Lecture Notes in Computer Science: Light-weight Ontology Reasoning Engine Model Research for Real-time Context-aware based on Smart Phone

Su-Kyoung Kim[1], Gui-Hyun Baek[1] and Gi-Tae Lee[1]

[1]125, Dongseo-daero, Yuseong-gu, Daejeon, 305-719
National Hanbat University, Korea
kimsk@hanbat.ac.kr, { beik2000, mm1023}@naver.com
[2]Electronics and Telecommunications Research Institute,
Gajeong-dong, Yuseong-gu, Daejeon, Korea
parkym@etri.re.kr

Abstract. This research suggest the big data enforcement system which utilizes context-aware and light-weighted ontology inference engine model which recognize the status data that occurs at smart phone in real time as context aware. The context-aware recognition method which will suggested at this research aim the inference which can able to manage at smart phone based on relation not based on an existing OWL-DL. Therefore we will improve the Jena into the light weighted Jena which can install and run into the smart phone. For this goal, in this research, we analyzed the inference mechanism of Jena, the model which loads the ontology model to memory, the relationship between API and SPI. Using like that analysis result, we suggest the new ontology inference model based on multiple thread which can recognize the many status data occurs in real time as a context-aware efficiently and improving of initialization step and running velocity and memory optimization which can save the status data that inference engine occurs in real time at smart phone environment.

Keywords: Big-data, Reasoning Engine, Jena, Ontology, Context-aware, Light-weight Reasoning Engine

1 Introduction

It's been 10 years to receive the 21century information- oriented society, the computerization of organization and company and department of web technology, the kinds of content became unlimited, so the IT equipment which use the contents developed so it's miniaturization and accessibility gets higher.

As a result, although thousands of data accumulate in real time, most of semiformal data are abandoned. Now the research which makes these data formal and useful information is on processing, so there are several successful case are appearing. Therefore, now we are live at big data era. The appearance of smart phone which causes the big data era make people's internet approach more easier so people can communicate the world. And also there are every kinds of sensors installs in the smart phone always the various kinds of data about the status of users and equipment occurs in real time, it uses this data and offers a various kinds and smart service such as a LBS, game, utility etc. to users. But this sensors, uses the status data which occurs at specific time and they throw away of the same time, the service is over. As this status data accumulate and they generate the big data, we can find out the context information which is formal information through analyze of the big data. Existing services are just specific status data using service, it is pretty difficult to define that this data are formalized context information at some time. In this reason, the importance about the recognition of status data as a context-aware is getting bigger. The effective method about context-aware recognition is inference methods which apply the ontology model.

This method divides into two methods; first method is inference method from upper level, second is inference method from lower level. The inference method from upper class transmits the status data to server, and activate the inference in the server, finally they transmits the result to smart phone.

This method occurs the overload when server manages all inference which all users request and inefficient traffic when occurring status data in real time transmit to server and receive the results. On the contrary, the strong point of inference method at lower level which performs the inference in smart phone which has processing ability, is there are no overload of server and data traffic. But if we load the existing inference engine into smart phone, it is hard to handle the inference of all data which occurs in real time with smart phone's ability, and also it is inefficient in expense aspect. To resolve these problems, in this research, we'll research and analyze the Jena inference engine which use the RBox(Relation-based) inference method based on existing relation and remove the problems and unnecessary logic and components. So we suggest the light weight from ontology inference engine which suitable for smart phone specifications improve the implement speed and optimize the memory to recognize the occurring status data in real time as a context. Jena which analyzed in this research, is based on Java API library, it starts research and develop at Hp laboratory, now it is process at the Apache foundation. This thesis is written according to 2.10.0 version which released at 2010.02.24. In addition, in this research, we suggest the big data enforcement system (3.1) and the context aware to enforce the more convenient service for users and context-aware ability of smart phone by forming the recognized contexts the form the standardized big data.

