SOFTWARE DESIGN QUALITY METRICS FOR WEB BASED APPLICATIONS

Shazia, M. Shoaib, A. Shah, and F. Majeed

Department of CS & E, University of Engineering & Technology, Lahore, Pakistan.
Corresponding author email: shoaib_uet@hotmail.com

ABSTRACT: Object-oriented Design Methodology (OODM) was proposed for the design of web applications. For an efficient design certain measurements are required during the design phase which OODM does not address. These measurements have been introduced in the form of metrics. Metric's validation proves that these metrics are useful for the design of any applications. Further research can be carried to extend this research for practical development of application using these metrics. Paper Extracted from thesis (Shazia A. and A.Shah 2010).

Keywords: - Reusability metrics**,** Navigational Accessing Time Metric, Interface Coherence Metric.

INTRODUCTION

Lots of users are adding into the World Wide Web (WWW)'s progression. (Hooi-Im *et al.*, 1998) On the WWW various sorts of information can be navigated upon (Berners-Lee *et al*., 2009). As web-applications differ from traditional applications, their design is facilitated by the providence of hypertext or hypermedia paradigm. (Norman and Pfleeger, 1997)

The Structured Analysis and Design (SADT) and Object-Oriented Techniques (OOT) do not provide a good solution for web based applications. As observed in (Norman, 1994), these techniques are unable to analyze, design, implement and test the web based applications. Design and measurement are two important activities that can affect the performance of any system. Software measurements help us to understand, control and improve the software products (Larry Peters, 1981).

A good measurement can clearly distinguish the characteristics of one entity from another by analysis and drawing the conclusion. Software Metrics are used to measure the attributes of an entity. It is generally accepted that quality of a system or software product is strongly dependent on the quality of its design (Norman, 1997). The usability factor, the ease of use of systems, the efficiency and the cost effectiveness are the issues those depend upon how good the design is (Norman, 1997).

Web-based applications are evaluated by multiple users and thus are high on the quality requirements. Therefore, good design and techniques are used to measure it in an early stage can help to produce an efficient design. Numerous methodologies for the design of Web-based applications have been proposed. There were only some parameters available for estimation of design quality of a web application. The reason was that there was a dearth of metrics for the design phase of web-based application. To overcome this shortcoming, we have proposed some design quality metrics for web based applications considering OODM (Shah, 2003).

The following four design metrics are available for Structural Approach: Coupling, Cohesion, Understandability and Adaptability. Quantification of the parameter, coupling, is only available in the form of Fan-in and Fan-out (Roger, 2001).

Thirteen design metrics are available to measure the design quality of software in Object Oriented Approach. Definitions of all parameters are available but how they can be calculated is not given for all. Some of the parameters are: Coupling, Cohesion, Understandability, Adoptability, Polymorphism, Inheritance, Reusability, Robustness, Maintenance, Abstraction and Modularity. By measuring these parameters quality of design can be estimated (Cook Steve, 1994).

The different approaches to develop Web- based applications are: Hypermedia Design Model (HDM), (LUNA, 2009), Relationship Management Methodology (RMM) (Isakowtis *et al*., 1995), the Object-Oriented Hypermedia Design Model (OOHDM) (Schwab *et al*., 1995) and an Object-Oriented Design Methodology (OODM) (Shah, 2003) . In HDM, RMM and OOHDM hypermedia methodologies there were some limitations like, design and implementation were intertwined. Analysis was minimal in such methodologies, not following the basic principles of software engineering. In 2003 (Shah, 2003) proposed a methodology, OODM (Object Oriented Design Methodology), following the principles of software engineering. Though all these design methodologies have been available, parameters to estimate the design quality of a web application were rare.

MATERIALS AND METHODS

In OODM (Shah, 2003) design of a web application is represented using Abstract Data Type

(ADT) undirected graph as in Fig. 1. In the figure, we have shown a general design of a web application. The root node represents home page of the application and other nodes of the graph represent pages, these can be either static pages or dynamic pages. Note that a page which does not take any input from the user, is referred to as the static page, and a page that takes input from the user, is referred to as the dynamic page (Tim Berners-Lee, 2009).

