

# Integrated Processes of SDR Data for Real-time Processing

Sang-Young Lee

*Department of Health Administration Namseoul University, 21 Maeju-ri,  
Seongwan-eup, Cheonan, South Korea  
Sylee@nsu.ac.kr*

## **Abstract**

*In this paper, relative data is classified using STDC which is an efficient classification process using the ontology technique. Classified data are saved at the storage according to its SDR type. Integrated processes are used to reuse the saved SDR data. Thus, relative data is constructed in a systematic reuse system applying total architecture. This overcomes the disadvantage of the past processes that required numerous joint computation when handling question and answer. SDTC Technique solves the weakness of old methods which required multiple join calculation that caused functional decline and allows normalized type of classification task.*

**Keywords:** *SDR, Integrated, Process, Reuse*

## **1. Introduction**

Recently, the concept of ontology has been receiving attention with its business strategy that is intellectual and which provides various services. With the use of XML (Extensible Markup Language) that utilizes Semantic Tag, convenient and efficient representation is possible[1, 2]. However, this kind of ontology only provides a meaningful modelling of the information. This is still insufficient when it comes to representing the characteristics of the concept and their interrelationships.

Thus, ontology as the basic component of knowledge base, needed an exclusive language to represent precise concept structure. The following are the Web Ontology Description Languages: RDF (Resource Description Framework), RDFS (RDF Schema), DAML (DARPA Agent Markup Language)+OIL (Ontology Interface Layer) and OWL (Web Ontology Language) [3]. Relative data applies the external concept of convention, and not internal relations. Therefore, relative data does not represent the concept of meaning-centered but the relationship among the entities in correlation to data[4, 5]. Following this, response time is prolonged as there are computations and complicated relationships in question/answer[6]. This concludes to the disadvantage that it is difficult to show the precise data that the user wants. At present, as the importance of this kind of relative data is emphasized, Semantic Data Repository (referred as SDR) and the likes that used Semantic Data Type Category (referred as SDTC) are preferred for reuse[7, 8, 9]. Most semantic repositories are designed to manage semantic models but do not deal with access rights. As in the case of databases that are traditional, data stored inside a semantic database should be protected by granting access to authorized users only. In the database area, over the past two decades, great advancements have been made regarding access control and the management of data privileges such as discretionary and mandatory access control models[10, 11, 12]. Currently, semantic data repositories require the same attention. The granting of authorization privileges to the semantic model's entities, and their individuals should be addressed to prevent knowledge leakage and to maintain high-quality services through secure access to the knowledge base[13, 14, 15].

In this paper, relative data is classified using STDC which is an efficient classification process using the ontology technique. Classified data are saved at the storage according to its SDR type. Integrated processes are used to reuse the saved SDR data.

## 2. Related Research

Looking at the research related to the improvement of XML data schema extraction processes to improve reuse, there was a research that evaluated cohesion and binding [16]. There is also another research that focuses not only on the reuse of actual components, but also having various final products released from component development processes as required basis[17, 18]. Research was done on the following: suggestion of systematic quality system to evaluate re-usability, identifying important characteristics to derive quality components [19]. Processing the present ontology for ontology reuse and changing it according to its usage was also a part of the research. However, there are still insufficiencies in the areas of relative data reuse during its actual use. Presently, there are numerous researches being held for next generation intellectual research so that computers can read, understand and process information instead of human beings. This will allow production of highly efficient new data.

One of the most famous definitions of the term “ontology” is “formal explicit specification of shared conceptualization”[20, 21]. Another famous definition is “classifications of the existing concepts”. In our study, each requirements statement should be interpreted based on the knowledge of atomic constituents of meaning, and ontology is used as such knowledge. Thus an ontology, which is used in requirements analysis, has to have atomic concepts that are interpreted in the same way by any stakeholder in a specific application domain.

Most ontology based researches make it possible to reuse the various requirements of final products released during the component development process. This process defines the relationship among the final products produced in the software development arising from the requirements. This is also advantageous because by tracking the requirements, it is possible to know whether the developed final products follow the requirement. However, this has problems since the concepts of function and non-functional requirements are not included [22, 23].

