Journal of Theoretical and Applied Information Technology
28th February 2019. Vol.97. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ONTOLGY-BASED MASHUP MODEL FOR CONTEXT-AWARE SERVICES IN INTERNET OF THINGS APPLICATION ENVIRONMENTS

YONGGOO CHOI

Department of Computer Information, Dong Seoul University, Seongnam 13117, Republic of Korea

E-mail: ygchoi@du.ac.kr

ABSTRACT

In the IoT (Internet of Things) environments, an integrated context model is mandatory for governing both context information and context awareness in a unified and coherent way. The core goal of the model is to enhance interoperability of heterogeneous contexts and reduce complexity of context-aware computing under the highly dynamic situations. For realizing this goal, this paper proposes an *OntoMash* model for adaptable context awareness and semantic reasoning services in the IoT information systems. The *OntoMash* model is an aggregation of multi-context information based on ontology technologies for managing overall situations in the IoT environments. The model is able to accommodate multiple context types from heterogeneous and distributed context sources and illuminate complex and diverse semantic relationships between context classes. In the *OntoMash* modeling scheme, frameworks and ecosystems are designed to represent its general and global models. The applications of *OntoMash* model are discussed in *OntoMash* adaptation for context-aware reasoning. Furthermore, in order to investigate the strengths and drawbacks of the *OntoMash* model, their performance issues are discussed according to the evaluation indices. Through simulation models, the evaluation results preliminarily demonstrate the adaptable benefits of the model in the IoT environments.

Keywords: *OntoMash, Ontology, Semantic Mashup, Context Reasoning, Context Awareness, Internet of Things (IoT)*

1. INTRODUCTION

The ultimate vision of science and information technologies is to create a better world for human beings. As a key task for realizing this vision, all of the things around human being make themselves govern overall matters concerning human activities with minimum human intervention. The Internet of Things (IoT) [1] has become a key concept in industry and academic fields to fulfill such a vision because it is completely integrated into the everyday life of people. So, IoT technologies have been widely applied to not only common workplaces, but also mission-critical applications such as healthcare service, air-traffic control, autonomous vehicle, smart factory and farm, security monitoring and control, and more. Despite the fact that the IoT is now pervasive and proliferated in every aspects of human life, there are increasing concerns that imperfection of its context information may lead to tremendous disasters beyond acceptable deficiencies.

Since IoT is a convergence of the things, internet context semantics and applications, a main goal of the IoT technology is to maintain the interoperability and adaptability between multiple and autonomous devices with sharing context information, processes and resources. The context information obtained from such the devices is very heterogeneous and distributed due to diverse information sources in the IoT environments. For instance, let's suppose that a patient is measured in his blood pressure, and is then given alert sounds when the patient's blood pressure exceeds a certain level. The following cases show various ways to get context information of the patient's blood pressure:

- It can be collected in real time from wearable blood pressure sensors attached on the patient,
- It can be retrieved from the patient's medical records into a healthcare database or a cloude system,
- Or it can be obtained by combining and computing results with both the patient's sensor data and their existing data in an information system.

Since multi-context information from such diverse sources makes a context awareness more

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

complicated, an integrated context model is mandatory for managing the information in a unified and coherent way. In fact, IoT information systems should enable intelligent things not only to provide an adaptive context-ware service for their interoperability, but also to give a decision-making service for insight interactions by reasoning and analyzing in their overall situations. These services should be supported by a well-designed context model because it is capable of reducing a complexity of the context-aware computing [2]. In the IoT application environments, a context model has to be able to accommodate all of the context information existing in both the physical devices and the virtual world. Due to the inherent complexity of context modeling schemes for the IoT information systems, implementation should be supported by adequate data engineering techniques.

In the academic fields, there are the most popular context modelling techniques that have been known as key-value, markup scheme, graphical, object-oriented, logic-based, and ontology-based models [2] [3]. In contrast to other modeling techniques, ontologies [4] [5] have been known as the most appropriate format for sharing and integrating heterogeneous context information from different sources. The main reason is because ontologies enable knowledge sharing and interoperability with semantically well-defined concepts and their relationships in highly open and dynamic environments [4].

Even though ontology-based modeling techniques have clear advantages regarding support for integration of multi-context information, it may not always be reasonable when considering the trade-off situations between expressiveness and complexity. Considering most previous studies, they have generally focused on service-oriented architectures under the specific application domains for M2M (Machine to Machine) communications. If their models are applied to the IoT application systems with open and dynamic characteristics, two issues have been observed: Most of the models have overlooked not only (a) the mashup which is a combination of contexts from other information sources to create a new type of context-aware services, but also (b) the representation of semantic relationships and dependencies among multiple context classes for context awareness computing. Thus, these issues must be dealt with a full-fledged context modeling scheme which encompasses all of the context information types for effective context-aware computing. For dealing with such issues this paper proposes an *OntoMash* model for adaptable context awareness and semantic reasoning services in the IoT information systems. The *OntoMash* model is an aggregation of multi-context information based on ontology technologies for managing overall situations in the IoT application environments.

The model is able to accommodate multiple context types from heterogeneous and distributed context sources and illuminate complex and diverse semantic relationships between context classes. The modeling philosophy of the *OntoMash* is an adaptability to adjust context-aware services according to diverse context sources in an open and dynamic IoT application environment. For this modeling scheme, *OntoMash* system framework and ecosystem are presented, and general and global models are illuminated in *OntoMash* modeling schemes. The applications of *OntoMash* model are discussed in *OntoMash* adaptation for context-aware reasoning. Furthermore, in order to investigate the strengths and drawbacks of the *OntoMash* model, their performance issues are discussed according to the evaluation indices. Through simulation models, the evaluation results preliminarily demonstrate the adaptable benefits of the model in the IoT environment.

The paper structure is as follows. Chapter 2 discusses the current approaches of the IoT context model and IoT mashup model. Chapter 3 illuminates *OntoMash* modeling scheme not only by showing *OntoMash* system framework and ecosystem, but by presenting *OntoMash* general and global modeling methods. *OntoMash* adaptation is addressed in Chapter 4 by showing context awareness and reasoning schemes. Performance evaluation is presented in Chapter 5 according to evaluation criteria. Finally, Chapter 6 underlines some conclusions and future works.

2. RELATED WORKS

2.1 IoT Context Models

For the IoT information system, a context information model is a well-defined framework that can realistically accommodate a concrete subset of multiple context information from diverse context sources such as sensors, contexts, applications and users. According to this definition, the notion of the context always refers to any information that can semantically abstract a situation of active entities in a specific domain [6]. Since a context-aware service employs a context management system in a unified and coherent way, context model explicitly should specify in the IoT information systems. As context

Journal of Theoretical and Applied Information Technology
28th February 2019. Vol.97. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

information sources, things of the IoT can be characterized into four categories: *Subjects* including *Users*, *Objects*, *Webs*, and *Applications* (Figure 1).

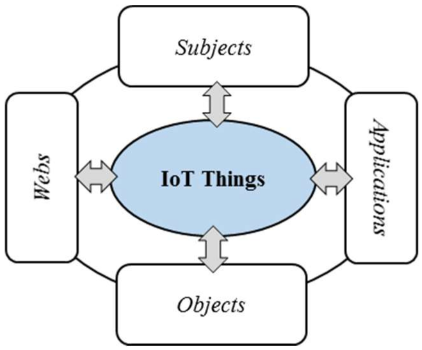

Figure 1: Framework of Tings on the IoT.

