ASOO 99

Proceedings

Argentine Symposium on Object Orientation

28 JAIIO

28 Jornadas Argentinas
de Informática
e Investigación Operativa

IFIP - Technical Committee 6
Communication Systems de IFIP
International Federation
for Information Processing

Sociedad Argentina de Informática
e Investigación Operativa.

Contents

SELECTED PAPERS

- *Implicit Object Persistence on a Reflective Abstract Machine* *1*
  F. Ortín Soler, A.B. Martínez Prieto, D. Álvarez Gutiérrez, J.M. Cueva Lovelle, University of Oviedo, Spain
- *Towards a Framework for Multi-Agents Systems* *19*
  H. Avancini; A. Amandi, UNCOMA/UNICEN, Argentina
- *Object Orientation in the TANGOW System: Working with Dynamic Task Trees* *35*
  R. M. Carro; E. Pulido; P. Rodríguez, UAM, Spain
- *A Framework for Distributed Task Management* *47*
  F. García; R. Moriyón, UAM, Spain
- *Reusable Coordinated Concurrent Objects with Coordinated Roles* *59*
  J. Murrillo; J. Hernandez; A. Gómez; F. Sánchez, University of Extremadura, Spain
- *Reuse of Pattern Scenarios in Business Processes* *73*
  M. Paludo; R. Burnett; E. Jamhour, PUC-PR, Brazil
- *Dynamical object generation during the execution of continuous simulation models* .. *89*
  M. Alfonseca; J. Lara; E. Pulido, UAM, Spain
- *Enhancing Framework Usability through Smart Documentation* *103*
  A. Ortigosa; M. Campo; R. Moriyon, UAM/UNICEN, Spain/Argentina
- *Measuring Complex Columns in Object-Relational Databases* *119*
  C. Calero; M. Piattini; M. Polo; F. Ruiz; M. Genero, University of Castilla-La Mancha, Spain
- C++ *Idioms for Concurrent Operations* *129*
  F. Náufel do Amaral; P. Garcês Rabelo; S. E. R. de Carvalho, PUC-RJ, Brazil
- *Workflow Management Systems. An Application of Smart Objects Paradigm* *141*
  S. G. Figueroa; D. Palliotto, UNSE, Argentina

INVITED TALKS

- *Software Product Lines: A New Paradigm for the New Millenium* *153*
  Dr. P. Clements. SEI, Carnegie Mellon University, USA
- *The ArchitectureTrade-Off Analysis Method (ATAM)* *154*
  Dr. Len Bass. SEI, Carnegie Mellon University, USA

Measuring Complex Columns in Object-Relational Databases[1]

Coral Calero, Mario Piattini, Macario Polo, Francisco Ruiz, Marcela Genero

Grupo ALARCOS
Departamento de Informática
University of Castilla-La Mancha
Ronda de Calatrava, 5
13071, Ciudad Real (Spain)
e-mail: {ccalero, mpiattin, mpolo, fruiz, mgenero}@inf-cr.uclm.es

Abstract: Object-relational databases are just being adapted by enterprises. This technology could raise some difficulties to users because of the richer semantics and higher complexity of the database schema, which has been characterised by its simplicity in the classical relational model. Table attributes in the object-relational model can be defined in a simple domain (simple columns) or in a user-defined class (complex columns). These classes can also be part of a generalisation lattice. In this paper we propose a suite of metrics for controlling the schema database complexity and we characterise the complex column size metric in the formal framework of Zuse [1].

Keywords: Object-Relational databases, Metrics

1 Introduction

In the last few years, we are witnessing important advances in database technology; a new "generation" of DBMS (Database Management System) is coming out, among which stand out the object-relational ones. Moreover, the object-relational databases will replace relational systems to become the next great wave of databases [2] so, it is very important to have metrics for this kind of databases.

Metrics for databases have been neglected in the metric community [3]. Most all of the metrics proposed from the McCabe famous cyclomatic number [4] until today have been centred in measuring programs complexity, quality, maintenance, etc. However, in modern Information Systems (IS) the database has become a crucial component, so there is a need to propose and study some measures to assess the database quality.