CORVMII, Renaissance, USRM, and MORALE are the reuse process study of legacy systems. CORVMII is the model used to integrate the re-engineering studies in architecture code base. Renaissance consists of 4 steps: evolution plan, realization, delivery, and distribution. This also provides re-engineering technical reports. USRM is composed of 4 steps: planning, analysis, positioning and changing. This is more of a data function approach method. MORALE consists of 3 steps: analysis, design and evolution. This is the process that extracts information by analyzing the present system[2, 7]. However, these aforementioned processes cannot support the whole process of software development and there are no systematic reverse engineering process and guidelines. Another problem is that legacy system information is configured based on the fact that it can be extracted.

In this paper, the SDTC technique is produced to solve these aforementioned disadvantages. This is to produce SDR of numerous information of relative data through SDTC technique. This aims to provide trustworthy data that the user wants through integrated processes. With the convenient management processes of SDTC technique, this method is proposing trustworthy measures to integrate high volume past data and real-time data. This can be done through continuously updating the SDR of the-real time data that is produced by the end user.

### 3. SDTC Technique

The SDTC technique categorizes the relative data and saves in SDR. Using SDR, the SDTC technique solves the problems of data reuse. In other words, the SDTC technique increases the trust of end users by using efficient Web OWL patterns in software in the process of unifying the reusable data.

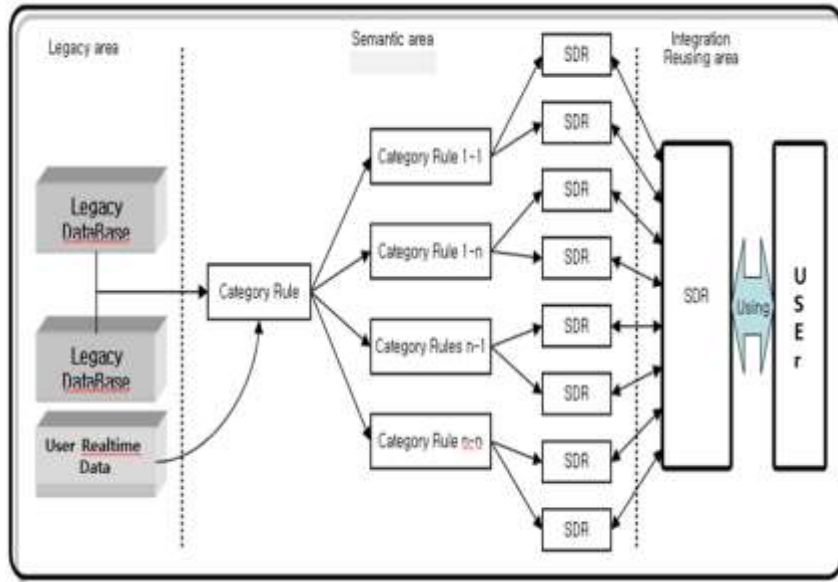
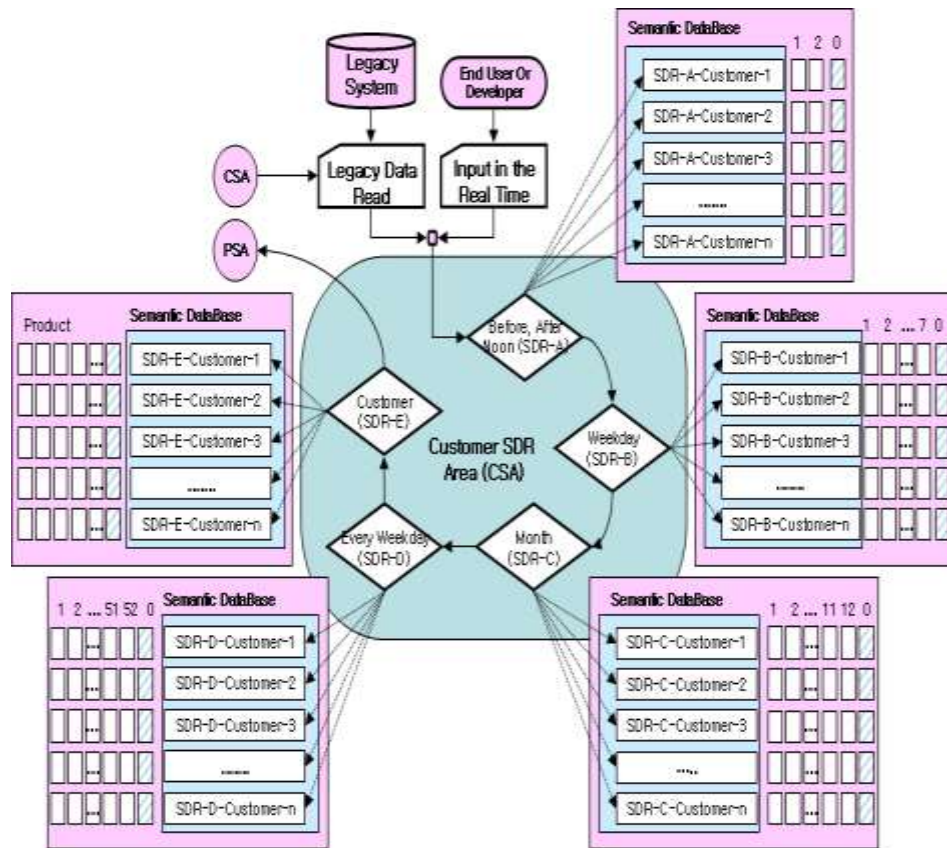