To achieve their inherent goals, such things in the IoT all are connected to the Internet and autonomously communicate with each other on behalf of humans. However, they are very heterogeneous due to structural diversities in both their semantic levels and update cycles. Despite this heterogeneity of the IoT information system, a well-defined context model can make the context-aware services easier. Moreover, a formal representation of context information within a model is easy to check consistently for governing context information as well as ability to provide sound reasoning for discovering knowledge that is more meaningful.

The context modeling techniques have been several surveys for context-aware computing and reasoning with their own strengths and weaknesses based on application domains. Wang, Xiao Hang, et al. [7] presented CONON (CONtext ONtology) which is an extensible context ontology for modeling context in pervasive computing environments. It only deals with an upper context ontology for capturing general concepts about basic context. Tao et al. [8] classified the existing context models into three categories: application-oriented approach, model-oriented approach, and ontology-oriented approach. Paea et al. [9] presented ontology database as semantic modeling methodologies using a relational database schema. Claudio et al. [4] surveyed context modeling and reasoning techniques to meet a variety of context information types and presented object-role based model and spatial models and highlighted ontology based models. Charith et al. [3] and Hatim et al. [4] addressed the six most popular context modeling techniques: key-value, markup schemes, graphical, objet based, logic based, and ontology based modeling. They also compared these models as advantages and disadvantages with context awareness.

For context-aware sustainable application in the IoT information systems, context-modeling approaches can be classified into key-value models and markup models. The key-value models formulate context information with simple key-value pairs that consist of attributes and their values. JSON (JavaScript Object Notation) [10] is well-known as a standard format of the key-value expression in the practical fields. Although this format can easily express context information, it lacks capabilities for semantic knowledge representation in order to exchange context information efficiently and provide reasoning soundly. To improve drawbacks of the key-value modeling technique, markup models with semantic markup tags have been widely applied for many application domains in order to clarify context information representation. XML (extensible markup language) [11] with the semantic markup tags has been recognized as a flexible context modeling format for effectively representing and exchanging knowledge. These XML modeling technologies have led to a wide interest in the *Semantic Web* [12] [13], which is an extension of the current web enabling web applications to understand the meaning of various information resources on the web and to search for them intelligently.

An important basis for many developments in the semantic web is RDF/RDFS (Resource Description Framework/RDF Schema) and OWL (Ontology Web Language), which provides some modeling schemes for representing metadata on the web resources [12]. While RDFS provides taxonomic relationships between web resources, ontology is more expressive for representing semantic classes and relationships with richer vocabulary [13]. Therefore, ontologies have clear advantages in the IoT context model due to enabling knowledge sharing and interoperability with semantically well-defined concepts and their semantic relationships in the open and dynamic environments [14]. Despite context modeling power of the ontologies, ontology modeling methodologies are still vague and general to accommodate a variety of context information types in the IoT application environments.

2.2 IoT Context Mashup Models

In general, mashups embody web services technology, fusing data from two or more web applications to create an integrated experience informed by the original data sources. Mashup creators pull data dynamically from one source and integrate it with another [15]. A context mashup is a process that brings together a variety of context information from multiple sources and combines them in a way that enhances reasoning processes for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

context awareness computing. The goal of the context mashup is to create more meaningful context information from ordinary context information for a more high-level context awareness. In the IoT information systems, context information sources can be divided into three categories: *Physical* (including *Sensors*), *Virtual* (including various *Web Contents* and *Legacy Data*), and *Logical* (including *Applications*) contexts (Figure 2).

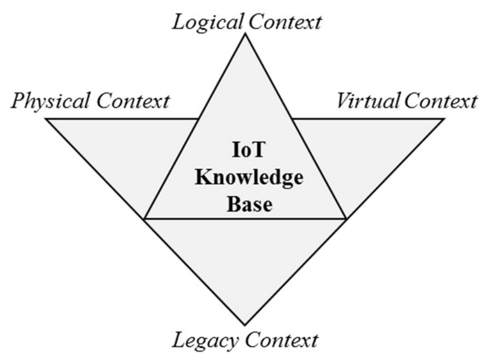

Figure 2: Context Information Sources for the IoT Knowledge Base

Context information generated from physical sensors is less meaningful and imperfect due to their dynamic and heterogeneous nature. To make context information more meaningful, they are inevitably combined with other context sources. Virtual context information, on the other hand, is typically collected from distributed web sources such as web contents, calendar, email, Twitter, online opinions and so on. They are mainly stored in data warehouses in the cloud computing to be shared and reused in several information systems. While the physical devices typically provide context information with real-time velocity, the virtual systems more rarely access context information from the clouds. Finally, logical context information is acquired by mixing and/or computing with the physical sensor data, the virtual contents, legacy data, and application logics. The reason for such manipulation is to create more meaningful context information or to excavate new context knowledge for high-level reasoning and business intelligence. For instance, meteorological information is a sort of logical context, which is obtained from a result of combining and processing thousands of physical sensor data and virtual contents related to various weather situations [2]. Therefore, in order to develop an efficient context model, it is important to understand their semantic levels, update rats, and context sources.

There have been several surveys [16] [17] [18] [19] [20] conducted in relation to context mashup models for context interoperability in the heterogonous and distributed information systems. More early, Wache et al. [16] surveyed ontology integration models that had three different approaches: single ontology, multiple ontologies and hybrid ontology. Single ontology approaches use one global ontology providing a shared vocabulary for the specifications of semantics; whereas multiple ontologies have their own ones for the semantics of an information sources but they maintain source ontologies for a combination of several other ontologies. Lenzerini [17] also comprehensively surveyed an ontology-based data management (OBDM) [9] which was a direction for the integration of data stored in different sources and the governance in a unified and coherent way. The key concept of OBDM is to a three-level schema, constituted by the ontologies, the sources and mappings between them. Kotis et al. [18] presented the engineering of an IoT-ontology to support several tasks of the automated deployment process of applications in heterogeneous IoT environments. The work of them particularly presented the ontology technology for the alignment of the extremely large amount heterogeneous IoT entities. Meng and Jinlong [19] proposed a mashup model based on both client and server side for the distributed data integration. It can reduce the processing overheads on mashup servers as distributing their roles between the client and the server. The client is responsible for representation, while the server takes charge of data and logic. Abiteboul et al. [20] introduced a formal model for capturing the essence of data management in mashups based on relation schema as in relational database systems. For mashlets that is components of a mashup, they consider three essential aspects: the various ways to use mashlets, the dynamic nature of the mashlet interaction, and the dynamic nature of the data.

Even though some models for multiple context integration address multiple aspects, an ideal modeling methodology that can mash up multiple context types from the diverse sources required in the open and dynamic IoT information system is yet to be design. If existing context mashup models apply for an open and dynamic IoT environment, most of the models have overlooked the semantic mashups [21] of open-world semantics of context information generated by multiple context sources, and the reconciliation of semantic relationships between multiple context types under dynamic situation changes. Therefore, it is very important to find out more attainable techniques for developing a context mashup model adaptive in open and dynamic IoT environments because it depends on expression powers, validation constraints, reasoning techniques, and performance costs.

3. *ONTOMASH* MODELING SCHEME

3.1 *OntoMash* System Architecture

For the sake of context awareness services based on ontologies in the IoT environments, it is necessary to integrate each of ontologies from multi-context sources for governing comprehensive tasks of the IoT information systems. In this paper such an integrated ontology is named *OntoMash* which is a composition of multi-context ontologies based on ontology modeling scheme. From the ontology modeling point of view, *OntoMash* system framework is composed of two layers: *ontology layer* and *service layer* (Figure 3).

