International Journal on Advanced Science Engineering Information Technology

Vol.8 (2018) No. 6
ISSN: 2088-5334

# Consistency Check between XML Schema and Class Diagram for Document Versioning

Rosziati Ibrahim[#], Hannani Aman[#], Richi Nayak[*], Sapiee Jamel[#],

*# Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, 86400, Malaysia*
*E-mail: rosziati@uthm.edu.my, hannani@uthm.edu.my, sapiee@uthm.edu.my*

**School of Electrical Engineering and Computer Science, Queensland University of Technology (QUT), Brisbane, QLD 4000, Australia*
*E-mail: r.nayak@qut.edu.my*

***Abstract*—A consistency check between design and implementation is usually done in order to check the correctness of the system' requirements. However, if the requirements are changed over time, then the document versioning occurred within the requirements. For XML Schema, document versioning exists when there is a change in the XML Schema from its previous Schema. In order to detect the versioning of both XML Schemas, consistency rules check need to be performed to both class diagrams produced by both Schemas. The consistency between two XML Schemas is checked based on transformation rules and versioning rules. Transformation rules are used for translating the XML Schema into the class diagram, and versioning rules are used for checking the existing of document changes between two XML Schemas. Once two XML Schemas are different the consistency rules will be used for the consistency check. This paper presents an approach based on transformation rules and versioning rules to check consistency between XML Schema and UML class diagram when document versioning exist. The approach is then used for the case study to show how the consistency is checked in order to detect the versioning of two different XML Schemas. Based on the case study, the approach shows that two XML Schemas can be checked for their consistency when document versioning exist.**

***Keywords*— consistency rules; XML schema; UML class diagram; document versioning.**

## I. INTRODUCTION

In software development life cycle (SDLC), four main phases are involved during the development of any system. They are analysis, design, implement, and testing. During analysis and design, UML specification is usually used to represent the system's requirements. Diagrams usually represent UML specification. Eleven main diagrams are used for UML specification [1]. One of the important diagrams in the UML specification is a class diagram. A class consists of a name of the class, its attributes for describing the class' data and the methods for representing the behavior of the class. A class diagram consists of a minimum of two classes that are usually connected by association.

XML Schema is usually used for developing web applications. XML Schema consists of rules that define the constraint of each element in the XML document. It provides a mean for defining the structure, content, and semantics of XML document in more detail [2]. XML Schema consists of 5 components as described in [3]. A UML class diagram can be formed based on the XML Schema in order to extract the requirements of the system. This class diagram contains the attributes and methods extracted from the XML Schema.

Document versioning exists when XML Schema has been changed over time due to changes of requirements. Once the versioning exists, the consistency rules check has to be performed between the two XML Schemas.

Consistency rules check is usually used for checking consistency between the schema and its implementation using the predefined rules. Chavez *et al.* [4], for example, introduced an approach to check consistency between the UML class model with its Java implementation. Ibrahim et al. [5], on the other hand, proposed a formal model for consistency rules between the UML use case diagram and activity diagram. This paper proposes an approach to check consistency between XML Schema and UML class diagram when document versioning occurred in XML Schema. The consistency check is performed based on several rules extracted for the consistency between the two XML Schemas.

## II. Material and Method

XML is used for the web applications. Any changes to the web application will reflect the changes in the XML schema. XML commercial tools offer editing XML document and schemas, for example, XSPath [6]. However, when there is a slight change in XML schema or document, the changes will be saved automatically once the document of XML is saved. For this reason, changes in the document cannot be traced. In order to manage the changes in the document, we need to keep track of the changes. This is the motivation for research for keeping track of the changes in the document and managing the changes in the document. Therefore, an approach to check consistency for document versioning between XML schema and UML class diagram using consistency rules (CR) is proposed using Fig. 1.