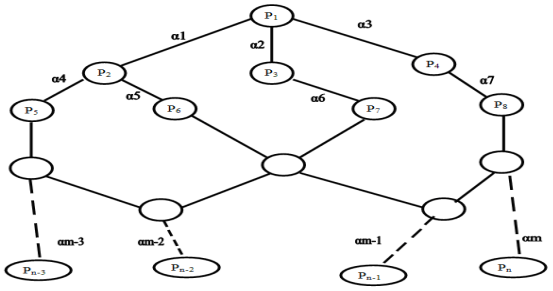

Fig. 1: Design of a web application-link tree

The design phase of OODM (Shah, 2003) works on the four aspects separately and develops corresponding four models, Component Model, Navigational Model, Operation Partitioning Model and User interface Model. In our ongoing research, design quality metrics for the four design phase models of OODM (Shah, 2003) have been proposed.

Component Model: Related multimedia attributes of a page-class are collectively known as Component Model. Its objectives are to identify page-classes of web application, components of a page-class and to identify multimedia attributes of a component of a page-class.

Reusability Metric: In this section we measure reusability of the design of a web application. Reusability and depth of the tree are directly proportional to each other and an inverse relationship holds between the depth of a tree and its complexity. A web application is a collection of pages. In the tree, nodes represent the pages of the application and edges/arcs represent links between the pages/nodes (Devpriya, 2009). However, this relation holds true only till the depth (D) ≤ 5 and can be maximum 8 but not beyond that as after that it becomes more complex (Chidamber *et al*., 1994).

Reusability of *jth* Access Sequence AS) is equal to the length of the Access Sequence

$$RASj = \sum_{i=1}^{n} R_{ij} \qquad \text{---------- (1)}$$

Where RASj =Reusability Access Sequence
i = Count from 1 to total no of links.
j = jth page of the link tree.
Assume there is N number of ASs in a web application then reusability metric for the web application (RWA) is

$$RWA = \sum_{j=1}^{N} \sum_{i=1}^{n} \beta_{ij} \qquad \text{---------- (2)}$$

where $\beta_{ij} = 1$

will attain value 1 always because
$\alpha = \beta$ if $\alpha = 1$

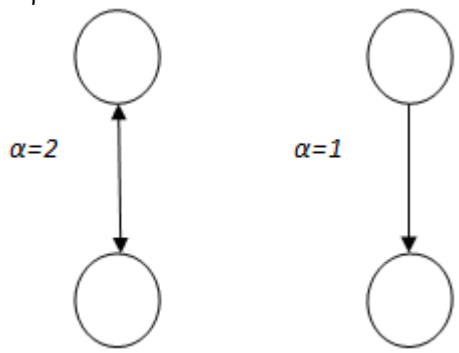

Fig. 2: Uni-directional and Bi-directional Links

In reusability metric when $\alpha_i = 1$ then we represent α as β to take a unique value of α**.** If it is required to measure the depth of a tree, then always take β=1. This is the only method to measure the depth of a tree; else it takes the form of a directed or cyclic graph which is not within our scope.

Navigational Model: The navigational model is used to design the information structure of a web based application. This information structure helps the user to navigate web pages. It provides various levels of navigation such as local navigation, global navigation, instance navigation and menu navigation [Shah, 2003].

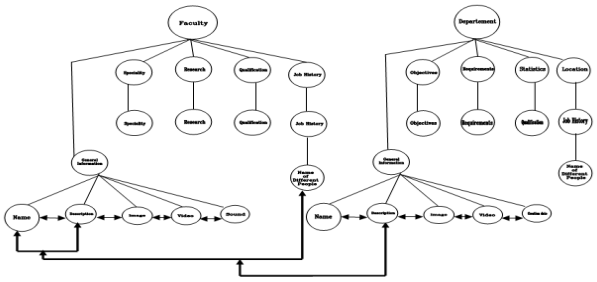

Figure 3.3: Navigational Model with Local, Instance and Global navigation

Fig. 3: Navigational Model with Local, Instance and global navigation

Navigational Accessing Time Metric: The main purpose of navigational accessing time metric is to help in calculating the total navigational time to reach the goal. Designer can develop a design by using this metrics which calculates the total time of navigation,

The time involved between two consecutive nodes is 1,
T = 1
As we mention earlier that α = 1 so
T = α.

The number of nodes involved are n, where $(1 \leq n \leq i)$

We can calculate the accessing time to reach the goal with the help of this formula:

T=(n-1) ---------- (3)

As the number of nodes involved in a path to reach the goal is increased, then the total accessing time to reach the specific node will also increase.

