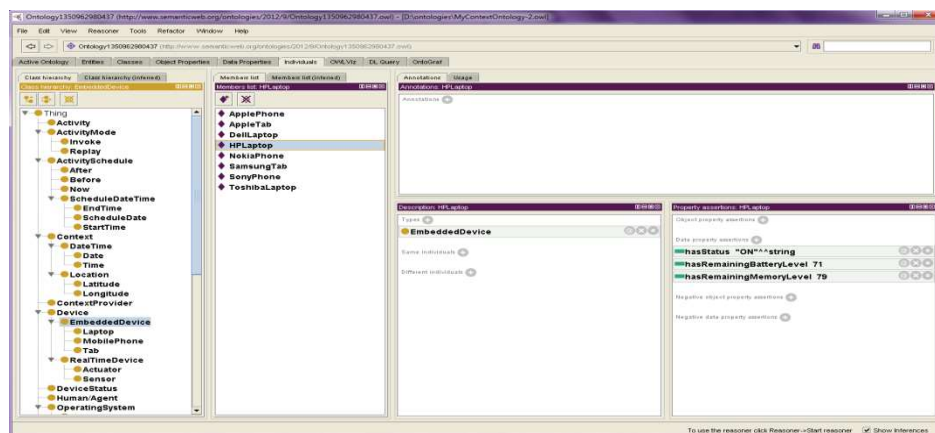


Fig. 5: Embedded device members' list in Protégé editor

5.2 Semantic Similarity Calculation

In literature, previous studies [39, 40, 41] have proposed different mechanisms for semantic similarity calculations and offer attributes of the services. For example, the approach proposed by Resnik and Lin [40, 41] calculates semantic similarity using probability rules. Skoutas et al. [39] also used similar assumptions in their proposed approach for calculating similarity since both approaches cannot consider taxonomic deviation effectively. This results in compromised semantic similarity for demand and attributes of services being offered.

$$\begin{aligned} \text{ContextMatch}(O, D) &= (O = D) \vee (O \subseteq D) \vee (D \subseteq O) & i \\ \text{Where} & \\ (O = D) &= (O.\text{IsNumeric} \wedge D.\text{IsNumeric}) \wedge (O.\text{valuetype} = D.\text{valuetype}) \\ (D \subseteq O) &= (O.\text{IsNamedConcept} \wedge D.\text{IsNamedConcept}) \wedge ((O.\text{level} \leq D.\text{level}) \\ (O \subseteq D) &= (O.\text{IsNamedConcept} \wedge D.\text{IsNamedConcept}) \wedge (O.\text{level} \geq D.\text{level}) \end{aligned}$$

The service concepts may be named or given as a number. In the case of named concepts, taxonomy will be used for relating both concepts. In relation to taxonomy, the relationship among these concepts may fall into one of three categories, which could be determined by using the rules of equation *i*. According to these rules, similarity among concepts depends on the distance between both concepts being used as a taxonomy. In this regard, the following equations can be used:

$$\begin{aligned} \text{Level}_O &= \sum \text{Level}(O) & ii \\ \text{If } \text{Level}(D) &< \text{Level}(O) \\ S_{\text{charAttrib}} &= (\text{CountAttributes}(D) / \text{CountAttributes}(O)) \end{aligned}$$

If the service concepts are numeric concepts, then the score will be calculated by using equations iii and iv, as given below. The numeric concepts may fall into one of two categories (positive or negative monotonic). In case of positive monotonicity (higher values are preferred), the following formula will be used.

$$S_{\text{numAttrib}} = 1 - ((\text{Demand.value} - \text{Offer.value}) / \text{Demand.value}) \quad iii$$

In the case of negative monotonicity (lower values are preferred), the following formula will be used.

$$S_{\text{numAttrib}} = 1 / 1 - ((\text{Demand.value} - \text{Offer.value}) / \text{Demand.value}) \quad iv$$

Offer.value is the value of the advertised service while *Demand.value* is the value of the user-requested service. Service similarity score based on its numeric and named concepts will be calculated by using the following formula:

$$\text{Total Score} = \max \sum (S_{j(\text{charAttrib})} W_D(\text{attrib}) + S_{j(\text{numAttrib})} W_D(\text{attrib})) \quad v$$

5.3 Experimental Evaluation

The key benefit of context ontology is that it can provide smooth interaction between DERTS devices since it concisely describes the properties of DERTS. An ontology explorer with a graphical user interface can allow different users and devices to discover and interact with each other semantically. The interaction between the entities is governed by their properties as defined in the context ontology. Moreover, this ontology enables humans and automated agents to specify the context-sensitive behavior of different entities.

We performed an experiment to determine the utility of the developed ontology in a health-care environment. In the case of hospitals, if management does not adequately maintain the humidity, temperature and air pressure, serious problems such as respiratory infections and allergies among patients, staff and visitors can arise. For this reason, management or automated agents always attempt to maintain temperature, humidity levels and air pressure in the hospital environment by using the services of different devices. Figure 6 [12] below is based on the same idea.

