International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:12, 2008

A Formal Suite of Object Relational Database Metrics

Justus S, and K Iyakutti

***Abstract*—**Object Relational Databases (ORDB) are complex in nature than traditional relational databases because they combine the characteristics of both object oriented concepts and relational features of conventional databases. Design of an ORDB demands efficient and quality schema considering the structural, functional and componential traits. This internal quality of the schema is assured by metrics that measure the relevant attributes. This is extended to substantiate the understandability, usability and reliability of the schema, thus assuring external quality of the schema. This work institutes a formalization of ORDB metrics; metric definition, evaluation methodology and the calibration of the metric. Three ORDB schemas were used to conduct the evaluation and the formalization of the metrics. The metrics are calibrated using content and criteria related validity based on the measurability, consistency and reliability of the metrics. Nominal and summative scales are derived based on the evaluated metric values and are standardized. Future works pertaining to ORDB metrics forms the concluding note.

***Keywords*—**Measurements, Product metrics, Metrics calibration, Object-relational database.

I. Introduction

IT is understood that the important constituent of product engineering is the ability to measure the internal quality of the product and extend them to assess the external quality of the product [1]. Most of the times the software product measures are concerned only with the complexity of the programs, its functionality and its runtime behavior, both in conventional and object oriented systems. As analysis, design and coding has already moved for object oriented approaches in developing Information Systems (IS) , databases too have become complex as they have opted to function with Object Oriented Information Systems. Hence, there arises the need to propose and study some measures to assess the internal quality of the database and thereby determine their external applicability and reliability. It is important that databases are evaluated for every relevant quality characteristic using validated or widely accepted metrics. Such metrics could help designers to choose the most maintainable, effectual, and semantically proficient schemata. Apart from the existing set of metrics [2], [3], [4] for object relational database (hereafter ORDB), few more were proposed in [5], [6], which focuses on the behavior of the database in real timeframes. Still ORDB is left unexplored on its varied temperament; like dynamism, componential traits and features of object oriented storage.

Justus S is with the K.L.N. College of Information Technology, Pottapalayam – 630611, Sivagangai DT, TN, India. (corresponding author phone: 91-9842115894; e-mail: justus.mku@gmail.com).

Iyakutti. K., is with the Department of Microprocessor & Computer, School of Physics, Madurai Kamaraj University, Madurai – 625021, TN, India.(e-mail: iyakutti@gmail.com).

A. Motivation

This work on the evaluation and the calibration of the ORDB metrics is motivated by the need for a formalized set of metrics that would be consistent, reliable and measurable. Hence we attempt here to consider simple ORDB schemas and make the metric evaluation process calculable. Also we extend the work to calibrate the metrics, to standardize and prove that the metrics measure what it intends to measure.

We see that database metrics are not given due importance in the software measurement community, might be for its static nature in functional platforms. But our earlier work in [6] discusses metrics to assess the dynamism of databases, and infers that databases too demand due attention from system engineers and designers. As far as we understand, for complex databases like ORDBs, the metrics need formalization and calibration to arrive at standardized scales, from which database designers would be able to assess their design. In this work we attempt to propose methodologies for evaluating the ORDB metrics and we aspire to calibrate them to achieve and arrive at standard scales for each metric. Simple OR schemas, based on the ontology given in appendix (figure 1), were considered for better understandability of the evaluation of metrics. Subsequently, sample experiments were conducted to make the metrics scalable and meaningful.

B. Relevant Works

The metric community possesses works of eminent researchers and scholars who have proposed suitable measures for the object oriented approach. Those measures linger around the design aspect of the real world entities and its operability. Chidamber & Kemerer [7] have proposed a suite for Object Oriented Design which helps in arriving at better design of the OO system. Works in [8], [9] states the dynamic nature of OO systems. Baroni Et.al [2], [3] and Piatini Et. al. [4] have given his major contribution exclusively to the Object Relational databases. They have considered the design metrics and have formally validated them based on the SQL:2003 Ontology.

The design complexity and the structural complexity are measured by the basic metrics like table size (TS), number of

weighted methods per class (CWM), depth in the relational tree (DRT) referential degree (RD), number of involved classes (NIC), number of shared classes (NSC) [2], [3], [10], [11]. Moris [12] proposed metrics for Object oriented software design and development which help the system designers to evaluate their design strategies and methodologies with which systems were architected. In [13] the authors attempted to test the structural complexity of OO software using cohesion and coupling metrics and in [14] three cohesion metrics are interpreted. In contrast, works in [15] and [16] quantify that cohesion and coupling are measures of behavioral aspect of OO systems.

The traditional metrics of object oriented design, which could be extended to ORDB, were criticized for their relevance in the functional platforms and their defects were reflected on the basis of their influence in the design process [17].

C. Outline

This research work is organized as follows: Section 2 details on the evaluation methodology, covering the basic concepts of ORDB, and formal definition and the evaluation procedures. Section 3 presents the formal validation of the metrics based on the content and criteria related validity. Calibrated scales are also derived for the metrics in this section. The formal calibration is performed using sample ORDB schemas and the metric values are calculated based on the schemas. Section 4 presents the concluding remarks and future works pertaining to the ORDB metrics.

II. EVALUATION OF ORDB METRICS

An object-relational database schema is comprised of a number of related tables that forms a connected structure of type constructs or classes (hereafter class-type). Class-types are the classes in the object relational paradigm. They possess all the properties of a class; data abstraction, encapsulation, inheritance and polymorphism. These traits of class-types are embedded in the relational nature of the database; data model, security, concurrency, normalization and so on. In more precise words, the relational database talks in terms of objects. It is also interesting to have a relational schema built with classes and these classes encapsulated with data and methods acting on the data.

Such as its description the object relational databases are more intricate and multifarious. When such a complex structure is proposed to be built, its design should be more systematic and elaborate taking into account the functional perception of the database along with its application. Hence a set of metrics are derived considering the object oriented epitome of a relational database. Consider the example shown in figure 1. The relational model on the left side uses class-types defined on the right. The relational tables are comprised of attributes of basic data type and also the class-types which are referred as User Defined Datatype (UDT). These UDTs are the complex components of challenge in the object relational database archetype [18], [19]. In this example ORDB Schema we use three class-types (UDT) for two relational tables.

