

Evaluations of Conceptual Models for Semi-structured Database System

Rita Ganguly¹

¹Department of Computer Applications, Dr.B.C Roy Engineering College, Durgapur, India

Anirban Sarkar²

²Department of Computer Applications, National Institute of Technology, Durgapur, India

ABSTRACT

In the latest years many different models for semi structured data have been proposed; most of them, however, are too specific to allow immediate comparison with other models, and do not easily support incremental model design. A number of features which can be considered interesting for a semi structured data model are listed. E.g. we review the more widely used models in Conceptual Modeling for Information Systems (Entity-Relationship and UML), and argue that they do not support effectively modeling of semi structured data. As a consequence, structured and semi structured data cannot be treated in an integrated, holistic way during requirements specification.

Research about similarity between semi structured documents (particularly for XML documents) has produced many works in the areas of Database Systems, Artificial Intelligence and Data Mining. In this work introducing a brief survey about it. At first introducing some basic properties. After that, some works are reviewed, highlighting their particularities and general approach. Concluding with a comparison of these works, analyzing their benefits and problems.

Keywords

Semi-structured Data, XML, XSD, Conceptual Modeling, Semi-structured Data Modeling, XML Modeling.

1. INTRODUCTION

The increasingly large amounts of data processing on the web based applications have led a crucial role of semi-structured database system. In recent days, semi-structured data has become prevalent with the growing demand of such internet based software systems. Semi-structured data though is organized in semantic entity but does not strictly conform the formal structure to strict types. Rather it posses irregular and partial organization [1].

The Extensible Markup Language (XML) is increasingly finding acceptance as a standard for storing and exchanging structured and semi-structured information over internet [12]. The Document Type Definition (DTD) or XML Schema Definition (XSD) language can be used to define the schema which describes the syntax and structure of XML documents [9]. However, the XML schemas provide the logical representation of the semi-structured data and it is hard to realize the semantic characters of such data.

To adopt the rapidly data evolving characteristics, the conceptual model of semi-structured data must support several properties like, representation of irregular and heterogeneous structure, hierarchical relations along with the non – hierarchical relationship types, cardinality, n – array relation, ordering, representation of mixed content etc. [13]. The participation of instances in semi-structured data model is not strict. In early years, Object Exchange Model has been proposed to model semi-structured data [2], where data are

represented using directed labeled graph. The schema information is maintained in the labels of the graph and the data instances are represented using nodes. Many of these approaches [3, 4, 5, 6, 7, 8] have been extended the concepts of Entity Relationship (ER) model to accommodate the facet of semi-structured data at conceptual level. The major drawbacks of these proposals are in representation of hierarchical structure of semi-structured data. Moreover, only two ER based proposals [7, 8] support the representation of mixed content in conceptual schema. In [7], a two level approach has been taken to represent the hierarchical relations. In *first level* the conceptual schema is based on extended concept of ER model and in *second level*, hierarchical organizations of parts of the overall conceptual schema are designed. In general, ER model are flat in nature [14] and thus unable to facilitate the reuse capability and representation of hierarchical relationship very efficiently. On the other hand, ORA-SS [11] proposed to realize the semi-structured data at conceptual level starting from its hierarchical structure.

Very few attempts have been made to model the semi-structured data using Object Oriented (OO) paradigm. ORA-SS [11] support the object oriented characteristic partially. The approaches proposed in [9, 10, 12] are based on UML. These approaches support object oriented paradigm comprehensively and bridge the gap between OO software design and semi-structured data schemata. These proposals have extended the UML stereotype definitions and notations. However, the UML and extensions to UML represent software elements using a set of language elements with fixed implementation semantics (e.g. methods, classes). Henceforth, the proposed approaches using extension of UML, in general, are logically inclined towards implementation of semi-structured database system. In [17, 18], a graph semantic based conceptual model for semi-structured database system, called Graph Object Oriented Semi-Structured Data Model (GOOSSDM), has been proposed. The model is comprehensively based on object oriented paradigm. Among others, the model supports the representation of hierarchical structure along with non-hierarchical relationships, mixed content, ordering, participation constraints etc.