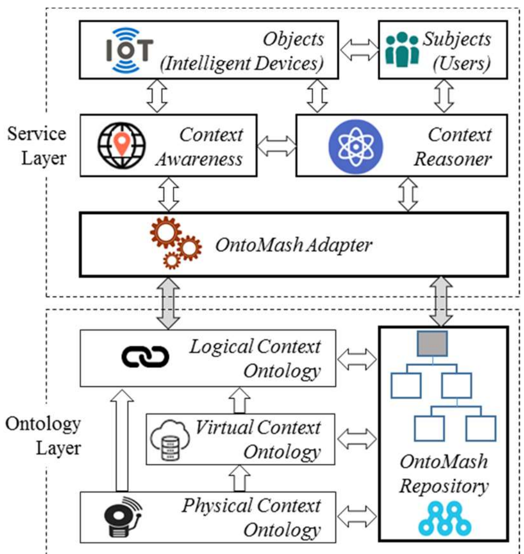

Figure 3: OntoMash System Framework

To provide service adaptations in unified and coherent ways, *OntoMash* repository is composed of physical, virtual and logical context ontologies.

- A physical context ontology is an explicit representation of their concepts and relationships to formalize entities in real world with the context semantics of intelligent objects which are all kinds of sensors and actuators, positioning devices, video and voice recognition facilities, and so on.
- A virtual context ontology is an explicit representation of their concepts and relationships to formalize entities in virtual world with the context semantics of distributed web data which are shared and reused in several information systems.
- A logical context ontology is an explicit representation of their concepts and relationships to formalize entities in logical world with the context semantics as a result of data mediation, aggregation, transformation, editing, computation, and so on.

These ontologies are mashed up by appropriate APIs (Application Program Interfaces) for the implementation of the *OntoMash* repository. The most popular protocols used by these APIs are SOAP (Simple Object Access Protocol), REST (Representational State Transfer), RSS (Rich Site Summary), and so on [22]. However, since this paper focuses on *OntoMash* modeling scheme, we suppose that the partial ontology blocks are inserted into *OntoMash* repository by such API protocols over a span with a certain period of time. *OntoMash* repository are necessary to drive the context-awareness services in the IoT application environments. In the *OntoMash* system, the context reasoning scheme provides ways to make high-level contexts which are more meaningful from the low-level context and used by the context reasoner. The context awareness is a task that recognizes a specific situation by combining the reasoning results with other contexts. Finally, *OntoMash* adapter plays a crucial role in making adaptive decision for the context awareness and reasoning process.

3.2 *OntoMash* Ecosystem

OntoMash is an aggregation of heterogeneous context components generated from multi-context sources for comprehensive context awareness in a unified and coherent way. For IoT information system, the *OntoMash* model typically combines physical sensor contexts with virtual context to create more meaningful context information. Therefore, *OntoMash* ecosystem that is a network of context mashups and APIs (Application Program Interfaces) is explicitly specified as a cube with three different perspectives: context types, reasoning levels and application environments (Figure 4).

Depending on the application environments of the IoT, context models can be classified as static or dynamic schema. While a static model schema has a predefined set of context information that is collected and stored, a dynamic model schema changes a data structure so as to be adapted to new platform configurations. In fact, the context types of the IoT are consisted of physical, virtual and logical contexts derived from multi-context sources.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Reasoning levels can be classified as low-level or high-level contexts [7] according to context usefulness.

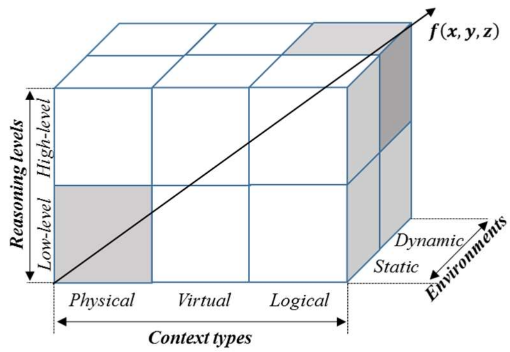

Figure 4: OntoMash Ecosystem

The *OntoMash* model is a mixing framework that contains three reflective dimensions as semantically relative domains. Therefore, in *OntoMash* ecosystem, a context reasoning scheme can be determined as a function of the independent variables of these components in the domain of interest. For instance, a function of a context-aware domain can be expressed as concatenation:

$$f(x,y,z) = \lim_{i,j,k=n} \sum_{t=1}^{\infty} (x_i \cdot y_j \cdot z_k)t, \qquad (1)$$

where x_i is a context type, y_j is a reasoning level, and z_k represents an application environment according to time t. Therefore, a full-fledged *OntoMash* model is able to provide flexibility enough to effectively make a fusion with such mashup components in an open and dynamic IoT environments.

3.3 *OntoMash* Classes

OntoMash classes are entities for representing context information in an IoT information system, and share common semantic features with each other for context-aware computing. There are three *OntoMash* context classes from the root *Things*: *Physical*, *Virtual* and *Logical* context classes (Figure 5).

Specifically, the logical context classes are derived from mixing the physical context classes with virtual context classes. Ontology semantic relationships between the context classes define as *isDerivedFrom* which distinguishes each of semantic relationships by the mashup components. The context-aware processes dynamically change and adapt to the object's behaviors based on context information derived from the context classes. In *OntoMash* model the ontology property, *isDerivedFrom*, plays a significant role in an ontology reasoning process to distinguish clearly relationships between context information sources.

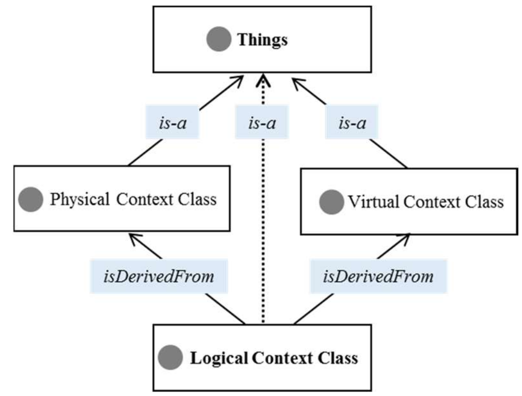

Figure 5: Semantic Relationships between OntoMash Context Classes

In this reasoning process, each attribute and its value is searched in *OntoMash* model to determine the context-aware information. In the IoT information system *OntoMash* model may have the same attributes and values from different context sources. In order to deal with the context inconsistency problems from different context sources in the context reasoning process, namespaces of OWL (Ontology Web Language) represent in the *OntoMash* model (Table 1).

Table 1: Namespaces Declaration for the OntoMash model

Prefix	Namespace
owl	http://www.w3.org.2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
xsd	http://www.w3.org/2001/xmlSchema#
xml	http://www.w3.org/XML/1998/namespace
ontomash	http://www.younglab.org/2018/07/ontomash#
phyc	http://www.younglab.org/2018/07/ontomash#pysical-context
virc	http://www.younglab.org/2018/07/ontomash#virtual-context
logc	http://www.younglab.org/2018/07/ontomash#logical-context

For uniquely distinguishing named elements and attributes in *OntoMash* model, namespaces use *http://www.young.org/ontomash#*{*physical-context*, *virtual-context*, or *logical-context*} as URI (Uniform Resource Indicator). To distinguish between elements or properties of ontology, it applies prefix *phyc*, *virc*, and *logc* respectively. A fragment of the default namespace *http://www.w3.org/202/7/owl#* uses an element *ontomash* for *OntoMash* model. For example, let us recall the example presented in the introduction section. In this case, the attribute of the same meaning related to a blood pressure value may exist in the physical context classes, the virtual

Journal of Theoretical and Applied Information Technology
28th February 2019. Vol.97. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

context classes, and the logical context classes respectively (Figure 6).