Figure 1. SDTC Structure

Figure 1 is the structure of SDTC technique. Legacy Area is realizing dispersed relative data and categorizing the usable real time data through a unifying process. Semantic Area categorizes relative data based on Category Rule. Data is categorized by classification types and it is divided into sub class by Category Rule 1-1 ~ Category Rule n-n before being saved in SDR. Moreover, real-time data that are produced are also categorized and saved in SDR. This process enables managing relative data more efficiently in real time. The Integration Reusing Area unifies the created SDR for end users' satisfaction.

SDTC Technique categorizes relative data by “Customer and Product” in the major class and “Before Noon, After Noon, Date, Weekday, Month, Every Week” in the sub class and these data are exported per category. By using categorized SDR data, it saves processing time and leads to a more efficient production. Data that are inputted by end-users are also saved in SDR using the SDTC Technique.

In Figure 2(Customer SDR Area / CSA), there are 5 stages (SDR-A ~ SDR-E) of classification process on Customers. SDR data that are produced after the classification process are saved in SDR-A-Customer-1 ~ SDR-E-Customer-n. Categorizing improves processing time, efficiency of data processing, and allows data to be managed more systematically. The created SDR is an important data that can be used in unifying the process of productivity improvement. Once CSA is finished, classification process is now moved to Product SDR Area / PSA. CSA is the category for Customer Code (n). Creating information map about resources so that relative data will not be readable by human beings but by computers. Through SDR integration process for reuse, a highly efficient service will be provided to the end user.



**Figure 2. Customer SDR Storage Hierarchy**

Figure 3 is SDR integrated structure. This is the part where the relativity of related data is increased and the efficiency of data reuse is increased. The integrated process of SDR classifies data with high specific gravity from those with low specific gravity. Then, data can be used selectively. With this, intellectual data usage to extract the data that the user needs will become possible. There are three types in the integrated process: No Event (NE), Customer Event In (CEI), Product Event In (PEI). NE is when there are no events, and CEI and PEI are cases when there are events. When there is no event, software is executed and needed data is extracted simultaneously. Then, data with high specific gravity is shown so that the user can use it. When there is an event, it is divided into partial input and total input. As the user selects one field of information, small range integration process is done with the remaining data.