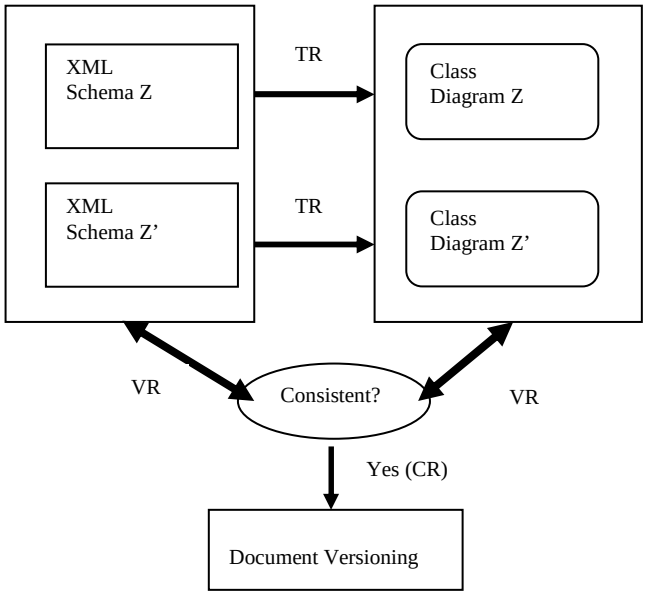


Fig. 1 Consistency Rules (CR) Check for Document Versioning

The transformation rules (TR) illustrated in Fig. 1 is used for extracting the information needed from the XML Schema. This information is used to translate the XML Schema into class diagrams as in [7]. The versioning rules (VR) are used to check the consistency between two XML schemas and two UML class diagrams as in [8]. Fig. 2 shows when the transformation rules and versioning rules are used.

From Fig. 1, three phases are involved. The first phase is schema parser, the second phase is reverse transformation using the transformation rules (TR), and the third phase is the versioning using the versioning rules (VR). For the first phase, schema parser is used to read the XML Schema in order to extract relevant information such as data structure and XML elements. In the second phase, the transformation rules are used for the extracted information from phase one in order to transform them to class diagram using the data structure and XML elements as described in [7]. The class diagram is used in order to capture the important information regarding the XML Schema. Any changes in XML Schema will be captured in a class diagram. These changes can be used for the detection of the XML versioning. In a situation when for the two XML Schemas of the same scenario are presented, two class diagrams will be produced. Both of these two class diagrams will be compared for checking the consistency using the versioning rules of the XML Schema.

In XML documents, changes occurred must be understood and documented. The issue of changes in XML documents is addressed by using the versioning of the XML documents. If XML document changes, it implies that the versioning occurs. Therefore, to address the versioning issue, two class diagrams are used from XML schemas for versioning purposes. The earlier XML Schema is used to generate a class diagram (class diagram Z). When maintenance process occurs, new XML schema is introduced. This new XML schema has been modified because of the number of XML schema components have been changed. The changes are visible in syntax. In order to capture the semantic changes of the new schema, the new schema is used for the transformation of the second class diagram, which is called class diagram Z'. The same transformation is used to generate the class diagram Z. The changes occurred in class diagram Z are captured and a new class diagram called class diagram Z' are produced. Both class diagrams (Z and Z') are then compared against each other in order to capture any additional class diagram elements and others.

For the third phase, both class diagrams (Z and Z') are compared and analyzed for the consistency in order to detect the versioning of the XML Schema. The analysis of the consistency between two diagrams are used to highlight the changes and are used for the traceability factor. The traceability factor is used for generating the software requirements documents (SRD) of the latest XML Schema.

In forward engineering, the transformation rules are used to generate UML class diagram from XML Schema and the versioning rules are used to check the document versioning between two XML Schema for their UML class diagram. Both transformation rules and versioning rules have been discussed in [7] and [8] respectively. The formalization of the transformation rules are presented in [7] and the formalization of the versioning rules are presented in [8].