A. Metric 1: Table Size (TS)

The metric is defined as the sum of complexities of simple columns and of the complex columns. Each of these complex columns can be a class or UDT or other class-type.

$$\mu(TS) = \sum_{i=0}^{n} SC_i + \sum_{j=0}^{m} CC_j \quad (1)$$

A simple column is constructed using basic data type, and no extra construction or maintenance cost is required and they have no runtime characteristics. Hence, the complexity weight is unity for a simple column. Whereas, a complex column (attribute) is an UDT (user-defined type), which exhibit dynamic behaviors during runtime. A complex column may be *Structured type, Composite type, Collection type or a Reference type* [20].

```
CREATE TABLE Tab_Staff
(
emp_no varchar(4),
person info person t,
do_joining date;
working department
department t,
ward_for varchar(7) FK
Tab Student(roll no)
inverse ref);
```

```
CREATE TABLE Tab Student
(
roll no varchar(7),
person_info person_t,
in department
department_t,
ward of varchar(4) FK
Tab_Staff(emp_no)
inverse ref);
```

```
CREATE TYPE person t (
Name varchar(20),
Gender varchar(1),
Birth_date date,
Address info address t,
MEMBER FUNCTION set_values()
     RETURN person t,
MEMBER PROCEDURE
print person()
);

CREATE TYPE department t(
Name varchar(15),
HOD varchar(4),
MEMBER FUNCTION set_values()
    RETURN department t,
MEMBER PROCEDURE
print object()
);

CREATE TYPE address_t (
Door no number,
Area_name varchar(20),
Zone name varchar(20),
City varchar(15),
Pin code number,
MEMBER FUNCTION set_values()
    RETURN address t
);
```

Fig. 1 Sample ORDB Schema

Size of address_t:	Size of person_t:	Size of department_t:
SC = 5, CC = 2	SC = 3,	SC= 2, CC=2+2 = 4
TS = 5+2 = 7	CC=7+2+2= 11	TS = 2+4 = 6
	TS = 3+11 = 14	

The size of the table student can be calculated as:	The size of the table staff can be calculated as:
SC = 2, CC = 14+6 = 20	SC = 3, CC = 14+6 = 20
TS(Tab_ Student)=2+20=22	TS (Tab_ Student)=3+20=23

Hence the structural complexity of a complex column in terms of its construction and maintenance should be considered [21]. For evaluation convenience, we assume the measure for methods and procedures in a complex column can be averaged to 2. Based on these assignments the metric value of the example schema shown in figure 1 is calculated as follows. The schema design is given in the appendix (figure 2)

This metric is a measure of the total size of a table in the object relational schema. However, TS is not just a measure of the sum of the attributes, but of the structural complexity of the complex columns. The metric will be useful in refining the relational database table design in restricting the tables' size to an optimal value, which will reduce effort and cost in the construction and maintenance of the database schemas. Larger the TS value, higher will be the effort in managing the attributes and the data in the tables, thus higher will be the maintenance effort.

B. Metric 2: Complexity of Weighted Methods (CWM)

Consider a class-type A, containing *m* number of methods whose complexity are C_1, C_2….C_m, then the metric is defined as the sum of the complexities of the methods, which is given as

$$\mu(CWM) = \sum_{i=1}^{m} C_i \quad (2)$$

This metric is not just the count of number of weighted methods as defined in earlier metrics definition [1], [7], but quantifies the complexities of all the methods. Cyclomatic complexity is a measure of complexity of a program unit or a procedure, but a member method of a class-type in ORDB does not have functional control paths. Hence, we formulate the complexity based on the FP calculation discussed by Pressman [22].

Count_total = {(no of tables accessed × cf) +
(no. of read queries × cf) +
(no. of write queries × cf) }

Where cf: complexity factor ranges from 1 to 5, as functional complexity increases.

CWM(Mi) = Count_total × [0.65 + 0.01 × F]

Where F is the functional weight of the method. F=1 for constructors and destructors, and other methods involving I/O operations. Based on the metric definition we evaluate the metric value considering the *person_t* class-type, defined in figure 2.

Person_t.set_values()

Tables accessed = 2, Read queries = 1, Write queries = 1

Count_total = {(2 × 3) + (1 × 2) + (1 × 2)} = 10

CWM(person_type.set_values)=10 × [0.65 + 0.01 × 3]=6.8

C. Metric 3: Cohesion between methods (COM)

In the view of Bunje [23], Cohesion is defined in the terms of similarity between two things as the intersection of the set of properties of the two things. This definition can be extended as the cohesion between the methods as the intersection of the set of instance variables.

TABLE I
CWM METRIC EVALUATION

Type Construct	No of Methods	CWM
person_t	2	6.8+0.66 = 7.46
department_t	2	5.4+0.66 = 6.06
address_t	1	2.7

$\gamma(M_1,M_2) = \{I_1\} \cap \{I_2\}$

where $\gamma(M_1,M_2)$ is the degree of similarity between methods M_1 and M_2 and {Ii} is the set of instance variables used by the method M_i.

```
CREATE OR REPLACE TYPE BODY person_t AS
   MEMBER FUNCTION set values ()
      RETURN person_t
   IS
      the_person person_t := SELF;
      the address address t := SELF;
  -- initialize local varialbes
   BEGIN
the_person.name:= & pn;
the person.gender:= &pg;
the_person.birth_date:= &pb;
the person.the address.setvalues();
insert into values the_person Tab_Student;
insert into values the person Tab Staff;

      RETURN the person;
   END;
MEMBER PROCEDURE print person
  IS
   BEGIN
DBMS_OUTPUT.PUT_LINE('Name : ' || name);
DBMS_OUTPUT.PUT_LINE('Gender  : ' ||gender);
DBMS_OUTPUT.PUT_LINE('Birth Date :'|| birth_date);
   END;
       END;
```

Fig. 2 Member Methods definition

Berard [11] defines cohesion between two methods as the ratio of set of common instance variables referred by the methods to the total number of instance variables in the class. Consider a class-type 'A' with *Vs* number of instance variables and with *n* number of methods. Then the metric definition can be stated as:

$$\mu(A) = \frac{I_i \cap I_j \cap I_k \cap \ldots I_n}{Vs} \quad (3)$$

where $(I_i \cap I_j \cap I_k \ldots I_n) \neq \phi$

This, Berard [11] refers to as the superglue for the methods in the given class-type.