2 Related Research.

2.1 Ontology and Reasoning

Ontology is the data model to save the standardized conceptions exist the actual word into the computer. The standardized conceptions of ontology model define the class and keep the characteristic properties of class property. There are instance exists which each of this classes includes actual form, between instances, the properties which inherent character of classes express the relation. Now ontology remodels with semantic web technology which has been developed with W3C as the central figure. There are language to express the semantic web is RDF/RDFS, OWL etc which based on XML, XML Schema. In semantic web, it expressed all kinds of conceptions, sentence form of subject, predicate, object. It is made up three things commonly, we call triple structure. There are protégé, Top braid, Allegro graph as an ontology modeling implement based on semantic web. Inference engine takes charge of fulfill the inference and load the ontology model as an actual memory. Inference engines inference the new reception of ontology model through the graph tour, and it is divided into two kinds of engines, one is Jena which is ontology inference engine based on RDF/ RDFS, the other is pellet engine which is ontology engine based on OWL, they are based on Java API library. Jena created and researched at Hp laboratory and now it process at the Apache foundation.

2.2 The Context- aware Framework based on Smart Phone

Precede research of context-aware frame work in smart phone circumstances is context model based on rule. This model tells the sensor of smart phone and all sorts of smart phone applications which is actual data offer. The knowledge administrator save and manage the status data which occurred by this, context administrator which include inference engine uses status data which managed in knowledge administrator and decides the context information, serve as a specific application, as a free of charge. At this moment, inference engine uses the IF-THEN ways rule, therefore there is a weakness that it is hard to maintenance and it is not intelligent. Practical using ontology and use the context information which inputs to smart phone's sensor and precedence research about context-aware is MOnCa [1] which is status recognition model based on OWL.

MOnCa makes instance based on ontology model and inference model to smart phone, and accomplish the Tableaux Algorithm which based on pellet, decides context information. But its weakness is time complexity and space complexity are large, because they inference the in real time status data based on smart phone. Besides also they don't lightening the pellet, so it is difficult to manage the consuming resource to perform the inference in smart phone.

2.3 The Reasoning Engine based on Smart Phone

There is a MiRE4OWL inference engine model [3] based on OWL-DL in precedence research of inference engine based on smart phone. This model loads the ontology model, modeling the rule and question to template which has XML for and input the RETE core, inference the new fact. This has a possibility of lower speed, cause of analyzing XML with analyzer in smart phone, also questions are managed by template, so they doesn't follow the SparQL which established in W3C, so the compatibility fall down.

3 Our Method Suggestion

3.1 Context-aware and Big data Strengthen System

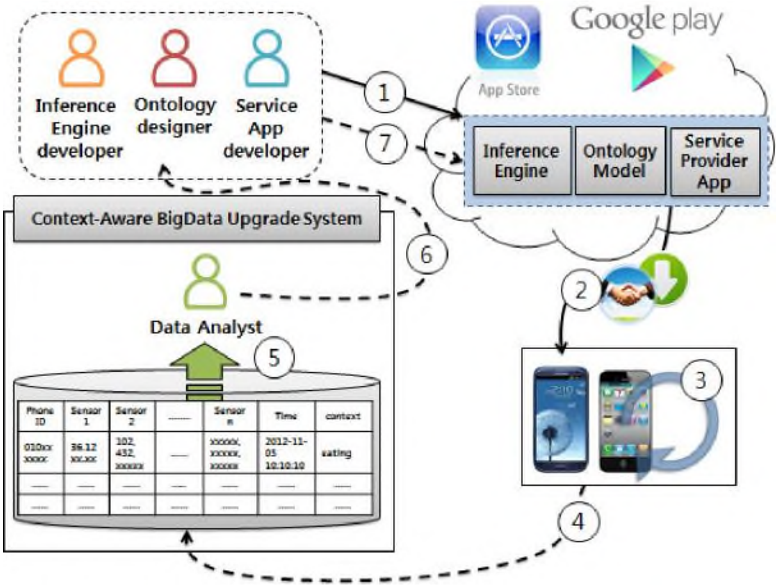

Fig. 1. Context-Aware Big Data Upgrade System

Fig. 1 is whole block diagram of model which recognize the status data as a context information and make a another standardization big data

① Jena inference engine developer, ontology designer and service provider app registers the developed inference engine and Ontology model and Service provider app at the smart phone app market.

② Some user of smart phone download this app and agree about status data and context information and collecting the personal information about cell phone's number.