This paper listed an extensive set of features for semi-structured data models which will be useful to lead towards the efficient database design for such data. Several available semi-structured data models are also evaluated based on these features and the study has pointed out the different advantages and disadvantages of those models. Moreover, this paper also pointed out a list of future research scopes on semi-structured database area.

2. RELATED MODELS SUMMARY

In this section summarized the related models for semi-structured database system. Majorly the models based on object oriented paradigm have been given emphasized for the descriptions. Semi-structured data is often explained as ".schema less 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 move from DTD document type definition a simple schema definition language to XML Schema a more expressive schema definition language highlights the importance of a schema definition for semi structured data applications.

(a) ERX

ERX (*Entity Relationship for XML*) model, an evolution of the Entity Relationship model that copes with the peculiar features of XML[5]. ERX is devised to provide an effective support to the development of complex XML processors for advanced applications, by giving a conceptual model to the collection of documents that puts in evidence classes of concepts and their relationships, even across different classes of documents.

(b) XML Conceptual Modeling with UML

Using UML for document design, we are able to combine object oriented s/w design with the xml document structures[12]. Hence the conceptual modeling with UML helps to improve a redesign and reveal possible structural weaknesses in the document design.

(c) ORA-SS

An object relationship attribute model for semi structured data is a semantically rich data model for semi structured data[11]. ORA-SS not only reflects the nested of structured data, but also distinguish between objects, relationships and attributes.

The definitions of a normal form ORA-SS schema diagram give the necessary and sufficient conditions for ensuring the corresponding set of nested relations in normal form for set of nested relations [23, 24].

(d) XER

Extensible Entity Relationship Modeling (XER) model[6], a conceptual modeling approach that can describe XML document structures in a simple visual form reminiscent of the ER model, and has the capability to automatically generate XML document type definitions and schema from such structures.

(e) EReX

ER extended for XML is a conceptual model [4]. Translating an EReX design to XML enables us to make use of the different features provided by XML. EReX is an extension to the E-R model proposed by Mani in [4]. The author introduces the following extensions to the E-R model:

- (1) Categorization of entity types can be modeled using category relationship types.
- (2) Total and exclusive coverage constraints can be specified for categories and for roles of entity types in relationship types
- (3) Order constraints can be specified for participants of a relationship type. An ordering on a participant E of a relationship type R is displayed by a thick solid line between R and E.

(f) XUML

XML a conceptual model , XUML [9], which inherits the merits of other models, and has major improvement in following aspects :1) can express the containment semantics more explicitly; 2) support the concept of business components ; 3) can specify the data dependencies in multiple contexts. These improvements make XUML more expressive, precise and understandable.

(g) XSEM

A new conceptual model for XML data called XSEM as a combination of several approaches in the area of the conceptual modeling for XML [7]. The model divides the conceptual modeling process of XML data to two levels. On the first level,

a designer designs an overall non-hierarchical conceptual schema of a domain. On the second level, designer derives different hierarchical representations of parts of the overall conceptual schema using transformation operators.

(h) GOOSSDM

A graph semantic based conceptual model for semi-structured database system, called Graph Object Oriented Semi-Structured Data Model (GOOSSDM). The model is comprehensively based on object oriented paradigm. Among others, the model supports the representation of hierarchical structure along with non-hierarchical relationships, mixed content, ordering, participation constraints etc.

The GOOSSDM reveals a set of concepts to the conceptual level design phase of semi-structured database system, which are understandable to the users, independent of implementation issues and provide a set of graphical constructs to facilitate the designers of such system. The schema in GOOSSDM [18] is organized in layered approach to provide different level of abstraction to the users and designers. In this approach a rule based transformation mechanism also has been proposed to represent the equivalent XML Schema Definitions (XSD) from GOOSSDM schemata [17]. The correctness of such transformation has been verified using the structural correlation mechanism described in [15].

3. MAJOR FEATURES OF SEMI-STRUCTURED DATA MODELS

Semi structured data has become prevalent with the growth of the internet. The development of new web applications that require efficient design and maintenance of large amounts of data makes it increasingly important to design “good” semi structured databases to prevent data redundancy and updating anomalies. Semi structured data though is organized in semantic entity but does not strictly conform the formal structured to strict types.