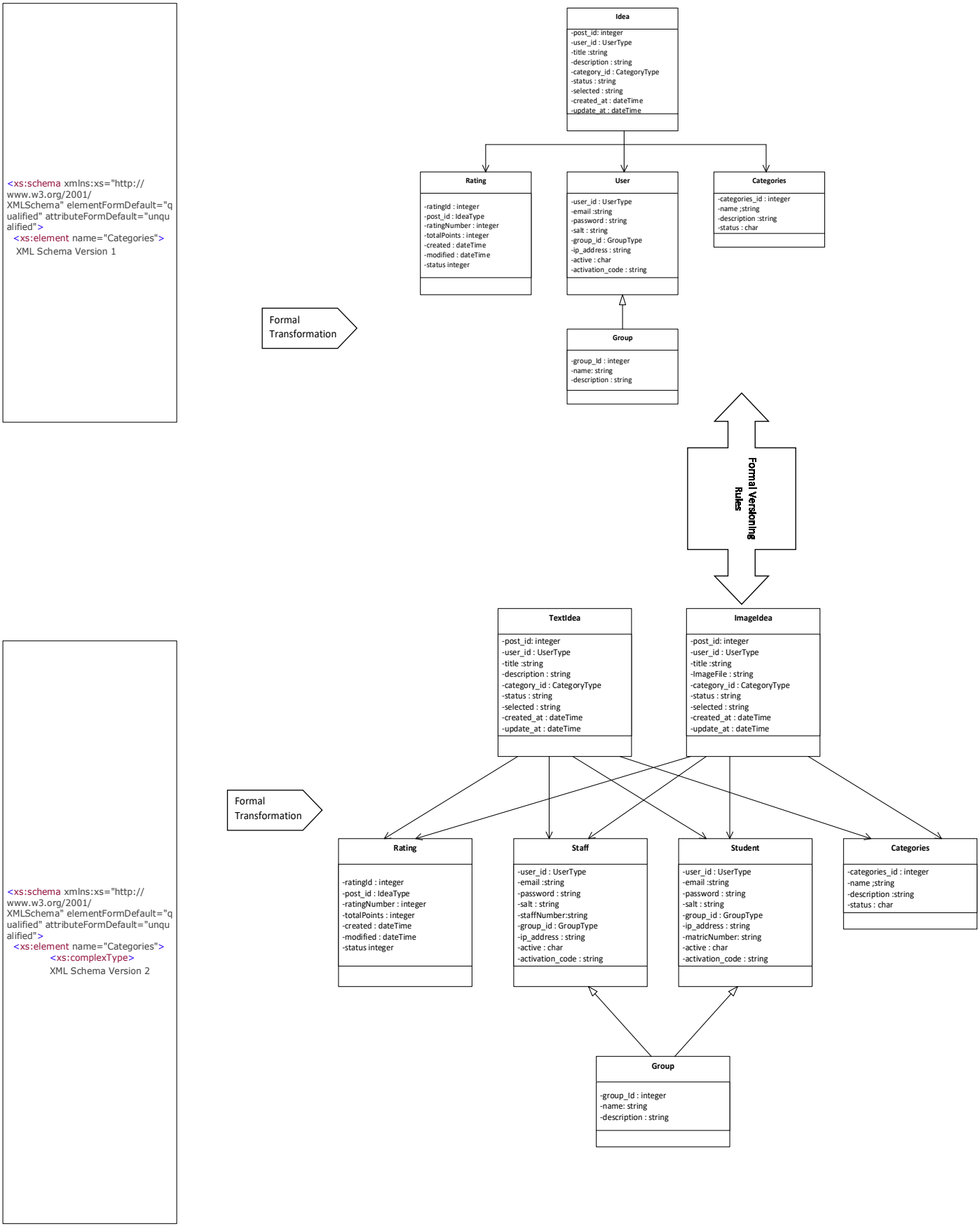


Fig. 2 Transformation rules and Versioning Rules for the Consistency Rules (CR)

Based on transformation rules from Fig. 1, the XML Schema is used to transform into a class diagram. Both class diagrams that have been transformed from the same XML schema when versioning exist will be checked for the consistency using versioning rules. The consistency check can be done in order to validate that the document versioning occurred within the two XML Schemas. Table 1 lists the consistency rules for the document versioning for XML Schema using UML class diagram.