TABLE II
COM METRIC EVALUATION

Type Construct	COM
person_t	3/4 = 0.75
department_t	1/2 = 0.5
address_t	0

Consider the sample of method definition in figure 2. for the class-type *person_t*. Method *set_values()* access three members and procedures, intersection gives the value of 3. The total number of instance variables accessed by both the methods is 4. Hence the COM for *person_t* is 3/4 = 0.75. Since *address_t* contains only one method the metric value will be zero.

The measure of COM in a class-type in ORDB gives the database designer an insight into how far the designed class-type is rigid in the given schema. If COM is high, then the fault prediction and change proneness of the class-type will be challenging and costly [7]. On the other hand, if COM is too low, then the class-type is loosely built, demanding its decomposition. Higher the number of decomposed classes, the structural complexity of the table increases as a whole [13].

D. Metric 4: Coupling between Objects (CBO)

An object is coupled to another object if one of them acts on the other, i.e., methods of one objects uses the methods or instance variables of the other. Since objects of the same class-type have same characteristics, two class-types are coupled only when methods declared in one class-type use methods or instance variables used by another class-type. Coupling is the sum count of instance variables and methods invoked from a class-type of an object of another class-type.

$$Count_{cp} = \sum_{i=0}^{n} I_i + \sum_{j=0}^{m} M_{INVj} \quad (4)$$

Method invocation (M_{INV}) involves calculations of number of methods called and the average number of arguments involved in each invocation. The count of coupling $Count_{cp}$ results in a positive value with which the coupling indicator can be derived.

$$Cob = \frac{k}{Count_{cp}}$$

Where k = 1 and is a proportionality constant which may be adjusted as experimental verification occurs. This implies that $Count_{cp}$ is inversely proportional of the metric *Cob*. Hence to have a coupling metric increase as the degree of coupling intensifies [22], a reversed coupling metric is defined as

$$\mu(CBO) = 1 - Cob \quad (5)$$

where the degree of coupling varies non-linearly between 0 and 1. This is a direct measure of number of instance variables and methods invoked, arguments passed to the methods of objects of other classes/tables or type constructs. The class-type *department_t* is reengineered and redefined so as to make it more complex. The *department_t* type is tightly coupled with *person_t* which in turn is coupled to *address_t*, where both are coupled using method invocation and not instance variables. In this definition *department_t* is coupled to *person_t* by one-method invocation, hence $Count_{cp}$ is 1. Also *person_t* is coupled to *address_t* by one-method invocation, hence $Count_{cp}$ is 1. But, since *department_t* type is under consideration and is coupled at two levels, we divide the second level couple by its level. Thus $Count_{cp}$ can be calculated as:

$Count_{cp}$ = CL1 + CL2 = 1 + (1/2) = 1.5

Where CL1=Coupling level 1, CL2=Coupling level 2

Cob = 1/1.5 = 0.667

CBO(department_t) = 1 – Cob = 1-0.667 = 0.333

The measure of coupling between objects in ORDB gives the insight into the compactness with which the class-type is entrenched in the schema. Removing or modifying a highly coupled class-type will affect the other coupled class-types, leading to higher effort in restructuring or reengineering the schema design.

Lower the coupling, better the design [13]. Hence, lower CBO value for a class-type is preferred as design and implementation goals. However, the practical challenge is in selecting the tolerable level of measure.

E. Metric 5: Referential Degree

The metric RD is the summation of the number of foreign keys and reverse reference keys the table or class contains. For a table with *n* number of foreign keys and *m* number of reverse references is given as follows:

$$\mu(RD) = \sum_{i=0}^{n} Fk + \sum_{j=0}^{m} Rvref \quad (6)$$

The concept of reverse reference is one of the best features of relation databases available in ORDB. This makes a relational table more smart and referenced to the contextually related tables by dual means, ie., the existence of the reference is authenticated double the times a relational model supports. The dual measure of the metric best admits the degree of entrenchment of the class-type or table in the database schema, subjecting the table/class-type to be normalized based on functional dependencies and key references. The metric is a mere count of the reference and reverse reference keys.

F. Metric 6: Depth in the Relational Tree

The metric DRT of a table/class is defined as the longest path in the tree that constitutes the tables that are related by means of reference keys and inheritance. Relationships formed between classes involve other classes or UDT or other data constructs like list, bag, array etc. Consider class A, and $d_1, d_2, \ldots, d_k$ are the distance between the related tables/classes. Then its depth in the relational tree can be given as

$$\mu(DRT) = \sum_{i=0}^{k} di \quad (7)$$

The metric determines how strongly the table is ingrained in the schema or in the relational tree. Hence change in the tree structure is influenced by the referential degree of the table or class.

G. Metric 7: Number of Inherited Properties

This metric relates to a notion of scope of properties of a class-type. It is a measure of how many properties are being inherited by the children and how many are not being inherited. The properties of a class-type include the instance variables and the methods. Consider a class-type 'A' having *n* number of instance variables and *m* number of methods to be inherited, then the measure can be stated as

$$\mu(NIP) = \sum_{i=0}^{n} I_i + \sum_{j=0}^{m} M_j \quad (8)$$

Where M_j= [no. of types x CWM] + [no. of methods x CBO]

Suppose the class-type definition of *person_t* allows all the instance variables and the member methods to be inherited by other two class-types, *student_t* and *staff_t*, then the NIP metric for *person_t* will be its size.

The count of number of properties being inherited determines the complexity with which the type can be reused. NIP is not just a direct count of properties but the coupling of a class-type with other types and the number of weighted

methods united cohesively. The type *person_t* has 3 instance variables, 1 UDT and 2 member methods. The complexity measure for UDT is its CWM and for methods is its CBO. This measures the component's reuse. Thus the metric value for person_t is calculated as follows:

NIP (*person_t*) = 3 + [1 × (CWM)] + [2 × CBO]

NIP (*person_t*) = 3 + [1×1.4)] + [2 × 0.5] = 5.4

This value of NIP obviously measures not just the number of properties but also their complexity with which the component can be reused in future schemas.

