Information System Decomposition Quality

Dr. Nejmeddine Tagoug
Computer Science Department
Al Imam University, SA
najmtagoug@yahoo.com

Journal of Digital Information Management

ABSTRACT: *Object-oriented design is becoming very popular in today's software development. An object-oriented information system is decomposed into subjects, each subject is decomposed into classes of objects. Good object-oriented system design should exhibit high cohesion inside subjects and low coupling among subjects. Yet, few quantitative studies of the actual use of cohesion and coupling have been conducted at the system level. These two concepts are defined qualitatively, and only at the class level, not at the system level. In this work, metrics are introduced for cohesion and coupling and used to define a quality metric at the system level. The feasibility of the approach is demonstrated by an example using a real information system.*

Categories and Subject Descriptors
D.2.1 [Requirements/Specifications]; Methodologies, object-oriented: **D.2.2 Design Tools and Techniques** [Object-oriented design methods]; **K.6.4 [System Management]** Quality assurance

General Terms: Information Systems Models, Decomposition quality, Object oriented data managment

Keywords: Object oriented information systems, Systems quality, System architecture

Received: 11 March 2011, Revised 19 April 2011, Accepted 23 April 2011

1. Introduction

Object-oriented approach is becoming popular in the software development community as an alternative to structured analysis and design methodologies. The object-oriented paradigm is often connected with the bottom-up approach. This approach is characterized by two basic steps, the creation of classes of objects out of smaller ones (composition) and the creation of more specialized objects out of more general ones (specialization). The opposite of the bottom-up paradigm is the top-down paradigm, based on steps of decomposition and generalization. The decomposition step is characterized by the decomposition of the system into subjects and the decomposition of subject into classes and the decomposition of bigger and complex classes of objects into smaller and simpler ones. Generalization is the opposite of specialization, where common characteristics of several specialized classes are abstracted and summarized in a more general class. A subclass inherits the attributes and methods of the super class.

In the design phase, a system is decomposed into subjects. Each subject is decomposed into classes. Courtois notes the importance of decomposition [5]: "Decomposition has long been recognized as a powerful tool for the analysis of large and complex systems."

Despite the importance of decomposition, there are few guidelines to help us decompose a system into subjects. Also, it is well known that how a system is decomposed has a great influence on its quality, in particular its maintainability. It would be beneficial if the quality of a software system could be estimated early in its lifecycle, during the analysis and the design phases, before the implementation phase.

Object-oriented software metrics can provide a quantitative means to control the software development process and the quality of the software products. Good system design should consist of subjects with high cohesion and low coupling among subjects. Unfortunately these two criteria are defined qualitatively and only at a program level, inside a class and among classes [1, 2, 4]. They are not defined at a system level. This work adapts the cohesion and coupling metrics defined elsewhere for the procedural information systems decomposition at the system level [12] to the object-oriented systems, taking into account the object-oriented characteristics. These two metrics measure decomposition at a system level, inside a subject and among subjects. Schach notes in [11] that there is no difference between the cohesion of an object and the classical cohesion, i. e, the module cohesion and also the coupling between two objects can be expressed in term of classing coupling, i. e, the coupling between two modules. There is one form of coupling that is specific to objects; we call this inheritance coupling.

This paper is organized as follows: Section 2 presents a very brief summary of the need for research in this area. Section 3 describes the object- oriented information system model. Section 4 defines the cohesion and coupling metrics at the system level. Section 5 illustrates the application of the metrics on a real information system is presented and discussed. Conclusion and future work are presented in section 6.

2. Problem Overview

Object-oriented analysis and design focuses on objects as the primary agents involved in a computation; each class of data and related operations are collected into a single system entity, then classes are grouped into subject according to the Coad and Yourdon methodology [3].

During the design phase, the analyst organizes the classes of objects into logical groups (subjects) based on some criterion like execution time or data exchange. How to know that a design is good or bad? The analyst needs to use a way to measure the top-level design. Rombach points out [9]: "We need a way to characterize software designs based on architectural design measures".

