International Journal of Innovation and Sustainability. 3 (2020); 47 - 51
ISSN Online: 0184-7937

The Semi-structured Data Model and Implementation Issues for Semi-structured Data

Dickson, Makasda Solomon* and Prof. P. O. Asagba

*Department of Information and Communications Technology, National Institute for Nigeria Languages, Aba, Abia State, Nigeria.
dicksonsolomon1988@gmail.com;
Department of Computer Science, Ignatus Ajuru University of Education, Rumuolumeni, Port Harcourt, River State, Nigeria.

Abstract
Semi-structured Data are becoming extremely popular in versatile applications including interactive web application, protein structure analysis, 3D object representation, personal lifetime information management. In order to meet the challenges of today's complex applications, a generic model is in demand. This paper therefore focuses to examine the Semi-structured Data Model and implementation issues for Semi-structured Data. The paper assumes that: fluidity in data structure makes it difficult to store and manage the semi structured data using conventional data models like Relational Database model; the main advantage of fully structured data is the strong typing which enables high performance and efficiency; unstructured and semi structured data allow a higher degree of flexibility; Graph based models (e.g OEM) can be used to index semi-structured data; data modeling technique in OEM allows the data to be stored in graph based model; the data in graph based model is easier to search/ index; and finally, XML allows data to be arranged in hierarchical order which enables the data to be indexed and searched as well.

Introduction

In semi-structured data, the information that is normally associated with a schema is contained within the data, which is sometimes called "self-describing". In some forms of semi-structured data there is no separate schema, in others it exists but only places loose constraints on the data. Semi-structured data has recently emerged as an important topic of study for a variety of reasons.

Citation: Dickson M. S. and Asagba, P. O. (2020). The Semi-structured Data Model and Implementation Issues for Semi-structured Data. *International Journal of Innovation and Sustainability*, 3: 47 – 51.

First, there are data sources such as the Web, which would like to treat as databases but which cannot be constrained by a schema. Second, it may be desirable to have an extremely flexible format for data exchange between disparate databases. Third, even when dealing with structured data, it may be helpful to view it as semi-structured for the purposes of browsing.

Semi structured means the schema and data structure cannot be easily predefined, and conventional database techniques have to be modified to manage these semi structured data, i.e., by adding flexibility in schema definition and loosening type checking. Furthermore, for some database applications, although the data is initially well structured, the structure evolves rapidly when new data gets integrated into the database.

Rolf, Unwald, and Stroka (2008) define Semi-structured data refers to the fact that no identifiable structure within this kind of data is available. Semi-structured data is also described as data that cannot be stored in rows and columns in a relational database. Storing data in a Semi-structured form without any defined data schema is a common way of filing information. An example for semi-structured data is a document that is archived in a file folder. Other examples are videos and images.

Semi-structured data is the data which does not conforms to a data model but has some structure. It lacks a fixed or rigid schema. It is the data that does not reside in a rational database but that have some organizational properties that make it easier to analyses. With some process, it can store them in the relational database. (McHugh, Abiteboul, Roy, Quass, and Widom, 2012). This

paper therefore focuses to examine the Semi-structured Data Model and Implementation Issues for Semi-structured Data.

The age of the Internet brought new data management applications and challenges. Data is now accessed over the Web, and is available in a variety of formats, including HTML (Hyper Text Markup Language), XML (Extensible Markup Language), as well as several applications for specific data formats. Often data is mixed with free text, and the boundary between data and text is sometimes blurred. The way the data can be retrieved also varies considerably: some instances can be downloaded entirely; others can only be accessed through limited capabilities. To accommodate all forms and kinds of data, the database research community has introduced the ”semi-structured data model”, where data is self-describing, irregular, and graph-like. The new model captures naturally Web data, such as HTML, XML, or other application specific formats.