To adopt the rapidly data evolving characteristics, the conceptual model of semi structured data must support several properties like, representation of irregular and heterogeneous structure, hierarchical relations along with the non hierarchical relationship types, cardinality, n-array relation, ordering, and representation of mixed content etc.

- based on labeled graphs rather than labeled trees
- used for data exchange among, and integration of, heterogeneous data sources
- schema information is in the edge labels
- sometimes called schema less or self-describing
- data stored at the leaves
- easy to discover new data and load it
- implement query without knowing data types

3.1 Meta-Model Level Features

Meta model means model about model. Meta-model can be applied for the translation in a common formalism of proposed data models , and is useful for an easy comparison and discussion of concrete models features, such as allowed sets of values, to handling of object identifiers, relationships representation; moreover, it supports effective inter-model translation and design. Features of meta models are list below-

- (1) **Language used to define model:** a logical language for reasoning with parameterized views over semi-structured data, and define the model.
- (2) **Extended Framework :** Semi-structured information in L can be merged in a logic-based framework .The framework has been extended to deal with uncertainty, in the form of probability values, degrees

of beliefs, or necessity measures, in the XML documents.

- (3) **Mathematical Formalism:** It provides a concept of mathematical operations that will operate on different semantic groups and consists of a set of operators.

3.2 Semantic Level Features

- (1) **No Strict Structure:** a) no strict format, but can have some structure with optimal elements attributes. b) Some attributes may be missing, some repeated and new ones can be added later, change unpredictably
- (2) **No Strict Participation / Instances:** it is important to realize the participation constraints of the instances in association with some relation or semi structured entity type.
- (3) **Hierarchical Structured:** A heading is linked to the level immediately above it and the level immediately below in the appropriate hierarchy. Hierarchical relationships may also be indicating by systematic presentations such as tree structures or graphic display.
- (4) **Non hierarchical Structured:** Nested relations in that all have limitations in modeling situations with non hierarchical relationships given their tree like structures.
- (5) **Ordering:** a) The instances of an object class can be ordered. b) the values of an attributes can be ordered, and c) the set of attributes of an object class can be ordered.
- (6) **Irregular structured of data:** Data with an irregular or changing organization often represented as a graph.
- (7) **Disjunction:** A characteristic of semi structured data is the attributes and objects are likely to be less homogeneous than in structured data.
- (8) **Self Evolving:** In semi structured data model, each label is a set of descriptive properties. A property is a kind of meta data.
- (9) **Mixed Content:** A content can contains attributes, elements and text.
- (10) **Abstraction:** Abstraction is the process of hiding the details and exposing only the essential features of a particular concept or object.
- (11) **Separation of Structure and content explicitly:** considering the text content as contiguous string and representing the logical structure of documents in a separate hierarchy.
- (12) **Partial Relationship/ Participation :(A relationship of degree 2)** one object is the parent and the other child and we distinguish between the participation constraint on the parent in the relationship and the participation constraint on the child in the relationship.
- (13) **Heterogeneous :** Heterogeneous collections can be found in a great variety of application domains, most of them related to the discovery and integrated of semi structured info coming from disparate and autonomous sources.(combining of 2 or more data.)
- (14) **N-Array relationship:** Degree of an n-array relationship leads to more efficient storage and access to the data.(An attribute is an attribute of an object.)
- (15) **Inheritance:** Inheritance hierarchies represent common properties of an object. (E.g.- an inheritance hierarchy can be used to show that the properties of a student are a superset of the properties of a person.)
- (16) **Reuse potential:** It is supported with inheritance mechanism using link relationship.

- (17) **Constraints:** If the constraints does not hold then the object should not be nested. Tertiary constraints.(in fact n-array where $n > 2$)
- (18) **Cardinality:-** instantiation in the relationship. E.g. 1:1 Total participation in the relationship. 0:1 Optional one instantiation in the relationship.
- (19) **Functional Dependencies:** F.D model real world constraints, showing how some of the attributes depend on other attributes.
- (20) **Symmetric Relationship:** A relationship is symmetric if “A” is related to “B”, the “B” is related to “A”.
- (21) **Recursive Relationship:** Recursive relationships are also possible, that is relationships between an entity and itself.