The relationship holds true only till the depth of the tree ≤ 5 and can be maximum 8 but not beyond that as after that it becomes more complex and for that reason consumes more time (Chidamber *et al*., 1994).

Operation Partitioning Model: This model is concerned with the following processing units: The operations represent services that are provided by the web pages of a WA, Building an object-interaction graph (OIG), design the execution pattern of each operation and message passing pattern among the objects. The main objective of this model is to identify objects and relationship among them (Shah, 2003).

Operation Performance Metric: We can calculate the complexity of a component by measuring the total complexity of its operations the complexity of an operation, and then in turn, depends on the complexity of the parameters it is comprised of. When compared, it is found out that the operations which have a large no. of parameters are more complex than those which cater to lesser no. of parameters.

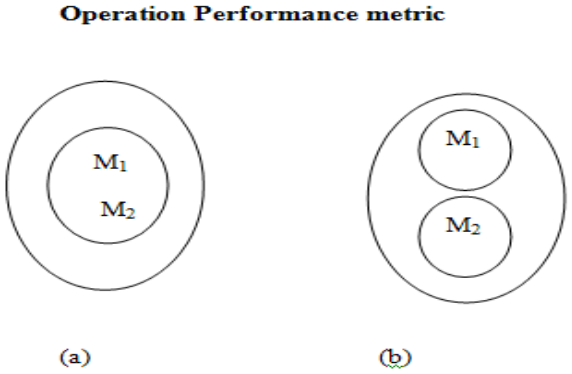

Fig. 4: Height of the operations

A same page having two operations can be displayed in 2 different ways. Firstly in Fig. 4(a) two operations are present in one component and in Fig. 4(b) two operations are present in two different components. In first case it is more cohesive. If the component is cohesive then in this case as an advantage, its efficiency increases but as a disadvantage its readability decreases (efficiency is inversely proportional to readability). Metrics can be defined as

$$Performance \propto \frac{1}{h_i} \ (hi \# 0) \qquad \text{---------- (4)}$$

The metric given above indicates that as the depth of the tree (value of hi) increases, performance (P) will decrease. Whereas, readability (R) increases.

$$Co \propto \frac{1}{h_i} \qquad \text{---------- (5)}$$

This metric (5) indicates that as the depth of the tree (value of hi) decreases, coherence (Co) increases. In contrast the lessening of the value of hi indicates that efficiency (E) increases, whereas readability (R) decreases.

$$E \propto \frac{1}{h_i} \qquad \text{---------- (6)}$$

$$P \propto \frac{1}{E} \qquad \text{---------- (7)}$$

In web application there are N operations, and then operational complexity is

$$Complexity = \sum_{j=1}^{N} \sum_{i=1}^{n} \frac{1}{h_{ij}} \qquad \text{---------- (8)}$$

*N represents total no of components in a web application.

*n represents total no of operations in component .

*hij tell about how many heights have these operations .

According to the metric (5) as the depth of the tree (value of hi) decreases, coupling (Co) increases. Moreover, decrease in the value of hi leads to an increase in coupling, an increase in readability(R) and a decrease in performance (P).

User Interface Model: This processing unit deals with the designing of user's perception and interaction with web application. Then the user interface design produces a consistent and predictable appearance of web application pages (Shah, 2003).

Interface Coherence Metric: Here we are going to measure the cohesion in terms of relationship between data on one screen. There are three modes of coherence:

- High coherence
- Medium coherence
- Low coherence

High Coherence: The cohesion is said to be high if data on interface is related to the attributes of the same component of a page-class. High cohesion for one page can be calculated as follows:

$$Co_h = \frac{P_s}{P_{WA}} \qquad \text{---------- (9)}$$

Where

Ps = Attributes of the same component

PWA = All page-classes of web application.

Medium coherence: The cohesion is said to be medium if data on interface is related to two or more interrelated components of the same page-class. Medium cohesion can be calculated as follows:

Where

Pr = related components of the same page-class ---- (10)

Low coherence: The cohesion is said to be low if data on interface is related to the different page-classes. Low cohesion can be calculated as following:

$$Co_r = \frac{P_{dr}}{P_{WA}} \quad (11)$$

Pdr = related components of different page-classes By summing all these cohesions, we can determine the overall cohesion of interface of a web application.
It can be calculated as:

$$Co_{WA} = \frac{Co_h + Co_m + Co_l}{P_{WA}} \quad \text{---------- (12)}$$

RESULTS AND DISCUSSION

Reusability Metric: Reusability metric helps to measure the depth of an independent path and overall complexity of web application.