In this regard, an administrator or an automated agent can efficiently retrieve the services through semantic matching services offered by these devices. However, for effective discovery, services must be described in a language which is supported by logical reasoning. Hence, we used Web Ontology Language [22] for describing the services whereas ontologies described in Figures 3 and 6 [12] are used to support the semantic matching process.

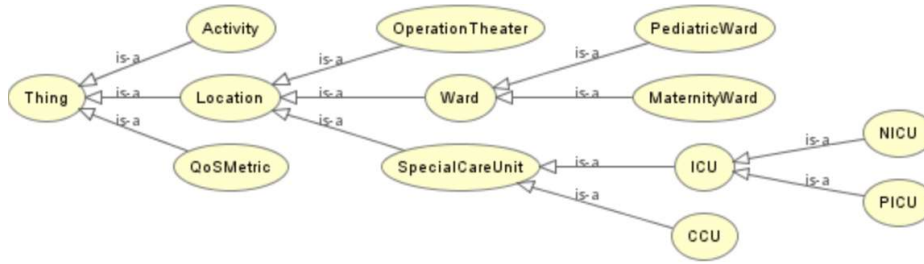


Fig. 6: Part of application ontology

In this context, it is supposed that a hospital administration has installed different devices in the hospital offering different services. Below, we presented the DL description of a few services while the more understandable descriptions are presented in Appendix A. Due to the shortage of space, the description logic notions for some of the services are provided.

$$\text{Demand} \sqsubseteq \text{srv} : \text{Service} \sqcap (\exists \text{srv} : \text{hasDeviceName.srv} : \text{HMT140}) \sqcap (\exists \text{srv} : \text{hasLocation.srv} : \text{NICU}) \sqcap (\exists \text{srv} : \text{hasRole.srv} : \text{HumidityMeasurement}) \sqcap (\exists \text{srv} : \text{hasResourceUsageLevel.srv} : \text{Low}) \sqcap (\exists \text{srv} : \text{hasBandwidth.srv} \geq 8000) \sqcap (\exists \text{srv} : \text{hasAvailability.srv} \geq 90)$$

Following is the description logic notions of the device services list in the questionnaire.

$$\text{Offer1} \sqsubseteq \text{srv} : \text{Service} \sqcap (\exists \text{srv} : \text{hasDeviceName.srv} : \text{DL2000}) \sqcap (\exists \text{srv} : \text{hasLocation.srv} : \text{NICU}) \sqcap (\exists \text{srv} : \text{hasRole.srv} : \text{HumidityMeasurement}) \sqcap (\exists \text{srv} : \text{hasResourceUsageLevel.srv} : \text{High}) \sqcap (\exists \text{srv} : \text{hasBandwidth.srv} : 7500) \sqcap (\exists \text{srv} : \text{hasAvailability.srv} : 74)$$

$$\text{Offer2} \sqsubseteq \text{srv} : \text{Service} \sqcap (\exists \text{srv} : \text{hasDeviceName.srv} : \text{T7510}) \sqcap (\exists \text{srv} : \text{hasLocation.srv} : \text{NICU}) \sqcap (\exists \text{srv} : \text{hasRole.srv} : \text{TemperatureMeasurement}) \sqcap (\exists \text{srv} : \text{hasResourceUsageLevel.srv} : \text{High}) \sqcap (\exists \text{srv} : \text{hasBandwidth.srv} : 6500) \sqcap (\exists \text{srv} : \text{hasAvailability.srv} : 84)$$

$$\text{Offer3} \sqsubseteq \text{srv} : \text{Service} \sqcap (\exists \text{srv} : \text{hasDeviceName.srv} : \text{zED-P}) \sqcap (\exists \text{srv} : \text{hasLocation.srv} : \text{CCU}) \sqcap (\exists \text{srv} : \text{hasRole.srv} : \text{AirpressureMeasurement}) \sqcap (\exists \text{srv} : \text{hasResourceUsageLevel.srv} : \text{High}) \sqcap (\exists \text{srv} : \text{hasBandwidth.srv} : 6500) \sqcap (\exists \text{srv} : \text{hasAvailability.srv} : 64)$$

$$\text{Offer4} \sqsubseteq \text{srv} : \text{Service} \sqcap (\exists \text{srv} : \text{hasDeviceName.srv} : \text{HMT140}) \sqcap (\exists \text{srv} : \text{hasLocation.srv} : \text{NICU}) \sqcap (\exists \text{srv} : \text{hasRole.srv} : \text{HumidityMeasurement}) \sqcap (\exists \text{srv} : \text{hasResourceUsageLevel.srv} : \text{Low}) \sqcap (\exists \text{srv} : \text{hasBandwidth.srv} : 7900) \sqcap (\exists \text{srv} : \text{hasAvailability.srv} : 85)$$

$$\text{Offer5} \sqsubseteq \text{srv} : \text{Service} \sqcap (\exists \text{srv} : \text{hasDeviceName.srv} : \text{DL1016}) \sqcap (\exists \text{srv} : \text{hasLocation.srv} : \text{PediatricWard}) \sqcap (\exists \text{srv} : \text{hasRole.srv} : \text{HumidityMeasurement}) \sqcap (\exists \text{srv} : \text{hasResourceUsageLevel.srv} : \text{High}) \sqcap (\exists \text{srv} : \text{hasBandwidth.srv} : 5900) \sqcap (\exists \text{srv} : \text{hasAvailability.srv} : 76)$$

