Effective Context-based BIM Style Description (BSD) of Query Results

Tae Wook Kang
ICT Lab. Korea Institute of Construction Technology
Seoul, South Korea
e-mail: laputa99999@gmail.com

Hyun Sang Choi
ICT Lab. Korea Institute of Construction Technology
Seoul, South Korea
e-mail: hyunsang@kict.re.kr

***Abstract*—The purpose of this study is to develop an effective Building Information Modeling (BIM) query visualization style description and a method for utilizing it based on context. Such BIM would influence usability in visualizing query results according to a user's use case context after querying the objects required from a BIM database. This study proposes a context-based structure for BIM style description (BSD) of query results and a language that defines it effectively. The proposed BSD is demonstrated through a case study, and conclusions are derived.**

Keywords-BIM; Query Language; Context; Visualization Style; BSD.

I. INTRODUCTION

Building Information Modeling (BIM) technology is a 3D-based building object database (DB) modeling and utilization technology specialized for the architecture, engineering, and construction (AEC) industries. BIM can be actively applied to unstructured building designs and building member prefabrication.

Recently, BIM technology has been used to conduct various studies on building Facility Management (FM), Building Automation Systems (BASs), Building Energy Management Systems (BEMSs), and Geographic Information Systems (GISs) for Smart Building and Urban Management (SBUM). This is because problems that cannot be solved by existing FM, BAS, BEMS, and GIS systems could be effectively addressed if 3D building object DB information from the BIM were well converged.

For BIM-based SBUM, BIM DB information must be queried effectively, and the queried results must be visualized according to the use case context for users. In particular, processing and visualizing the queried results according to the use case context can influence the usability of the BIM-based SBUM system. However, few studies related to this area have been conducted because BIM technology is relatively new, and FM, BAS, and BEMS, as well as technologies for integration and utilization of their information have emerged within the last two to three years.

This study proposes a context-based BSD structure that can effectively visualize queried results obtained from the BIM DB by a user according to the use case context, and a language that can define the structure effectively. The proposed BSD is demonstrated through a case study, and conclusions are derived.

Here, context-based BSD refers to a method for displaying queried information effectively in various application contexts based on BIM. For example, in some applications, only important query results among all results are preferred to be displayed on a screen. Furthermore, a representation mode (color, visualization style, etc.) for certain results may be emphasized depending on the context in which a user searches the virtual space. The context-based BSD mode supports these features, so that it can help users to recognize query results effectively in BIM information that includes a large number of BIM objects and make a decision.

We described the content for BSD research as follows.

1.The Introduction section describes research background and motivation.
2.The Study methodology section describes the study workflow.
3.The Related literature review surveys the recent research.
4.The Development of the BSD query result structure section describes the BSD method design.
5.The Implementation case study section develops the prototype to validate BSD.
6.The Discussion section describes the benefits of BSD
7.The Conclusion and future work section explains the results of our study and draws conclusions.

II. STUDY METHODOLOGY

The research scope of this study is the development of a context-based BSD structure. The study's conclusions are derived through case studies.

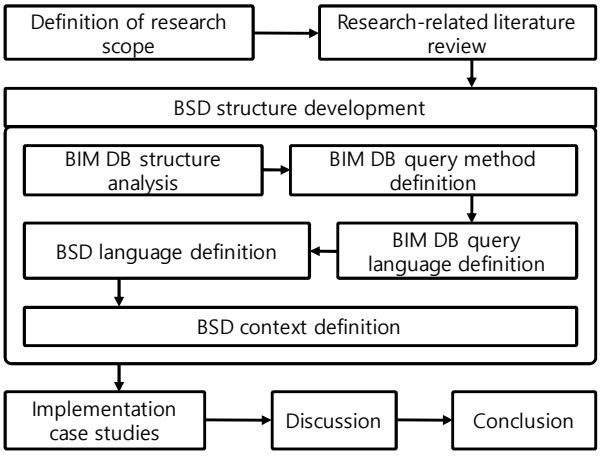


Figure 1. Research flow.