Since inheritance in a form of reuse, this metric, NIP, contributes to the componential nature of the database. Hence the designing of classes need more attention for their future usage and their contribution in the componential database design.

III. CALIBRATING THE METRICS

Measurements for OOsystems, in general, have been proposed by CK [7], and other research works [1], [10], [15], [16], and for ORDB, in particular, metrics have been proposed by Piatini [2], [4], [5]. Metrics for ORDB have formally validated in [20], [24], but the metric values have to be calibrated to a scale, where their applicability could well be appreciated. In this work we have attempted to define the metrics for ORDB in the context of its design, that is, at the schema-level. The calibration of these metrics is formalized by testing the device (metrics) for its 1) construct validity, 2) criterion based validity and 3) reliability in measuring the intended attribute.

A. Content Validity

The measurement device (metrics) has to consider the global set of data to test its consistency in formulating the evaluating methodology. In the previous sections we have dealt in detail the evaluating methodology for each of the metrics. This validity is proved by arriving at consistent values for the metrics using a wide variety of database schemas.

Three object relational database schemas were considered for this research work. We name them as ORSx, ORSy and ORSz.

The schema ORSx is backing a financial application, where the specification is to maintain day-to-day transactions of the organization and process data, to generate reports and memos according to the customers' requirement. The schema ORSy houses the information of a shipping corporation. The realtime ship movements, radar information from the traffic cell, generation of reports and memos are the primary function of the application. ORSz supports an SCM system for an electronic production company.

TABLE III
EXPERIMENTAL DATA SET

Schemas	No of Tables	No. of types	ANIV	ANMT	ANRT	RT
ORSx	140	84	12	34	7	14
ORSy	226	127	20	43	12	28
ORSz	174	97	15	30	10	22
Total	540	308	47	107	29	64

ANIV: Avg no of Instance variables per type, ANMT: Avg no of Methods per type, ANRT: Avg no of References per type, RT: No of Relational Trees

TABLE IV
OBJECT ORIENTED AND RELATIONAL CONCEPTS

Schemas	Object Oriented Concepts				Relational Concepts		
	ABS	AGN	PMS	IHT	NRM	RI	ACID
ORSx	Y	Y	N	Y	P	Y	P
ORSy	Y	Y	Y	P	P	Y	Y
ORSz	Y	Y	P	Y	Y	P	Y

ABS: Abstraction, AGN: Aggregation, PMS: Polymorphism, IHT: Inheritance, NRM: Normalization, RI: Referential Integrity;
Y – Yes, N – No; P – Partial

The experimental dataset is given in table 3. The schemas covered a wide variety of concepts in object oriented system and relational system. Table 4 presents the various concepts covered by the experimental dataset.

The schemas were very appropriate for this study that they are applicable in testing the metric device in more than one dimension; Measurability, Consistency and Reliability. The design-level metrics, proposed here, are tested on these four dimensions.

B. Criterion related Validity

Criterion related validity can be performed by means of testing

- The measurability of the metrics and
- The consistency with which they measure the attributes.

1) Measurability

Measurability of the metric includes other factors like understandability and calculability. Understandability of these metrics for ORDB is done in our earlier works [18]. Calculability is the ability to derive complete ontology based standard formulas for estimating the measured values. That is, using the formula we arrive at definite numbers that gives us insight into the design-level measurability of the database schema.

This is proved by arriving at definite numerical values that convey the semantics of the metrics, which are detailed in the previous sections.

2) Consistency

Consistency of the metric is defined as the ability of the device (metric) to produce coherent values at different iteration of measurements [17], [26]. An iteration is one instance of executing the formulas for the given schema design. The three object relational schemas ORSx, ORSy and ORSz are the dataset where the metrics are applied to measure their attributes. We have conducted totally 18 iterations with 6 iterations for each schema. The metrics TS, DRT, RD, CBO, COM, NIP, CWM are measured using the formulas given for

each measurement. A total of (7 × 6 =42, and 42 × 3 = 126) 126 metric values are obtained. For a single metric we have 6 iterations and for the 7 metrics it totals to 42 and for the three schemas it is 126.

We have adopted correlation analysis to study the consistency of the metric values and have found that they correlate between iterations of experiments. All possible pairs of iterations are analyzed using Karl Pearson's Coefficient Correlation [26] using equation 9 and the results are given in table 5.

$$Correl(x,y)=\frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2\sum(y-\bar{y})^2}} \quad (9)$$

TABLE V
CORRELATION ANALYSIS BETWEEN ITERATIONS

Metrics	1,2	2,3	3,4	4,5	5,6	6,1
TS	0.872	0.846	0.812	0.896	0.833	0.846
DRT	1	1	0.998	1	1	0.997
RD	1	0.978	0.986	0.998	1	1
CBO	0.975	0.963	0.957	0.978	0.932	0.912
COM	0.911	0.927	0.923	0.935	0.926	0.917
NIP	0.750	0.787	0.769	0.791	0.812	0.768
CWM	1	1	0.996	0.988	1	0.992

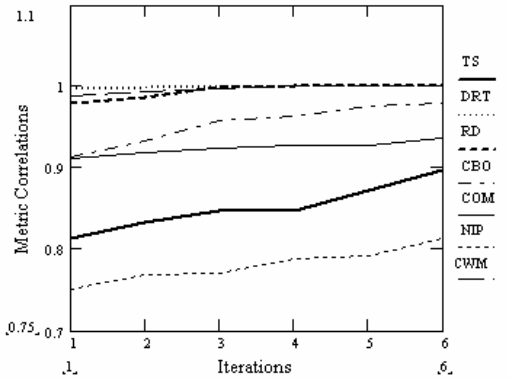

Fig. 3 Correlation among the different Iterations of measurement

The results show that the six iterations for each metric show only positive correlation and are nearly tending to unity. In some cases, the values are repeated thus showing unit correlation. The values are plotted in a graph to show a more clear understanding of the correlation among the iterations, shown in figure 3.

The scales for the metrics are calibrated based on these evaluated metric values. TS metric is a standalone scale of measurement; COM and CBO are integrated because of their range of measurement; NIP is a measurement for assessing the reusability of the class-type. The optimal value is from 8 to 10, as maximum number of classes fall in this range, in our three schemas. This metric is commended by CWM and CBO, whose acceptable range of values are from 0 to 1. The left out metrics RD, DRT and CWN are supporting metrics for the other four metrics. The metric distributions for the three schemas were given in figures 4 (a), 4(b) and 4(c).