3.3 Semi-Structured Query Language

The main query language for semi structured data models are XML Query Language and RDF Query Language. In case of XML, it may be subdivided into 2 parts-1) XPath: XPath is a declarative language for specifying paths in trees, such as XML documents, using path syntax similar to the one used in file system directory and file hierarchies.

2) XQuery: XQuery is used for a more expressive query language for XML. The path expressions of XQuery are based on the abbreviated syntax of XPath. XQuery employs a type system based on XML Schema and shares a set of built-in Functions and operators with XPath.

RQL is a typed functional language. It combines querying of schema and data.

- (1) **Algebra or Formalism:** It provides a concept of algebra that will operate on different semantic groups and consists of a set of operators.
- (2) **Basic Operations:** It provides the concept of basic operations that will operate on semi structured data model. Like union, intersection, Cartesian product, join, aggregation, retrieve etc.
- (3) **Advanced Operations:** It provides the concept of higher level operations that will operate on semi structured data model like drill across operation and drill through operation.

3.4 Feature related to Implementation

We have to extent features of semi structured data model to take advantages of all facets that semi structured data model offers. Additionally, these extensions enable the tuning of the models for guiding the transformation process.

- (1) **Logical Model:** Logical data models represent the abstract structure of some domain of information. They are often diagrammatic in nature .Logical data models should be based on the structures identified in a preceding conceptual data model.
- (2) **Transformation/Specification Language** The main goal of transformation specification language is to provide a mechanism for importing into a transformation arbitrary objects and edges between these objects from a source database.
- (3) **Query Language Implementation:** Traditional data models and query languages are inappropriate, since semi structured data often is irregular, some data is missing, similar concepts are represented using different types, heterogeneous sets are present or object structure is not fully known.
- (4) **Prototype Case tools:** Implement a set of tools which constitute a Computer aided Software Engineering (CASE) Environment together with other necessary tools.

4. EVALUATIONS OF CONCEPTUAL LEVEL SEMI-STRUCTURED DATA MODELS

In this section, we compare the different semi structured data models briefly described in this paper. The comparison is made against the general requirements and the modeling construction requirements.

There are four comparative tables .Table 1: Meta model level features, Table 2: Semantic level features, Table 3: Semi structured query language features, and Table 4: Features related to implementation.

Anyone in the world is not able to decide, which of the previous approaches much better modeling in Semi Structured data model is. ER model allow user to create a schema with no metadata redundancy, but there is the problem with the modeling of specific XML features. Also the hierarchical structure of XML arises problems such as data and metadata redundancy .modeling of attributes of relationship types, and modeling of n-array relationship types. The important requirement on the integration of all properties of semi structured is solved only by GOOSSDM model. None of the models solve the problem of the integration with semantic web technologies.

To adopt the rapidly data evolving characteristics, the conceptual model of semi-structured data must support several properties like, representation of irregular and heterogeneous structure, hierarchical relations along with the non – hierarchical relationship types, cardinality, n – array relation, ordering, representation of mixed content etc. [1]. Beside these, it is also important to realize the participation constraints of the instances in association with some relation or semi-structured entity type. The participation of instances in semi-structured data model is not strict. In recent past, several researches have been made on conceptual modeling of semi-structured data as well as XML. Many of these approaches have been extended the concepts of Entity Relationship (ER) model to accommodate the facet of semi-structured data at conceptual level. The major drawbacks of these proposals are in representation of hierarchical structure of semi-structured data. On the other hand, ORA-SS proposed to realize the semi-structured data at conceptual level starting from its hierarchical structure. But the approach does not support directly the representation of no-hierarchical relationships and mixed content in conceptual level semi-structured data model.