The recent work in metrics of object-oriented development of by Bieman et Al. in [2] or by Lee et Al. [8] or by Hitz and Montazeri in [6] measure the class cohesion and coupling. These metrics focus on internal object structures that reflect the complexity of each individual entity, such as cohesion, and on external complexity that

measures the interactions among entities, such as coupling and inheritance. A major criticism of these metrics that they measure class cohesion and coupling at the end of the algorithmic design phase from program-design language documents. They are defined only at a class level and not at the system level.

In this paper, a quality metric at the system level is defined and illustrated by an application on real information system decomposition.

3. Object-Oriented System Design Model

The model is inspired from the Coad and Yourdon object-oriented analysis and design model [3] and is influenced by Courtois' quasi-decomposable systems model [5]. The components are defined below and illustrated with the graphical model in Figure 1.

3.1 Object-Oriented System Model

Object: An instantiation of some class which is able to save state and which offers a number of methods to examine or affect this state.

Class: a set of objects that share a common structure and a common behavior manifested by a set of methods.

Attribute: Defines the structural properties of classes.

Method: An operation upon object, defined as part of the declaration of a class.

Instance connection: The process of creating an instance of the object and binding or adding the specific data that one class needs with other classes.

Message connection: A request that a class makes to another class to perform a method.

Inheritance: A relationship among classes, wherein an object in a class acquires characteristics from one or more other classes.

Link: a link between two classes. There are two kinds of links: physical and conceptual.

A physical link represents a message connection or instance connection or inheritance (generalization specialization structure) or a whole part structure between two classes.

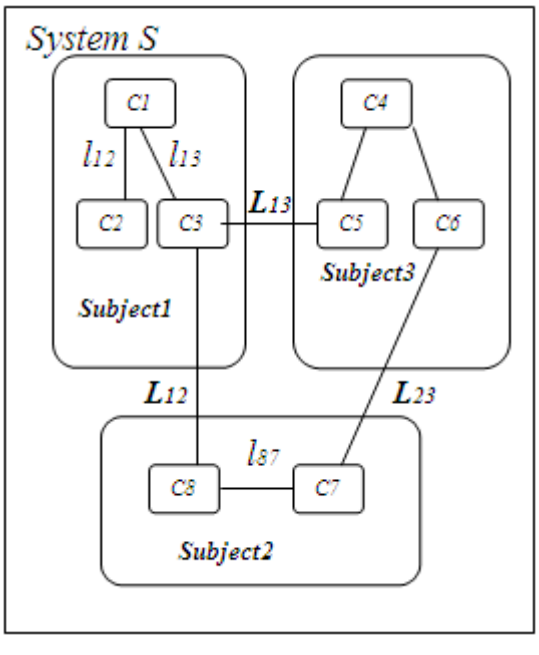

Figure 1. Example of an object-oriented system decomposition

There is a conceptual link between two classes if they share some property. For example if two classes are instantiated in the same slice time, they are linked by a conceptual time link.

System: a set of classes, linked together and contributing to a common goal.

Subject: a system linked to other subjects in a larger system.

Object-oriented system decomposition: a set of subjects such that each class is part of one subject only, and each internal link is either internal to a subject or between two classes of different subjects.

3.2 Decomposition Criteria

A decomposition criterion is some property of classes of objects allowing the analyst to decompose the system into subjects.

The most frequently used decomposition criteria are defined below:

- ***Data criterion***: all classes with share the same data are grouped into the same subject.
- ***Business function criterion***: all classes, which contribute to the same business function, are grouped into the same subject.
- ***Time criterion***: all classes, which are executed at the same slice of time, are grouped into the same subject.
- ***Organizational structure criterion***: all classes belonging to the same organizational unit are grouped together into the same subject.
- ***Behavior criterion***: each class behaves like a finite state machine. If a state change in one class induces state changes in other classes, these are grouped together in one subject.