5.4 Evaluation by Experts

It is a fact that research should demonstrate the achievement of results which have been derived through careful analysis of the data. The evaluations are also very important as they ensure that the proposed work will produce sound results when used in a proper way. Such results must be evaluated carefully by experts before their implementation in the environment. Therefore, a critical part of any research work is the evaluation of the results. To evaluate the correctness of the results produced by our ontology, expert judgment is used. In this regard, we conducted a survey where the details of which are given in Appendix A.

5.4.1 Survey development

For the survey development, we selected four participants; each one holds a graduate or master degree and has at least two years of experience in the development, use of or research into DERTS. Based on the discussion, the most important factors for the survey were determined and necessary questions were identified. On the consensus, these questions were framed on the survey randomly. The domain experts developed ten (10) questions which were considered most important and these are listed in the questionnaire (see Appendix A).

5.4.2 Participants' selection

We selected ten participants with at least one year of experience in DERTS environments. Most of the participants were selected from the EReTSEL lab of the Department of Software Engineering of Universiti Teknologi Malaysia. First of all, we approached participants verbally and through email. Following their confirmation, we submitted the questionnaire to the participants and finally received their completed questionnaires. We received completed questionnaires from six participants within one week and found no written comments on the questionnaires.

5.4.3 Results

The developed questionnaire was handed to the participants for the collection of their responses. We requested them to rank service as follows: number 1 awarded to properties which are very near to the demand and ranking of service; number 2 awarded to the one which came second in relation to the demand and so on. By analysing the responses of different participants, we observed that different participants ranked the same service with different numbers. This variation may have been due to the participant's assumptions (one property is more important than others) or it could be because of human error. To minimize such variations, we calculated the average score of the participants as shown in Table 5. By sorting the services in increasing order, the following Ordered Service List was obtained: {O4, O10, O7, O1, O8, O2, O3, O5, O9, O6}.

Table 5: Services ranking by human participants

Service Id	Participants						Average Score
	P1	P2	P3	P4	P5	P6	
O1	4	4	4	4	4	4	4.00
O2	6	6	6	7	6	9	6.67
O3	8	8	8	6	8	5	7.17
O4	1	1	1	1	1	1	1.00
O5	7	7	7	10	7	10	8.00
O6	10	10	10	8	10	7	9.17
O7	3	3	3	3	3	3	3.00
O8	5	5	5	5	5	8	5.50
O9	9	9	9	9	9	6	8.50
O10	2	2	2	2	2	2	2.00

In Table 6 below, a semantic similarity score of services is given. For calculating the similarity score, we used the formulas (ii, iii, iv and v) discussed in Section 5 and considered the same set of services as presented in the questionnaire (description logic notions of the device services is provided in Section 5.2).

Table 6: Computed score of services

Source code	Device	Location	Role	Resource usage level	Bandwidth Required	Availability	Total score
O1	0	1.00	1	0.0	0.94	0.82	3.76
O2	0	1.00	0	0.0	0.81	0.93	2.74
O3	0	0.50	0	0.0	0.81	0.71	2.02
O4	1	1.00	1	1.0	0.99	0.94	5.93
O5	0	0.00	1	0.0	0.74	0.84	2.58
O6	0	0.00	0	0.5	0.79	0.69	1.98
O7	1	0.75	0	1.0	0.94	0.71	4.40
O8	1	0.00	0	0.5	0.94	0.92	3.36
O9	0	0.00	0	0.0	0.79	0.56	1.35
O10	1	0.00	1	1.0	0.96	0.71	4.67
D	1	1.00	1	1.00	1.00	1.00	6.00

In order to identify that the questionnaire was understandable to all respondents, we used a bar chart (Figure 7) to present their responses as stated in Table 6. The objective of Figure 7 was to determine the values representing the non-understandability of the respondent (the larger or smaller the value with respect to the average value means more or less understandability). In Figure 7, we compared the responses of six respondents against each service average score (light blue color) to show the differences between distributions (portraying extreme values).