Metrics could be used for building prediction systems for database projects [5], for understanding and improving database development and maintenance projects [6], for maintaining the quality of the systems [7], for zooming in on problematic areas [8], and for determining the best ways to help practitioners and researchers in their work [9].

---

[1] This research is part of the MANTICA project and is partially supported by the CICYT and the European Union (CICYT-1FD97-0168)

Because of these reasons, we think it is very important to measure databases and understand their contribution to the overall IS complexity. We must be conscious, however, that a general complexity measure is "*the impossible holy grail*" [10]. Henderson-Sellers [11] distinguishes three types of complexity: computational, psychological and representational, and for psychological complexity three components are considered: problem complexity, human cognitive factors and product complexity. The last one is our focus.

In this paper different metrics to measure object-relational databases are proposed. In the next section the object-relational databases are presented. In section 3, we describe the proposed metrics. Complex column size metric verification in the framework of Zuse [1] is presented in section 4. Finally, section 5 summarises the paper and draw our conclusions.

2 Object-Relational Databases

Object-relational databases combines the traditional database characteristics (data model, recovery, security, concurrency, high-level language, etc.) with object-oriented principles (e.g. encapsulation, generalisation, aggregation, polymorphism,...). These products offer the possibility of defining classes or abstract data types, in addition to relations, domains and constraints[2], as relational databases.

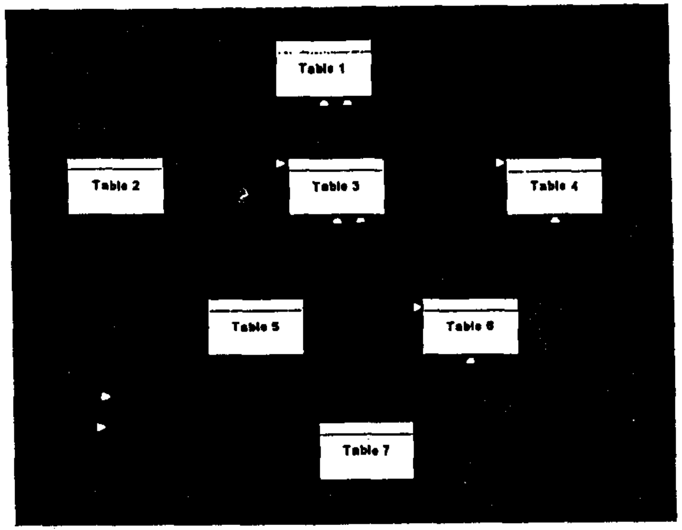


Figure 1. Example of an database object-relational schema

Also, generalisation hierarchies can be defined between classes (super and subclasses) and between tables (CREATE TABLE subtable UNDER supertable). Then, two types of associations can be established between tables, as shows figure 1.

Object-relational databases support usually multiple inheritance (a subtable can be defined with more than one supertable). Table columns can be defined in a simple domain, e.g. char (25), or in a user-defined class as complex number or image, see figure 2. These classes can also be part of a generalisation lattice. Classes are composed by attributes and methods. Attributes can be simple or complex (references to other classes).

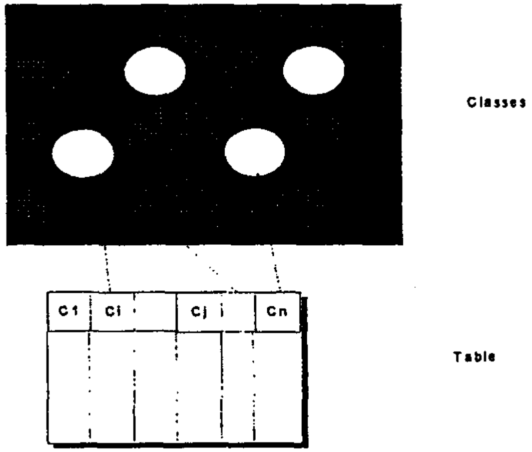


Figure 2. Example of complex column definition

3 Proposed Metrics

We define the total size of complex columns defined as the sum of each complex column size (CCS):

$$TSCC = \sum_{i=1}^{NCC} CCS_i$$

Where NCC the number of complex columns in the table.