According to Bernard (2019).Beyond structured and unstructured data, there is a third category, which basically is a mix between both of them. The type of data defined as semi-structured data has some defining or consistent characteristics but doesn’t conform to a structure as rigid as is expected with a relational database. Therefore, there are some organizational properties such as semantic tags or metadata to make it easier to organize, but there’s still fluidity in the data. Email messages are a good example. While the actual content is unstructured, it does contain structured data such as name and email address of sender and recipient, time sent, etc. Another example is a digital photograph. The image itself is unstructured, but if the photo was taken on a smart phone, for example, it would be date and time stamped; geo tagged, and would have a device ID. Once stored, the photo could also be given tags that would provide a structure, such as ‘dog’ or ‘pet.’

A lot of what people would usually classify as unstructured data is indeed semi-structured, because it contains some classifying characteristics.

Semi-structured data can contain both the forms of data. Semi-structured data as a structured in form but it is actually not defined with e.g. a table definition in relational DBMS (Data Base Management System). Example of semi-structured data is a data represented in an XML file.

Conceptualizing Semi-Structured Data

Semi-structured data is often explained as "schemaless or self-describing, terms that indicate that there is no separate description of the type or structure of the data" Semi-structured data does not require a schema definition. This does not mean that the definition of a schema is not possible, it is rather optional. The instances do also exist in the case that the schema changes. Furthermore, a schema can also be defined according to already existing instances (posteriori). The types of semi-structured data instances may be defined for a part of the data and it is also possible that a data instance has more than one type.

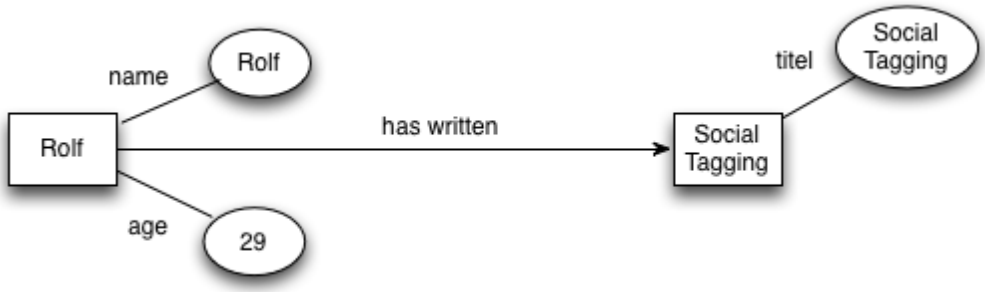

Figure 1: Sample Resource Description Framework (RDF) Graph**.**

One of the strengths of semi-structured data is “the ability to accommodate variations in structure. This means that data may be created according to a specification or close to a type. For instance, fields can be duplicated, data can be lacking or there may exist minor changes. Figure 1illustrates a graph representation of semi-structured data (Papakonstantinou, Garcia-Molina, and Wisdom, 2010).

Characteristics of semi-structured Data:

Data does not conform to a data model but has some structure, data cannot be stored in the form of rows and columns as in databases, Semi-structured data contains tags and elements (Metadata) which is used to group data and describe how the data is stored, Similar entities are grouped together and organized in a hierarchy, entities in the same group may or may not have the same attributes or properties, does not contains sufficient metadata which makes automation and management of data difficult, size and type of the same attributes in a group may differ, and due to lack of a well-defined structure, it cannot used by computer programs easily

Sources of semi-structured Data:

Semi-structured data get created. A few instances of semi-structured data sources are email, XML and other markup languages, binary executable, TCP (Transmission Control Protocol) /IP (Internet Protocol) packets, zipped files, integration of data from different sources and web pages. The growing volume of semi-structured data is partly due to the growing presence of the web, as the need for flexible formats for data exchange between disparate databases. In addition, certain scientific databases that require a broader mix of structural and text data, along with annotations and attribute extensibility, also create this kind of data. Where application data does not have a rigidly and predefined schema, semi-structured data is created. The schema may be descriptive, partial, evolving, and very large. Web pages (Treehousetechgroup, 2019).