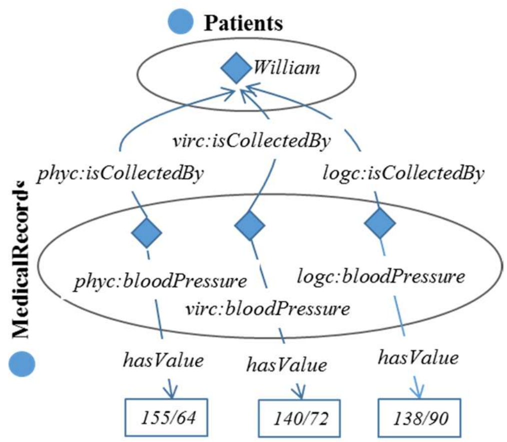

Figure 6: Example of the Different Ontology Properties with Same Semantics

In an IoT application environment, *OntoMash* model is divided into conceptual model for upper ontology mashups and logical model for domain ontology aggregations. The conceptual model is a high-level ontology which represents general concepts of basic context classes. The logical model is a collection of domain ontologies which defines the detail of general concepts and their features in each sub-domain. With this approach, an integrated view that is provided to context reasoning and awareness applications is not only a context model that accommodates the various context information from multiple sources, but also a semantically rich description of the relevant context classes in a domain of interest. Therefore, the high-level ontology classes are fully unified in *OntoMash* model for adaptation of context awareness and reasoning adaptations.

3.4 General *OntoMash* Model

The notion of *OntoMash* model is to formalize the context information classes with a more high-level organization to provide a common understanding of their terms and meanings between things in the IoT. Depending on an abstract level of knowledge concepts, ontologies can be divided into upper-level ontology and lower-level ontology in a broad sense [17]. Since the upper-level ontology deals with general and universal concepts applicable to various application domains, it is not dependent on a specific domain and situation. On the other hand, the lower-level ontology aims to formalize a specific domain and situation in the real world without pursuing universality. First of all, this paper will focus on a general *OntoMash* model which is an upper-level ontology to discuss the conceptual schema of *OntoMash* model.

In the IoT application environments, *OntoMash* model includes several kinds of vocabulary for expressing ontology classes and their relevant properties that are related to *Thing*: *Subject*, *Object*, *Context*, *Service*, and *Resource* (Figure 7).

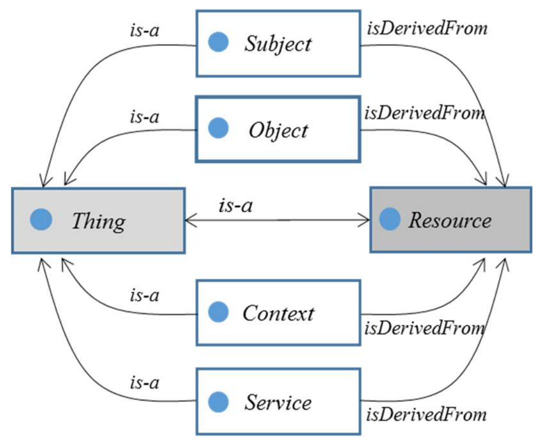

Figure 7: General OntoMash Model

In the Figure 7, the general *OntoMash* model is concisely composed of *Subject*, *Object, Context*, and *Service* classes. They is a subclasses of *Thing* and is indicated from *Resource*. Especially, *Resource* class is used to clearly distinguish the context types (physical, virtual, and logical types) from the *OntoMash* model. Thus, to enhance a search performance for reasoning, this class is located in the top level of *OntoMash* model. The terminology of the relation property for *Resource* class is *isDerivedFrom*, which is a semantic property for distinguishing context classes with each other in the *OntoMash* model.

- *Object* is physical devices such as sensors, devices, or actuators in the IoT application environments. They can interact with one another through communication channels. The sensors are used to observe the status of physical devices, and the actuators are used to control or operate intelligent computing devices. They have context information types such as name, time, location, status, owner, and more.
- *Subject* is users who can own the objects or define the context-aware services in needs. They have personal information, context awareness information, service definition information, security information, and so on.
- *Context* is any information type that can be used to characterize the situation of subjects,

Journal of Theoretical and Applied Information Technology
28th February 2019. Vol.97. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

objects, places, or services. For interoperability through the reasoning process, the context class is used to share or reuse the things in the IoT.

- *Service* provides information and any behaviors requested by the subjects or objects through a context-aware process. Service classes have service conditions and service behaviors. The service condition defines the criteria for determining the situation and the conditions to provide the service. A service process defines an action to be provided with the object's state or function.

The *OntoMash* model is widely referred to a well-defined aggregation format that is related to multi-context classes and their relevant properties. With this approach, a context information model merely has not only a context structure to accommodate the various context from multi-context resources with ontology semantics [23], but also a semantical relationship between such classes. The semantic mapping rules for representing relationships between classes in general *OntoMash* model in OWL are shown in Table 2.

Table 2: Parts of Ontology Semantic Mapping Rules for OWL

Resource	*rdfs:subClassOf*	***Thing***
?S	*rdfs:subClassOf*	***Thing***
?S	*rdfs:isDerivedFrom*	***Resource***

Each class in the model is able to make a synergistic effect through organically mutual combinations with each other. A reference model between ontology classes is depicted in Figure 8.

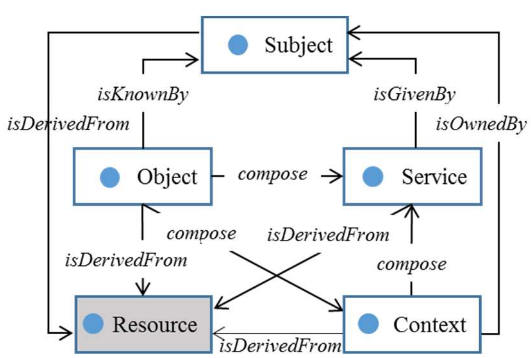

Figure 8: Reference Model between General OntoMash Classes

In the reference model between upper-level *OntoMash* classes, *Object*, *Subject*, *Context*, and *Service* all are derived from a *Resource* class. *Object* are owned by *Subject*, and they compose a class of *Service* and *Context*. *Context* is composed of *Service* and are known by *Subject*. *Service* is given by *Subject*. Each of them is modeled into a lower-level ontology to be suitable for a specific domain and application in an IoT application environment.

3.5 Global *OntoMash* Model

The global *OntoMash* Model integrates general *OntoMash* model into the specific local ontologies: physical ontology, virtual ontology, and logical ontology. In other words, the global ontology [24] is a synthesis of upper-level ontology and lower-level ontologies [7] for a unified and coherent reasoning. The upper-level ontology is a high-level ontology to capture general features from basic context classes. On the other hand, the lower-level ontologies is domain-specific ontologies to collect ontology sets to define in details of general concepts and their features in each local domain for any specific goal [25]. Hence, *OntoMash* global structure (Figure 9) is comprehensive of repetitive blocks of the lower-level ontologies according to general *OntoMash* schema rules.