③ 1] Service provider app collects the sensor figure and requires the inquiry to applying ontology model
to inference engine
2] Inference engine decides the user’s status by performing the inference which recognized the status
data which occurs the smart phone in real time as context information.
3] Return to service provider app. and service provide app offers the relevant service about user’s status.

(4) Transmit the inference context information and context information used in inference to context information big data strengthen system.
⑤ Data analyst analyzes the collected context information.
⑥ Send the analyzed result to Jena inference engine developer, ontology designer, service app provider.
© Update the service provide app, ontology model, inference engine based on analyzed result and update the new version.

In smart phone, they inference the big data called status data as a context information, and transmit that to context information big data strengthen system and analyzes again and updates the service provider app, ontology model, inference engine which are based on analyzed result, so they can offer the more enforced service to user.

3.2 Jena Engine Analysis

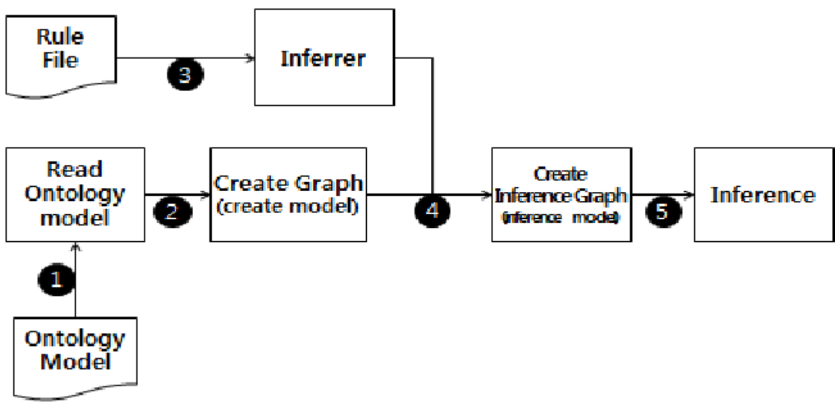

Fig. 2. The Process Analysis of Jena Engine for Reasoning

In order to light-weight reasoning engine, we analyze the structure of Jena Platform and trace the reasoning process by Jena's API. Above all, if you see the approximate Jena's process like fig. 2. For reasoning ontology, first, we run Jena API on Java Platform. As you can see fig. 2, Jena has process to inference through each role of process and handled the form of memory database.

They uses the parser to load the turtle form, N-Triple form, XML form of RDF file, OWL which is ontology model which based on semantic web expressed Unicode. Parsed triple object saved in triple store of graph. Analyzing the rule file and saves the rule object into reasoner, and the reasoner creates inference graph by binding analyzed rule object from ® and schema graph (TBox), Instance graph (ABox) made in (2), they can perform the inference using SparQL of made in inference graph.

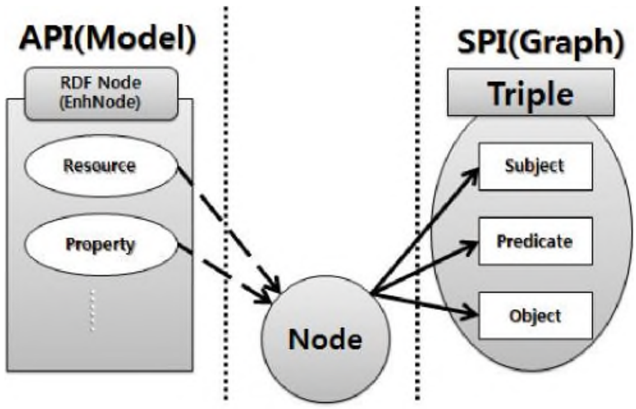

Fig. 3. Node form consist of API (Model-base) and SPI (Graph-based) in Jena

If you look at the fig. 3 in Jena, to express the ontology to semantic web, it is divided into APJ field and SPI field. In general, API field is the classes relate to model type and the classes relate to model type classes, SPI field is the classes relate to graph type and the classes relate to graph type. In commonly, model type class manages the RDF Node such like a resource, property, Literal which included in RDF graph aspect. Graph type classes includes inference engine and the class which manages the triple engine and the class which manages the triple store, so this is divided inference graph which support the inference. Model which is API field include the Node to RDF Node, property, resource in RDF graph. Node usually saves the Variable which gets into URI, Literal, SparQL etc.