The Nature of Semi-Structured Data

The semi-structured data have a typical nature. It is organized into semantic entities and similar entities are grouped together. It is not necessary that entities in the same group have the same attributes within a group any differ. There are many different ways to extract information or data from semi-structured data. Graph – based models, or object exchange models (OEM), can be used to index the data. OEM (Object Exchange Model) data modeling techniques enable the data to be stored in graph-based models that are easier to search and index. Another option is XML, which allows hierarchies to be created and facilitates index and search.XML allows data to be arranged in hierarchical order which enables the data to be indexed and searched Use of various data mining tools (Chaki and Cortesi, 2011). Data mining tools are also used to extract information from semi-structured data.

Challenges of Handling Semi-Structured Data

While semi-structured data increases flexibility, the Lack of fixed, rigid schema make it difficult in storage of the data also creates storage and indexing challenges. The schema and data are tightly coupled and inter dependent and a query update both. It is also challenging to run queries. OEM and XML formats help to store and exchange semi-structured data, and can overcome some of these challenges.

The volume of semi-structured data continues to grow; new ways to manage, collate, integrate, store and analyze it will evolve. Semi-structured data can help to capture and process data as it really is, without forcing it into an unnatural structure. Knowledge about the nature of semi-structured data and ways to use it is extremely important considering, interpreting the relationship between data is difficult as there is no separation of the schema and the data and queries are less efficient as compared to structured data.

Implementation issues for semi-structured data

An older technology is another possible solution to the semi-structured data problem. After XML was proposed by the W3C in 1998, the academic work on semi-structured data was almost halted, as there was hope that XML was the answer. Almost a decade later, XML is universally accepted and embraced by a variety of communities for many reasons, yet it is now clear that while XML solves a large number of schema-related challenges, it does not solve the general problem of semi-structured information. (In this article, XML refers to the entire standard XML infrastructure developed by W3C, including a set of technologies and languages designed in a consistent fashion: abstract data models, such as Infoset, XQuery 1.0, and XSLT 2.0; XML Schema; and the declarative processing languages such as XPath, XQuery, and XSLT.)

XML offers some major advantages; as a standard syntax for information, XML is able to model the entire range of information, from totally structured data (e.g., bank account information) to natural text. Having a single model for the entire spectrum of information has tremendous benefits for modeling, storage, indexing, and automatic processing of such information. There is no need to

switch from system to system and make inconsistent systems communicate with each other while we increase or decrease the level of structure in information.

Another major advantage of XML is the ability to model mixed content. Having an abstract information model that goes beyond the entity-relationship model opens the door to a large volume of information that was impossible to model with prior techniques. The fact that XML schemas are decoupled from data is also essential for data and schema evolution; data can exist with or without schemas, or with multiple schemas. Schemas can be added after the data has been generated; the data and schema generation can be freely interleaved.

While providing significant advantages for managing semi-structured information, XML-based technologies in their current form and in isolation are not the magic bullet. Most information is still not in XML form, and some of it never will be. The advantages of XML (e.g., complex schemas, mixed context, schema-independent data) unsurprisingly bring an extra level of complexity and many challenges. Finally, XML-related technologies today don't offer a complete solution. For example, while XSLT (Extensible Stryllesjeet Language) and XQuery provide good query and transformation (i.e., read-only) languages, there is still no good way of expressing imperative logic over such schema-flexible data, or language to describe complex integrity constraints and assertions. Such limitations will eventually be eliminated, but not immediately.

A solution to the general problem of semi-structured information will need to draw ideas and techniques from many fields, including knowledge representation, XML and markup documents, information retrieval, the Semantic Web, and traditional data management techniques. No single method will provide the answer to this problem. Search engines will improve solving one aspect of the semi-structured data problem: the human consumption of information. Contextual, semantic, and structural information can be exploited to increase the relevance of results of simple textual queries.

Decoupling data from schemas also has a large impact on all the aspects of data processing: storage, indexing, querying and updating, providing transactional support, and so on. Most current techniques rely on static schema information to achieve performance for such tasks. We need to revisit such techniques to guarantee their correctness and performance, even in the absence of schema information or with constantly evolving schema information (Daniela, 2005).