The value for CCS is obtained with:

$$CCS = \frac{SHC}{NCU}$$

Where SHC the size of the hierarchy above which the column is defined and NCU is the number of columns defined above this hierarchy. This expression is due to the fact that the understandablility is less if more than one column is defined above the same class. If the number of columns that are defined above a class is greater than one, the complexity of this class decreases (respect to each column, but not for the

---

are not considered for measures

total columns) and this fact must be appointed when we calculate the complexity of a class.

The SHC may be defined as the sum of each class size in the hierarchy (SC):

$$SHC = \sum_{i=1}^{NCH} SC_i$$

Where NCH the number of classes in the hierarchy.

The size of a class is defined as:

$$SC = \frac{SAC + SMC}{NHC}$$

Where SAC the sum of the size attributes of the class, SMC the size methods of the class and NHC the number of hierarchies to which the class pertain (if we consider that multiple inheritance is allowed).

The attributes of a class may also be simple or complex (which can be a class or an UDT), then the SAC is defined as the sum of the simple attributes size (SAS, that have size equal to one then the metric corresponds with the number of simple attributes) and the complex attributes size (CAS) in the class.

And the SMC is calculated with the version of the cyclomatic complexity of McCabe given by Li and Henry [12]:

$$SAC = SAS + CAS$$

$$SMC = \sum_{i=1}^{NMC} V_i(G)$$

Where NMC the number of methods in the class

Other metrics can de very useful to manage object-relational database complexity. In one hand, we have propose two metrics related to referential integrity: DRT (Depth referential tree) and RD (referenciability Degree), which are analysed in [13].

In the other hand, table generalisation hierarchy can be characterised using Chidamber and Kemerer [14] DIT(Depth Inheritance Tables), CBO (Coupling Between Objects) and NOC (Number of Children). These metrics are analysed in [1].

4 Formal Description of the Proposed Metrics

In this paragraph we present the formal description of the proposed metrics in the formal framework of Zuse who defines a set of properties for measures, which characterise different measurement structures. This framework is based on an extension of the classical measurement theory, which gives a sound basis of software measures, their validation and criteria for measurement scales.

An empirical relational system is represented as:

A = (A, • >=, o)

Where A is a no empty set of objects, • >= is a empirical relation in A and o is a binary closed operation (concatenation) in A.

The most important structures defined by Zuse are the following:

4.1 Modified Extensive Structure

Axiom1: (A, • >=) (weak order)
Axiom2: A1 o A2 • >= A1 (positivity)
Axiom3: A1 o (A2 o A3) ≈ (A1 o A2) o A3 (weak associativity)
Axiom4: A1 o A2 ≈ A2 o A1 (weak commutativity)
Axiom5: A1 • >= A2 ⇒ A1 o A • >= A2 o A (weak monotonicity)
Axiom6: If A3 • > A4 then for any A1, A2, then there exists a natural number n, such that A1o nA3 • >A2 o nA4 (Archimedean axiom)
As we know, binary relation • >= is called weak order if it is transitive and complete:
A1 • >= A2, and A2 • >= A3 ⇒ A1 • >= A3
A1 • >= A2 or A2 • >= A1

4.2. Independence Conditions

The independence conditions are assumed by the extensive structures and are a prerequisite for the existence of a combination rule:

C1: A1 ≈ A2 ⇒ A1 o A ≈ A2 o A and A1 ≈ A2 ⇒ A o A1 ≈ A o A2
C2: A1 ≈ A2 ⇔ A1 o A ≈ A2 o A and A1 ≈ A2 ⇔A o A1 ≈ A o A2
C3: A1 • >= A2 ⇒ A1 o A • >= A2 o A, and A1 • >= A2 ⇒ A o A1 • >= A o A2
C4: A1 • >= A2 ⇔ A1 o A • >= A2 o A, and A1 • >= A2 ⇔A o A1 • >= A o A2
Where A1 ≈ A2 if and only if A1 • >= A2 and A2 • >= A1, and A1 • > A2 if and only if A1 • >= A2 and not (A2 • >= A1).

4.3 Modified Relation of Belief

Zuse introduces this new structure in order to characterise object-oriented measures. These measures do not assume the independence conditions or even the extensive structure. In this case the measure is u: ℑ -> ℜ, where ℑ is the set of finite subsets of a countable set X.