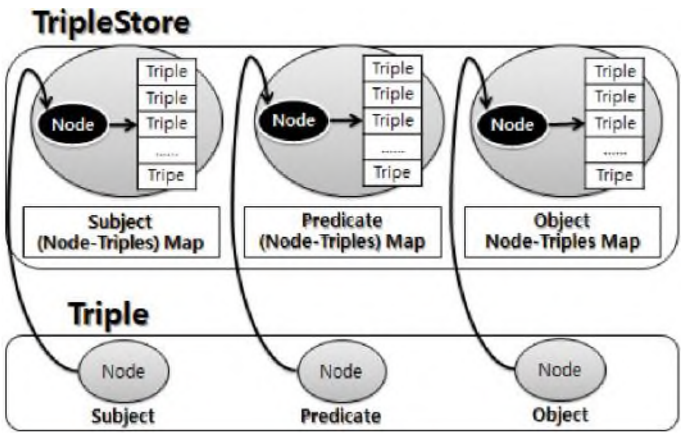

Fig. 4. Triple Store structure in Graph-based using SPI form

In Fig.4, we can see the structure, triples saves into triple store of graph. In triple store, it has structure of three-map which decide triple as value, Node as key. Three map structure decide each Node which relevant to Subject, Predicate, Object as key, and decide the triple which relevant to key. This is a structure which found the triple which decide specific Node as key and decide the specific Node of found triple as key and find again.

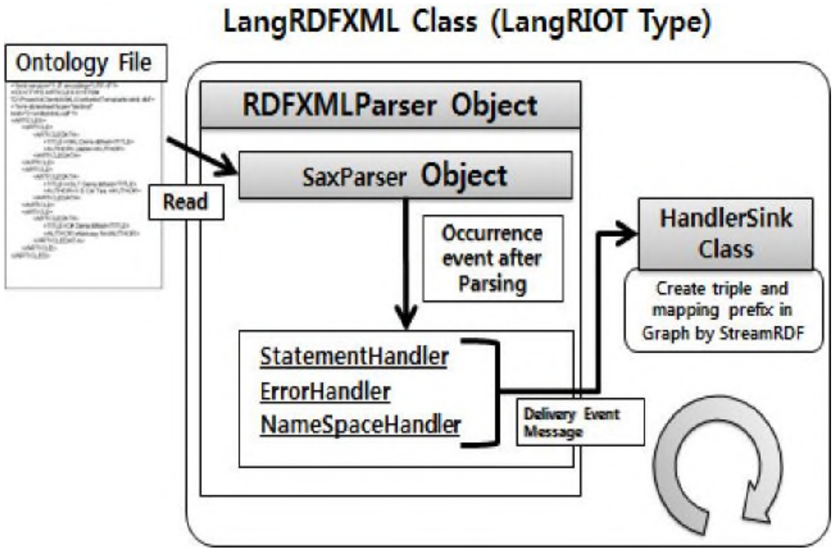

Fig. 5. Process of Loading Ontology Model using RDFXMLParser Class in Jena

Fig. 5 is shows the process which loads the ontology model. When it calls the RDFDataMgr class's load model method, in XML form, LangRDFXML object is occurred and analyzes using SAX parser and HandlerSink. SAX parser occur the event by analyzing an ontology file. HandlerSink handles the event. Triples are saved in graph's triple store, a namespace and prefix space saves in graph's prefix mapping object.

Made graphs at Fig. 6 divided into schema graph (TBox) and instance graph (ABox). To using this and inference in Jena, they should make an inference graph like picture 6. Inference graph itself is a graph which to exist the gathering of inference engine and rules, not an ontology model. Application program calls the createInfModel method using schema graph, Instance graph to the class which make model which called ModelFactory.