Table 1: Features of Meta-Model level

Model name	Language used to define model	Extended Frame Work	Mathematical Formalism
ERX	ER for XML	ER	Algebra
Xml conceptual modeling with UML	UML	Relational Model	
ORA-SS	Object related	Objected related	Graph
XER	ER Based Xml	ER	Algebra
EReX	ER Based Xml	ER	Algebra
XUML	UML	XML	Algebra
XSEM	ER for Semi structured data	ER	Algebra
GOOSSDM	Graph	GDM	Graph

Table 2: Features of Semantic Level

Model Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
ERX	√	√	√	-	√	x	x	x	x	x	-							√			
XML conceptual modeling using UML	√	√	√	P	√	x	x	x	x	x	x	x	x	x	x	x	x	x			
ORA-SS	√	√	√	P	√	-	√	√	√	-	-	√	√	√	√		√	√	√		
XER	√	√	√	x	√	x			√			√	√				√				
EReX	-	-	√	√	-	√	√		√	√			√	√		√		√			
XUML	√	√	√	P	P	√	√	x	√	-	-	-	√	√	√	√	V	√			
XSEM	√	√	√	√	√	√	x	-	√	-	-		√	-	-	-	√	√			
GOOSSDM	√		√	√	√	√	√	√	√		√		√		√	√	√	√			

√=Fully-Supported; x=Fully Not Supported; P=Partially-Supported

Table 3: Semi Structured Query Language

Model name	Algebra	Basic Operations	Advanced operations
ERX	√	Specialization	
Xml conceptual modeling with UML		Generalization Aggregation	
ORA-SS	√	Selection, Projection, Join	If-Then
XER		Generalization	
EReX	√	Conjunction Disjunction	
XUML	√	Aggregation	
XSEM	√	Join Projection	Clustering of hierarchical relationship
GOOSSDM	×	×	×

√= Fully Supported. ; ×= Fully Not Supported. ; P= Partially Supported.

5. FUTURE RESEARCH DIRECTIONS

Table 4: Features Related to Implementation

Model name	Logical Model	Transformation Specified Language	Query Language Implemented	CASE
ERX	RM	DTD	ERX-QL	XML-ERX Mapping
Xml conceptual modeling with UML	XML Based	DTD		
ORA-SS	RM	DTD		PVS
XER	RM	DTD		VBA
EReX	RM	X-Grammar	X-Query	
XUML	XML Based	WXS		XUML Editor UML 2 XUML
XSEM	XML Based	XSEM-H		
GOOSSDM	XML Based	XSD		GME

(1) **Algebra for Semi-structured Query** In classical databases, a query system could detect all type errors statically, before accessing the data. Believing this is no longer the case in

SSD: Schemas may allow quite irregular structure; sometimes, no schema is given.

(2) **Fuzzy query system for GOOSSDM:** Three important features of a good query language, especially from the view of users:

- (1) Expressiveness: a good query language should be able to express most user queries and the expressions should be clear and concise without ambiguity.
- (2) Completeness: a good query language should not only support information extraction but also data manipulation (e.g. INSERT etc.), data definition and data control.
- (3) User-friendliness: Most query languages are developed for human users and most users are not experts in database. Thus, a good query language should be easy to learn, easy to write and easy to read.

Fuzzy SQL provides a powerful, easy to implement, and robust tool for searching corporate databases as well as private data sources (such as local Access databases, Excel spreadsheets, and flat files).

GLASS (Graphical Query Language for Semi-Structured Data), is developed as a graphical language for users to extract information from semi structured data. It should support aggregation functions, negation and other XQuery standards.

(3) **Query Evaluation of Semi structured databases:** By semi structured, we mean that although the data may have some structure, the structure is not as rigid, regular, or complete as the structure required by traditional database management systems.

Query optimization as considered here involves modifying a query such that semantics is preserved but performance is enhanced. The first step is to replace the name of the decomposed relation by an expression that recovers the table from the decomposition. It can also give the answer of an SQL query can be implemented in many ways, but which one is best? Overview of Query Evaluation

SQL query is implemented by a query plan

Tree of relational operators

- 'Pull' interface: when an operator is 'pulled' for the next output

tuples, it 'pulls' on its inputs and computes them.

- Can change structure of tree
- Can choose different operator implementations

Two main issues in query optimization:

For a given query, what plans are considered?

- Algorithm to search plan space for cheapest (estimated) plan.

How is the cost of a plan estimated?

Ideally: Want to find best plan.

Practically: Avoid worst plans!

Two parts to optimizing a query:

Consider a set of alternative plans.