When the metric do not accomplishes the extensive structure, it will be necessary to verify everyone of the modified structure of belief axioms:

MRB1: ∀ A, B ∈ ℑ: A • >= B or B • >= A (completeness)
MRB2: ∀ A, B, C ∈ ℑ: A • >= B and B • >= C ⇒ A • >= C (transitivity)
MRB3: ∀ A ⊇ B ⇒ A • >= B (dominance axiom)

MRB4: $\forall (A \supset B, A \cap C = \phi) \Rightarrow (A \bullet >= B \;\Rightarrow A \cup C \;\bullet > B \cup C)$ (partial monotonicity)
MRB5: $\forall A \in \Im: A \bullet >= 0$ (positivity)

It is important to note that when a metric accomplishes the weak order of the extensive modified structure axiom, it also accomplishes the completeness and the transitivity axioms of the belief structure.

Finally, we must remember that there exist five scale types, which in hierarchical order are: nominal, ordinal, interval, ratio and absolute. Each scale type is defined by admissible transformations. Software measurement starts with the ordinal scale.

Measures may be classified in a scale type, depending on whether they assume an extensive structure or not. When a measure accomplishes this structure, it also accomplishes the independence conditions and can be used on the ratio scale levels.

If a measure does not satisfy the modified extensive structure, the combination rule (that describes the properties of the software measure clearly) will exist or not depending on the independence conditions. When a measure assumes the independence conditions but not the modified extensive structure, the scale type is the ordinal scale (the characterisation of measures above the ordinal scale level is very important because we cannot do very much with ordinal numbers).

In the next paragraph we present the formal description of the complex column size metric (TSCC), in order to made it, the steps that we must to follow are: define the concatenation operation and the combination function, and prove the modified extensive structure, the independence conditions and the structure of belief.

4.4 TSCC metric

The TSCC measure is a mapping: TSCC: $R \rightarrow \Re$ such that the following holds for all relations Ri and Rj $\in$ R: Ri $\bullet >=$ Rj $\Leftrightarrow$ TSCC(Ri) >= TSCC(Rj).

The number of attributes when we combine two tables with natural join may be the sum of the attributes of the two tables (cartesian product) or may be this sum minus the number of common attributes (natural join by columns or by foreign key-primary key). So, we can define the combination rule for TSCC as:

$$\text{TSCC}(Ri \circ Rj) = \text{TSCC}(Ri) + \text{TSCC}(Rj) - \text{TSCC}(Ri \cap Rj)$$

Where TSCC(Ri $\cap$ Rj) is the number of attributes which are common to (belong to the intersection[3] of) Ri and Rj.

TSCC fulfils the first axiom of weak order, because if we have two relations R1 and R2, it is obvious that TSCC(R1) >= TSCC(R2) or TSCC(R2) >= TSCC(R1) (completeness) and let R1, R2 and R3 three relations, transitivity is always fulfilled: TSCC(R1) >= TSCC(R2) and TSCC(R2) >= TSCC(R3), then TSCC(R1) >= TSCC(R3). TSCC do not fulfil positivity, because if we combine a relation R1 with itself without cycles: TSCC(R1 o R1) is not greater than TSCC(R1). But it fulfils weak positivity , because it is always true that: TSCC(R1 o R2) >= TSCC(R1) for all R1, R2 $\in$ R. TSCC fulfils axiom 3, because the natural join operation is associative and so TSCC(R1 o (R2 o R3)) $\approx$ TSCC((R1 o R2) o R3). TSCC also fulfils weak commutativity because natural join is commutative. TSCC but not fulfils weak monotonicity as we see in the figure 3.