4. Cohesion And Coupling Metrics

4.1 Introduction

Object-oriented analysis considers the systems behavior and data by looking for system entities that combine them. Object-oriented analysis and design focuses on objects as the primary agents involved in a computation; each class of data and related operations are collected into a single system entity. In the literature, many metrics are proposed to evaluate the OO concepts: methods, classes, cohesion, coupling, and inheritance [4, 2, and 8]. Many guidelines for object-oriented systems make use of coupling and cohesion as indexes to help construct good design [8], but two criticisms may be applied to current cohesion and coupling metrics. Fist they are defined only at the program level, not at the system level, while the metrics focus only on internal object structure for cohesion, and external measures of the interactions among entities for coupling. In this paper, to define system cohesion and system coupling metrics, the idea is based on a Courtois's Model [5] when he remarks: In general, interactions inside subsystems are stronger and more frequent than interactions among subsystems. This remark can be subdivided into two parts corresponding to the interactions of components inside and among subjects. (A subject corresponds to a subsystem).

- The interactions of components inside subjects are stronger and more frequent. That is **system cohesion.**
- The interactions of components among subjects are relatively weak and infrequent. That is **system coupling**.

The approach taken here is to develop metrics for measuring the quality of object-oriented decomposition information system. The metrics are defined in [11] for the procedural information system. In this work, the metrics are generalized for the object-oriented

information system taking into consideration some aspects of the object-oriented paradigm.

4.2 Subject Cohesion

Let S be a system and E be any set of classes of S. Lij is the set of all links between classes Pi and Pj. Wij is the sum of the weights of links in Lij $Wmax = \max\{Wij\}$ in system S: $Wmax$ is the largest weight of links between any two classes in S.

$$n = |E|$$

The cohesion of E is defined by the following formula (Formula (1)) for $n > 1$

$$C(E) = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Wij}{W\max * (n * (n-1)/2} \quad (1)$$

The term $n(n\text{-}1)/2$ represents the maximum number of possible links between classes in E. Cohesion values are therefore between 0 (no links between classes) and 1 (every class linked to all others by a link of maximal weight).

4.3 Cohesion And Coupling : Same Property

If no assumption is made on the nature of components and on the nature of links, cohesion and coupling measure the same thing, namely the strength of the relationships between the components.

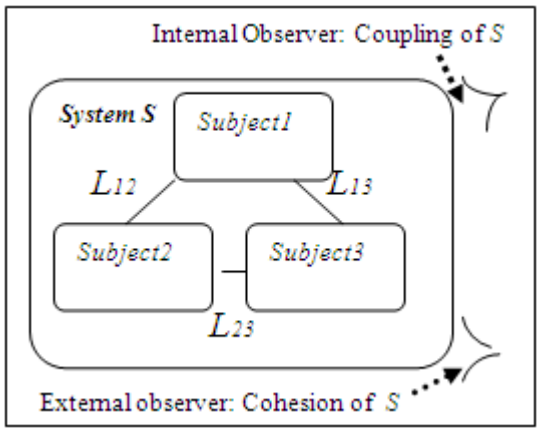

Figure 2. Cohesion and coupling: Same thing

Formula 1 measures also system coupling. Let us consider system S in Figure 2, and suppose that there is some formula for calculating the strength of the relationships between *Subject1*, *Subject2*, and *Subject3*. For an observer inside system S, the formula calculates the extent to which the components are tied together: that is component coupling. But an outside observer does not see the inside details of system S. For him, the formula calculates how "tight" system S is, how cohesive. In both cases, the property is the strength of the relationships. In one case (cohesion), it characterizes a whole (system S), in the other case, it characterizes a set of elements (coupling).

4.4 Subjects Coupling

Let us consider the decomposition shown in Figure 1.