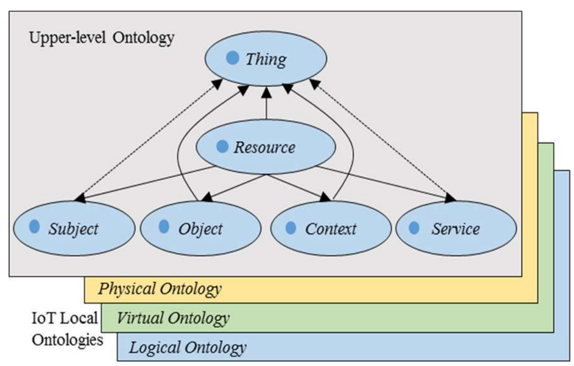

Figure 9: Global OntoMash Structure

In the Figure 9, the global *OntoMash* (GO) structure is composed of an upper-level ontology (UO) and several local ontologies (LOs): physical, virtual, logical ontologies. It can be expressed as follows:

$$GO = UO \circ \sum_{i=0}^{m} \sum_{j=0}^{n} LO_{ij}, \qquad (2)$$

where *i* is the number of a specific ontology domain, and *j* is the number of an ontology block in the block *i*. According to equation 2, Global *OntoMash* model is shown as $GO = \{ UO \circ \{ \{ LO_{11}, LO_{12}, ..., LO_{1n} \}, ..., LO_{ij}, ..., LO_{mn} \} \}$, where LO_{ij} is the context mashup blocks in global *OntoMash* model, and each LO_{ij} has seamlessly semantic relationships with each other based on *UO* schema rules.

Definition 2 (*Context integration and integrity constraints of the OntoMash model*): *OntoMash*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

model preserves (a) context integration that completely converges the lower-level (or domain) *OntoMash* to share the *OntoMash* global model among the things in the IoT, and (b) context integrity that entirely contains all of the essential context information pertaining to context reasoning.

Lemma 1(*New-fashion placing in OntoMash Model*): If an ontology transaction intends to access a target context mashup block in the *OntoMash* model, the access point should be placed in the most recently inserted ontology block in a time sequence.

Proof) The IoT applications are typically based on prompt interoperability that uses context information generated in a real-time fashion from various context sources on the IoT. Let's recall Definition 1. Suppose that the lower-level *OntoMash* sets { LO_{11}, LO_{12}, …., LO_{1n} } in global *OntoMash* model are listed in chronological order, and $LO_{11} > LO_{12}$ in time sequence is established. If an ontology transaction intends to traverse a lower-level *OntoMash* block LO_{1j} for context reasoning, the LO_{11} always becomes the target ontology block. In the same manner, if an ontology transaction intends to insert a new ontology block from local ontologies, it is inserted into the front of LO_{11}. This new-fashion placement follows a stack data structure that has a "last-in, first-service (LIFO)" fashion. Thus, this placement scheme of each local block makes a context awareness computing easier and faster in reasoning process.

OntoMash model can provide interoperability between ontology mashups into the model when different local ontology blocks cannot be integrated or merged because of mutual inconsistency of their context information. It is useful for highly dynamic, open and distributed environments and also reduces the complexity and overheads of integrating multi-context sources. Compare to mappings between global ontology and local ontologies, the *OntoMash* model has more maintainability and scalability because the changing (adding, updating, removing) of local ontology is done locally without regarding other mappings. However, finding mappings between local ontologies may not be easier than between an integrated ontology and local ontologies because of the lack of common vocabularies.

4. *ONTOMASH* ADAPTATION

4.1 Context Awareness Scheme

Context awareness refers to situational reorganization around subjects and/or objects in IoT environments. In fact, a definition of context awareness has been widely referred to many studies and described as "A system is context-aware if it uses context to provided relevant information and services to the user, where relevancy depends on the user's tasks [26]". Simplifying this definition, context awareness can streamline the context's applications to provide some context-aware services to any subject or object. In the *OntoMash* adaptation, context awareness is to use *OntoMash* context blocks to provide relevant context services to users and/or objects in the IoT application environments. It figures out how to retrieve context information from a large amount of *OntoMash* repository. For the sake of context-aware performances, it always depends on how to categorize context information types and apply context awareness computing techniques.

A categorization of the context types will be useful in reasoning processes and context awareness for fast exploration in *OntoMash* model. The important aspects of the context types are to determine "where you are, who you are with, and what resources are nearby [4]". Thus location, user, activity and computational entity are most fundamental context types for capturing the information about the executing situation [7]. Depending on semantic levels for context awareness, context types are divided into primary contexts and secondary contexts [27]. Primary contexts are any information retrieved without using any contexts, whereas the secondary contexts are any information that can be retrieved or computed using primary contexts. For example, if a blood pressure level of a patient needs to be collected, it is important to know the information of resource, identifier, location, time, and status related to the patient.

A quick and easy way to retrieve a large amount of context information in the IoT depends on how to formalize these primary contexts in a model. Even though primary context types are very controversial in research, this paper only defines the primary context as *Rid (Resource identification)*, *Identifier*, *Location*, *Time*, and *Status* (Figure 10).

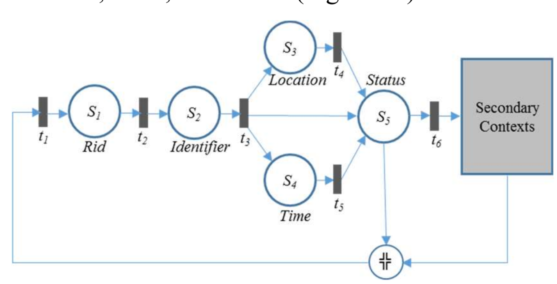

Figure 10: Partial Petri Nets for State Transitions of Primary Context Types

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In partial *Petri nets* [28] of Figure 10, *OntoMash Adapter* first finds *Rid* of an entity in state S_1 at time t_1 to figure out a context source, and then *Identifier* of an entity in state S_2 at time t_2 to figure out adaptive context information. In the *OntoMash* adapter, an entity is always a subject or an object. If there is an entity in a static IoT environment, the IoT information system directly retrieves the *Status* of the entity in the state S_4 at time t_3, and then the secondary contexts will be retrieved in the system. On the other hand, if there is a dynamic IoT environment, it retrieves the *Location* and *Time* of the entity in state S_3 and state S_4 at time t_3. Afterwards, the IoT information system finds out their status information in state S_5 at times t_4 and t_5 each, and then the secondary contexts will be retrieved. These procedures are repeatedly executed in a real-time or sequence fashion based on a context-aware process.

4.2 Context Reasoning Scheme

In the IoT information system, *OntoMash* adapter typically monitors in real time whether there are situation changes to the IoT things that use context information from multiple context sources. The *OntoMash* adaptation makes a decision on whether any service is adequate to the changes. The context reasoning scheme is applied for inferring higher context information or new knowledge from *OntoMash* repository. The context reasoning procedure consists of analyzing the ontology context model in order to infer adaptive context information as following steps:

- *Reasoner*: Allows new situational inferences from relevant context properties based on defined semantic relations and inference rules.
- *Application*: Defines processes associated with each action in the reasoning results.
- *Repository*: Stores a new context information or behavior status in the reasoning results of the context model.

For example, in a context mashup model based on medical care, the discrete behavior of each diagram's component is usually defined through finite state mechanisms. The reasoning process can be simply defined with the following transitions (Figure 11).