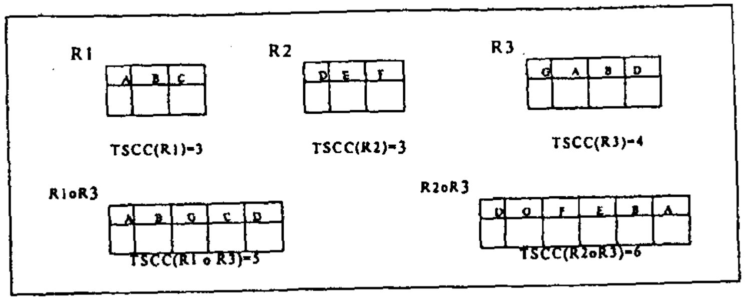
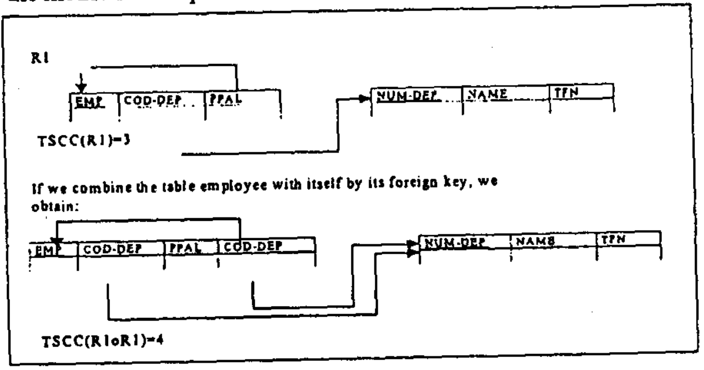


Figure 3.- TSCC does not fulfil weak monotonicity

Previously to verify if TSCC accomplish the Archimedean axiom, we will verify, with the figure 4, if the metric is idempotent:



If we combine the table employee with itself by its foreign key, we obtain:

Figure 4. Idempotency of a table

It is easy to see that the number of attributes vary when we combine one table with itself, so it is not idempotent and is necessary to prove the Archimedean axiom. When two tables are combined by natural join successively, the number of attributes vary and may be possible to obtain two tables R1, R3 that do not accomplish the Archimedean axiom, as we can see in figure 5, where the result of the first concatenation is shown (the tables obtained in successive concatenations will be the same). Measure TSCC does not assume an extensive structure.

---

[3] This intersection is different from the relational algebra operation of Intersection

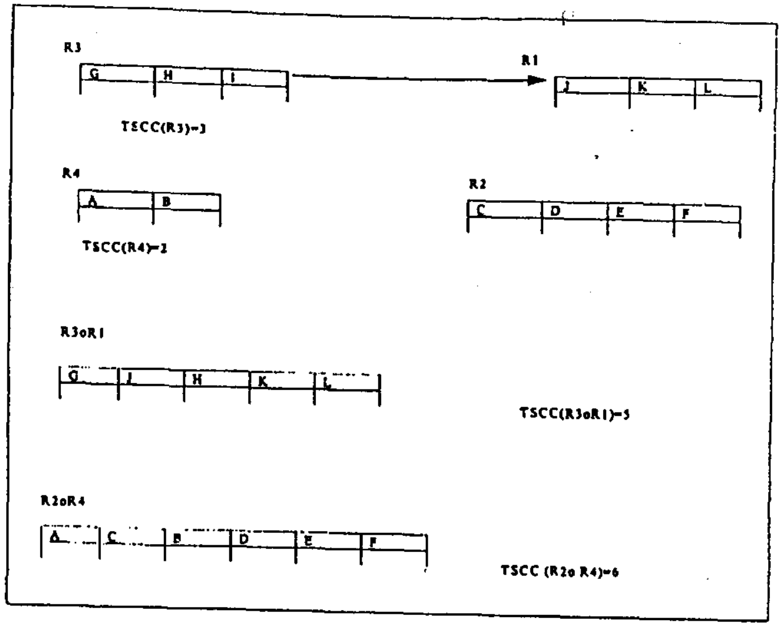


Figure 5. Archimedean axiom

Would TSCC verify the independence conditions?. With figure 4 we can ensure that TSCC do not accomplish the first condition, or the second one. TSCC do not accomplish the weak monotonicity, so it cannot accomplish the third and the fourth independence conditions. In fact, this type of combination rules do not assume the independence conditions. The part $-TSCC(R_i \cap R_j)$ rejects the condition C1 that implies the rejection of the axiom of weak monotonicity, monotonicity and extensive structure.