Thus, we conclude that for the number of iterations for each metric value, the device is consistent to measure the attributes, proving the consistency test for the device (metric). As far as the metric is consistent in measuring the intended attribute it is expected to be reliable enough to satisfy the measurement properties.

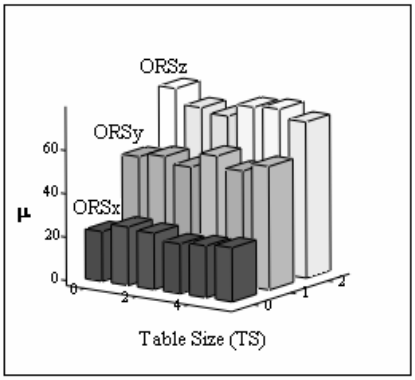

Fig. 4(a) Table size metric for three schemas

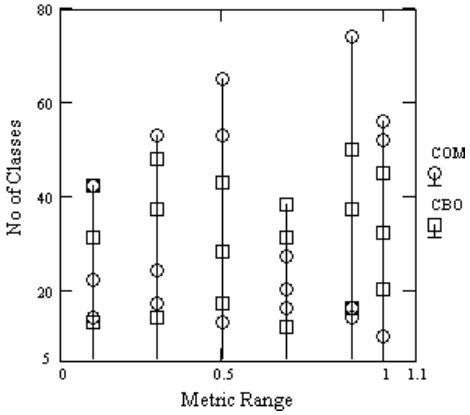

Fig. 4(b) COM and CBO Metric Ranges

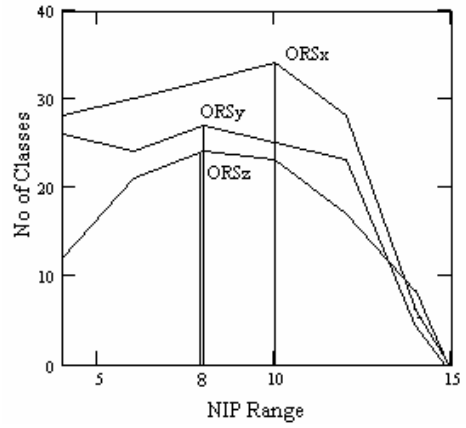

Fig. 4(c) NIP Metric for the three schemas

C. Reliability of the metrics

The reliability of the metrics can be tested using two approaches: Formal analytical approach and Scale Calibration approach.

1) Analytical Approach

The metrics for object-oriented design proposed in [7] are based on the six properties of metric evaluation. The definitions of the properties given in [27], [28], [29] for OOD are redefined here in relevance to the ORD schema design requirements, retaining the essence of the properties. Based on the six properties we have framed four properties, each at the design, functional, atomicity and componential interaction. The properties are:

1. Design-level Non-uniqueness: the metric for two different class-types may have same numerical value at the design-level
2. Functional-level uniqueness: the metric for two different class-types may have different numerical value at the functional level. This is because the realtime parameters involved in calculating the metric varies at several instances
3. Atomicity: The metric for individual class-type is always less than or equals the metric when two or more class-types are joined.
4. Interaction complexity: the metric of each class-type varies in interaction with other class-types, as they are dynamically determined by external programming factors.

Table Size (TS): This metric measures the design-level complexity of the schema [5]. The design-level complexity measure involves the fields of the table, both simple and complex. As complex tables include the UDTs, class/types, complexities of the tables are not just count of attributes, but the complexity of the attributes. Hence property 1 holds good. As well, the metric for the two tables A and B reveals the atomic existence of itself, and hence property 3 holds good.

$\forall$ *tables A, B;* $\mu(A) = \mu(B)$ [Property 1]

ie., the two tables are equally complex at the design level.

$\forall$ *tables A, B;*

$\mu(A) \leq \mu(A+B), \quad \mu(B) \leq \mu(A+B)$ [Property 3]

No. of Weighted methods per class (CWM): This metric shows uniqueness when the functional complexity of a class-type is considered and non-uniqueness for design level complexity of a class-type. For a class type with n methods, the CWM metric calculates the inputs, outputs, queries and other methods invoked, where all these parameters are calculable during design time and runtime. Hence property 1, 2 and obviously 3 holds good.

$\forall$ *Class C,* $\exists$ *method M, N;*

then $\mu(M) = \mu(N)$ [Property 1]

$\mu(M) \neq \mu(N)$ [Property 2]

$\mu(M) \leq \mu(M + N), \quad \mu(N) \leq \mu(M + N)$ [Property 3]

Cohesion between methods (COM): This functional metric involves the uniqueness and the interaction complexity of the class-type. Cohesiveness of a class-type is the interactions among methods, which the metric intends to measure. Hence properties 2 and 4 holds good. Obviously property 3 is ignored as cohesion complexity of pairs of methods or an individual class is non-equivalent.

$\forall$ *Classes A and B;* $\mu(A) \neq \mu(B)$; *and* [Property 2]

$\exists$ *A,* $\exists$ *B and* $\exists$ *C, such that* $\mu(A) = \mu(B)$

$\mu(A + B) \neq \mu(B + C)$ [Property 4]

Coupling between Object (CBO): This functional metric involves the uniqueness and the interaction of the complexity of the class-type with other class-types, showing the level of coupling. Coupling of the class-types in the schema determines the reusability of the component class and the rigidity with which the component class-type be modified. In addition to properties 2 and 4, property 3 also holds good for CBO as the class-types can be combined.

$\forall$ *Classes A and B;* $\mu(A) \neq \mu(B)$; *and* [Property 2]

$\exists$ *A,* $\exists$ *B and* $\exists$ *C, such that* $\mu(A) = \mu(B)$

$\mu(A) \leq \mu(A+B), \quad \mu(B) \leq \mu(A+B)$ [Property 3]

$\mu(A + B) \neq \mu(B + C)$ [Property 4]

Number of Inherited Properties (NIP): In contract to NOC metric in OO design [7] this metric measures number of properties of a class-type that could be inherited. The properties declared public are inherited and thus interaction among the class-types in the inheritance tree. Since inheritance is a form of reuse [7] the properties 1, 2 and 3 hold good, satisfying the componential trait.