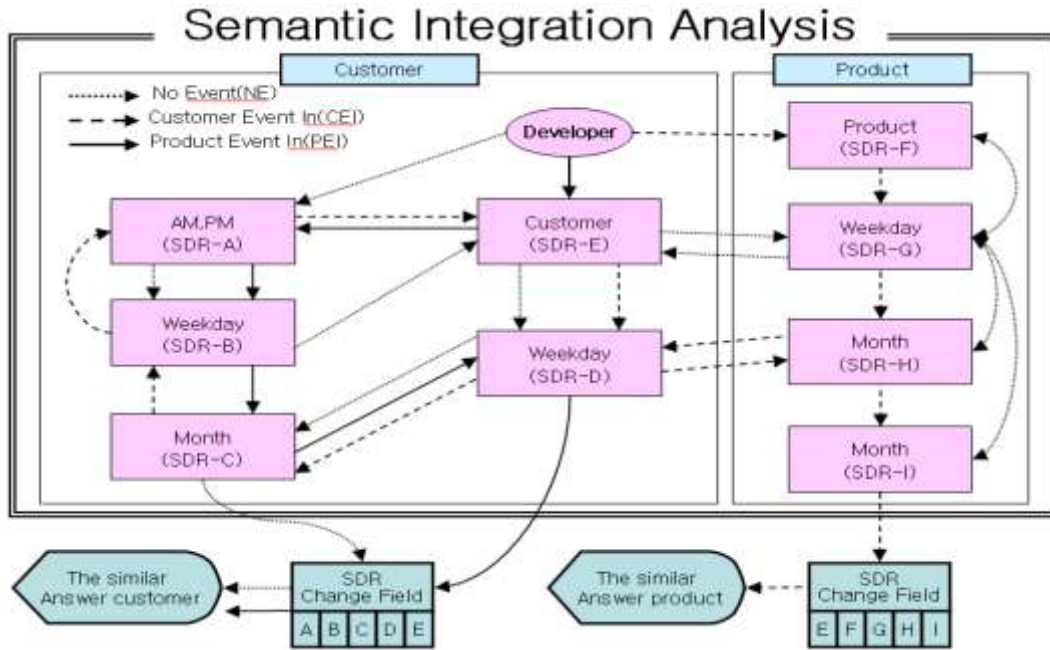


Figure 3. SDR's Integrated Structure

**Type 1: No Event (NE)**

In this type, software extracts data from SDR without the event of the user. This is an integrated loop that shows the user its client and product information. Total SDR will be searched.  $R$  and  $S$  are relations while  $r$  and  $s$  are tuples, and finally,  $\theta$  is the operator.

$$\begin{aligned}
 CST_n &= \{t | t \in SM_1 \wedge t \in SM_2 \wedge t \in SM_3 \wedge PST_n \\
 &\quad t \in SM_4 \wedge t \in SM_5\} \\
 PST_n &= \{t | t \in SM_1 \wedge t \in SM_2 \wedge t \in SM_3 \wedge CST_n \\
 &\quad t \in SM_4 \wedge t \in SM_5\} \\
 R_{SM_3 \theta SM_4} S &= (r \cdot s | r \in R \wedge s \in S \wedge (r.SM_1 \theta s.SM_n)) \\
 T &= \{t | t \in (R_{SM_3 \theta SM_4} S) \vee t \in SM_0\}
 \end{aligned}$$

**Type 2: Customer Event In (CEI)**

This type is when user enters a partial data of the client as this is compared with NE loop. Inserted data is the default value and data is extracted from SDR. Also, when full client data is entered, data can be extracted from SDR through CEI's integrated loop. CEI integrated loop considers the availability of product integrated loop's partial or full data before extraction. Then, CEI's integrated loop is processed considering the data.

All input:  $\{t | t \in SM_6 \wedge t \in SM_7 \wedge t \in SM_8 \wedge t \in SM_9\}$

Some input:  $\{t | t \in SM_5 \wedge t \in SM_6 \wedge t \in SM_7 \\ \wedge t \in SM_8 \wedge t \in SM_9\}$

$$R_{SM_6 \theta SM_9} S = (r \cdot s | r \in R \wedge s \in S \wedge (r.SM_6 \theta s.SM_n))$$

### **Type 3: Product Event In (PEI)**

This type considers the relative data when client's partial or full data is inserted. This considers the availability of relative data. Then, PEI's integrated process will be implemented.

All input, Some input:

$$\{t | t \in SM_5 \wedge t \in SM_1 \wedge t \in SM_2 \wedge t \in SM_3 \wedge t \in SM_4\}$$

$$R_{SM_1 SM_4} S = (r \cdot s | r \in R \wedge s \in S \wedge (r.SM_1 \theta s.SM_4))$$

SDTC Technique categorizes relative data by "Customer and Product" in the major class and "Before Noon, After Noon, Date, Weekday, Month, Every Week" in the sub class and these data are exported per category. By using categorized SDR data, it saves processing time and leads to a more efficient production. Data that are inputted by end-users are also saved in SDR using the SDTC Technique.