Then we must study if TSCC fulfil some of the modified relations of belief. MRB1 is fulfilled, because giving two relations R1 and R2 $\in \Im$ ($\Im$ is the set of all the possible relations made with the attributes of the relational schema) TSCC(R1) >= TSCC(R2) or TSCC(R2) >= TSCC(R1). MRB2 is also fulfilled (transitivity of natural join). For MRB3 we will consider that a relation R1 $\supseteq$ R2 if all the attributes of R2 are present in R1. In this case it is evident that TSCC(R1) >= TSCC(R2), and MRB3 is fulfilled. MRB4 is fulfilled because if a relation R1 $\supset$ R2 then TSCC(R1) > TSCC(R2) and TSCC(R1 U R3) > TSCC(R2 U R3), where R1 $\cap$ R3 = ϕ. If the relations R1 and R3 do not have any attribute in common, adding the attributes of R3 to both R1 and R2, (if R1 subsumes R2), then the number of attributes of R1 and R3 is greater than the number of attributes of R2 and R3. MRB5 is fulfilled because a relation must always have zero or more attributes. Then, we can conclude that TSCC accomplish all the structure of belief axioms as says Zuse respect to the metrics that have function of belief (like TSCC metric). In summary, we can characterise TSCC as a measure above the level of the ordinal scale, assuming the modified relation of belief.

4.5 Rest of metrics

In table 1, we summarise the results obtained for the rest of metrics presented in this work. The formalisation of the metrics marked with a asterisk (*) is included in [1].

	CONCAT OPER	COMBINATION RULE	MOD EXT STRUCTURE						INDEP COND				STRUC BELIEF					SCALE
			1	2	3	4	5	6	1	2	3	4	1	2	3	4	5	
SCC	Natural join	TSCC(RioRj)=TSCC (Ri)+TSCC(Rj) - TSCC(Ri ∩ Rj)	Y	N	Y	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Ordinal
CCS	Integration	CCS(RioRj)=CCS (Ri)+CCS(Rj)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Ratio
SHC	CUNI	*	Y	Y	Y	Y	N	N	N	N	N	N	Y	Y	Y	N	Y	Ordinal
SC	CUNI	*	Y	Y	Y	Y	N	N	N	N	N	N	Y	Y	Y	N	Y	Ordinal
SAC	Natural join	SAC(RioRj)=SAC (Ri)+SAC(Rj)- SAC(Ri ∩ Rj)	Y	N	Y	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Ordinal
SAS	Natural join	SAS(RioRj)=SAS (Ri)+SAS(Rj)- SAS(Ri ∩ Rj)	Y	N	Y	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Ordinal
CAS	Natural join	CAS(RioRj)=CAS (Ri)+CAS(Rj)- CAS(Ri ∩ Rj)	Y	N	Y	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Ordinal
SMC	Vij(G)	*	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Ratio
DRT	Natural join	DRT(RioRj)=max (DRT(Ri),DRT(Rj)) - v	Y	N	Y	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Ordinal
RD	Natural join	RD(RioRj)=RD(Ri)+ RD(Rj)-v	Y	N	Y	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Ordinal

Table 1. Results for other metrics

5 Conclusions and future work

We have presented a first approximation for measuring object-relational databases. A metric for measuring the complex columns size of a table is defined.

However, the framework used is not the only one, see for example [15, 6, 16] and it is not generally accepted [17]. So we must validate these metrics using other axioms definition.

The presented measures do not assume an extensive structure but can be characterised above the ordinal scale by fulfilling all the properties of the modified structure of belief, so, the object-relational databases measures assume, as object-oriented measures [1], more complex properties related to concatenation operation than classic measures.

In this moment, the metrics for object-relational databases are complemented with others for active databases [18].

Also empirical validation is being carried out (Calero et al., 1999), not only to prove metrics validity, but also to give some limits which can be useful for database designers. We must be conscious however, as [8] remarks that "associating with numeric ranges the qualifications *good* and *bad* is the hard part". Adaptation of these

improve the metric suite [19].

Finally, an automatic tool is being developed in order to automate the metric gathering for ORACLE 8 DBMS.