$\exists$ *A and* $\exists$ *B, such that B is the root and A is the leaf node,*

Then $\mu(A) \neq \mu(B)$ [Property 2]

If A and B are siblings, then $\mu(A) = \mu(B)$ [Property 1]

If B is a subclass of A, then

$\mu(A) \leq \mu(A+B), \quad \mu(B) \leq \mu(A+B)$ [Property 3]

Depth of Relational Tree (DRT): The metrics is a design-level attribute satisfying property 1, considered to determine the reusability of the class-type, and hence satisfies the property 4. As they don't have functional significance property 2 is ignored. Combination of two class-types or tables, will remove them from the tree and hence the metric becomes insignificant during combination.

Referential Degree (RD): This design-level metric obviously satisfies property 1, since the references are made during the design of the class-type. However, during combination of individual component classes, $\mu(A+B) = \mu(A) + \mu(B) - \lambda$, Where λ is the common references, thus satisfying property 3.

$\exists$ *A and* $\exists$ *B, such that* $\mu(A) = \mu(B)$, *then*

$\mu(A) \leq \mu(A+B), \quad \mu(B) \leq \mu(A+B)$ [Property 3]

Using the four evaluation properties, we say that each of the metric is reliable enough to measure the relevant attributes. These properties are helpful in identifying the purpose of the metric and in removing redundancy in the metric values.

TABLE VI
SCALABLE VALUES FOR THE ORDB METRICS

	TS	CWM	COM	CBO	NIP	DRT	RD
Max	65.24	55.32	0.98	0.96	10.7	6	12
Min	59.12	54.58	0.66	0.68	7.82	2	4
Mean	62.18	54.95	0.82	0.82	9.26	4	8
Std Dev	3.06	0.37	0.16	0.14	1.44	2	4

2) Scale Calibration Approach

The reliability of the metrics is assessed by deriving scales, indicating the maximum and minimum range of measurement for each individual metric. The results presented by C&K [7] and Annie Mitchel [15], [16] metrics for OO system were validated against scales, and in this work we relatively propose nominal and summative scales for the evaluated ORDB metrics. Table 6 shows the scalable values for each metric.

The metrics TS, CWM and RD are design metrics which measures the design level structural complexity of the schema, COM and CBO measures the class-level functional complexity of the class-types, and the metrics NIP and DRT measures the componential nature of a class-type. Hence we arrive at the following scales for the metrics.

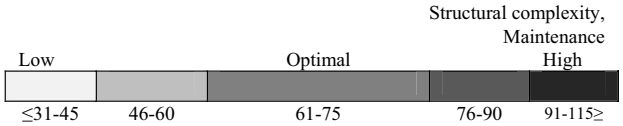

Figure 5(a): TS nominal scale

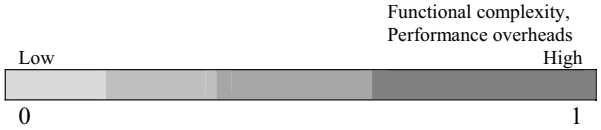

Figure 5(b): COM, CBO summative scale

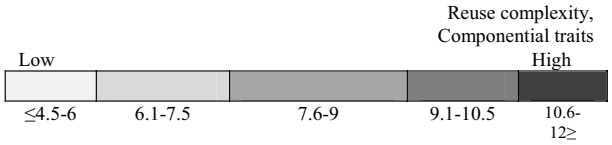

Fig. 5 (c): NIP nominal scale

In figures 5(a), (b) and (c), we carefully suggested the calibrated scale for the four metrics for object-relational databases. The design-level structural metrics serve as indicators for maintenance cost and internal structural complexity of the class-type. The functional level metrics, cohesion and coupling, serve as indicator for tuning the performance of the database. This is discussed widely in our earlier works. The componential metrics, NIP, serve as indicators for reusability of a class type, or a database table. The remaining metrics CWM, DRT and RD serve as supportive measures in calculating TS and NIP metric, and hence scale calibration to a scale is not required for these metrics.

The metrics are assessed and calibrated based on their content and criteria related validity which validates the measurability, consistency and the reliability of the metrics. Scaling of the metrics is also derived and proposed against which the measures can be assessed in their scope of applicability.

IV. CONCLUSION AND FUTURE WORKS

Object-relational database metrics require formal validation and verification as they determine the structural, functional and componential nature of the ORDB schemas. They are used in assuring the internal quality of the database design, which on further study indicate the usability, maintainability, understandability and reliability that assure the external quality of the ORDB schema. In this work we have formally defined seven ORDB metrics and described the calculability of the metrics, thus providing the evaluation methodology. Then we validated the metrics based on its content validity, criteria related validity, which includes the verification of the measurability, consistency and the reliability of the metrics. The verification and validation are carried out using three ORDB schemas, which are designed for realtime applications. Finally we calibrated the metrics and derived scales for each metric indicating the usability of the metric values. This work can be extended by considering the following future issues.

Future Works: Apart from formalizing the ORDB metrics on the three traits, structural, functional and componential traits of ORDB, the following open issues can be considered for future works: 1) The compatibility of the ORDB objects with the programming languages, 2) The ability of the ORDB objects to represent knowledge models using XML, CG constructs, 3) The complexity of the objects with regard to performance and storage, and 4) Additional ORDB metrics to assess the time optimization and disk space utilization need to be formally defined

The features of object-relational database and models need to be explored, so that its power of handling data through objects may be appreciated. The componential trait of the OR schema and class-types has to be considered for measurement so as to produce componential OR schemas. The evaluation and the calibration of the metrics presented in this work is an instigation of research in object-relational databases. This work on the calibration of the device (metric) for ORDB will benefit further researches in the metric area and serve as a direction guide for assessing and scaling future metrics.

APPENDIX

Figures are enclosed at the end of the references

ACKNOWLEDGEMENTS

The authors wish to thank the K.L.N.C.I.T labs for providing experimental space and metric values. Also we thank Syras Technologies for providing technical support and metric values.