SDR data that are produced after the classification process are saved in SDR-A-Customer-1 ~ SDR-E-Customer-n. Categorizing improves processing time, efficiency of data processing, and allows data to be managed more systematically. The created SDR is an important data that can be used in unifying the process of productivity improvement. Once CSA is finished, classification process is now moved to Product SDR Area / PSA. CSA is the category for Customer Code (n).

#### **Step 1: Creation of SDR-A**

In this stage, Time and Date from Customer Record is extracted and they are recorded in Before Time (1) and After Time (2) category based from Customer Code (n). To save the data of real-time renewal and deletion of information, Change Field (0) of SDR-A is placed. Customer Code is created and saved when there is an existing field in Customer Code (n). If there is no existing field, a field is created in Customer Code (a) in SDR-A and the base value is set to 0 (zero).

#### **Step 2: Creation of SDR-B**

In this stage, SDR-B is created by categorizing the data fields of relative data. They are categorized from Monday to Sunday, and saved as SDR-B (1) ~ SDR-B(7) in SDR-B. If there is no existing Customer Code Record, Customer Code is created just like Step 1. SDR-B (0) is set to renew and delete the information.

#### **Steps 3 and 4: Creation of SDR-C, SDR-D**

In stages 3 and 4, relative data is categorized from January to December and saved in SDR-C(1) ~ SDR-C(12). Each category is sub categorized per week, from week 1 to week 52. If there is no existing Customer Code Record, Customer Code is created just like Step 1. SDR-C (0) and SDR-D (0) are set to renew and delete the information.

#### **Step 5: Creation of SDR-E**

In this stage, relative data are categorized and saved by Product Code (n) related to Customer Code (n). Customer Code (n) that is saved in SDR-E is set as base key. A two-dimensional table is created using the Customer Code (n) in column and the Product Code (n) in row. Product Code (n) can be repeatedly created depending on the Recode of Customer Code (n). When data is categorized, value is accumulated by finding the Product Code (n) included in the Customer Code (n).

### Step 6: Creation of SDR-F

In this stage, Time and Date are extracted from the Product category after reading the relative data of the Legacy System. There is a Before Time (1) and After Time (2) category based on the Product Code (n). It has a Change Field (0) in SDR-F in order to renew and save real-time information. Product Code is created and saved in a related category when there is an existing field in Product Code (n). If there is no existing field in the product code field, a field will be created in the Product Code(n) of SDR-F and will have a base value of 0 (zero).

### Step 7: Creation of SDR-G

In this step, relative data is categorized into data items and create SDR-G ~ I. Extract items of relative data. Classify the extracted item into Monday - Sunday and accumulate it in SDR-G(1) ~ SDR-G(7) of SDR-G then save it. When there is no Product Code, create with the same steps as Step 6. Then, assign SDR-G (0) to renew, delete and save the information.

### Steps 8, 9: Creation of SDR-H, SDR-I

In this step, relative data is categorized into January - December, then saved in SDR-H (1) ~SDR-H (12). Items are classified according to weeks, Week 1- Week 52, then saved in SDR-I (1) ~ SDR-I (52). When there is no record of Product Code, create one with the same steps in Step 6. Then, assign SDR-H (0) and SDR-I (0) to renew, delete and save the information.

### SDR Integration

Creating information map about resources so that relative data will not be readable by human beings but by computers. Through SDR integration process for reuse, a highly efficient service will be provided to the end user.

This is the part where the relativity of related data is increased and the efficiency of data reuse is increased. The integrated process of SDR classifies data with high specific gravity from those with low specific gravity. Then, data can be used selectively. With this, intellectual data usage to extract the data that the user needs will become possible.

There are three types in the integrated process: No Event (NE), Customer Event In (CEI), Product Event In (PEI). NE is when there are no events, and CEI and PEI are cases when there are events. When there is no event, software is executed and needed data is extracted simultaneously. Then, data with high specific gravity is shown so that the user can use it. When there is an event, it is divided into partial input and total input. As the user selects one field of information, small range integration process is done with the remaining data.

## 4. Case Study and Implementation

In this paper, actual data of sales management system which is relative data are applied to accurately and expeditiously enter sales data for the evaluation of SDTC techniques

$L_n$  is the constant value to be applied in the SDR item. This applies different weight in the items of relative data. This is also used to grant higher weight on selected SDR. On the other hand, this can also limit the application of specific items.

$$L_n \times S_n \xrightarrow{T} CS_n \text{ or } PS_n$$

$S_n$  is the item used to read the items in relative data and classify the matching characteristics.