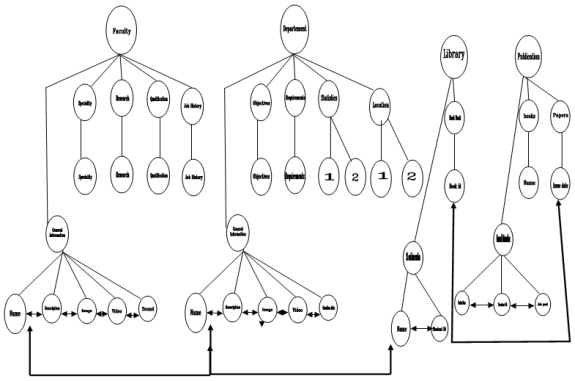

Fig. 5: Usability of an independent path

According to the fig. 5 the total no. of nodes involved in an independent path is 9.By putting this value in equation
Page/node = 9

$$RAS_j = \sum_{i=1}^{j} R_i \qquad \text{RASj} = 9$$

It means depth of a tree is 9, where depth of the tree should be ≤ 5 and can be maximum 8.

Reusability decreases the space, cost and time of a project. By measuring the reusability at the time of design the designer can save the valuable resources by keeping ($h_i \leq 5$) of its components and thus can save valuable resources like time , cost and space. (Chidamber and Kemerer, 1994).

Navigational Accessing Time Metric: Calculate time involved between two consecutive nodes of an access sequence.

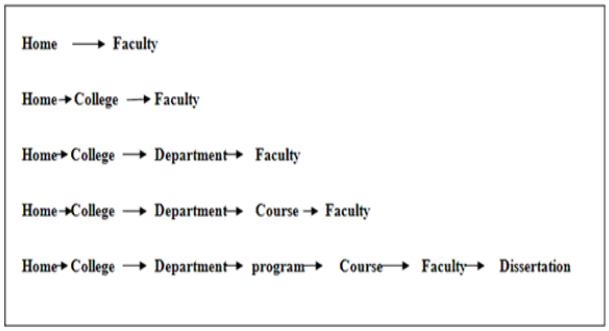

Fig. 6: Different access sequence of a link tree

By putting the values of access sequences in metric
For the first case: **home → Faculty**

n = 2
T = (2-1)
T = (1)

For the second case: **home → College → Faculty**

n = 3
T = (3-1)
T =(2)

For the third case: **home → College → Department Faculty →**

n = 4
T = (4-1)
T =(3)

For the forth case:
home → College → Department → program → course → Faculty →

n = 6
T = (6-1)
T = (5)

For the fifth case:
home → College → Department → program course → Faculty → dissertation →

n = 7
T = (7-1)
T = (6)

As the number of nodes involved in a path to reach the goal is increased, then the total accessing time to reach the specific node will also increase. From this metric we can calculate how much time an operation takes to complete a task, so that complexity remains under limits. The case was moving towards complexity as its value of nodes is 6.

Operation Performance Metric: By taking two cases of Faculty page-class:

The Faculty page-class has five components and their height is 1 ($h_i = 1$) and the height of their attributes is 2. Putting these values in the proposed metric for the first case we get:

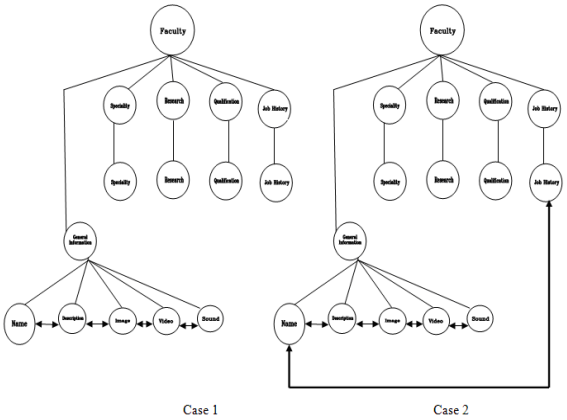

Fig. 7: Different cases for operation performance

$$P \propto \frac{1}{h_i} \qquad P = k\frac{1}{h_i} \qquad (0 \le P \le 1)$$

Here assume k=1 as k is the proportionality constant.
Putting k=1 in the metric, we get:

$$P = \frac{k}{h_i}$$

Now putting $h_i = 1$

$$P = \frac{1}{h_i}$$

$$P = \frac{1}{1}$$

$$P = 1 \qquad (0 \le P \le 1)$$

In this case, as value of P is 1 means efficiency has increased, readability has decreased.
This also indicates that our application is more cohesive.