TABLE I
CONSISTENCY RULES FOR DOCUMENT VERSIONING

| **Consistency Rule (CR)** | **Details** |
|---|---|
| CR1 | Each class from both class diagrams has been compared and checked (level by level of the diagrams for the consistency of type definitions, element declaration, attribute declaration, attributes group definitions and model group definition) |
| CR2 | Based from both class diagrams, each added classes and removed classes are compared and checked for the consistency |
| CR3 | Based from both class diagrams, classes which have the same attribute and same relationship are consistently checked for similarity in order to detect the new version of class diagram. |

Based on Table 1, three consistency rules are used in order to detect the versioning of two XML schemas exist or not. Both XML schemas are compared first using the first consistency rules. Then the second consistency rules check on the possibility of any classes that has been added or removed from the previous class diagram. Finally, the third consistency rules detect on the existent of the versioning from two class diagrams that have been compared.

Versioning occurred when there exist changes in one XML schema from the preceding XML schema. The preceding XML schema is used to generate the above class diagram. When the maintenance process occurs, new XML schema is introduced. This new XML schema has been modified from the preceding XML schema. Therefore, the new XML schema is used to generate the new class diagram. The new class diagram is compared with the preceding class diagram for detection of document versioning.

Document versioning is important to keep track of the changes in XML Schema. Tan and Goh in [9] described the XML specifications for changes in XML Schema. The importance of keeping track for the changes in XML documents and schemas are also discussed in [10]. Cavalieri [11] introduced a tool for keeping track of the changes from XML Schema and its documents. Brahmia *et al.* [12], on the other hand, proposed an approach for detecting changes in XML Schema for fundamental elements.

## III. RESULTS AND DISCUSSION

This section discusses a case study on a house of the ideas web page from [13]. The house of ideas represents a web page that can share a new idea and browse ideas that have been stored in the database. Based on the case study, the transformation rules and versioning rules are applied to the case study and the final results are shown in Fig. 2. The XML schema is extracted from [13]. The extracted XML schema is shown in Fig. 3. From Fig. 3, the XML schema consists of 5 global elements, 20 local elements and 10 attributes as described in Table 2.

```
<xs:schema xmlns:xs="http://
www.w3.org/2001/
XMLSchema" elementFormDefault="q
ualified" attributeFormDefault="unqu
alified">
 <...element name="Categories">

 < ...element  name="Idea">

  < ...element  name="Rating">

  < ...element  name="User">

  < .. element name="Group">
        ...
 </xs:schema>
```

Fig. 3 Extracted XML Schema version 1 [13]

TABLE II
ELEMENTS AND ATTRIBUTES [13]

| **Global Element** | **Categories, Idea, Rating, User, Group** |
|---|---|
| Local Element | created_at, selected, title, updated_at, created, modified, rating_number, total_points, activation_code, active, created_on, email, forgotten_password, ip_address, last_login, nomatrik, password, remember_code, salt, username |
| Attributes | categories_id,groups_id, ideas_id, user_id, rating_id, post_id, description, status, username, name |

From the extracted schema in Fig. 3 and the elements and attributes in Table 2, the class diagram is developed using the transformation rules. The developed class diagram is shown in Fig. 4. The actual XML Schema can be downloaded from [13[]. The global elements from Table 2 are transformed into classes in Fig. 4. Over time, during the maintenance phase, the XML schema has been changed, and a new class diagram is developed and shown in Fig. 5. Based on Fig. 4 and Fig. 5, both class diagrams are compared using the consistency rules in Table 1 to check whether the document versioning occurs or not.