 ISBN: 978-1-61208-363-6

The study method is shown in Fig. 1. The differentiation of this study can be derived by reviewing the research scope and related literature. For BSD structure development, first, the Industry Foundation Classes (IFC)-based BIM DB structure is analyzed, and a simplified BIM query language structure is developed.

Then, the context-based BSD language is defined and conclusions are derived via the implementation of case studies.

III. RELATED LITERATURE REVIEW

Studies related to BIM DB queries were conducted to develop query methods in accordance with their utilization purpose, and they are based on the IFC DB proposed by buildingSMART alliance [8] or on commercial building modeling system DBs, such as Autodesk Revit [11] and Graphisoft ArchiCAD software [12].

Research on object interaction query [1] proposed a Query Volume (QV) method that is specialized for the purpose of building design. Research on automated extraction from and querying of the BIM DB [2] proposed a building component query, location query, and space query for the BIM DB query operations, and implemented them. Research on spatio-semantic consistency checking for the BIM DB [3] proposed a semantically queryable measure for building spaces. Research on the spatial query language [4] defined the operators for space query of GIS and building objects, and developed a viewer that can perform space queries with regard to objects represented by CityGML [9]. Research on construction-specific spatial information [5] proposed a query language that can check constructability from the designed building model information. Another study on the GIS model information representation proposed online mapping [6] using HTML5. In that study, GIS model information was represented graphically using HTML5. In a study on theme-based mapping and graphical representation of GIS model information [7], the direction for the concept of the Extensible Markup Language (XML) theme-based graphic symbol creation and its extension were proposed. Many studies related to this are based on GIS.

The literature review related to the present study shows that most research focused on query methods and language development for building object query, design intention check, space check, and constructability. However, few studies have been conducted on a method for effectively visualizing queried results from the BIM DB according to the use case context, and there is no related tool to perform this. In addition, the development of BIM technology is still underway. Nonetheless, few studies have been done on how to express query results for BIM shape and attributes effectively.

IV. DEVELOPMENT OF THE BSD QUERY RESULT STRUCTURE

A. Analysis of the BIM DB structure

The BIM DB in this study is based on the IFC [10]. The IFC is a BIM international standard file format that can represent a building structure as relationships among building objects, such as building, story, slab, floor, column, door, and window. The IFC is stored in EXPRESS or XML mode, and verification rules are defined for each object class in order to conduct object information integrity checks. Recently, buildingSMART presented IFC4 that improved the extendibility of file formats and removed ambiguity.

The IFC can search an entire IFC object hierarchy structure and extract required information using the IfcProject object that defines a building project. Each object can search topological information in relation to the corresponding object through the IfcRelationship of the IfcObjectDefinition. The IfcRelationship is redefined as it is derived into classes of IfcRelAggregates or IfcRelContainedInSpatialStructure, depending on the type of relationship with the objects. These classes represent whole/part and reference relationships between objects. The relationships that are used most to search the IFC objects are Relating/Related, Decomposed, Contained, and Referenced.

The building space and member objects have an inclusion relationship with each other. For example, IfcBuilding that includes IfcProject includes an aggregate relationship of IfcBuildingStorey using the IfcRelAggregates object. Conversely, IfcBuildingStorey references objects, in which they are included, as an inverse relationship (INV). IfcBuildingStorey manages IfcProduct using the relationship object called IfcRelContainedInSpatialStructure. Fig. 2 shows the IfcBuilding class relationship.

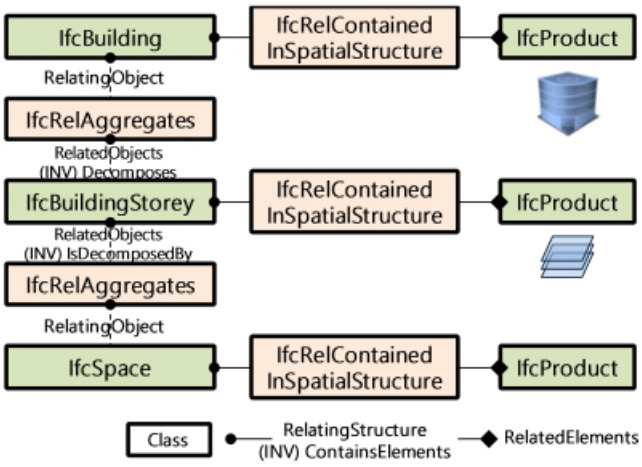


Figure 2. IfcBuilding classes relationship [10]