Now putting $h_i = 2$,

$$P = \frac{1}{2}$$

$$P = 0.5$$

In this case, as performance has decreased, readability has increased. Application has become less cohesive. As height is increasing, efficiency goes on decreasing. In Operation performance metric, as value of P is 1 means efficiency has increased and readability has decreased. This also indicates that our application is more cohesive. But as value decreases, performance also decreases, whereas, readability increases. In this case, the application becomes less cohesive. Also, as height increases, efficiency decreases. Basically with the Operation Performance Metric we are able to calculate the overall performance and efficiency of a web-application.

Interface Coherence Metric:
High Cohesion:

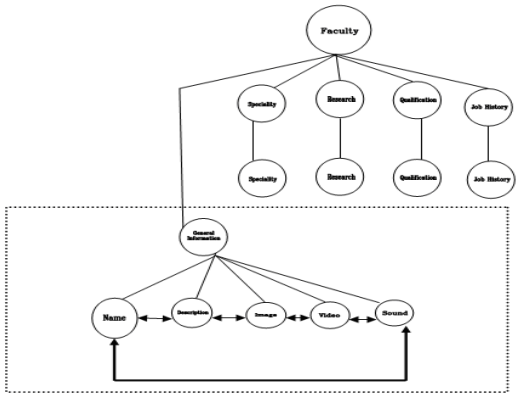

Fig. 8: Interface for navigation between attributes of the same component

The cohesion is said to be high if data on one interface shows the navigation between attributes of the same component.
High cohesion can be calculated as:

$$Co_h = \frac{P_s}{P_{WA}}$$

$$Co_h = \frac{1}{5}$$

$$Co_h = 0.2$$

Medium Cohesion:

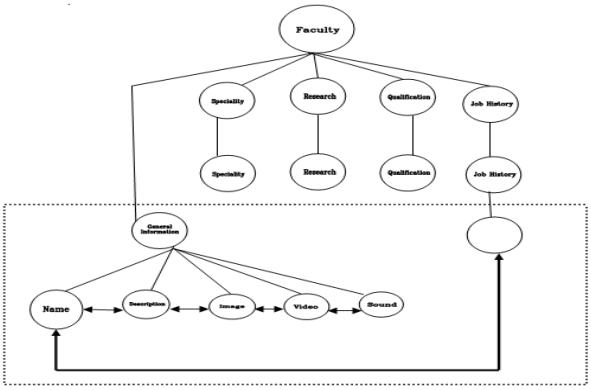

Fig. 9: Interface for navigation between attributes of different components

$$Co_m = \frac{P_r}{P_{WA}}$$

$$Co_m = \frac{2}{5}$$

$$Co_m = 0.4$$

Low Cohesion:

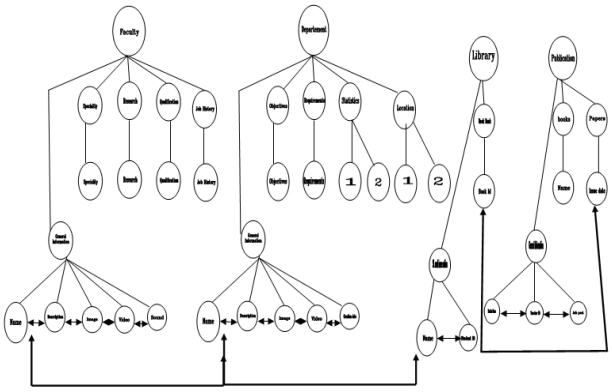

Fig. 10: Interface for navigation between components of different page-classes

Low cohesion of the given interface is given as under:

$$Co_l = \frac{P_d}{P_M}$$

$$Co_l = \frac{4}{5}$$

$$Co_l = 0.8$$

High, medium and low cohesion is

$$Co_h = 0.2$$

$$Co_m = 0.4$$

$$Co_l = 0.8$$

If coherence lies between $(0 \leq C \leq 1)$ it shows, that if value is near to 1, it means coherence is low and if far from 1 means high coherence.