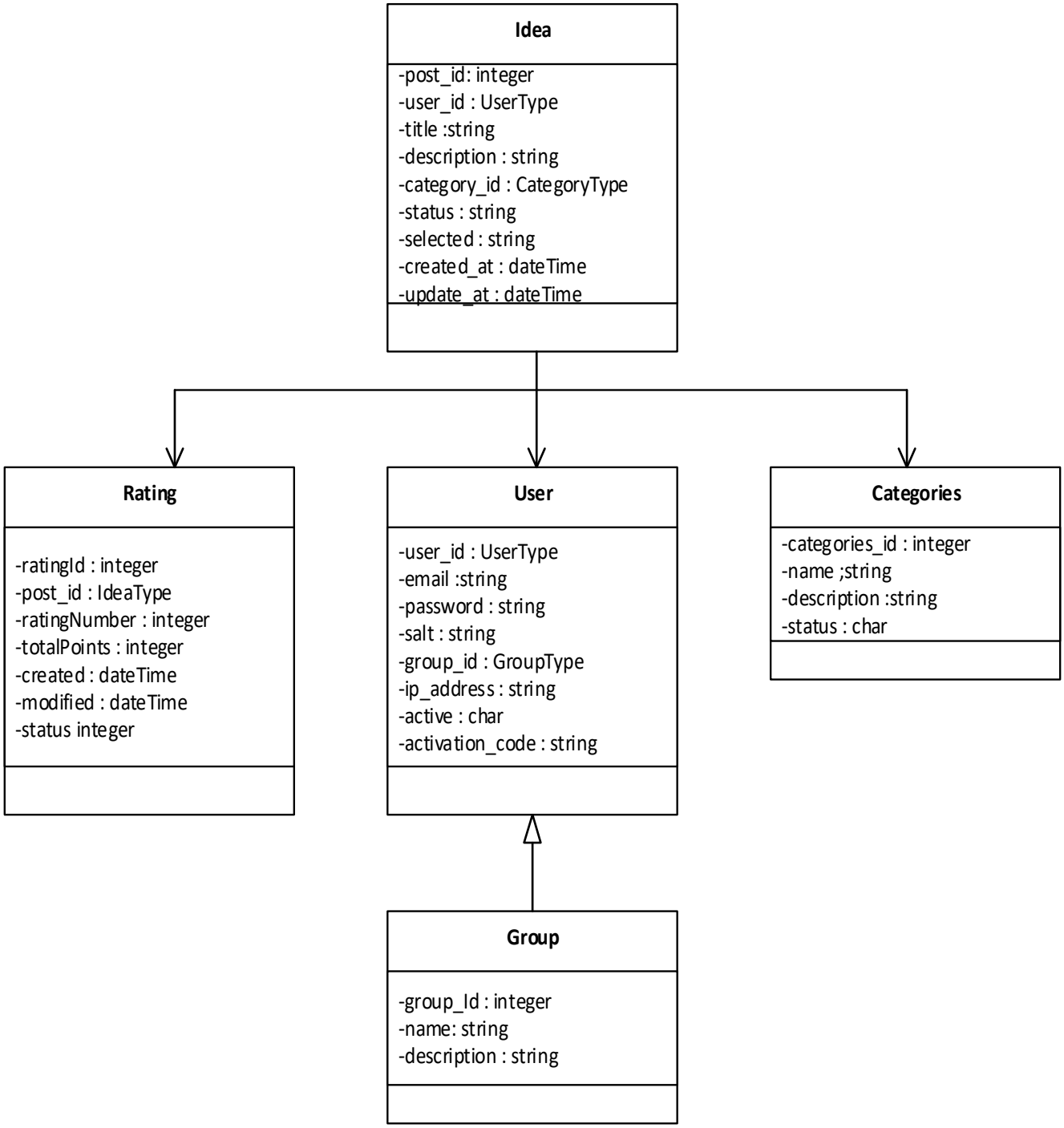


Fig. 4 A class diagram is generated from an earlier version in [13]