REFERENCES

[1] Henderson-Sellers, B. Object-oriented Metrics - Measures of complexity, Prentice-Hall, Upper Saddle River, New Jersey, 1996.
[2] Baroni. A.L, C, Calero, F.B. Abreu and Mario Piatini, "Object relational Metrics Formalization", in Proc. of the Sixth International Conference on Quality Software. 2006. Available: doi.ieeecomputersociety.org/10.1109/QSIC.2006.44.
[3] Baroni.A.L, C. Calero, F. Ruiz and F. Brito eAbreu, "Formalizing Object-Relational Structural Metrics", in Proc. 5th Portuguese Association of Information Systems Conference, November, 2004. http://ctp.di.fct.unl.pt/QUASAR/Resourses/Paper/2004/baroni5CAPSI.pdf
[4] Piattini M., Calero C., Sahraoui. H., Lounis H., "Object-Relational Database Metrics", L'object, March, 2001. Available: www.iro.umontreal.ca/~sahraouh/papers/lobjet00_1.pdf
[5] Justus S., "Metrics for Object Relational Databases", in Proc. International Conference on Recent Trends in Information Systems, ISBN: 81-7764-954-X, 483-496, 2006.
[6] Justus S, and Iyakutti. K., "Assessing the Object-level behavioral complexity in Object Relational Databases", Conf. Rec. 3rd IEEE sponsored International Conference on Software Science, Technology and Engineering, Israel, 2007, pp. 48-59, Available:doi.ieeecomputersociety.org/10.1109/SWSTE.2007.6
[7] Chidamber. S.R and Kemerer. C.F., A metrics suite for object oriented design. IEEE Transactions on Software Engineering, Vol. 20, 1994, pp.476-493.
[8] Ammar. H.H., S.M. Yacoub and T. Robinson, "Dynamic metrics for object-oriented designs," in Proc. The 5th International Software Metrics Symposium, Boca Raton, Florida, USA, 1999, pp. 50–61.
[9] Arisholm. E, L.C. Briand, and A. Foyen, 2004. "Dynamic coupling measures for object-oriented software", IEEE Transactions on Software Engineering, Vol. 30, No. 8, pp. 491–506, 2004.
[10] Baroni A. L, "Formal Definition of Object-Oriented Design Metrics", Master Thesis. Vrije Universiteit Brussel – Belgium, 2002.
[11] Berard E. "Metrics for Object-Oriented Software Engineering," an Internet posting on Computer Software Engineering, January 28th, 1995.
[12] Moris. K., "Metrics for object oriented software development", Masters Thesis, M.I.T Sloan School of Management, Cambridge, MA, 1998
[13] David P, C.F. Kemerer, Sandra A.S., and James E.T., "The Structural Complexity of Software: An Experimental Test", IEEE Trans on Software Engineering, Vol. 31, No. 11, 2005, pp.982-995.
[14] Steve Counsell and Stephen Swift, "The Interpretation and utility of three cohesion metrics for object-oriented design", ACM Trans on Software Engineering and Methodology, Vol.15, No.2, 2006, pp.123-149.
[15] Mitchell and J.F. Power, "Toward a definition of run-time object-oriented metrics", in Proc. 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'2003), Darmstadt, Germany, July 2003.
[16] Mitchell and J.F. Power, 2004. "Run-Time Cohesion Metrics: An Empirical Investigation", in Proc. International Conference on Software Engineering Research and Practice, Las Vegas, Neveda, USA, June 2004, pp.532-537.
[17] Tobias Mayer, Tracy Hall, "A Critical Analysis of Current OO Design Metrics", Software Quality Journal, Vol. 8, No.2, Jun 1999.
[18] Elmasri. R and S. Navathe, Fundamentals of Database systems, Fourth edition, Addison-Wesley, Massachussets, 1999.
[19] Wook-Shin Han, Kyu-Young Whang, Yang-Sae Moon, "A Formal Framework for Prefecting based on the Type-Level Access Pattern in Object-Relational DBMSs", IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.10, 2005, pp.1436-1448.
[20] Calero, C, Ruiz, F., Baroni, A., Brito e Abreu, F., Piattini, M., "An Ontological approach to describe the SQL: 2003 Object-Relational Features", International Journal Computer Standards & Interfaces, Vol. 28, 2006, pp.695-713.
[21] Michura. J, Capretz. M.A.M., "Metrics Suite for Class Complexity", in Proc. International Conference on Information Technology: Coding and Computing (ITCC'05), 2005.
[22] Pressman R.S., Software Engineering: A Practitioners' Approach, McGraw-Hill International Edition, Fifth Edition, 2001.
[23] Bunge. M., Treatise on Basic Philosophy: Ontology II: The World of Systems, Boston, Riedel, 1979.
[24] Baroni, A.L., Coral. C., Mario Piattini and Abreu. F.B., "A Formal Definition for Object Relational Database Metrics", Url: http://:ctp.di.fct.unl.pt/QUASAR/Resources/Papers/2005/baroniICEIS.pdf, Visited: January, 2008.
[25] Justus S, and Iyakutti. K., "An Empirical Investigation on the Understandability of ORDB metrics", in Proc. International Conference on Modeling and Simulation. India, 2007.
[26] Kothari. C.R., Research Methodology, Methods and Techniques, Wiley Eastern Limited, ISBN 0852264771, 1990.
[27] E. Weyuker, "Evaluating software complexity measures," IEEE Trans. Software Eng., vol. 14, 1988, pp. 1357-1365.
[28] Zusc. H., "Properties of Software measures", Software Quality Journal, Vol. 1, 1992.
[29] Cherniavsky. J.C and C.H. Smith, "On Weyuker's axioms for software complexity measures", IEEE Transactions on Software Engineering, Vol. 17, 1991, pp.197-211.
[30] Justus S, and Iyakutti. K., "Object Relational Database Metrics: Classified and Evaluated", in Proc. of International Workshop on Software Engineering, ISBN-10: 3-8322-5611-3, Potsdam, Germany, 2006, pp. 119-131.
[31] Justus S, and Iyakutti. K., "The Theory of Time and Space in Object Relational Databases", Conf. Rec. IEEE International Conference on Computing Intelligence and Multimedia Applications, India, 2007, pp.575-580, doi.ieeecomputersociety.org/10.1109/ICCIMA.2007.410

Justus S received his Masters in Computer Applications from Madurai Kamaraj University, Madurai, India, and currently pursuing his PhD in the same University. His areas of interest are knowledge management, software process improvement and software metrics. He has research publications in International Journals and in IEEE/ACM organized International Conferences. He has visited Germany and Israel to deliver his research lecturers in International Conferences. He is a member of IEEE and is also a member of reviewers in International Journals. Currently, he is serving as Assistant Professor in the Department of Computer Applications, KLN College of Information Technology, India.