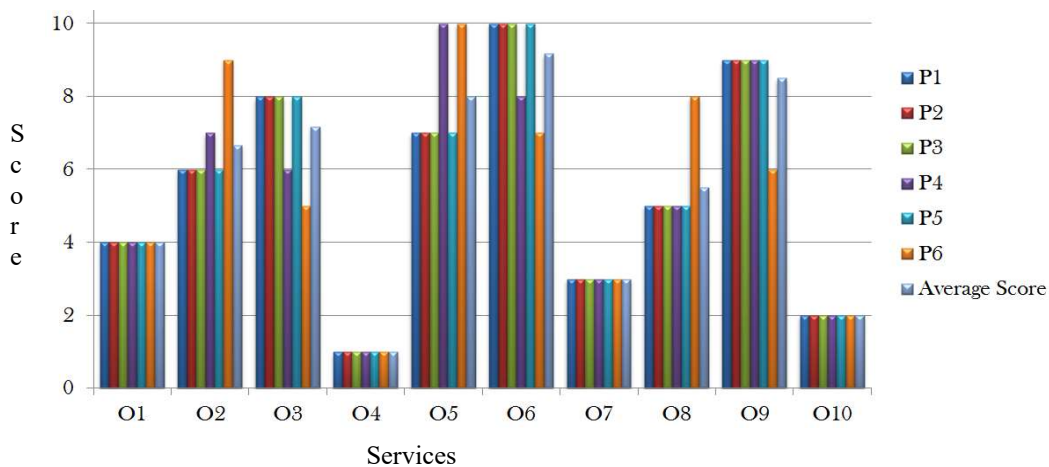


Fig. 7: Participants' responses against each service

When analyzing Figure 7, it was observed that Respondent 6 appears confused (more than others) while responding to services 2, 5, 8 and 9. However, by comparing the score averages (average score of Person 6 for all ten services and the median score of averages of all 6 persons), it can be observed that the overall understandability of Person 6 regarding the questionnaire was 84% (average response score of five persons other than Person 6 towards the abovementioned four services was 6.95 while the average response score of Person 6 against services 2, 5, 8 and 9 is 8.25). Experts consider this to be within a satisfactory range of acceptability. Hence, it was concluded that the questionnaire was properly designed by experts.

Ontological concepts may consist of a name or a number. In the case of named concepts, the closeness between the demand (D) and the offered (O) concepts will be judged through taxonomy. The numeric concept belongs to the QoS Metric shown in Figure 1.

To perform the experiment, we considered a small set of ten services. To create a context specification of service, several domain experts were consulted. A high level description of services is proved in a tabular form in Appendix A, while the description logic of some services is presented in Section 5.3. Table 6 presents the computed scores services based on the formulae *i*, *ii*, *iii*, *iv* and *v*. For example, the fifth and sixth attributes (bandwidth and availability) of demand offered in Table 3 consist of a number and a positive monotonic. These attributes are related to the QoSMetric concept as shown in Figure 1. Since both concepts have the same value type direction and belong to the same QoSMetric concept, it is considered that both attributes are semantically equal. Similarly, the demand and offer attribute (second attribute) is a named attribute (location). The similarity score of this attribute is calculated through a taxonomy relationship by using equations *i*, *ii* and *v*. Finally, the total score of the services is calculated using equation *vi*. By sorting the services of Table 6 in increasing order, the following Ordered Service List was obtained: { O4, O10, O7, O1, O8, O2, O5, O3, O6, O9 }.

Upon analyzing the results produced by the proposed approach as well as the results obtained from the participants through their questionnaire responses, it was observed that the first six results were the same. These were produced by both approaches. Meanwhile, slight variations were noted in the last four results only since the services are ordered with respect to the request of the user. This means that the user will select the first service to use and then the second if the first service is not available and so on. Hence, we can say that the proposed approach produces much more consistent results according to human perception.

5.5 Significance test

In order to determine whether or not the proposed approach can produce consistent results with human perception, a *t* test experiment was designed in SPSS Version 16. The usage of the exact type of statistical test depends upon many factors including the field, the type of data and sample size, as well as other factors. The independent samples *t* test is a small sample test and can be used if the size of the samples does not exceed 30. In this case, it was decided to use the *t* test, which compares the population means of two variables. It computes the difference between two variables to identify whether or not the average differences are significantly different from each other by using the means from randomly-drawn samples.

The vast majority of scientific research is ultimately tested by statistical methods, all providing a degree of confidence in the results. For most disciplines, the researcher looks for a significance level of 0.05, signifying that there is only a 5% probability that the observed results and trends occurred by chance. The significance level determines whether the null hypotheses are accepted or rejected, which is the crucial part of hypothesis testing.

Before executing the *t* test, we set our alternative (H1) and null (H0) hypothesis for the test and these are given below:

H1: ConOntDERTS ontology can produce consistent results with human perception.

H0: ConOntDERTS ontology cannot produce consistent results with human perception.

To perform the above test, a group of ten ($n = 10$) services are selected randomly from the population to investigate whether the proposed approach really improves the correctness of results. Before starting the experiment, it is required to be aware that the samples are normally distributed in the population, hence, the normality is checked with the help of the Shapiro-Wilk test.

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Score	.134	20	.200*	.929	20	.145

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Fig. 8: Normality test statistics of ConOntDERTS

For a dataset smaller than 2000 elements, the Shapiro-Wilk test is used, otherwise, the Kolmogorov-Smirnov test is applied. In this case, since there are only 10 elements, the Shapiro-Wilk test was used. Since the *p*-value of the Shapiro-Wilk test is 0.145 (Figure 8), which is higher than .05, it was concluded that the data came from a normal distribution.