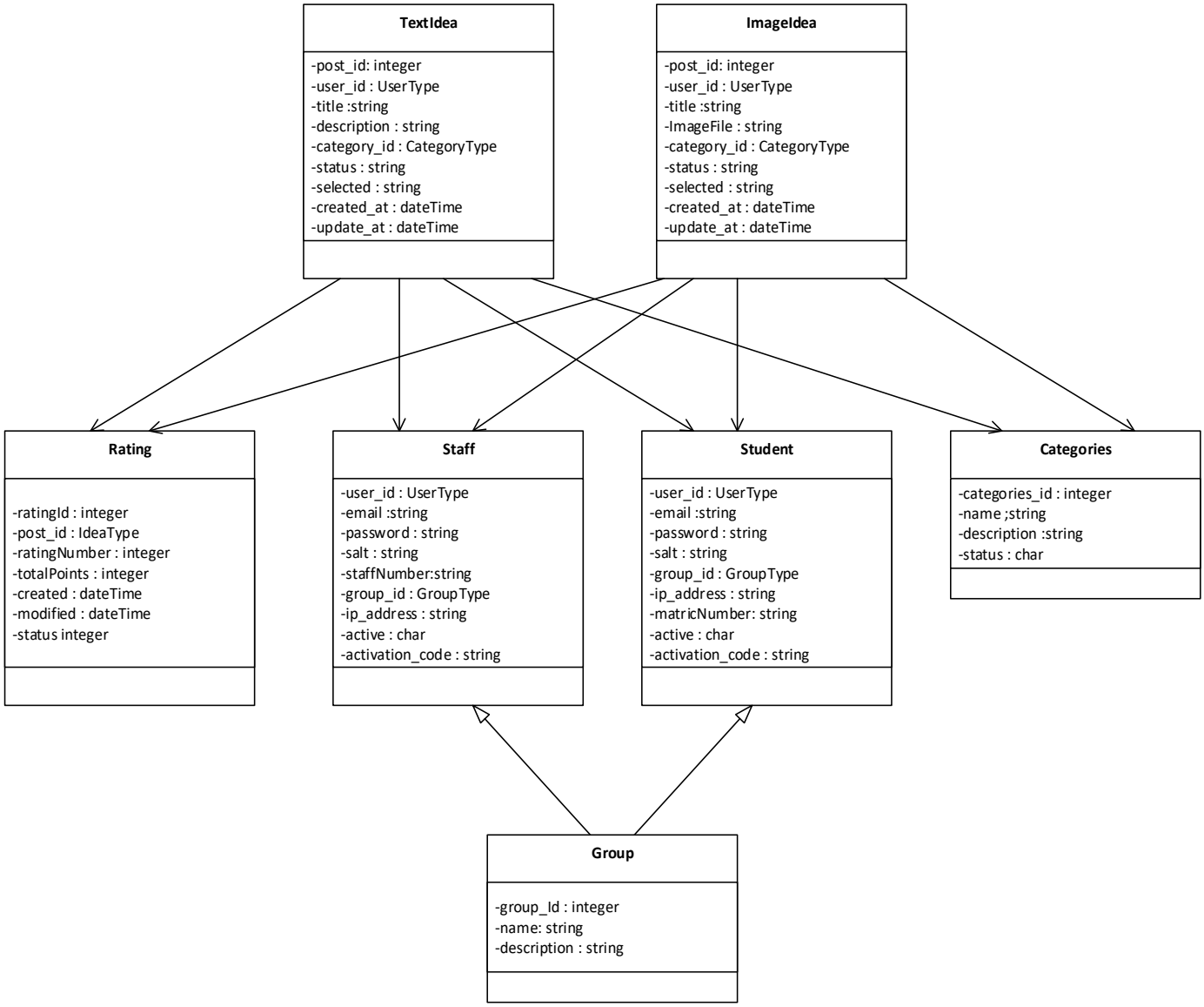


Fig. 5 A new class diagram is generated for the newer version in [13]

From Fig. 3, the class diagram is generated based on the XML Schema using the transformation rules. Schema parser is used to extract every class from the XML Schema including the attributes and methods. Then the transformation rules are used to generate the class diagram. The generated class diagram from XML Schema is shown in Fig. 4. The house of ideas system has gone into a second cycle, and updated XML Schema has been produced. This updated XML Schema is once again used to generate a new class diagram. Using the transformation rules and versioning rules, a new class diagram is generated and shown in Fig. 5.

From the generated class diagram in Fig. 4, two main classes are used for the house of ideas system. They are class *Idea* and class *User*. However, when a new class diagram is generated as in Fig. 5, these two classes do not exist anymore. New classes have been detected in the new class diagram. Table 3 shows the added classes, and removed classes are detected based on CR2 from Table 1.