The IFC stores building information as shape information and attribute information. The shape information is recorded in boundary representation (B-Rep) format or parametric-based primitive format. If the IFC

 ISBN: 978-1-61208-363-6

shape information is analyzed, space-based objects can be queried.

The attribute information is provided via IfcPropertySet in association with objects. An object can be related to multiple attribute sets. IfcPropertySet has IfcProperty which manages individual attributes. Therefore, an attribute value of a particular object can be acquired via access to IfcProperty; in addition, an object with a specific attribute value can be searched.

In IFC2x3, the IFC structure consists of more than 700 classes based on extendibility. This is highly complex because it defines relationships between classes semantically. In this paper, the Simple BIM (SBIM) DB structure shown in Fig. 3 was designed to make querying of IFC DB information more convenient.

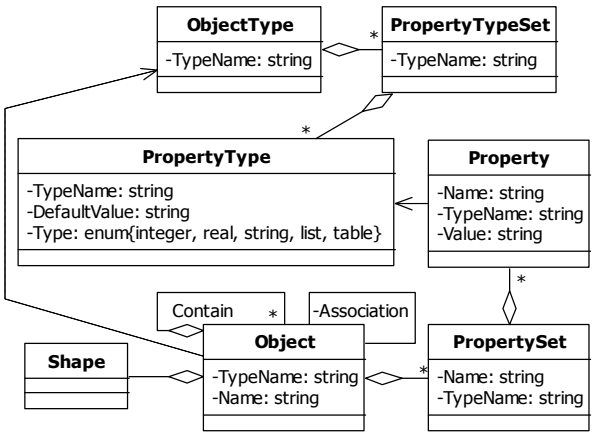


Figure 3. Simple BIM DB architecture.

B. Definition of the BIM DB query method

For a BIM DB query, IFC is parsed to be stored in the simple BIM DB structure. Building classes such as slab, wall, and window are mapped to objects, whereas types of building classes are mapped to ObjectType. A Whole/Part structure of an object is represented by a Contain Relationship. Other relationships can be represented by an Association Relationship.

Since Object classes and ObjectType classes have a Dependency relationship, a specific building element can be queried. In addition, Object classes include PropertySet classes such that objects with a specific attribute value can be searched. Searching according to object relationships can be done using the Contain and Association relationships. Fig. 4 shows the BIM DB query process, including BSD processing.

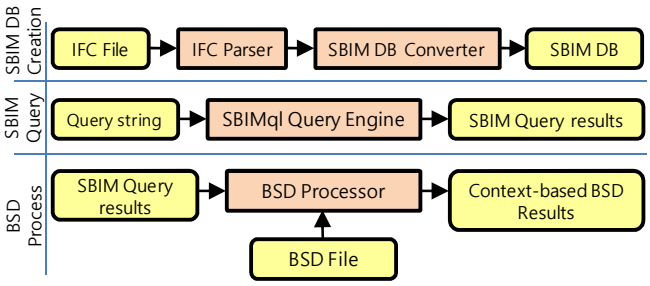


Figure 4. BIM DB query process.

C. Definition of the BIM DB query language

In this study, Simple BIM DB Query Language (SBIMql) is developed to define an information query format for SBIM DB. SBIMql uses a token analyzer and a parser generator such as LEXical analyzer (Lex) or Yet Another Compiler Compiler (YACC) in order to check syntax errors automatically. The language structure is defined as being similar to that of Structured Query Language (SQL). The SBIMql definition for YACC is as follows:

```
Variable
   : PROPERTY_NAME

Value
   : STRING

ConditionOP
   : '=' | '<' | '>' | '<=' | '>='

Func
: IsIn
| IsCross
| IsOut

SimpleExpression
: Variables ConditionOP Value

LogicOp
: 'AND'
| 'OR'

Expression
: SimpleExpression
| SimpleExpression LogicOp Expression
```

Using SBIMql, the following query can be defined syntactically.

```
SELECT * FROM IfcObject WHERE
      Type=IfcWindow AND Name='Window xyz'
```

 ISBN: 978-1-61208-363-6

Figure 5. SBIMql process algorithm.