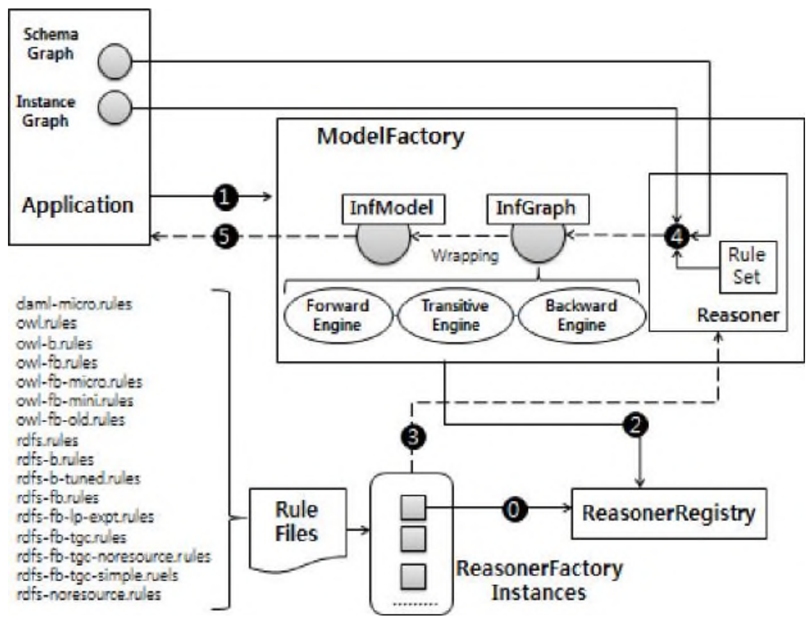

Fig. 6. Process of Inference Graph make from ModelFactory Class in Jena

In ModelFactory class, they bring objects which makes reasoner called ReasonerFactory and makes a reasoner. The reasoner reads the rule file which is relevant rule set which doesn't read when it is made, and analyzes and saves on rule set list statically. The public interest of rule file and made reasoner was binding the instance graph of ModelFactory and rule gathering of schema graph to inference graph and return.

At last, they returns the inference model which wrapping an inference graph. In inference graph, Forward engine which support the forward inference, Transitive engine which support the transitive inference, and Backward engine which support the backward engine which support the backward inference are created. In reference graph, they go through the preparation process before using this inference engine and performing inference.

They checks the triple which has the predicate called subClassOf, subPropertyOf, domain, range of schema graph, instance graph and Transitive Engine saves triple which has predicate subPropertyOf and subClassOf which is fulfillment relation to each two set. And in the LPBRuleEngine of rule store which is kind of Backward Engine, they saves the rule made in RETEEngine object and backward rule of rule gathering. The RETEEngine which is kind of Forward-Engine can remake rules. Gathering of rule axioms is correct the closure rules and saves in the rule store.

After approximate preparation process, the inference of Jena performs into LPBRuleEngine object. LPBRuleEngine supports the hybrid inference method, performs the cruise, backward, fulfillment inference with a LPInterpreter object unit. LPInterpreter object inference and investigate the subClassOf prepared in preparation process, set of triple which has subPropertyOf as the predicate and new rules which made instance graph, schema graph in RETEEngine and backward rules to standard. Reasoning utilizes the SparQL query smpily or call the find method of inference graph directly, besides all of them utilize the Iterator. When Iterator calls the hasNext method, they deducts the next result, if the result is exist, they return the true. When they calls the next method, they return the deducted result from hasNext method. Like this way of Iterator doesn't command list which include all result, they command the working of next inference.

3.3 Graph Model to Optimization of Memory

In Jena, Node is one of the smallest units which consist of ontology model like Fig. 3. One ontology model maneuvers to one graph object. Most of Nodes, they has the string value of URI form which distinguish the resource with name space and identifier. In here, the name space doesn't need to include all Node, because their string length is long and same name space of some model. In other words, like graph 1, URI form Nodes gives the integer type ID to name space part, and saving the name space and prefix is more effective in memory aspect. Look at Fig. 8, prefix mapping objects which dependent to existing graph locates global part of inference engine as a PrefixMapper object. This object confers this object when URI form Node is create, request the name space's ID. In addition, when loading the ontology model which based on 3.4 binary, if the ID value about between same name space and prefix are different, they will manage it consistently. Therefore, the memory doesn't waste causing of redundant string value about name space.