Group Statistics									
Spec		N	Mean	Std. Deviation	Std. Error Mean				
Score	SDFD	10	3.2217	1.32986	.42054				
	Human	10	5.5000	2.84605	.90000				

Independent Samples Test										
		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Score	Equal variances assumed	7.969	.011	-2.293	18	.034	-2.27830	.99340	-4.36537	-.19123
	Equal variances not assumed			-2.293	12.751	.039	-2.27830	.99340	-4.42868	-.12792

Fig. 9: Test statistics of t test

As shown in Figure 9, the mean for the ConOntDERTS ontology is 3.221 and the mean for a human is 5.500. The standard deviation (measuring the degree of uncertainty and advising how the data is distributed around the mean value) for the ConOntDERTS ontology is 0.42054 while for a human it is 0.90000. The Sig. (2-tailed) values before and after the consideration of variances are 0.034 and .039 respectively which are much less than 0.05. Hence, the null hypothesis is rejected and it is therefore concluded that ConOntDERTS ontology produces consistent results with human perception and can be used for the generation of correct results.

6.0 Criteria-Based Ontology Evaluation

In-order to develop an evaluation criteria, we used a brainstorming technique. For this purpose, we invited our domain experts to participate in a brainstorming session. Brainstorming, basically, provides a platform to a group of people in which to meet so as to generate new ideas around a specific area of interest. All the ideas were noted down and all participants were encouraged to contribute fully. When the brainstorming session was over, the ideas raised were evaluated. During the ideas evaluation session, we used a voting technique (show of hands). We asked our experts to raise their hands (each raised hand means one point) for each idea and then we sorted the ideas based on the votes (total points). Finally, we selected the top six ideas (requirements) to develop and evaluate our ontology. These criteria requirements (ideas) are discussed in detail in Section 3. In this section, we used the same criteria requirements for the evaluation of the existing and our developed ontology. The detail of the evaluation is provided below.

Table 7: Comparative evaluation based on DERTS criteria

Properties Ontologies	Activity	Resources	Schedule	Location	Timeliness	Role
SOUPA [13]	√		√	√	√	√
COMANTO [14]	√			√	√	√
CoOL [6]						√
CONON [17]	√		√	√		√
SOCOM [19]	√			√	√	√
CoDaMoS [20]						√
DeviceOntology [42]		√		√		
OUR-K [43]	√			√		
OASys [44]	√	√		√		
ConOntDERTS	√	√	√	√	√	√

A detail of comparison is given in Table 7 while Figure 10 is based on the data given in Table 7. By analysing the data of Table 7 carefully, it was observed that SOUPA [13] is the highest-scoring ontology, as it met five requirements out of six hence making its total participation 83% (5 out of 6) of the score.

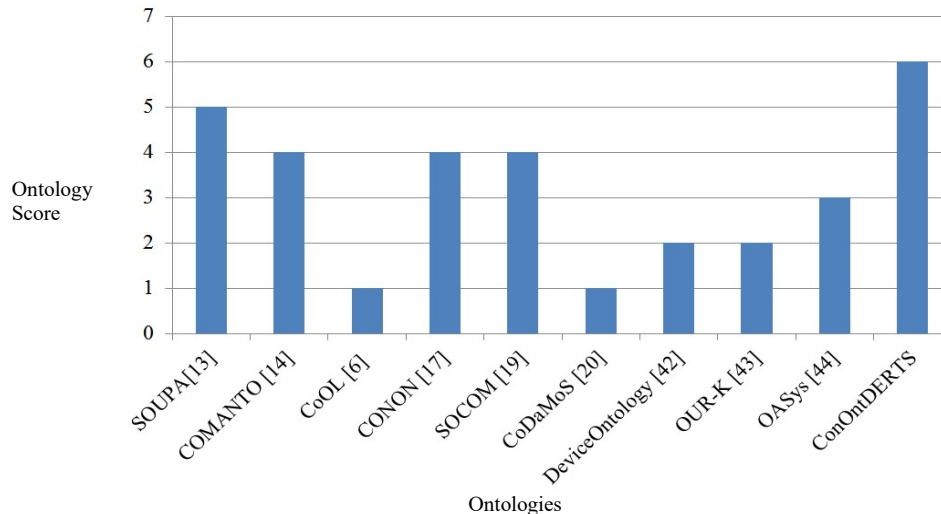


Fig. 10: Comparative evaluation of context-aware ontologies

Meanwhile, CoOL [6] and CoDaMoS [20] were noted as the lowest-scoring ontologies, since they met only one requirement out of six, hence their score was 16%. Similarly, it is also observed that only the location requirement had been well-studied by researchers (as shown in Table 7) whereas the resources and schedule are the most neglected requirements.

Based on the above analysis, it is recommended that an ontological work which considers both the most neglected and the most important requirements of DERTS may be beneficial for the industry. Hence, this conclusion, along with other motivational factors, motivated us to work on the context-aware ontology for DERTS. Accordingly, we developed and presented our context-aware ontology for DERTS as ConOntDERTS in this paper. The new ConOntDERTS with the help of experimental and comparative evaluation has been proven as being the best ontology among the remaining ontologies of its category. Moreover, by analyzing computed results, it was concluded that one of the major reasons behind the success of ConOntDERTS is the involvement of domain experts.