The semantics of this query statement is for querying all the objects of Type IfcWindow and the Name attribute value of Window xyz. Fig. 5 shows the algorithm of the SBIM query engine that processes SBIMql.

D. Definition of the Context-based BSD structure

In order for a user to recognize query results effectively, it is highly important to support features that provide color schema according to attribute values based on the use case context while visualizing the query result. It is also important to emphasize specific objects and specific values of objects, which influences usability directly. To implement this feature, context-based BSD language is defined to display SBIMql query results effectively based on the use case context. The BSD must support a variety of use case contexts for users.

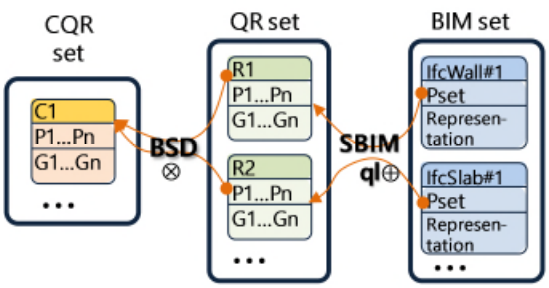


Figure 6. Set Mapping from QR to CQR.

The issue described in the previous paragraph can be defined as a problem of mapping elements in the query results (QR) set into the context-based QR (CQR) described by the context-based BSD, as shown in Fig. 6. As can be seen from Fig. 6, BSD plays a role as a conversion function between sets.

The BSD consists of a number of BIM styles (BSs) of query results that define the view style of the query result. Depending on the context in which the BS is applied, an attribute and shape style are applied to the BS. The context consists of specific style application conditions under which the style is applied. The condition has filters that define detailed conditions to process the attribute or shape styles in accordance with the context. The attribute style defines how to represent the attribute values in the query result to a user, whereas the shape style defines the materials that are used to visualize the object shape in the query result. The material can define color, transparency, and texture. For example, if the context is a space program, the color schema can be developed and applied accordingly. Table 1 describes the context-based BSD definition.

$$BSD = \{BS^*\}, BS = \{BS_{Arg}, C^*\}, \quad (1)$$

$$BS_{Arg} = \{Ctx, Desc\}, Ctx = Context, Desc = Description, \quad (2)$$

$$C = \{C_{Arg}, F^*\}, C_{Arg} = \{T, N\}, \quad (3)$$

$$F = \{F_{Arg}, L\}, F_{Arg} = \{Var, V, V_{From}, V_{To}\}. \quad (4)$$

TABLE I. CONTEXT-BASED BSD DEFINITION

Element	*Description*
BSD	BIM query results Style Definition
BS	BIM Style Definition
BS_{Arg}	Arguments of BIM Style like below Ctx = Context name Desc = Description
BS_{Arg}	BIM Style Definition Arguments
Ctx	BSD Context Definition
$Desc$	BSD Description
C	BSD Context Condition Definition
C_{Arg}	BSD Context Condition Arguments
T	Context-based Condition Type
N	Context-based Condition Name
F	Context-based Condition Filter Definition
F_{Arg}	Filter Arguments that have a variable, value, and equation.
L	BSD Condition Logic script to describe the BIM Property and Geometry Transformation depending on Context
Var	Context-based Condition Filer Variable such as Type, DistanceFromCameraToObjectCenter and OrderIndex
V	Filter Variable value
V_{From}	'From value' to evaluate the Filter condition
V_{To}	'To value' to evaluate the Filter condition

 ISBN: 978-1-61208-363-6

Fig. 7 shows the algorithm related to rendering the query results considering BSD.