If high coherence is achieved, the user can be restricted to one screen for accessing its components. The interface coherence metrics enables the designer to measure the coherence and then improve it, which was not possible in case of OODM or any other methodology. The relevancy and time saving has been increased and navigational effort has been reduced as more and more information from the same page can be obtained.

CONCLUSION: We have proposed metrics for different models of OODM's design phase to evaluate design quality of a web application, before going to implementation. Because of these metrics important decisions can be taken in the design phase, saving development cost and effort. Using these proposed metrics we can also tell which design is overall better than others. We can compare different aspects of both designs using the model evaluating metrics and can compare them aspect-wise.

The methodology, OODM, is believed to be one of first web application development methodologies. It provides a complete, detailed and step-wise development process. Therefore, it needs to be automated. In future, work in this direction may be done. There can be another future direction of this work: to upgrade this methodology OODM for the development of semantic web applications. Also OODM can be upgraded keeping security issues on the forefront.

Acknowledgement: This research paper is extracted from the Ph.D thesis of Ist author which has been published at UET Lahore in 2010.

REFERENCES

Abad S., OODM: An Object-Oriented Design Methodology for Development of Web Applications, King Saud University, Kingdom of Saudi Arabia, 189-229 (2003).

Berners-Lee, T., Cailliau and Robert. "WorldWideWeb: Proposal for a hypertexts Project". http://w3.org/Proposal.html. [Retrieved July 27, 2009] (1990).

Berners-Lee, T. Pre-W3C Web and Internet Background. World Wide Web Consortium. http:// w3.org / 2004/Talks/w3c10-How It All Started/?n=15. [Retrieved April 21, 2009].

Chidamber, S. and Kemerer. A metric suit for Object Oriented Design, IEEE transactions on Software Engineering, 20: (1994).

Devpriya, S., S. Ritu and M. Kumar. A framework for validation of object-oriented design metrices, (IJCSIS) International Journal of Computer Science and Information Security, 6 (3): (2009).

Everald, E. M. Metrics in the software engineering curriculum, 6(1-4): 181 – 200 (1999).

Everald, E. M. Metrics in the software engineering curriculum, Annals of Software Engineering, 6 (1-4): 181-200 (2004).

Horst Z., History of Software Measurement, (1995). http://irb.cs.tu-berlin.de/~zuse/metrics/3-hist.html

Isakowitz, D., E. Stohr and P. Balasubramanian. RMM: A Methodology for structured hypermedia design, Communication of the ACM, 38(8): 34–44 (1995).

Judith B. A new reusability metric for object-oriented software, Software Quality Journal 7: 35-50 (1998).

Karl, E. W., A Software Metrics, Primer",Copyright © 1999. http://www.processimpact.com/articles/metrics_primer.pdf

Larry P. Software Design: Methods and Techniques. New York: YOURDON Press (1981).

LUNA, A. M. Geração de Interfaces RIA Dirigida por Ontologias. Rio de Janeiro, . MSc Dissertation – Dept. of Informatics, Pontifícia Universidade Católica do Rio de Janeiro. (in Portuguese) (2009).

Norman, E. F. and S. L Pfleeger. Software Metrics: A Rigorous and Practical Approach, 2nd edition International Thomson Computer Press (1997).

Norman, F. Software Measurement: A Necessary Scientific Basis, IEEE Transactions on Software Engineering, 20 (3): (I994).

Roger, S. P. Software Engineering A Practitioner's Approach, Fifth Edition (2001).

Seyyed M. J. Object oriented metrics, A Survey Approach (2006).

Schwabe, D. and G. Rossi. Building Hypermedia Applications as Navigational Views of Information Models, Proceedings of the 28th Annual Hawaii International Conference on System Sciences, 231–240 (1995).

Shazia, A. and A. Shah. Software Design Quality Metrics For Web Based Applications, Ph.D Thesis, University of Engineering and technology Lahore (2010).

Tom, D. Controlling software projects: management, measurement & estimation, Yourdon Press (1982). http://ce.sharif.edu/~m_jamali/resources/ObjectOrientedMetrics.pdf

Tim B. Design Issues for the World Wide Web. public domain. World Wide Web Consortium. (2009).