Table 1. Saving from of PrefixMapper Object

IDl	Namespace	Prefix
0x70000000	http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdf
0x70000001	http://www.w3.org/2000/01/rdf-schema#	rdfs
0x70000002	http://www.w3.org/2002/07/owl#	owl
0x7000000n	http://www.lightening.com/lightontology	lightonto

3.4 Binary Conversion Model to fulfill speed Advancement

The ontology model of Jena reads through graph like Fig. 5. As they are input of string file, they support the reading / writing model of XML, N-Triple, N-turtle form ontology file. If they changes ontology file model into binary file, the running speed of inference engine will be faster than existed engine. Inference engine running speed became slow, because the ontology model which is XML form based on string have to analyze through the SAX parser, when they loads to inference engine. This is general common sense, binary form saving method is more effective than saving method based on string in cost aspect. In Java, they supports the binary saving method of object form, if they save triple unit like Fig. 7, and uses the saving way which name space of Nodes in Triple about ID separately, they can improve the saving effectiveness and loading effectiveness.

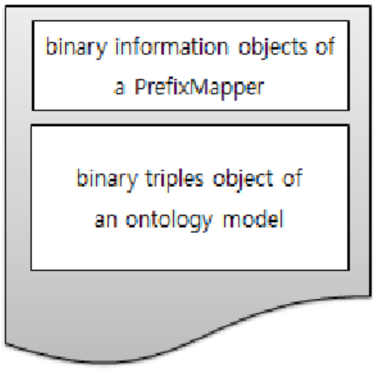

Fig. 7. For Efficient Process of Ontology Model based on Binary Form

3.5 Suggestion of Inference Engine Model based on Smart Phone

This research suggests the model like Fig. 8 finally.

① In context-aware service provider, State Listener load and deliver the context information to the SparQL Creator. To receive context information, SparQL Creator sends an Ontology model and query to the light-weighted Jena inference engine.

② Lighten Jena inference engine analyzes the query which transmit from the SparQL Creator by using SparQL Parser.

③ If the transmitted model is existed, the ontology model manager requests the binding to reasoner. If the ontology model doesn't exist, SparQL Parser responds to context aware service provider app that there is no apply ontology model.

④ After finishing the analysis about transmitted query, SparQL parser request inference performance to reasoner by using Interpreter. At this moment, reasoner runs the inference by using the rules which made from RETEEngine and pre-defined rules.

⑤ Light Weighted Jena Inference Engine transmits the inferenced context information to Context-aware Service-Provider, and they offers the appropriate service to application which registered at smart phone. Existing Jena inference mechanism was subordinate to inference graph, and handling about SparQL was also subordinate to inference graph and inference engine, so it is impossible to handle the parallel, but lightening Jena inference engine model which suggested in picture 8, is the model which can handle each part parallel, and it is absolutely independent process which is not context aware service provider, so many application process can use the inference engine.