In Figure 11, we depict a context reasoning process of the *OntoMash Adapter* for a medical care system. The reasoning process inputs multiple context types such as physical sensor, virtual data, and logical data, and then output high-level context information after adaptive reasoning processes according to a predefined inference rules. The event function of an application determines the blood pressure level (BPL) of the patient in the results of context reasoning process. If it is less than *140*, then a message of "*Normotensive level!*" will be alerted. If not, a message of "*High blood pressure!*" will be alerted, staring alarm sound.

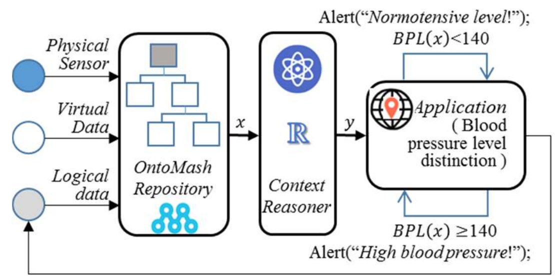

Figure 11: Context Reasoning Process of the OntoMash Adapter

In this way, if there are multiple context types in the *OntoMash* repository, the adaptive result can be reasoned by applying an appropriate selection pattern in the reasoning process. According to the system configuration of the IoT, the reasoning pattern can be set by the user or by the IoT computing system through a machine or deep leaning process. Let us recall the partially ontology mashup model depicted in Figure 6. In the *OntoMash* model, an ontology inference rule that can retrieve the logical context of *William*'s blood pressure level is applied by following ontology axioms:

$$
\begin{aligned}
&\textit{WilliamBloodPressureLevel} \cong \textit{Identifier.William} \ \cap \\
&\quad \forall \textit{isDerivedFrom.Logical-Context} \ \cap \\
&\quad \forall \textit{Status.Active} \ \cap \\
&\quad \exists \textit{hasLocation.(Room123} \cap \textit{AbcHospital)} \ \cap \\
&\quad \exists \textit{hasTime.2017-05-20-T10:22:12} \ \cap \\
&\quad \exists \textit{hasBloodPresureValue.138/90}
\end{aligned}
$$

To retrieve *William*'s blood pressure level, *OntoMash* adapter first creates a *William*'s partial ontology where the value of the attribute *Identifier* in the primary contexts is *William*, and then creates a *William*'s logical ontology where the value of the primary context's attribute *Status* is *Active*. Afterwards, the adapter retrieves the value of property *BloodPresureValue* from *William*'s logical ontology where the value of the attribute *Location* is *Room123* and *AbcHospital*, and the value of the attribute *Time* is *2017-05-20-T10:22:12*. Finally, *William*'s blood pressure level can be detected as *138/90*.

The context reasoning aspects for *OntoMash* adaptation are often emphasized by an ontology inference rule so that at least one situation must be

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

active at one time. For organizing all possible situations, their relationships and transition courses have to include *OntoMash* repository as particular cases in informal reasoning patterns as well as formal reasoning processes. This would provide a more abundant knowledgebase, better stable reasoning, and best performance of *OntoMash* adaptation.

5. EVALUATION

5.1 Evaluation Criteria

Since evaluation of an information model is an integral part of the modeling process, it helps to find the best model with well-defined concepts and how well the model will works in the real field. No matter how good a model is, it is difficult to develop an ideal model considering all situations and characteristics. In addition, it is very complicated to develop the IoT information model for representing multi-context types in unified and coherent way due to its inherent heterogeneity and dynamic natures. The proposed *OntoMash* model was considered to deal with these characteristics. Since any modeling method always has performance trade-offs in various situations, to discuss the strengths and drawbacks of the proposed model, evaluation indices are presented in Table 3.

Table 3: Evaluation Criteria of OntoMash Model

Criteria	Comments
Expressiveness	A model formulates all the knowledge explicitly.
Interoperability	A model is completely understood in implementation
Adapterbility	A model is anticipated by its applications.
Conciseness	A model is free from any unnecessary and useless axioms.
Efficiency	A model achieves maximum effects under a condition
Occupancy	How much space does a model occupy?

First of all, since *OntoMash* model is to propose a method to mashup various context types based on ontology, it would encompass all strengths and drawbacks of the ontology modeling scheme. With respect to simpler approaches such as key-value and markup models, ontologies have been known to provide clear advantages for both expressiveness and interoperability because it is a formal explicit description of consensual knowledge in a domain of discourse. In addition, *OntoMash* model is able to accommodate multiple context types from heterogeneous and distributed context source and provide an adaptability to integrate multiple local ontologies which are composed of physical, virtual, and logical ontologies and identified their ontologies distinctively.

In the IoT environment, the things operate autonomously with each other by sharing knowledge depending on intelligent information communication. This process is mainly done through context-aware reasoning. However, the context-aware reasoning process is very complicated in real world. Let's consider $\forall p_i \in \{p_1, \dots, p_m\}, \forall v_j \in \{v_1, \dots, v_n\}, and\ \forall l_k \in \{l_1, \dots, l_o\}$, here $p_i, v_j,\ and\ l_k$ are a number i of the physical ontology block p, a number j of the virtual ontology block v, and a number k of the logical ontology block l. In the *OntoMash* model, the time T required to find a specific context x for context awareness is as follows:

$$T = mC_a(\log_2 x)^a + nC_b(\log_2 y)^b + oC_c(\log_2 z)^c, \quad (3)$$

where a, b and c are the number of selected blocks in each ontology block, and x, y and z are the number of triples in an ontology block when aassuming a binary search algorithm. Therefore, the efficiency of context-aware reasoning depends on the type of ontology blocks, the number of ontology blocks, and the number of triples in *OntoMash* model.

5.2 Experimental Environments

For the simulation, ontology knowledge model and reasoning rules assume the blood pressure management with diverse IoT things. The things consist of wearable blood measurement devices, medical cloud systems, and aggregation logics and they are connected relatively close to each other on the Internet. The test dataset referred to a NCD Risk site [29], which is a network of health scientists around the world that provides rigorous and timely data on major risk factors for non-communicable diseases for all of the world's countries. To acquire high-level context information, rules are used to infer temporal relationships between sequential observations received from the IoT things and aggregated by resource identifiers. The context information describes high blood pressures, normal blood pressures, and low blood pressures.

Simulation program for performance evaluation is implemented by *simpy* at *https://www.sympy.org*, which is a discrete-event simulation library for Python version 3.6. Since *sympy* only comes with data collection capabilities in simulation results, this simulation is used along with other library such as *numpy* at *https://www.numpy.org* and *scipy* at https://www.scipy.org for statistics and *matplotlib* at *https://matplotlib.org* for plotting. Moreover, these performance experiments are performed on an *Intel® Core™ CPU 3.40 GHz* with *8GB* of *RAM*, using *Windows 10* as operating system. Under the system configuration, a simulation program for data-driven evaluation [30] [31] is implemented on the three components:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- *RQ* (Reasoning Query) *generator* dynamically generates reasoning queries at regular time intervals, and send them to *Context reasoner* one by one.
- *Context reasoner* is responsible for accepting reasoning queries from *RQ* (*Reasoning Query*) *Generator* and for translating their queries depending on *Reasoning Rules*, and forwards the axioms of the query to the *Axiom_Queue*.
- *OntoMash adapter* inputs each axiom from the *Axiom_Queue* in *FIFO* (First-In-First-Out) manner for fairness and makes a decision for finding out a *TOBs* (target ontology blocks) from *OntoMash repository*.