References

[1] Zuse, H.: A Framework of Software Measurement. Berlin, Walter de Gruyter (1998)

[2] Stonebraker, M. and Brown, P.: Object-Relational DBMSs tracking the next great wave, Morgan Kauffman Publishers, California (1999)

[3] Sneed, H.M. y Foshag, O.: Measuring Legacy Database Structures. Proc of The European Software Measurement Conference FESMA 98, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.) (1998) 199-211

[4] McCabe, T.J.: A complexity measure. IEEE Trans. Software Engineering Vol 2, N° 5, (1976) 308-320

[5] MacDonell, S.G., Shepperd, M.J. and Sallis, P.J.: Metrics for Database Systems: An Empirical Study. Proc. Fourth International Software Metrics Symposium – Metrics'97, Albuquerque. IEEE Computer Society, (1997) 99-107

[6] Briand, L., Morasca, S. and Basili, V.: Property-Based Software Engineering Measurement. IEEE Transactions on software Engineering, Vol. 22, No 1, January, (1996)

[7] Graham, I.: Migrating to object technology. Wokingham, Addison-Wesley (1995)

[8] De Champeaux, D.: Object-oriented development process and metrcis. Upper Saddle River, Prentice-Hall (1997)

[9] Pfleeger, S. L.: Assessing Software Measurement. IEEE Software. March/April, (1997) 25-26

[10] Fenton, N.: Software Measurement: A Necessary Scientific Basis. IEEE Transactions on Software Engineering, Vol 20, N° 3, (1994) 199-206

[11] Henderson-Sellers, B.: Object-Oriented Metrics. The object-oriented Series. Prentice Hall. (1996)

[12] Li, W. and Henry, S.: Object-Oriented metrics that predicts maintainability. *J.Sys.Software*, 23, (1993) 111-122

[13] Calero, C. , Piattini, M., Polo, M., Ruiz, F.: Validating referential integrity as a database quality metric. Proceedings of the First International Conference on Enterprise Information Systems (1999) 45-50

[14] Chidamber, S. and Kemerer, C.: Towards a metric suite for object.oriented design. Proc of OOPSLA (1991) 197-211

[15] Weyuker, E.J.: Evaluating software complexity measures. IEEE Transactions on Software Engineering Vol. 14, N° 9 (1998) 1357-1365

[16] Fenton, N. y Pfleeger, S. L.: Software Metrics: A Rigorous Approach 2nd. edition. London, Chapman & Hall (1997)

[17] Kitchenham, B.A., Stell, J.G.: The danger of using axioms in software metrics. IEEE Proc.-Soft. Eng., Vol. 144, N° 5-6 (1997) 279-285

[18] Díaz, O and Piattini, M.: Metrics for active databases maintainability. Accepted in CAISE'99. Heidelberg, June (1999) 16-18

[19] Churcher, N.J. and Shepperd, M.J.: Comments on A Metrics Suite for Object-Oriented Design. IEEE Trans. on Software Engineering Vol. 21, N° 3 (1995) 263-265

C++ Idioms for Concurrent Operations

Fernando Náufel do Amaral, Patricia Garcês Rabelo, Sergio E. R. de Carvalho*

Laboratório de Métodos Formais, Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente 225, Rio de Janeiro, RJ, 22453-900, Brasil
email: {fnaufel, rabelo, sergio}@inf.puc-rio.br

Abstract. Well-integrated development tools, allowing automatic code generation from visual representations of analysis and design decisions, are important assets in handling the complexities of today's software. This paper describes several message passing semantics for the expression of concurrency in a new object-oriented visual development system, along with the C++ idioms generated for asynchronous messages.

Keywords: Message semantics, concurrent operations, automatic code generation, development tools, C++ idioms

1 Introduction

Expected increases in developer productivity, thus reducing time to market, correctness by construction, and design and code reuse suffice to justify large investments in the construction of object-oriented CASE tools. The automatic generation of quality, executable code from visually represented designs is one of the most desirable aspects of such tools. In this paper we present C++ code automatically obtained from several message passing semantics available in a new development environment, 2GOOD/DDL.

2GOOD (Second Generation Object-Oriented Development) (Carvalho+98) is a new object-oriented design tool. 2GOOD designs are automatically represented as code in DDL (Design Description Language) (Carvalho97), a very high level intermediate language, to smooth out transformations from visual designs to C++. The translation from DDL to C++ is accomplished with TXL (Tree Transformation Language) (Cordy+95), a programming language whose basic paradigm involves the use of correctness-preserving transformation functions and rules to process input data after it has been converted to tree format.