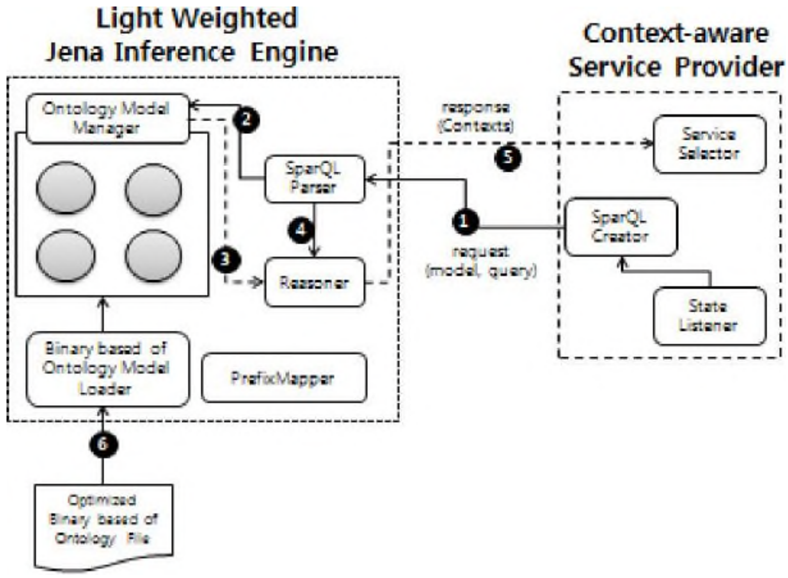

Fig. 8. Light weighted Reasoning Engine and Context-aware Provider Structure

4 Conclusion

The main purpose of this research is to improve the existing Jena engine into the light-weighted engine by analysis of Jena inference engine and utilize the smart phone's context aware. In this research, the analysis about approximate functions of inference engine and inference graph, the way which loading and saving ontology model was completed. The detailed content about SparQL and inference algorism of inference engine doesn't analysis yet, but we found out they were subordinate to inference-graph, and they has a structure which can't achieve the parallel handling. Therefore the inference engine of Jena in this research can optimize the memory like 3.3, and it also can support the loading the ontology model like 3.4 more fast.

Besides it divides the dependency which concentrates on inference graph like 3.5, and consist the lightened inference model. This model was a singular thread existing way but now they changed it can the parallel handling of multiple thread way through this research. With these, more lightened and effective structure's Jena inference engine loads the smart phone. So it will highly contribute the big data creation later.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology(2012-
0004360)

References

1. Kim-jae min, Kim-mi hwa, Park-young taek,"MOnCa: For context aware smart phone application based on ontology." Frame work, information-science society journal: software and application and application book 38, unit 7, pp.369--381 (2011)
2. Taehun Kim, Insuk Park, Soon J. Hyun, and Dongman Lee, “MiRE4OWL : Mobile Rule Engine for OWL”, 34th Annual IEEE Computer Software and Applications Conference Workshops, pp.317--322 (2010)
3. Yun Her, Su-Kyoung Kim, Young-Taek Jin, "A Context-Aware Framework using Ontology for Smart Phone Platform," International Journal of Digital Content Technology and its Applications, vol.4, no.5, pp.159--167, Aug (2010)
4. S.Kim, K.Hong, “A Study of Dynamic Web Ontol ogy for Comparison- shopping Agent based on Semantic Web”, Journal of Intelligence and Information Systems, Korea Intelligent Information System Society Quarterly, pISSN : 1229-4152, v.11, no.2, pp. 31--45 (2005)
5. S. Kim, "Using Description Logic and Rule Language for Web Ontology Modeling", Proceedings of Korea Intelligent Information System Society (KIISS) spring conference, pp.277-- 285 (2007)
6. Kim-su kyoung, Kim-gun woo,"customized searching service which considerate the user's intention based on ontology." Intelligence information research, v.18, no.4, pp.19--28, December (2012)
7. Lee-jin woo, Kim-su kyoung, "Analysis and Supplement Research of Hand Model", Korea Science Computer Information Society (KSCI review), Vol.20 No.2 (2012)
8. Su-kyoung Kim, Yoon-sung Ch, Ki-Hong Ahn, "Mobile Search Interaction Framework based on User Experience Reusing", JNIT: Journal of Next Generation Information Technology, Vol. 4, No. 2, pp. 20--28 (2013)
9. Sukyoung Kim, Kihong Ahn: User-intent custom adaptive search service based on ontology and similar-word retrieval algorithm composition. ICITST pp.121--126, December (2012)
10. Jurgen Angele, Michael Kifer, and Georg Lausen, “Ontologies in F-Logic”, International Handbooks on Information Systems, DOI 10.1007/978-3-540-92673-3, pp.45--70 (2009)
11. Jeff Z. Pan, “Resource Description Framework”, International Handbooks on Information Systems, DOI 10.1007/978-3-540-92673-3, pp.71--90 (2009)
12. Pascal Hitzler and Bijan Parsia, “Ontologies and Rules”, International Handbooks on Information Systems, DOI 10.1007/978-3-540-92673-3, pp. 111--132 (2009)
13. Bo-Hyun Yun, Dea-Ho Back, Sang-Bum Kim, Kyung-Soo Han, Hea-Chang Lim, An Information Retrieval Method of Alleviating Semantic Term Mismatches with Interactive Query Expansion, the Korea Institute of Information Science, (1999)
14. Chantal Taconet, Zakia Kazi-Aoui, "Context-awareness and Model Driven Engineering", E-Commerce application scenario. ICDIM (2008)
15. MobiLife, http://www.ist-mobilife.org, (2008)
16. Apache Jena http://jena.apache.org/