These components and their interconnections for the simulation model are depicted in Figure 12.

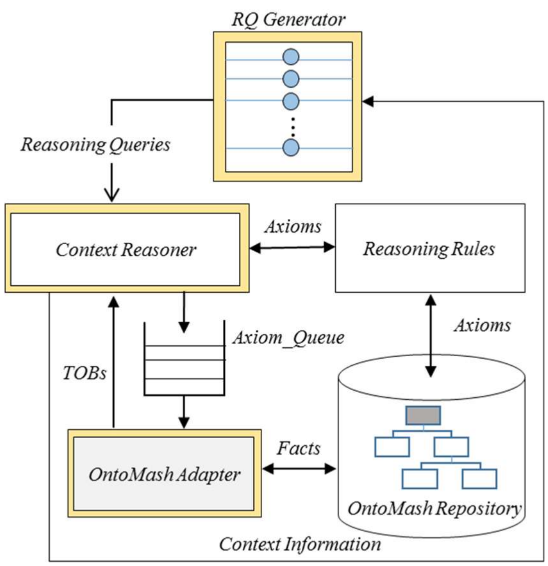

Figure 12: OntoMash Simulation Model

During simulation time, the program computes each throughput according to independent variables such as *num_triples*, *num_reasoning_queries*. The reasoning throughput are the main performance index in the experiments. Throughput can be informally defined as the amount of tasks performed by the experimental program in a given unit of simulation time. In this paper, its value is measured as the number of reasoning queries that successfully perform per simulation time. The performance of the *OntoMash* model is measured using the throughput factor which is defined as throughput $X = \frac{\sum R}{T}$, where R is the number of reasoning queries successfully completed in the time interval of T. The throughput factor will always be accompanied by creating an environment *env = simpy.Environment()* and staring the setup process *env.process(setup(env, num_triples, ext_think_time))* and they are evaluated by *env.run(until= simulation_time)* with *Python 3.6*.

5.3 Experimental Results

Although a context knowledge model for expressing a specific-domain knowledge in interest is extremely subjective according to design situations, it would be meaningful to conduct in a way of performance comparison with other relevant modeling schemes. In this experiment, the relevant schemes are *XML/RDF* [12] and *JSON* [10] which have well known as knowledge modeling technologies in the IoT environments. The *JSON* format formulates context information with simple key-value pairs that consist of attributes and their values. On the other hand, the *XML/RDF* modeling technologies represent context information in hierarchy manners to understand the meaning of various information resources on the web and to search them intelligently. For measuring space occupancy, file sizes of *OntoMash*, *XML/RDF*, and *JSON* are shown in Figure 13.

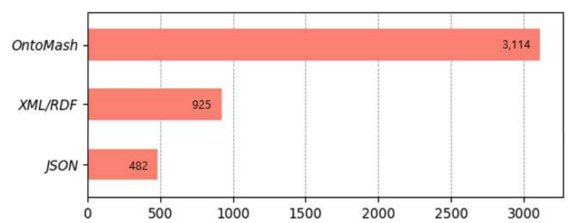

Figure 13: Comparison of File Sizes of Context Modeling Formats

In the case of storing the blood pressure measurement results of *10* persons, the file sizes of *JSON*, *XML/RDF*, and *OntoMash* models were investigated as *482*, *925*, and *3114* bytes respectively. The *OntoMash* model based on ontology requires more storage space than other modeling format because it has to include a lot of vocabularies to represent formal semantics. So it would take a lot of overhead to search a specific context triples as well as a parsing speed for context-aware reasoning.

To evaluate the efficiency of the proposed modeling technique, the throughput of reasoning queries is investigated in changing the number of triples, which are a unit of expressing specific knowledge to express as *SPO* (*Subject-Predicate-Object*). For this experiment, simulation time, *simulation_time* was set to *1,000* seconds and time interval, *ext_think_time* of reasoning queries was set to 5 seconds. The result of this experiment is shown in Figure 14.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

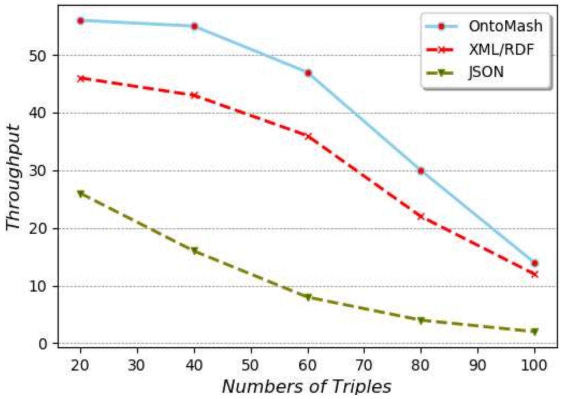

Figure 14: Throughput of the Number of Triples

As shown in Figure 14, *OntoMash* model was investigated as showing higher throughputs in all of the triples with the comparison of *XML/RDF* and *JSON*. It is because the *OntoMash* model applied a depth-first search algorithm under the indexed graph structure, *XML/RDF* used the binary tree structure search algorithm, whereas *JSON* assumes a sequential search based on key values. The results of these experiments show that the *OntoMash* model also decreases the reasoning throughputs by increasing the number of triples, but the throughput converges to that of *XML/RDF* from *90* or more of the number of triples. Therefore, it is necessary to develop a new knowledge storage model to efficiently perform reasoning queries with number of many triples in *OntoMash* model.

On the other hand, in order to find the optimal point of performance of the *OntoMash* model, throughputs by changing the number of ontology blocks and reasoning queries is investigated by simulation model shown in Figure 12. For this experiment, simulation time, *simulation_time* was set to *1,000* seconds and time interval, *ext_think_time* of reasoning queries was set to *5* seconds. In case that a reasoning query searches ontology blocks, it assumed that the probability of choosing physical ontology blocks, virtual ontology blocks, and logical ontology blocks is *70*, *20*, and *10* percent, respectively. In addition, the number of triples of a physical ontology is assumed to be 8 to 5, and randomly selected in a reasoning queries. The number of triples of a virtual ontology block and a logical ontology is set to *3* and *2* respectively. Of the reasoning queries that are issued during the entire simulation period, computational queries using physical ontology and virtual ontology assume to set *3* percent, the result of the queries are inserted in the *OntoMash* repository. The computing time and inserting time assume to be *20* and *5* seconds respectively. Algorithm to find a specific ontology block and target contexts in a reasoning query uses Equation 3. The results of this experiment are shown in Figure 15.