Iyakutti K obtained his PhD in 1974. His wide area of research interests has brought out 70 international publications in journals and conferences. His current research interests are software metrics, process improvement, intelligent systems and knowledge management systems. He has been to countries like Italy, Canada, Japan to present his research projects.
Presently, he is the Senior Professor in the Department of Microprocessor and Computers, School of Physics and also the Dean in Madurai kamaraj University, Madurai, India.

APPENDIX

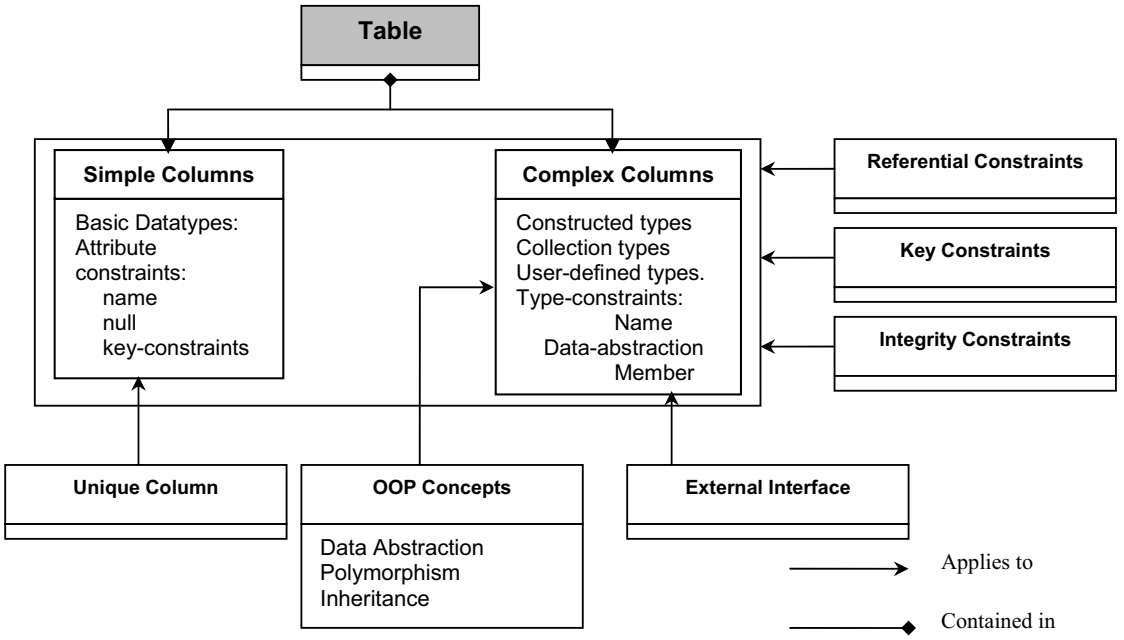

Fig. 1 ORDB Schema Ontology

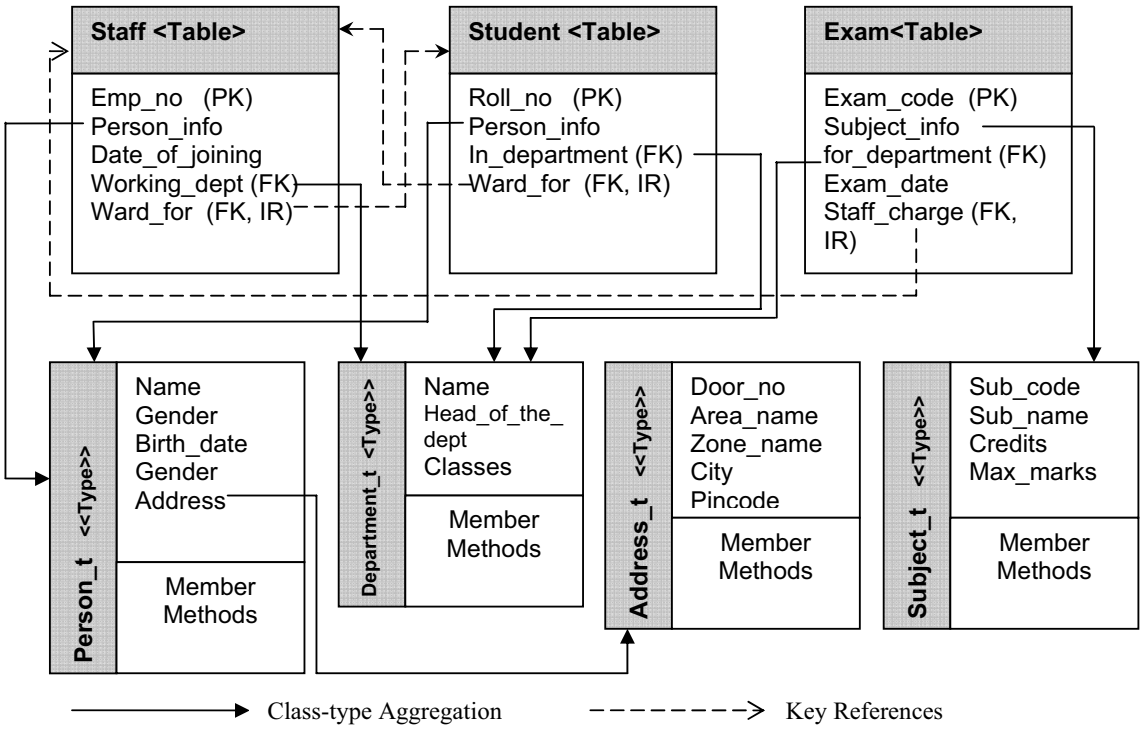

Fig. 2 Sample Schema for the Metrics Evaluation