7.0 CONCLUSION

The trend to use smaller, more intelligent and cost-effective embedded systems continues to provide much greater functionality and complexity. These systems are increasingly being connected with the help of wireless networks including the internet. In order to operate and maintain such complex systems, SOA is the best methodology which can be used for the development of loosely-coupled, flexible, distributed and cost-effective applications for the dynamic environments of DERTS. SOA enables different device applications to exchange data and participate in different processes regardless of the complexity of the applications and facilitate different operating platform components to work together through exposing their functionalities as services. Semantic web services, with the help of ontologies (representing domain knowledge in an application-independent way), competently handle the issue of device heterogeneity. This takes the form of different languages, operating platforms and data management standards which restrict the ability of the devices to interact with each other, and improve interoperability. In this paper, a context-aware ontological model (ConOntDERTS) for organizing and sharing of domain knowledge in the heterogeneous domain of distributed embedded and real-time devices is presented. This ontology was also used for effective semantic matching of resource requests and advertisements in the dynamic environment of DERTS. For checking the effectiveness of ConOntDERTS, we developed a questionnaire which was provided to domain experts. The comparative evaluation of results showed the consistency and feasibility of the ConOntDERTS; moreover, the statistical test (independent samples *t* test) results show that ConOntDERTS ontology can produce consistent results with human perception.

REFERENCES

- [1] A. Sehmi. "On Distributed Embedded Systems." *Advances in Computer Science: an International Journal* 2.1 (2013): 30-39.
- [2] D. Guinard et al., "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services." *IEEE transactions on services computing*, vol. 3, no. 3, july-september 2010.
- [3] E. Fleisch et al., "Das Internet der Dinge. Ubiquitous Computing und RFID in der Praxis: Visionen, Technologien, Anwendungen, Handlungsanleitungen." *Springer-Verlag*, 2005.
- [4] A. DEY. "Providing Architectural Support for Building Context-Aware Applications." *PhD thesis, Georgia Institute of Technology*, 2000.
- [5] C. Bettini et al., "A Survey of Context Modelling and reasoning Techniques." *Pervasive and Mobile Computing*, Volume 6, Issue 2, April 2010, Pages 161–180.
- [6] T. Strang et al., "Applications of a context-ontology language." in *Proceedings of International Conference on Software, Telecommunications and Computer Networks*, Split/Croatia, Venice/Italy, Ancona/Italy, Dubrovnik/Croatia, 2003, pp. 14-18.
- [7] A. Dey et al., "Towards a better understanding of context and context awareness." *Proceedings of the workshop on the What, Who, Where, When and how of Context Awareness*, ACM Press, New York (2001).
- [8] B. Schilit et al., "Context-Aware Computing Applications." In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, pp 85-90, *IEEE Computer Society Washington*, DC, USA, 1994.
- [9] R. Want et al., "The Active Badge Location System." *ACM Transactions on Information Systems*, pp. 91 – 102, 1992.
- [10] F. Gandon. "Ontologies in Computer Science: These New Software Components, of Our Information Systems. Ontology Theory, Management and Design: Advanced Tools and Models." *IGI Global*, 2010. 1-26. Web. 13 Jul. 2013. doi:10.4018/978-1-61520-859-3.ch001
- [11] J. Wassim et al., "Ontology Theory, Management and Design: An Overview and Future Directions. Ontology Theory, Management and Design: Advanced Tools and Models." *IGI Global*, 2010. 27-77. Web. 13 Jul. 2013. doi:10.4018/978-1-61520-859-3.ch002
- [12] F. Zeshan et al., "Medical Ontology in the Dynamic Healthcare Environment." *Procedia Computer Science*, Volume 10, 2012, Pages 340-348, ISSN 1877-0509.
- [13] H. Chen et al., "SOUPA: Standard ontology for ubiquitous and pervasive applications." In *Proceedings of the First International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2004.
- [14] M. Strimpakou et al., "A contextontology for pervasive service provision." in *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*, Vienna, Austria, 2006, pp. 775-779.
- [15] T. Strang et al., "A Context-Modelling survey." *First International Workshop on Advanced Context Modelling, Reasoning and Management*. UbiComp 2004.
- [16] A. Dey. "Understanding and Using Context." *Personal and Ubiquitous Computing*, Vol. 5. No. 1, 2001, pp. 4-7.
- [17] X. Wang et al., "Ontology based context modeling and reasoning using OWL." in *Proceedings of IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Orlando, Florida, USA, 2004, pp. 18-22.