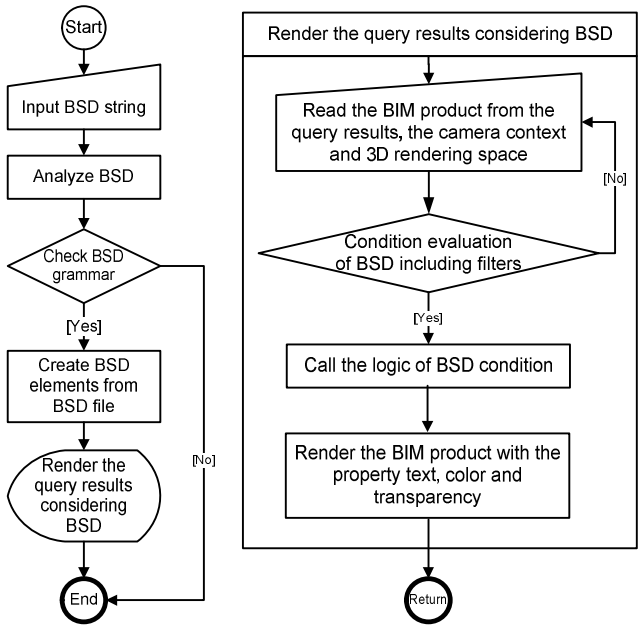


Figure 7. BSD process algorithm.

Fig. 8 shows the BSD elements class diagram. It consists of conditions which can contain the filters. The filter contains the other filter or the logic to represent how to render the queried BIM product. The filter consists of *Var* , V , V_{From}, and V_{To} to evaluate the condition. This logic can be executed when the condition including the filters is satisfied.

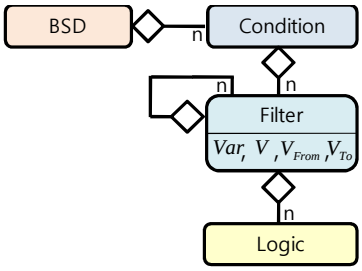


Figure 8. BSD elements class diagram in Unified Modeling Language (UML).

The following code is an XML definition example to visualize the BIM query information in the context of the Space Program Management using the proposed BSD.

```
<BSD>
 <BS Context="Facility management"
      Desc="Information highlight">
  <Condition Type="Property" Name="Length of object">
   <Filter Var="Type" Value="IfcWallStandardCase">
    <Filter Var="DistanceFromCameraToObjectCenterPosition"
          FromValue="0" ToValue="10">
     <Logic>
          Value = string.format("Length of %s = %.2f",
               Property.PSet("Name"), Property.PSet("Length"))
          TextHeight = 30.0 -
               DistanceFromCameraToObjectCenterPosition
     </Logic>
    </Filter>
   </Filter>
  </Condition>
  <Condition Type="Geometry" Desc="Object highlight">
   <Filter Var="Type" Value="IfcWallStandardCase">
    <Filter Var="SortIndex" Value="0">
     <Logic>
      Color = RGB(0.0, 1.0, 1.0, 1.0)
     </Logic>
    </Filter>
   </Filter>
  </Condition>
  <Condition Type="Geometry" Desc="Transparent schema">
   <Filter Var="Type" Value="IfcWallStandardCase">
    <Filter Var="DistanceFromCameraToObjectCenterPosition"
          FromValue="0" ToValue="10">
     <Filter Var="SortIndex" ValueFrom="1"
          ValueTo="infinite">
      <Logic>
       Ratio = (QR.Count - SortIndex) / QR.Count
       Color = RGB(0.0, 1.0, 0.0, Ratio)
      </Logic>
    </Filter></Filter></Filter>
  </Condition>
 </BS>
</BSD>
```

V. IMPLEMENTATION CASE STUDY

SBIMql and the BSD were implemented using the open-source eXtensible Building Information Modeling (xBIM) Toolkit for this case study via the context-based BSD implementation. Note that xBIM is an open-source-based software development BIM tool that supports IFC read-and-write as well as mesh processing using Open CASCADE, and visualization via a 3D rendering engine for IFC shape visualization.