System S is composed of subjects *Subject1*, *Subject2*, and *Subject3*. Links like l_{12}, l_{13}, are inside subjects while links like *L12*, *L13*, *L23* are between subjects because they connect classes located in different subjects.

Coupling between the subjects of S is obtained in two steps. First, an equivalent model of the decomposition is obtained as follows (see Figure 2 for the result).

1. Each subject is collapsed into a single element (like a class). Links internal to a subject are not considered anymore, links between subjects, which are now defined as links between the collapsed components, remain.
2. The coupling of the subjects of the system *S* is obtained by calculating the cohesion of the set of components resulting from the procedure described above, and by using the value of $Wmax$ as defined above.

4.5 Decomposition Quality Metric

A decomposition quality metric, inspired by the works of Karimi and Kosynski [7] is now defined, as the difference between the cohesion of the elements and their coupling (Formula 2):

$$Q(D(S)) = \frac{(\sum_{i=1}^{i=K} Cohesion_i) - Coupling}{K} \quad (2)$$

where:

$Cohesion_i$: cohesion of subject *i*.
$Coupling$: coupling between all the subjects.
K : number of subjects in decomposition $D(S)$.

It is easy to show that that the quality is between 0.5 and 1 ($0.5 \leq Q(D(S)) \leq +1$). The lowest value is obtained if K is minimal, i.e., $K = 2$, all cohesions are null and coupling is equal to 1. In this case, $Q(D(S)) = -0.5$. A negative value of $Q(D(S))$ indicates a "catastrophic" decomposition: inter-system coupling is stronger not only to the internal cohesion of each subject but worse, it is stronger than the sum of all cohesions. A quality value of 1 on the other hand, is obtained when all subjects have a maximal cohesion of 1 and there is no coupling at all between the subjects.

5. Applying The Quality Metrics

To illustrate the application of the metrics defined in this paper, a real information system based on a motor vehicle registration and title system is presented below. The system issues registration renewal notices for vehicle. Clerks are accountable for registrations and titles issued (plus the fees accepted). The system maintains information on 12 different classes.

Figure 3 depicts the graphical representation of the decomposition of the system based on a data criterion:

The system (S) is decomposed into 2 subjects:

- *Subject 1* is the Person subject, contains the following classes: *C1*: organization, *C2*: Person, *C3*: ClerkPerson, *C4*: OwnerPerson, *C5*: ClerkOwnerPerson.
- *Subject 2* is the *Vehicle* subject, contains the following classes: *C6*: *LegalEvent*, *C7*: *TitleLegalEvent*, *C8*: *RegistrationLegalEvent*, *C9*: *Vehicle*, *C10*: *TruckVehicle*, *C11*: *MotorcycleVehicle*, *C12*: *TrailerVehicle*.

The links inside *subject 1* are:

- There is a whole part link between classes C1 and *C3*: The class *Organization* (C1) is composed by many classes *Clerks* (*C3*).
- There are 5 message connection links between: *C2* and *C3*; *C1* and *C3*; *C3* and *C5*; *C4* and *C2*; *C4* and *C5*. For example, the class *Clerk* (*C3*) sends a message to the class *Person* (*C2*) requesting to access one method in *C2*.