Creates and data as the conversion rule to SDR.

- Formula for  $CS_n$  improvement of accuracy

$$CST_n = L_1SM1_{(1:2)} + L_2SM2_{(1:7)} + L_3SM3_{(1:12)} + L_4SM4_{(1:52)} + L_5SM5_{(1:n)} < L_1, L_2, \dots, L_n >$$

- Formula for  $PS_n$  improvement of accuracy

$$PST_n = L_5SM5_{(1:n)} + L_6SM6_{(1:2)} + L_7SM7_{(1:7)} + L_8SM8_{(1:12)} + L_9SM9_{(1:52)} < L_5, L_6, \dots, L_n >$$

Formula 1 grants the clear values of SDR data. The range being used will vary depending on the data with higher priority as it is saved. is the client's SDR while is the product's SDR.

The item with high priority among the integrated items and can improve the reusability and productivity. It is the constant value that is applied differently according to the SDR item. This can give or deduct weight to each item. This should be applied differently according to the environment for amplification of SDTC technique efficiency. Produced SDR of changes in real time depending on the current system status being used by the user. Frequencies are applied respectively and it is calculated based on the item with high frequencies.

It is used after finding the field with matching conditions. For it is usually operated after is selected, and for, after is selected. In this case, it is possible to extract accurate values more quickly and this also allows shortened inquiry processing time.

Constant value is applied to integrate the high frequency SDR among the SDR values until SDR-A~I that are created according to the relative data type classification. Values with the highest frequencies are extracted by applying these constant values and integrating SDR. Then, client information is extracted through NE loop and CEI loop. Product information values are extracted through NE loop and PEI loop. At this point, priorities are set according to the value being applied in the respective SDR.

Legacy Area is realizing dispersed relative data and categorizing the usable real time data through a unifying process. Semantic Area categorizes relative data based on Category Rule. Data is categorized by classification types and it is divided into sub class by Category Rule 1-1 ~ Category Rule n-n before being saved in SDR. Moreover, real-time data that are produced are also categorized and saved in SDR. This process enables managing relative data more efficiently in real time. The Integration Reusing Area unifies the created SDR for end users' satisfaction.

It is possible to search for the result needed according to the client's needs. According to the priority, if the result that meets the client's requirements shows 1~5, it is labeled as Strong Combination. If it falls under 6~10, it is Medium Combination and it is Weak Combination when it falls above 11. The application of SDTC technique, in these cases, is very effective in the aspects of improving reusability and extracting data that matches the requirements of the client.

Weight was applied in the data until SDR-A ~ SDR-I where Semantic data information is more structured, that it can be filtered according to individual needs and which can be applied for individual service.

Among the respective SDR values that were created up to SDR-A ~ SDR-I according to the classification type of relative data, constant value is applied for data filtering fit for individual characteristics. The weight within the same set showed similar results



## 5. Conclusion and Future Studies

In this paper, relative data is classified using STDC which is an efficient classification process using the ontology technique. Classified data are saved at the storage according to its SDR type. Integrated processes are used to reuse the saved SDR data. Thus, relative data is constructed in a systematic reuse system applying total architecture. This overcomes the disadvantage of the past processes that required numerous joint computation when handling question and answer. SDTC Technique solves the weakness of old methods which required multiple join calculation that caused functional decline and allows normalized type of classification task. By applying semantic data extraction technique to integrate heterogeneous relative data, the unsolved problem of compatibility was solved. Through this, relativity in the relative data of legacy system's reuse was increased, as well as efficiency and the disadvantages of existing processes were overcome.