- Must prune search space; typically, left-deep plans only.
- Must estimate cost of each plan that is considered.
- Must estimate size of result and cost for each plan node.
- Key issues: Statistics, indexes, operator implementations.

(4) **Requirements Engineering in current Web Methodologies** Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish.

Requirements Engineering Techniques

A requirement is defined as a condition or capability that must be met or fulfilled by a system to satisfy a contract, standard, specification, or other formally imposed documents (IEEE Standard 610.12-1990). The requirements defined for a system

should be: correct, consistent, verifiable and traceable. It is an iterative and co-operative process with the objective to analyze the problem, to document the results in a variety of formats and evaluate the precision of the results produced [20]. The iterative process of requirements engineering consists of three main activities [21]:

- requirements elicitation
- requirements specification
- requirements validation

On the one hand, many different kinds of stakeholders participate in the development process: analysts, customers, users, graphical designers, marketing, multimedia and security experts, etc. On the other hand, the main features of these systems are the navigational structure, the user interface and the personalization capability. The structure requires an intuitive guide to avoid that the user “gets lost in the navigational space” [22]. The design of the user interface often has to take into account multimedia and marketing aspects. These special design aspects not only have to be handled differently during design, but already be considered during the requirements specification [19].

- (1) **Functional requirements:** are capabilities that a system must exhibit in order to solve a problem. Functional requirements can be sub-classified in:
- (2) **Requirements Engineering for Web Applications:** A Comparative Study Data requirements also known as conceptual requirements, content requirements or storage requirements.
- (3) **Interface requirements (to the user):** also known as interaction requirements or user’s requirements. They give an answer to how the user is going to interact with the Web application.
- (4) **Navigational requirements-** represent users’ navigation needs through the hyperspace.
- (5) **Personalization requirements:** also known as customization or adaptation requirements. They describe how a Web application has to (dynamically) adapt itself, depending on the user or environment profile.
- (6) **Transactional requirements:** also known as internal functional requirements or service requirements, express what the Web application has to compute internally, without considering interface and interaction aspects.
- (7) **Non-functional requirements:** act to constraint the solution, e.g. portability requirements; reuse requirements, usability requirements, availability requirements, performance requirements, etc.

(5) An Approach to Quality Evaluation Process of Semi Structured Data Model:-

Quality Management has specific characteristics. From these characteristics it was defined some points that identify a methodology to evaluate the Quality Management. Based on these two sets - characteristics and methodological proceedings - it were proposed several models, that integrate the global evaluation system, centered in the analysis of Quality Management reflexes in the organization. The categories are:

- (1) **Leadership:** it involves high direction leadership; management for the quality and public responsibility and the relations of the enterprise with the social community;
- (2) **Information and analysis:** it involves range and data and information management about quality and performance; comparisons with the concurrence and also with excellence references; analysis and use of data;

- (3) **Strategic planning and Quality:** it involves strategic planning process of quality and performance of the enterprise and plans for Quality and performance improvement;
- (4) **Development and management of human resources:** it involves planning and management of human resources; employees involvement; employees education and training;
- (5) **Quality Management processes:** it involves project and introduction of products and services in the market;
- (6) **Results obtained related to the quality and operations:** it involves obtained results related to the quality of the products and services;
- (7) **Focus on the client and the satisfaction:** it involves clients’ expectation: present and future; clients relationship management; compromise with clients; determination of the clients’ satisfaction;

Evaluation Methodology

The data quality evaluation methodology is designed to evaluate the equipment’s specified data by collecting and analyzing the requested data using a standard interface using the proper evaluation tool. The two areas are keys to the measurement and evaluation of the data quality that the equipment provides.

1. Qualitative Evaluation

The first step in the evaluation is to ensure that the equipment interface provides data at a level of quality .

1.1. Documentation

If the equipment interface documentation, the manuals, or the software version in the equipment under evaluation does not match, knowing what data are available for collection or what the equipment is capable of executing and/or reporting will be difficult.

1.2. Interface Software Defects

When the equipment has software defects, the equipment affects factory normal operations and forces the user to place additional tools to ensure that the equipment is functional and continues to report data normally.