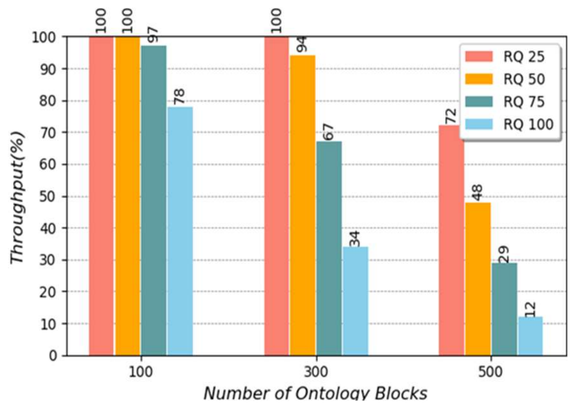

Figure 15: Through of the Number of Reasoning Queries

As shown in Figure 15, throughputs for reasoning queries *25*, *50*, *75*, and *100* were investigated as *100*, *100*, *97*(*=73/75*100*), and *78*(*=78/100*100*) percent respectively at the number of *100* ontology blocks during simulation time *1000* seconds. At the number of 300 ontology blocks, throughputs for reasoning queries *25*, *50*, *75*, and *100* were investigated as *100*, *94*, *67*, and *34* percent respectively. On the other hand, at the number of *500* ontology blocks, throughputs for reasoning queries *25*, *50*, *75*, and 100 were investigated as *72*, *48*, *29*, and *12* percent respectively during the simulation time. The experimental results show that *90* percent or more of the throughput was in *RQ 25*, *50*, and *75* at the ontology block *100*, and in *RQ 25* and *50* at the ontology block *300*. These results provide important clues to find an optimal point of performance behavior of *OntoMash* model. In addition, they show that the throughput of varying reasoning queries is more sensitive than that of varying ontology blocks. Therefore, it is necessary to discuss the optimization method of reasoning query rather than improvement of modeling method.

6. CONCLUSIONS

In this paper, an *OntoMash* modeling scheme for the diverse context mashup based on ontology was introduced to be adapted to open and dynamic IoT environments. The key point of this scheme is an adaptive context model that can mash up various context classes that use ontology in open and dynamic IoT environments. This model is useful for supporting a context-aware process through multiple context types and reasoning processes under the model. For realizing this model, we presented semantic classes to represent various context entities

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

from multiple context sources and clearly expressed semantic relationships between the classes. The proposed model has a common ontology class, named *Resource* that can distinguish the duplicated context properties from multiple context sources. This approach not only enables the model to accommodate various context information from multiple sources in a unified and coherent way, but also provides it with semantically rich relationships between the relevant context classes.

This paper also presented semantic levels in accordance with a context awareness scheme that governs context types and context reasoning. The reasoning aspects of context awareness are often emphasized by an ontology inference rules so that at least one situation must be active at a time. To organize all possible situations, their relationships and transition courses are included in an ontology mashup model as particular cases in informal or formal reasoning processes. This provides a more abundant knowledge base, better stable reasoning, and best performance for the IoT information systems.

Although the proposed model is adequate in an open and dynamic IoT environment, additional discussion is needed on how to prove validation and present a detailed implementation specification of *OntoMash* model based on an actual application conditions. This is why context-aware services can be sensitively changed in open and dynamic IoT environments. In addition, the IoT information system is needed for synchronization schemes in concurrent access to a context information model at the same time. Despite the limitations of this research, these studies can become a useful reference model for a significant evolution for novel context models and efficient context-aware services.

REFRENCES:

[1] IStrategy, I. T. U., and Policy Unit. "ITU Internet Reports 2005: The internet of things." Geneva: International Telecommunication Union (ITU) (2005).

[2] Perera, Charith, et al. "Context aware computing for the internet of things: A survey." IEEE communications surveys & tutorials 16.1 (2014): 414-454.

[3] Guermah, Hatim, et al. "An Ontology Oriented Architecture for Context Aware Services Adaptation." arXiv preprint arXiv: 1404.3280 (2014).

[4] Bettini, Claudio, et al. "A survey of context modeling and reasoning techniques." Pervasive and Mobile Computing 6.2 (2010): 161-180.

[5] Maarala, Altti Ilari, Xiang Su, and Jukka Riekki. "Semantic reasoning for context-aware Internet of Things applications." IEEE Internet of Things Journal 4.2 (2017): 461-473.

[6] Abowd, Gregory D., et al. "Towards a better understanding of context and context-awareness." International Symposium on Handheld and Ubiquitous Computing. Springer Berlin Heidelberg, 1999.

[7] Wang, Xiao Hang, et al. "Ontology based context modeling and reasoning using OWL." Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on. IEEE, 2004.

[8] Gu, Tao, et al. "An ontology-based context model in intelligent environments." Proceedings of communication networks and distributed systems modeling and simulation conference. Vol. 2004. 2004.

[9] LePendu, Paea, et al. "Ontology database: A new method for semantic modeling and an application to brainwave data." International Conference on Scientific and Statistical Database Management. Springer, Berlin, Heidelberg, 2008.

[10] Crockford, Douglas. "The application/json media type for javascript object notation (json)." (2006).

[11] W3C, Extensible Markup Language (XML), http://www.w3.org/XML.

[12] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web." Scientific American 284.5 (2001): 28-37.

[13] Ankolekar, Anupriya, et al. "The two cultures: Mashing up Web 2.0 and the Semantic Web." Proceedings of the 16th international conference on World Wide Web. ACM, 2007.

[14] Hachem, Sara, Thiago Teixeira, and Valérie Issarny. "Ontologies for the internet of things." Proceedings of the 8th Middleware Doctoral Symposium. ACM, 2011.

[15] Palfrey, John, and Urs Gasser. "Case Study: Mashups Interoperability and eInnovation." (2009).

[16] Wache, Holger, et al. "Ontology-based integration of information-a survey of existing approaches." IJCAI-01 workshop: ontologies and information sharing. Vol. 2001. 2001.

[17] Lenzerini, Maurizio. "Ontology-based data management." Proceedings of the 20th ACM

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

international conference on Information and knowledge management. ACM, 2011.

[18] Kotis, Konstantinos, and Artem Katasonov. "An ontology for the automated deployment of applications in heterogeneous IoT environments." Semantic Web Journal (SWJ) (2012).

[19] Meng, Jian, and Jinlong Chen. "A mashup model for distributed data integration." Management of e-Commerce and e-Government, 2009. ICMECG'09. International Conference on. IEEE, 2009.

[20] Abiteboul, Serge, Ohad Greenshpan, and Tova Milo. "Modeling the mashup space." Proceedings of the 10th ACM workshop on Web information and data management. ACM, 2008.

[21] Malki, Abdelhamid, and Sidi Mohamed Benslimane. "Building Semantic Mashup." ICWIT. 2012.

[22] Anjomshoaa, Amin, and A. Min Tjoa. "Towards semantic mashup tools for big data analysis." Information and Communication Technology-EurAsia Conference. Springer, Berlin, Heidelberg, 2014.

[23] Jarrar, Mustafa, Jan Demey, and Robert Meersman. "On using conceptual data modeling for ontology engineering." Journal on data semantics. Springer Berlin Heidelberg, 2003. 185-207.

[24] Brena, Ramon F., and Hector G. Ceballos. "Combining Global and Local Ontology handling in a Multiagent System." FLAIRS Conference. 2004.

[25] Hachem, Sara, Thiago Teixeira, and Valérie Issarny. "Ontologies for the internet of things." Proceedings of the 8th Middleware Doctoral Symposium. ACM, 2011.

[26] Schilit, Bill, Norman Adams, and Roy Want. "Context-aware computing applications." Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on. IEEE, 1994: 85-90.

[27] Chen, Harry, et al. "Soupa: Standard ontology for ubiquitous and pervasive applications." Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on. IEEE, 2004.

[28] Murata, Tadao. "Petri nets: Properties, analysis and applications." Proceedings of the IEEE 77.4 (1989): 541-580.

[29] NCDRISC. http://www.ncdrisc.org/http://www.ncdrisc.org/downloads/bp/individual-countries/South%20 Korea.csv

[30] Brewster, Christopher, et al. "Data driven ontology evaluation." (2004).

[31] Brank, Janez, Marko Grobelnik, and Dunja Mladenić. "A survey of ontology evaluation techniques." (2005).