TABLE III
DIFFERENCE CLASSES DETECTED

| Class Removed | Class Added |
|---|---|
| Ideas | ImageIdeas |
| | TextIdeas |
| User | Student |
| | Staff |

Based on Table 3 and using the consistency rules in Table 1, attributes in each removed class are compared with the attributes in class added. If the attributes remain more than 1 in the new classes added, it means that the new schema has a new version from the old schema. Class *Ideas* is evolved to a new version of *ImageIdeas* and *TextIdeas.* Fig. 6 shows class *TextIdea* has the same attributes as class *Idea.* While class *ImageIdea* has nearly all attributes as class *Idea*. The relationship of class *TextIdea* and *ImageIdea* remains the same as the relationship with class *Idea*. This shows that classes *TextIdea* and *ImageIdea* have evolved from class Idea.

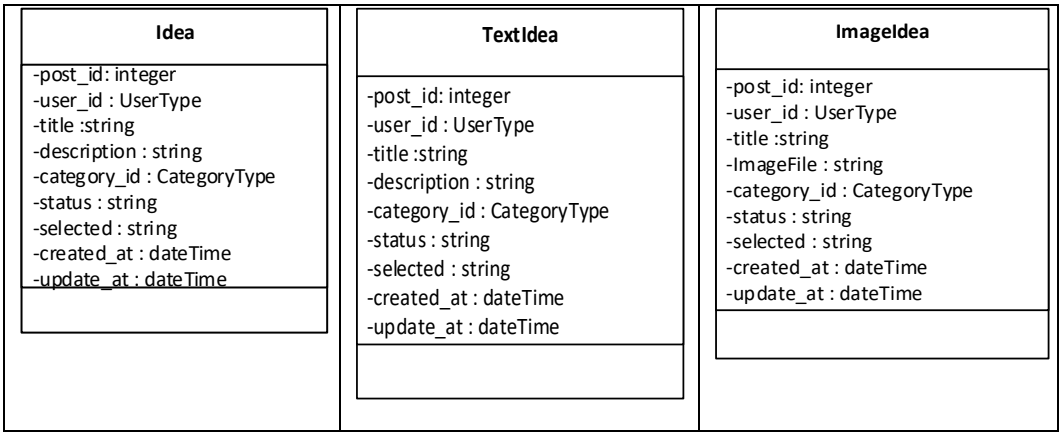


Fig 6. Class Idea has evolved.

The second class is examined with the same procedure. The CR2 in Table 1 stated that for both class diagrams, each added classes and removed classes are compared and check for the consistency. Based on Fig. 4, a class *User* has been used for the house of ideas system. However, this class *User* does not exist anymore in Fig. 5. Instead, a new two classes are introduced which are classes *Student* and *Staff*. Both classes have nearly the same attributes and relationship as *User* class as shown in Fig 7.