1.3. Data Reporting

Data reporting is sometimes viewed as a software defect but in actuality it is a lack of data availability in the interface implementation.

1.4. Timestamps

Timestamps and time-stamping are two areas that are easy to address and verify using common manufacturing or business practices. Not setting the date and time can result in incorrect timestamps being reported.

1.5. Qualitative Evaluation Process Flow

When accepting a new piece of equipment into the factory or qualifying the equipment after a major software update, the user must determine the quality of the software interface and the data available for use.

2. Quantitative Evaluation

Depending on the interface used, the user will run the interface-provided services to retrieve all data available for process, control, performance, and equipment history tracking.

2.1. Data Collection Interface

The user must first be able to identify whether the needed data are available, the user has authorization access to the data, and the data are provided by the equipment. For this purpose, a specific method or evaluation mechanism will need to be selected or used to assess the data quality.

2.2. Data Collection Mechanism

For each parameter to be evaluated, the data collection mechanism must be specified as described below.

2.3. Data Protocol

The data protocol provides the rules by which the data needs to be interpreted. It pertains to the formal structure/patterns in which something is expressed.

2.4. Data Format

Semantics is the meaning or interpretation of a parameter when it is properly defined or when it includes information regarding its use and application.

2.5. Data Units

To be quantifiable means to be able to determine, express, or differentiate data from one tool to another. *Units* are used to quantify data.

2.6. Data Order

In many cases to properly identify a problem or analyze a set of data it has to be received in the order it was requested. In many cases order is defined when the data follows a specified pattern or sequence.

2.7. Data Accuracy

Accuracy is the closeness of the agreement between the result of a measurement and a true value of the measured. Since the true value cannot be determined exactly, the measured or calculated value of highest available accuracy is typically taken to be the true value [25].

2.8. Data Precision

Precision is the closeness of agreement between indications obtained by replicate measurements on the same or similar objects under specified conditions.

2.9. Data Frequency

Sampling error is that portion of the difference between the value derived from observations and the value it is supposed to be estimated; attributed to the fact that samples represent only a portion of the population [26].

2.10. Data Latency

In many cases, response time is critical to successful interdiction and control of the process. The time lag between sending a request and receiving the data or sense the desired action can be critical. Response time is directly related to the system *latency*. Latency is an accumulation of time delays from multiple sources.

6. CONCLUSION

In this paper presenting the semi structured data model properties into a unified frame work. The main applications are inter model comparison, translation, fundamental toward easy mediation between heterogeneous data sources. A related though distinct line of research is followed a unified framework for the management and the exchange of semi structured data, described according to a variety of formats and models. According to the comparison the most user friendly semi structured data model is GOOSSDM. This is a comprehensive object oriented conceptual model and the entire semi-structure database can be viewed as a Graph (V, E) in layered organization. Future studies will be concentrating on a fuzzy query system for this GOOSSDM.