## Acknowledgments

Funding for this paper was provided by Namseoul university

## References

- [1] D. Jeong, M. Choi, Y. Jeong, S. Han " Implementation and Evaluation of a Web Ontology Storage based on Relation Analysis of OWL Elements and Query Patterns" *Journal of Kiise D : Database*, vol.35, no.3, (2008), pp.231-242.
- [2] R. Fikes, D. L. McGuinness, "An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL" *Knowledge Systems Laboratory Computer Science Department Stanford University March1*, <http://www.daml.org/>, (2010).
- [3] O. Gotel and A. Finkelstein, " An Analysis of the Requirements Traceability Problem," *Proceedings of the First International conference on Requirements Engineering*, Colorado Springs, Colo., pp.94-101, (1994).
- [4] J. Lee, H. Cho, I. Ko, " A Method for Requirements Traceability for Reuse of Artifacts using Requirements-Ontology-based Semantic Tagging" *Journal of Kiise : Software and Applications*, (2013), pp.357-365.
- [5] D. Abadi, "Aurora: A New Model and Architecture for Data Stream Management ," *VLDB Journal* , vol. 12 , no. 2 , (2003).
- [6] S. On, J. Her, "A Quality System for Evaluating Reusability of Core Assets in Product Line Engineering", *Journal of Kiise: Software and Applications*, vol.33, no.3, (2014), pp.277-288.
- [7] J. Brokestra, A. Kampoman and F. van Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema," *Springer Verlag, ISWC 2002*, vol. LANCIS 2342, (2002).
- [8] A. Jain and C. Farkas, "Secure Resource Des-cription Framework: an Access control Model," *SACMAT'06*, Lake Tahoe, California, USA, (2006).
- [9] K. Crowston, "Information Systems Success in Free and Open Source Software Development: Theory and Measures", *Software Process: Improvement and Practice (Special Issue on Free/Open Source Software Processes.)*, (2006).
- [10] T. Akidau, " The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing", *PVLDB* , vol. 8, no. 12, (2015), pp. 1792 – 1803.
- [11] R. C. Fernandez " Liquid: Unifying Nearline and Offline Big Data Integration", *CIDR*, (2015).
- [12] J. Meehan , N. Tatbul , S. Zdonik , C. Aslantas , U. Cetintemel , J. Du , T. Kraska , S. Madden , D. Maier , A. Pavlo , M. Stonebraker , K. Tufte and H. Wang , "S-Store: Streaming Meets Transaction Processing", *PVLDB*, vol. 8 , no. 13, (2015), pp. 2134-2145.
- [13] M. Stonebraker , " SciDB: A Database Management System for Applications with Complex Analytics ," *Computing in Science and Engineering* , vol. 15 , no. 3 , pp. 54-62, (2013).
- [14] B. Fitzgerald, "The Transformation of Open Source Software", *MIS Quarterly*, vol. 30, pp. 587-598, (2006).
- [15] W. Scacchi, K. Lyytinen, "Understanding Requirements for Open Source Software", *Design Requirements Engineering - A Multi-disciplinary perspective for the next decade*, Springer-Verlag, (2009).
- [16] J. Cleland-Huang, "Automated support for managing feature requests in open forums", *Communications of the ACM*, vol. 52, (2009), pp. 68-74.

- [17] A. van Lamsweerde, "Goal-Oriented in Requirements Engineering", Requirements Engineering - From System Goals to UML Models to Software Specifications, (2007).
- [18] S. Konrad, B. H. C. Cheng, "Requirements Patterns for Embedded Systems", Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02), (2002).
- [19] M. Fernández-López, A. Gómez-Pérez, "Searching for a Time Ontology for Semantic Web Applications", Formal Ontology in Information Systems, Turín, Italy, (2004).
- [20] A. Gómez-Pérez, M.C. Suárez-Figueroa, "NeOn Methodology: Scenarios for Building Networks of Ontologies", 16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW 2008). Conference Poster, Italy, (2008).
- [21] A. Gómez-Pérez, M. Fernández-López, O. Corcho, "Ontological Engineering", Advanced Information and Knowledge Processing series. Springer, Heidelberg, (2003).
- [22] A. Gómez-Pérez, F. Ortiz-Rodríguez, B. Villazón-Terrazas, "Legal Ontologies for the Spanish e-Government. In: Current Topics in Artificial Intelligence", Springer, Heidelberg CrossRef, (2006), pp. 301–310.
- [23] J.M. Gómez-Pérez, T. Pariente, C. Buil-Aranda, G. Herrero, A. Baena, "NeOn Deliverable D8.3.1.", Ontologies for pharmaceutical case studies, (2007).

## Author



**Sang-Young Lee**, Professor, Dept. of Health Administration,  
Namseoul University, South Korea.