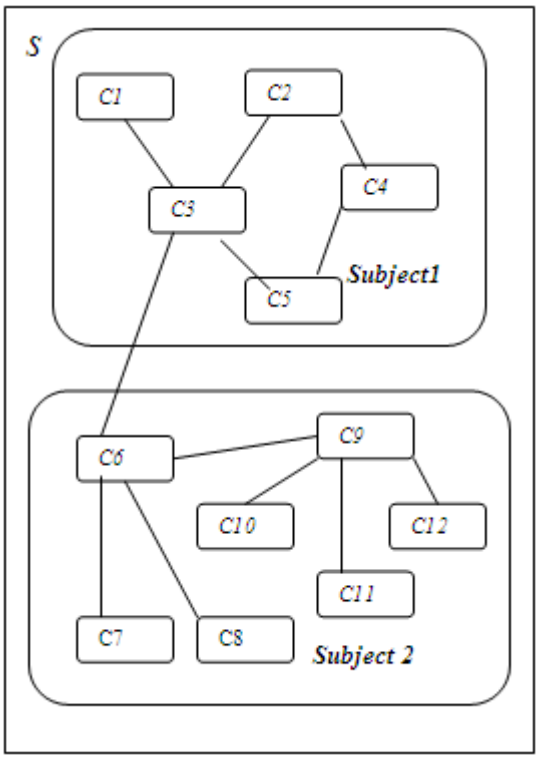

Figure 3. Vehicle Registration and Title System Data Decomposition

- There are 4 generalization specialization links between: *C2* and *C3*, *C2* and *C4*, *C3* and *C5*, *C4* and *C5*. The class *Clerk* (*C3*) inherits all attributes and methods of the *class Person* (*C2*).
- There are 5 instance connection between *C1* and *C3*; *C2* and *C3*; *C2* and *C4*; *C4* and *C5*; *C3* and *C5*. For example, the class *Organization* (*C1*) needs to create a new *Clerk* object (*C3*).
- There are 10 conceptual data links between each pair of classes of *subject 1* = (n (n-1)/2) links; n = 5: represents the number of classes inside *subject 1*.

The links inside *subject 2* are:

- There are 5 inheritance links between: *C9* and *C10*, *C9* and *C11*, *C9* and *C12*, *C6* and *C7*, *C6* and *C8*. For example the class *MotorCycleVehicle* (*C10*) inherits all the methods and attributes of the class *Vehicle* (*C9*).
- There are 6 message connection links between: *C6* and *C9; C6* and *C7; C6* and *C8; C9* and *C10; C9* and *C11; C9* and *C12.* The class *MotorcycleVehicle* (*C10*) sends a message to the class *Vehicle* (*C9*) needing to access an inherited method.
- There are 6 instance connection links between *C6* and *C9; C6* and *C7; C6* and *C8; C9* and *C10; C9* and *C11; C9* and *C12*. For example, for example the class *LegalEvent* (*C6*) needs a method inside the class *Vehicle* (*C9*) to calculate the registration fees.
- There are 21 conceptual data links between each pair of classes of *subject 2*.

The type of links between *subject 1* and *subject 2* are:

- There is one message connection link between *C6* and *C4. The class LegalEvent* (*C6*) needs to send access a message to the class *OwerPerson* (*C4*) to calculate the title fees.

In order to assign an interdependency weight (*Wij*) to the links joining each pair of classes, first a weighting scheme should be developed. Second, appropriate weights should be assigned to the appropriate link in the [0, 1] range.

The objective of assigning weight to links joining any two classes is to encourage or discourage their grouping in a single subject. Subjects should be designed with the objective of a high level of cohesion and low level of coupling. A high level of cohesion results when classes within a subject have physical links such as whole part structure or inheritance. There are weights of link which might result from grouping different classes inside a subject. These are shown, in order, in Table 1.

The order indicates the degree of association with respect to data, or methods or property shared between two classes, according to the object-oriented expert designers.

Links Type	**Weights (*Wij*)**
Whole Part Structure	0.9
Inheritance	0.8
Instance Connection	0.7
Message Connection	0.6
Conceptual Link	0.5

Table 1. Links Weights

Applying Formula 1, we obtain the following cohesions for *subject 1* and *subject 2*:

For *Wmax*, we have only 4 type of links (because we can not have a whole part structure and a specialization generalization structure at the same time between two classes):

$$C(Sub1) = \frac{(5\times0.7)+(4\times0.8)+(5\times0.6)+(0.5\times10)+0.9}{(0.9+0.7+0.6+0.5)\times10} = 0.55$$

$$C(Sub2) = \frac{(6\times0.7)+(5\times0.8)+(6\times0.6)+(0.5\times21)}{(0.9+0.7+0.6+0.5)\times21} = 0.39$$