- [18] E. Baralis et al., "Cas-mine: providing personalized services in context-aware applications by means of generalized rules." *Knowledge and Information Systems*, August 2011, Volume 28, Issue 2, pp 283-310
- [19] R. Tan et al., "SOCOM: Multi-Sensor Oriented Context Modelbased on Ontologies." *Eighth International Conference on Intelligent Environments*, IEEE 2012.
- [20] CoDaMoS development team. The codamos project, 2003.
- [21] D. Preuveneers et al., "Towards an extensible context ontology for ambient intelligence." *In Proc. 2nd European Symp. Ambient Intelligence*, LNCS 3295, pages 148–159, 2004.
- [22] H. Chen et al., "An Ontology for Context-Aware Pervasive Computing Environments." *The Knowledge Engineering Review* 18.03 (2003): 197-207.
- [23] S. Pokraev et al., "Context Aware services" WAP/D3.0, November 2003.
- [24] G. Chen et al., "A Survey of Context-Aware Mobile Computing Research", *Technical Report TR 2000-381, Dept. of Computer Science, Dartmouth College*, 2000.
- [25] Z. Salvador et al., "Architectures for ubiquitous environments." *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, WiMob 2005, Volume 4, pp. 90-97, Montreal, Canada, August 2005.
- [26] R. Aversa et al., "A simulation model for localization of pervasive objects using heterogeneous wireless networks." *Simulation modeling practice and theory*, vol. 19/8, pp. 1758–1772–1772, 2011.
- [27] M. Gruninger et al., "Methodology for the design and evaluation of ontologies." *In: Proceedings of International Joint Conference of Artificial Intelligence Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, 1995, pp. 1–10.
- [28] M. Fernández et al., "Methontology: from ontological art towards ontological engineering." in: *AAAI 97 Spring Symposium on Artificial Intelligence in Knowledge Management*, Stanford University, California, USA, 1997, pp. 33–40.
- [29] O. Corcho et al., "Building Legal Ontologies with METHONTOLOGY and WebODE." *Springer-Verlag*, LNAI No. 3369, pp. 142-157, 2005 ISBN: 3-540-25063-8.
- [30] N. Noy et al., "Ontology development 101: a guide to creating your first ontology." *Knowledge Systems Laboratory* 01 (2001) 05.
- [31] S. Ahmed et al., "A methodology for creating ontologies for engineering design." *Journal of Computing and Information Science in Engineering* 7 (2) (2007) 132–140.
- [32] S. Châabane et al., "Méta-modélisation du processus de construction d'ontologies géographiques: Application au domaine routier." *In Proceedings of ICWIT '2009*, Kerkennah, Tunisia, (pp. 653-661).
- [33] L. Leclère et al., "Construction of an ontology related to the projective geometry." *In RFIA 13th congrès des Reconnaissance des Frames et Intelligence Artificielle*, France(2002).
- [34] J. Nanda et al., "A methodology for product family ontology development using formal concept analysis and web ontology language." *Journal of Computing and Information Science in Engineering* 6 (2) (2006) 103–113.
- [35] V. Chaudhri et al., Open Knowledge Base Connectivity 2.0.3. *Technical Report*. <http://www.ai.sri.com/~okbc/okbc-2-0-3.pdf>
- [36] Y. Liu et al., "Imbalanced text classification: a term weighting approach." *Expert Systems with Applications* 36 (1) (2009) 690–701.

- [37] Protege <http://protege.stanford.edu/download/ontologies.html>
- [38] Pellet OWL reasoner. URL: <http://www.mindswap.org/2003/pellet/index.shtml>.
- [39] D. Skoutas et al., "A ranking mechanism for semantic web service discovery." In: *IEEE SCW*. (2007) 41–48.
- [40] P. Resnik. "Using information content to evaluate semantic similarity in a taxonomy." In: *IJCAI*. (1995) 448–453
- [41] D. Lin. "An information-theoretic definition of similarity." In: *Proc. 15th International Conf. on Machine Learning, Morgan Kaufmann*, San Francisco, CA (1998) 296–304
- [42] A. Bandara et al., "A pragmatic approach for the semantic description and matching of pervasive resources." *International Journal of Pervasive Computing and Communications* Vol. 6 No. 1, 2010, pp. 19-46.
- [43] G. Lim et al., "Ontology-based unified robot knowledge for service robots in indoor environments." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(3), 492-509.
- [44] L. Paull et al., "Towards an Ontology for Autonomous Robots." *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, vol., no., pp.1359,1364, 7-12 Oct. 2012.
- [45] Zeshan, F., Mohamad, R., Ahmad, M. N. "Service Discovery Framework for Distributed Embedded Real-Time Systems." *Handbook of Research on Emerging Advancements and Technologies in Software Engineering*, 126.
- [46] Zeshan, F., Mohamad, R., Ahmad, M. N. "Ontology for Autonomous Mobile Robot System" In *Ebook Series of Frontiers in Artificial Intelligence and Applications, Volume 265: New Trends in Software Methodologies, Tools and Techniques*; Pages 1073 – 1085. DOI: 10.3233/978-1-61499-434-3-1073
- [47] Zeshan, F.; Mohamad, R., Ahmad, M. N. "Quality of Service Ontology Languages for Web services Discovery: An overview and Limitations" *Springer: LNCS (PART 1 - 8016)*, pp. 400-407.
- [48] Zeshan, F., Mohamad, R., Ahmad, M. N. "Comparative Evaluation of Semantic Web Service Composition Approaches" *ICSECS 2011, Part III, CCIS 181*, pp. 283–290, *Springer-Verlag Berlin Heidelberg* 2011.
- [49] Zeshan, F., Mohamad, R., Ahmad, M. N. "Services Discovery Frameworks for Dynamic Environments: An overview and Limitations" in the *Proceedings of the World Congress on Engineering 2013*.
- [50] Zeshan, F., Mohamad, R., Ahmad, M. N. "Semantic Web Service Composition Approaches: Overview and Limitations" *International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(3): 640-651, The Society of Digital Information and Wireless Communications*, 2011 (ISSN: 2220-9085)
- [51] A. Elhag et al., "A Systematic Composite Service Design Modeling Method Using Graph-Based Theory." *PLoS ONE*, pp: 1-26, Vol: 10, Issue: 4, Impact Factor: 3.53. doi:10.1371/journal.
- [52] A. Qazi, R. G. Raj, M. Tahir, M. Waheed, S. U. R. Khan, and A. Abraham, "A Preliminary Investigation of User Perception and Behavioral Intention for Different Review Types: Customers and Designers Perspective," *The Scientific World Journal*, vol. 2014, Article ID 872929, 8 pages, 2014. doi:10.1155/2014/872929.
- [53] T. Strang et al., "A Context Modeling Survey." *Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing*, Nottingham/England, September 2004.
- [54] T. Gu, et al., "Towards an OSGi-Based Infrastructure for Context-Aware Applications in Smart Homes," *IEEE Pervasive Computing*, 3(4) 2004, 66-74.
- [55] H. Chen, et al., "Intelligent Agents Meet the Semantic Web in Smart Spaces." *IEEE Internet Computing*, (November 2004):69-79.