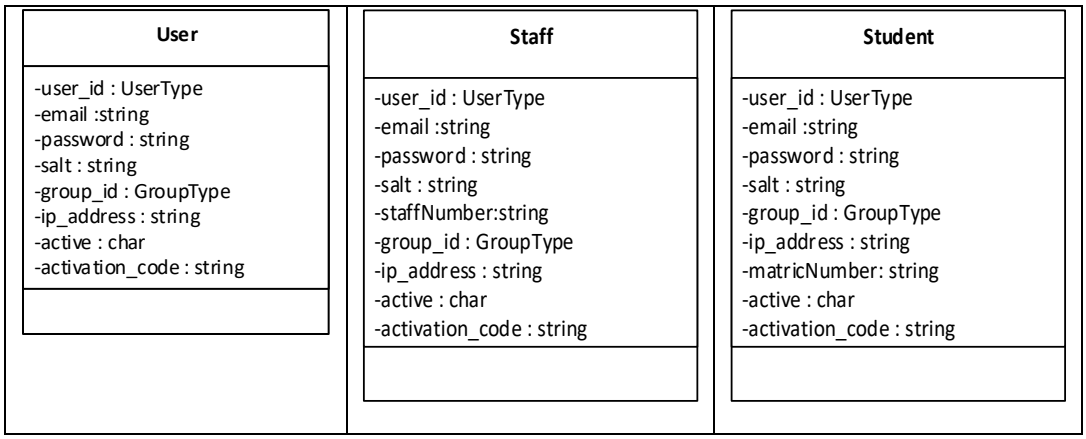


Fig 7. Class User has evolved

Based on Figure 7, the attributes for class *Staff* are the same as the attributes for class *User* and the attributes of class *Student* have almost the same attributes for class *User* with one extra attribute *matric number*. Based on CR3 from Table 1, both class diagrams are compared and checked in order to detect a new version of the class diagram. The consistency check using the rules from Table 1 has been done for both class diagrams in Fig. 4 and Fig. 5. Based on the three consistency rules, it shows that the new version schema has been consistently checking with the old schema for document versioning.

## IV. Conclusion

This paper has presented an approach for consistency check between two XML schemas for the document versioning. The class diagram is used as a mechanism for the document versioning. The class diagram is developed based on the XML Schema using Schema parser, transformation rules, and versioning rules. Then two class diagrams are compared using consistency rules for the checking of the existing of document versioning.

## Acknowledgment

The authors would like to thanks Malaysian Ministry of Higher Education (MoHE) and Universiti Tun Hussein Onn Malaysia (UTHM) for supporting this research under the Fundamental Research Grant Schema (FRGS) and E15501, Research Management Centre (RMC).

## References

[1] UML2.5.1 (2018). OMG UML Specification. [Online]. Available: http://www.omg.org/spec/UML/2.5.1/

[2] W3C Primer (2004). XML Schema Part 0: Primer 2nd Edition. [Online]. Available at: http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/

[3] W3C Structure (2004). XML Schema Part 1: Structures 2nd Edition. [Online]. Available at: http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[4] H. M. Chavez, W. Shen, R. B. France, B. A. Mechling, and G. Li. (2016). "An Approach to Checking Consistency between UML Class Model and Its Java Implementation," *IEEE Transactions on Software Engineering*, vol. 42, no. 4, pp. 322-344, April 2016.

[5] Ibrahim, N., Ibrahim, R., Saringat, M.Z., Mansor, D and Herawan, T. (2011). "Consistency rules between UML use case and activity

diagrams using a logical approach." *International Journal of Software Engineering and its Applications*, 5 (3), pp. 119-134, 2011.

[6] Cavalieri, F., Guerrini, G., and Mesiti, M. (2014). " XSPath: Navigation on XML Schemas Made Easy." IEE Transactions on Knowledge and Data Engineering, Vol 26, No. 2, pp. 485-499, February 2014.

[7] Aman, H., Ibrahim, R. (2014). "Formalization of Transformation Rules from XML Schema to UML Class Diagram." *International Journal of Software Engineering and Its Application*, *8*(12), pp.75-90, 2014.

[8] Aman, H., Ibrahim, R. (2017). "Formalization of versioning rules for XML schema using UML class diagram." Journal of Theoretical and Applied Information Technology, 95 (15), pp. 3652-3661, 2017.

[9] Tan, M., & Goh, A. (2004). Keeping Pace with Evolving XML-Based Specifications, 280–288.

[10] Cavalieri, F., Guerrini, G., and Mesiti, M. (2011). "Updates on XML documents and schemas." *2011 IEEE 27th International Conference on Data Engineering Workshops*, pp. 308–311, 2011.

[11] Cavalieri, F. (2010). "E X up: An Engine for the Evolution of XML Schemas and Associated Documents." *In EDBT Workshop Proceedings*. 2010.

[12] Brahmia, Z., Grandi, F., Oliboni, B., & Bouaziz, R. (2012). "Versioning of Conventional Schema in the tXSchema Framework." *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*, pp. 510–518, 2012.

[13] IDEAS. (2018). House of Idea system. [Online]. Available: http://ideas.uthm.edu.my/