The coupling of *S*, between *subject 1* and *subject 2* is obtained by the same formula (Formula 1):

$$C(S) = \frac{0.6}{(0.9+0.7+0.6+0.5)\times1} = 0.22$$

The quality of the system *S* is obtained by Formula 2:

$$Q(S) = \frac{(0.55+0.39).-0.22}{2} = 0.36$$

The values of the cohesion metric for the two subjects (0.55 and 0.39) are higher than the value calculated for the coupling between them. This finding is consistent with the basic principle of systems design that encourages high cohesion within a subject and low coupling across subjects. The value of the quality metric was calculated as 0.36, which appears to be quite low. This is only an indication of the quality of the actual decomposition. Other decompositions of the system may lead to a higher quality value. The analyst should test different others decompositions, and then selects the best decomposition (with the highest value of quality).

Based on the results obtained, the following proposition can be made: The quality metric based on cohesion and coupling and available early in the development cycle, is a good indicator of the quality of the architectural object-oriented design.

6. Conclusion

In this work two metrics for measuring the quality of object-oriented system architecture have been defined. The metrics are based on two internal system criteria namely cohesion and

coupling. These metrics take into consideration the number and also the nature of the links between the different classes of the system. An application of these metrics on a real information system has demonstrated their use at the early phase of the life cycle development process, the analysis and design phases. They are good indicators of decomposition quality and can help object-oriented designers to make decisions among competing design solutions. For object-oriented system, others aspects may be considered for example polymorphism as defined by Rumbaugh: the same operation may behave differently on different classes [10]. Work is in progress to experiment its impact on the quality of the decomposition. Future work is to generalize the quality metric to any number of levels of subjects, taking into account the subject hierarchy.

References

[1] Basili, V R, Briand, L C., Melo, W.L (1996). A validation of object-oriented design metrics as quality indicators, *IEEE Transactions on Software Engineering*, October, p751-761.

[2] Bieman, J.M., Ott, L.M. (1994). Measuring Functional Cohesion*, IEEE Transactions on Software Engineering,* 644-657.

[3] Coad, P., Yourdon, E. (1991). Object-Oriented Analysis*.,* Prentice Hall.

[4] Chidamber, S.R., Kemerer, C.F.(1994). A Metrics Suite for Object-oriented Design*, IEEE Transactions on Software Engineering,* 476-493.

[5] Courtois, P.J. (1985). On Time and Space Decomposition of Complex Structures, *Com. of the ACM*, 28 (6) June, p. 590-604.

[6] Hitz, M., Montazeri, B. (1995). Measuring Product Attribute of Object-Oriented Systems, *In*: Proc. Of the ESEC'95, Sitges, Spain, p. 124-136.

[7] Karimi, J., Konsynki, B. (1988). An Automated Software Design Assistant*, IEEE Transactions on Software Engineering*, 14, 2, February, p. 194-210.

[8] Lee, Y., Liang, B., Wu, S. (1995). Measuring the Coupling and Cohesion of an Object-Oriented Program Based on information Flow, *In*: Proc. Of the ICSQ'95, Slovenia, p. 81-90.

[9] Dieter, Rombach. (1990). Design Measurement: Some Lessons Learned*, IEEE Software*, p. 17-25.

[10] Rumbaugh, J. et al. (1991). Object-Oriented Modeling and Design, Prentice Hall Publ.

[11] Schach, S R. (1995). The cohesion and coupling of objects, *Journal of Object-Oriented Programming*, July-August.

[12] Tagoug, N. (2001). Information Systems Decomposition Based on Cohesion and Coupling, *In: Proc. Of the PDPTA'2001*, Las Vegas, Nevada, USA, June.