 ISBN: 978-1-61208-363-6

Fig. 9 shows a result that performs the SBIMql query statement "SELECT * FROM IfcWallStandardCase WHERE IfcWallStandardCase.Length > 5000." Fig. 9 shows the model before (above) and after (below) the SBIMql query.

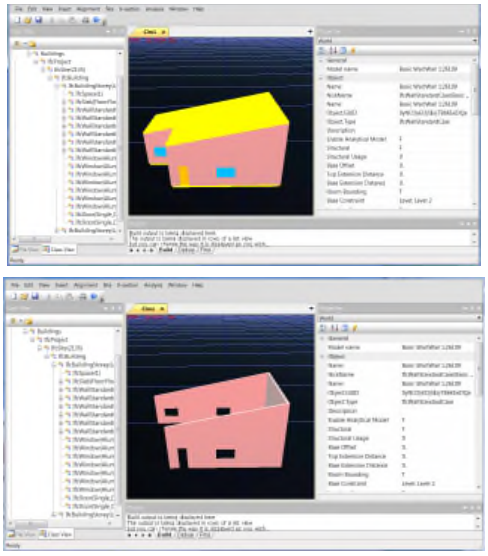


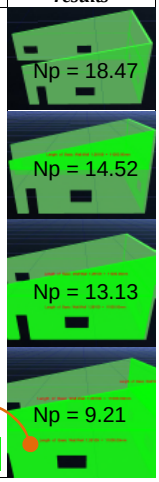
Figure 9. SBIMql query and results.

The following code shows the result of applying the context-based BSD according to the use case context. It was verified that the query result was effectively visualized using the BSD according to the use case context.

```
SELECT BSD(*, 'Context-based BSD.xml') FROM IfcWallStandardCase WHERE IfcWallStandardCase.Length > 5000' ORDERBY IfcWallStandardCase.Length
```

Table 2 lists the BSD results for how the SBIMql-processed result can be different depending on the context condition.

TABLE II. CONTEXT-BASED BSD RESULTS

Context-based BSD Condition and BSD Logic Script	*SBIMql+BSD results*
Var = DistanceFromCameraToObjectCenterPosition Np = Near distance from camera to object center position *If* (0 ≤ *Var* < 15) → Value = string.format("Length of %s = %.2fm", Property.PSet("Name"), Property.PSet("Length")) TextHeight = 30.0 - DistanceFromCameraToObjectCenterPosition Color = RGB(0.0, 1.0, 0.0, 0.8) *Else* → Value = "" Color = RGB(0.0, 1.0, 0.0, 0.3) *End*	

Context-based BSD Condition and BSD Logic Script	*SBIMql+BSD results*
Var = *SortIndex* *If* (*Var* = 0) → Color = RGB(1.0, 1.0, 0.0, 0.0)	
Var = *SortIndex* *If* (0 ≤ *Var* < inf) → Ratio = (QR.Count - SortIndex) / QR.Count Color = RGB(0.0, 1.0, 0.0, Ratio)	
Var = *Length* *If* (0 ≤ *Var* < 10000) → Color = RGB(0.0, 1.0, 0.0, 1.0) *If* (10000 ≤ *Var* < 12000) → Color = RGB(0.0, 1.0, 1.0, 1.0) *If* (12000 ≤ *Var* < 15000) → Color = RGB(0.0, 0.0, 1.0, 1.0) *If* (15000 ≤ *Var* < inf) → Color = RGB(1.0, 0.0, 1.0, 1.0)	

VI. DISCUSSION

To evaluate the usability of the BSD, a five-point scale Likert survey was conducted with three BIM practitioners and three ordinary persons by asking them how much they understood the query intentions after showing the results of SBIMql and SBIMql+BSD.

TABLE III. CONTEXT-BASED BSD RESULTS

Usability Item	*SBIMql*	*SBIMql+ BSD*	*Difference*
Understanding (U) – easy to understand the query result.	1.7	3.7	2.0
Search (S) – required objects can be searched quickly.	1.7	4.0	2.3
Recognition (R) – information on the screen can be recognized comfortably.	1.7	3.7	2.0

The interview participants provided positive answers for the BSD usability, shown in Table 3, which distinctively demonstrate the difference between the two groups of SBIMql and BSD. Particularly, respondents recognized that the improved ability to search for a needed object within the queried objects results set.