7. REFERENCES

- [1] S. Abiteboul, P. Buneman, and D. Suciu, "Data on the Web: From Relations to Semi structured Data and XML", Morgan Kaufman, 1999.
- [2] Jason McHugh, Serge Abiteboul, Roy Goldman, Dallas Quass, Jennifer Widom "Lore: a database management system for semi structured data", Vol. 26 , Issue 3, PP: 54 - 66, 1997
- [3] Badia, A.; "Conceptual modeling for semi structured data", In Proc. of the Third International Conference on Web Information Systems Engineering, PP: 170 – 177, 2002.
- [4] M. Mani. "EReX: A Conceptual Model for XML", In Proc. of the Second International XML Database Symposium, PP 128-142, 2004.
- [5] G. Psaila, "ERX: A Conceptual Model for XML Documents", In Proc. of the ACM Symposium on Applied Computing, PP: 898-903, 2000.
- [6] A. Sengupta, S. Mohan, R. Doshi, "XER - Extensible Entity Relationship Modeling", In Proc. of the XML 2003 Conference, PP: 140-154, 2003.
- [7] Martin Necasky, "XSEM: a conceptual model for XML", In Proc. of 4th ACM International Asia-Pacific conference on Conceptual Modeling, Vol. 67, PP: 37 - 48, 2007.
- [8] Bernadette FariasLócio, Ana Carolina Salgado, Luciano do RêgoGalvDo, "Conceptual modeling of XML schemas", In Proc. of the 5th ACM International Workshop on Web Information and Data Management, PP: 102 – 105, 2003.
- [9] HongXing Liu, YanSheng Lu, Qing Yang, "XML conceptual modeling with XUML", In Proc. of the 28th International Conference On Software Engineering, PP: 973 – 976, 2006.
- [10] Carlo Combi, Barbara Oliboni, "Conceptual modeling of XML data", In Proc. of the ACM Symposium On Applied Computing, PP: 467 – 473, 2006.
- [11] Xiaoying Wu; Tok Wang Ling; Mong Li Lee; Dobbie, G.; "Designing semi structured databases using ORA-SS model", In Proc. of the 2nd International Conference on Web
- [12] Rainer Conrad, Dieter Scheffner, J. Christoph Freytag, "XML International Conference On Conceptual Modeling", PP: 558-574, 2000.
- [13] Necasky, M. (2006), "Conceptual Modeling for XML: A Survey", Tech. Report No. 2006-3, Dep. of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague, 2006.
- [14] Choudhury S., Chaki N., Bhattacharya S., "GDM: A New Graph Based Data Model Using Functional Abstraction", Journal of Computer Science and Technology, Vol. 21, Issue 3, PP: 430 – 438, 2006.
- [15] Sarkar A., Choudhury S., Chaki N. and Bhattacharya S., "Conceptual Level Design of Object Oriented Data Warehouse: Graph Semantic Based Model", International Journal of Computer Science (INFOCOMP), Vol. 8, Issue 4, PP: 60 – 70, 2009.
- [16] ÁkosLédeczi, Arpad Bakay, MiklosMaroti, Peter Volgyesi, Greg Nordstrom, Jonathan Sprinkle, and GáborKarsai, "Composing Domain-Specific Design Environments," IEEE Computer, November 2001, pp. 44-51.
- [17] AnirbanSarkar, SesaSingha Roy, "Graph Semantic Based Conceptual Model of Semi-structured Data: An Object Oriented Approach", 11th International Conference on Software Engineering Research and Practice (SERP 11, WORLDCOMP 2011), Vol. 1, PP 24 – 30, Las Vegas, USA, July 18 – 21, 2011.

- [18] Anirban Sarkar, “Conceptual Level Design of Semi-structured Database System: Graph-semantic Based Approach”, *International Journal of Advanced Computer Science and Applications*, The SAI Pubs., New York, USA, Vol. 2, Issue 10, PP 112 – 121, November, 2011. [ISSN: 2156-5570(Online) & ISSN : 2158-107X(Print)]
- [19] Escalona, M.J., Mejías, M., Torres, J. (2002). “Methodologies to develop Web Information Systems and Comparative Analysis”. *Informatik/Informatique*. núm. 2/2002 de I/I.
- [20] Ferreira, M.J., Loucopoulos, P. (2001).”Organisation of Analysis Patterns for effective Re-use.Proceedings “of the International Conference on Enterprise Information Systems. ICEIS 2001. Setubal, Portugal.
- [21] Lowe, D., Hall, W. (1999). *Hypermedia and the Web. An Engineering approach*. John Wiley & Son.
- [22] Olsina, L. (1999). “Metodología Cualitativa para la Evaluación y Comparación de la Calidad de Sitios Web.” Ph. Tesis. Facultad de Ciencias Exactas. Universidad de la Pampa. Argentina.
- [23] T. W. Ling. A normal form for sets of not-necessarily normalized relations. In *Proceedings of the 22nd Hawaii International Conference on System Sciences*, pp. 578-586. United States: IEEE Computer Society Press, 1989.
- [24] T. W. Ling and L. L. Yan. NF-NR: A Practical Normal Form for Nested Relations. *Journal of Systems Integration*. Vol4, 1994, pp309-340
- [25] American Society for Quality. *The History of Quality*. ASQ Press. Milwaukee, WI. Dec. 2006.
- [26] Hillmer, Steve; Kocabasoglu, Canan. Improving Quality and Productivity. *Quality management Journal*. Volume 14,number 1. January 2007.