- [56] E. Dejene et al., "An ontology-based approach to context modeling and reasoning in pervasive computing." *Pervasive Computing and Communications Workshops*, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on. IEEE, 2007.
- [57] T. Gu et al., "An ontology-based context model in intelligent environments." *In Proceedings of communication networks and distributed systems modeling and simulation conference* (Vol. 2004, pp. 270-275).
- [58] A. Qazi, R. G. Raj, G. Hardaker and C. Standing, "A systematic literature review on opinion types and sentiment analysis techniques: tasks and challenges", *Internet Research*, Vol. 27 No. 3, pp. 608-630, doi: 10.1108/IntR-04-2016-0086.
- [59] H. Guermah et al., "An Ontology Oriented Architecture for Context Aware Services Adaptation." *IJCSI International Journal of Computer Science Issues*, Vol. 11, Issue 2, 2014.
- [60] B. S. Nathalie, and C. R. Newman. "Context awareness in health care: A review." *international journal of medical informatics* 76.1 (2007): 2-12
- [61] M. G. Wamamu, and H. O. Nyongesa. "Context awareness in mobile computing: A review." *International Journal of Machine Learning and Applications* 2.1 (2013): 5-pages
- [62] K. E. Kritikos (2008). *Qos-based web service description and discovery*(Doctoral dissertation, University of Crete).
- [63] Zeshan, F., Mohamad, R., Ahmad, M. N., Hussain, S. A., Ahmad, A., Raza, I., ... & Babar, I. (2017). Ontology-based service discovery framework for dynamic environments. *IET Software*, 11(2), 64-74.

Appendix A

Questionnaire: Services Ranking in DERTS Environment

Assume you are looking for a service according to your requirements (D) from the published services (O_i) given in the Table below.

You are kindly requested to order each published service (O_i) against the demand (D) given in the last row of the Table: e.g. Rank 1 for the best choice; 2 for the second best, etc.

Please rank services based on the following attribute priorities.

- | | |
|-------------|-------------------------|
| 1: Device | 4: Availability |
| 2: Location | 5: Available Bandwidth |
| 3: Role | 6: Resource usage level |

Service Code	Device	Location	Role	Resource usage level	Bandwidth Required	Availability	Rank
O1	DL2000	NICU	Humidity Measurement	High	7500	74	
O2	T7510	NICU	Temperature Measurement	High	6500	84	
O3	zED-P	CCU	Air pressure Measurement	High	6500	64	
O4	HMT140	NICU	Humidity Measurement	Low	7900	85	
O5	DL1016	Pediatric Ward	Humidity Measurement	High	5900	76	
O6	DL1016	Operation Theater	Air pressure Measurement	Medium	6300	62	
O7	HMT140	PICU	Temperature Measurement	Low	7500	64	
O8	HMT140	Maternity Ward	Air pressure Measurement	Medium	7500	83	
O9	zED-P	Ward	Temperature Measurement	High	6300	50	
O10	HMT140	Operation Theater	Humidity Measurement	Low	7700	64	
D	HMT140	NICU	Humidity Measurement	Low	8000	90	

Please write down any of your comments on data quality, selection and distribution etc.