The participants suggested that an increased variety of operators is required for the BSD, and that more improvements are needed in terms of queries on the BIM objects and BSD processing performance. There were the discussions related to BSD results such as the meanings and effects as described below.

First, there are meanings when the BSD is used for the object classification between the BIM products which have similar shapes but different property values.

Second, the query results can be understood as representing an object by using BSD intuitively. It is possible to validate the query results regardless of whether the search result is correct.

 ISBN: 978-1-61208-363-6

Third, the BSD can be used to represent the color schema depending on the property value of the BIM product.

Forth, the query results can be visualized considering the navigation context including the variables such as the camera distance from the observed object in the 3D rendering space.

However, there were some issues as described below.

First, if there are many objects in a BIM model, it seems that the BSD query process performance is slow. In case of this, it is necessary to improve the performance.

Second, the various BSD operations such as "touch", "in", and "out" between BIM products are needed to represent the query results considering the model space context.

In addition, to satisfy the suggested opinions, optimization and efficient space indexing for SBIMql and BSD processing are required.

VII. CONCLUSION AND FUTURE WORK

For context-based BSD structure development, the IFC-based BIM DB structure was analyzed, and a simplified BIM query language structure was developed. The BSD proposed in the implementation case study showed the feasibility of visualizing query results based on the use case context of a user.

By using the proposed BSD method, tools that can increase the usability of query results can be provided to the AEC industries according to various use case contexts, such as SBUM.

The survey results cannot be verified statistically, because the number of the samples was small, which is the limitation of this study. This limitation will be overcome in the subsequent study. Moreover, we will extend the mapping operators for the proposed BSD and study how to analyze the information graphically in the future.

ACKNOWLEDGMENT

This research was supported by a grant from the "BIM/GIS Platform-based Construction Spatial Information Integration Operation Technology Development" project, which is one of the main projects in 2014 of the Korea Institute of Construction Technology.

REFERENCES

[1] C. Soto and M. Carlsson, "Object Interaction Query: A context awareness tool for evaluating BIM components' interactions." bimForum, projects.buildingsmartalliance.org/files/?artifact_id=5372, 2013.

[2] M. Nepal, "Automated extraction and querying of construction-specific design features from a building information model," doctoral thesis, University of British Columbia, 2011.

[3] S. Daum and A. Borrmann, "Checking spatio-semantic consistency of building information models by means of a query language", Thirteenth International Conference on Construction Applications of Virtual Reality, 2013, pp. 492–501.

[4] A. Borrmann, "From GIS to BIM and back again—a spatial query language for 3D buildings and 3D city models," Fifth International 3D GeoInfo Conference, 2010, pp. 19–26.

[5] M. Nepal, S. Staub-French, R. Pottinger, and A. Webster, "Querying a building information model for construction-specific spatial information," Advanced Engineering Informatics, vol. 26, pp. 904–923, 2012.

[6] M. N. K. Boulos, W. Jeffrey, G. Jianya, and Y. Peng, "Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping," International Journal of Health Geographics, vol. 9, no. 1, pp. 1–13, 2010.

[7] T. S. Abson and E. Olivier, "Towards web services dedicated to thematic mapping," OSGeo Journal, vol. 3, no. 1, 2007.

[8] buildingSMART alliance, www.buildingsmart.org/, [retrieved April 2014]

[9] CityGML, www.citygml.org/, [retrieved April 2014]

[10] Industry Foundation Classes, buildingsmart-tech.org/ifc/IFC2x4/rc3/html/index.htm, [retrieved April 2014]

[11] Autodesk Revit, http://en.wikipedia.org/wiki/Autodesk_Revit, [retrieved April 2014]

[12] ArchiCAD, http://en.wikipedia.org/wiki/ArchiCAD, [retrieved April 2014]

 ISBN: 978-1-61208-363-6