Data usually has an irregular and partial structure. Some sources have implicit structure of data, which makes it difficult to interpret the relationship between data. Schema and data are usually tightly coupled i.e. they are not only linked together but are also dependent of each other. Same query may update both schema and data with the schema being updated frequently. Distinction between schema and data is very uncertain or unclear. This complicates the designing of structure of data Storage cost is high as compared to structured data. (Dobbie, Xiaoying, Ling, and Lee. 2009).

How to use semi-structured data

The use of semi-structured data assists to integrate data from various source or exchange data between different systems. Applications and systems need to evolve with time, but of work purely with structured data. Its web forms helps to modify forms and capture different data for different users. Using a traditional relational database, the database schema needs to be changed each time a new field is needed, and fields cannot be left empty. Semi-structured data can allow you to capture any data in any structure without making changes to the database schema or coding. Adding or removing data does not impact functionality or dependencies.

When you work with semi-structured data, you get a flexible representation, and you do not need configuration or code changes if the data evolves over time. Data from multiple sources with differences in notation and meaning can be collected and used. Relationships are described as references and are incorporated completely into parent object (tree). Semi-structured data makes it possible to maintain and support complex query types of data structure and storage, while keeping the relationships between objects and complex schema. Queries and reporting over many systems and data types are now possible, Data can be stored in DBMS specially designed to store semi-structured data, XML is widely used to store and exchange semi-structured data. It allows its user to define tags and attributes to store the data in

hierarchical form, Schema and Data are not tightly coupled in XML, Object Exchange Model (OEM) can be used to store and exchange semi-structured data. OEM structures data in form of graph, while RDBMS (Relational Database Management System) can be used to store the data by mapping the data to relational schema and then mapping it to a table.

Conclusion

The main advantage of fully structured data is the strong typing which enables high performance and efficiency. On the other hand, unstructured and semi structured data allow a higher degree of flexibility. Logically and conceptually, many attempts were made to model SSD (Semi Structured Data). However, a widely accepted generic model for SSD is yet in demand. This extensive survey aims to analyze the state of art works on semi-structured data representation to identify the expected features of such a generic model. This is going to be the basic objective set towards building generic SSD model. The identified characteristics of SSD formally define access structures including artificial intelligence and linguistic support for the next generation database language as well as uniform view of Structured Data, Semi-Structured Data and Unstructured Data SD, SSD and It is difficult to describe the structure of semi-structured data because the data is from heterogeneous sources and may not have static form. Lack of knowledge about information content and complicated relationship between data elements also add troubles when users try to predefine a fixed schema, as they do in traditional relational databases. Enabling semi-structured information processing that is flexible, cheap, simple, and effective is an important goal.

References

Bernard, M. (2019). https://www.forbes.com/sites/bernardmarr/2019/10/18/whats-the-difference-between-structured-semi-structured-and-unstructured-data/#5c8f49ee2b4d. (Retrieved on August 28th 2020)

Chaki, N. and Cortesi, A. (2011). *A Survey on the Semi-Structured Data Models.* Berlin Heidelberg: Springer-Verlag. Pp. 257–266.

Dobbie, G., Xiaoying, W., Ling, W. T., Lee, L. M. (2009). An object-Relationship Attribute Model for Semi-Structured Data. Technical Report, School of Computing, Singapore. Pp.7-9.

Dobbie, G., Xiaoying, W., Ling, W.T., Lee, L. M. (2009). Designing Semi-structured Database using ORA-SS Model. In: 2nd International Conference on Web Information Systems Engineering, vol. 1, p. 171. IEEE.

Daniela F. O. (2005). Semi structured? https://queue.acm.org/detail.cfm?id=1103832. (Retrieved on July, 6th 2020).

McHugh, J., Abiteboul, S., Roy, G., Quass, D., Widom, J. and Lore, E. (2012). A Data base Management System for Semi-structured Data. SIGMOD Record. Pp. 54-66.

Papakonstantinou, Y., Garcia-Molina, H. and Wisdom, J. (2010). Object Exchange Across Heterogeneous Information Sources. 11th International Conference on Data Engineering, Taiwan, pp. 251–260.

Treehouse Techgroup (2019). https://treehousetechgroup.com/what-is-semi-structured-data-5-key-things-to-know/ (Retrieved on September 6th 2020).