

# Semantic Metrics, Conceptual Metrics, and Ontology Metrics: An Analysis of Software Quality Using IR-based Systems, Potential Applications and Collaborations

Letha H. Etkorn  
Computer Science Department  
University of Alabama in Huntsville  
Huntsville, AL 35899 USA

## *Abstract*

*Similarities and differences between “semantic metrics” (metrics defined on a knowledge-based IR system) and “conceptual metrics” (metrics defined on a Latent Semantic Indexing-based IR system) are discussed. Potential collaboration areas between research groups are identified. Potential application and collaboration areas of a new research area called “ontology metrics,” metrics calculated on the ontologies that form part of an ontology-based software system, are also discussed. Currently ontology metrics are calculated using techniques similar to semantic metrics, but other semantically-based expansions, including some similar to conceptual metrics are possible.*

Application areas of Information Retrieval (IR) based approaches have in the past included concept location [13], clone detection [8], traceability link recovery [11], and software reuse [5,6], among others. One relatively new application area is the following: in the past few years, the use of IR based techniques has been extended to examine software quality, through metrics based on the output of IR-based systems. This new field, introduced by Etkorn et al. [7,8] and furthered by Stein et al. [16,17,18] and Cox et al.[3], and called “semantic metrics,” has recently been extended by Marcus and Poshyvanyk [12,20], and by them called “conceptual metrics.” “Conceptual metrics” are different from “semantic metrics” in that semantic metrics are defined in the context of comment and identifier analysis within a program understanding system based on knowledge-based natural language understanding, whereas the “conceptual metrics” are defined in the context of comment and identifier analysis within a program understanding system based on Latent Semantic Indexing. Latent Semantic Indexing (LSI) is a corpus-based statistical method for representing the meaning of natural language words and passages (sentences, paragraphs, etc.). With LSI, meanings are derived from usage rather than from a predefined dictionary[11].

Both types of IR-based metrics, semantic metrics and conceptual metrics, have the advantage that they can be used to measure some aspects of software design, such as cohesion, which are difficult to measure [7] using traditional syntactic metrics [2] based solely on counting items in the source code. Both types of IR-based metrics could also be used to analyze design qualities in, for example, software design documents [18] as well as in source code, which would expand metrics capability far beyond that provided by most traditional syntactic metrics.

The advantages and disadvantages between semantic metrics and conceptual metrics would seem to be primarily based on the differences between the underlying natural language-based program understanding systems: knowledge-based or LSI based. For example, one primary disadvantage of a knowledge-based natural language understanding system is the necessity to create knowledge bases for each new domain examined (or possibly one large knowledge base to cover several domains). An additional related disadvantage of the knowledge-based system approach would be a potential lack of scalability due to the difficulties involved in word sense disambiguation in large knowledge-bases. Therefore, the LSI approach is possibly more easily scalable than the knowledge based system approach. (We note here, however, that there are a number of variations of Word Sense Disambiguation (WSD) techniques used for IR that do not depend on complete knowledge bases. Ozcan and Aslandogan [15] performed a study that employed various WSD and LSI techniques separately with success.)

However, LSI can be criticized for not making use of word order, morphology, or the syntactic relations between words [13], so although overall it has shown to work well [13], potentially it could work better on some input data than others. For example, Deetwester et al. [4] found a substantial improvement over traditional term (keyword style) indexing using latent semantic indexing in one corpus (13%

improvement on the MED corpus) of two that they tested, while they found no improvement on the other corpus (the CISI corpus). They suggested the poor performance in the CISI corpus was potentially due to the homogeneous nature of the dataset.

However, in areas where both semantic metrics and conceptual metrics can be collected easily (data areas where both knowledge-based program understanding and LSI-based program understanding will easily operate; for example, where knowledge bases are available and the corpus involved is non-homogeneous), a comparison between semantic metrics and conceptual metrics intended to measure the same software qualities would be interesting. Each type of metric has, in the past, been used to measure the same quality. For example, the software quality “cohesion” has been measured using semantic metrics [3,17] and conceptual metrics [12]. In these papers, the semantic and conceptual metrics have been compared to existing syntactic metrics, but not (to the author of this paper’s knowledge) to each other.

One interesting expansion of both kinds of metrics would be to investigate applying variations of already proposed semantic or conceptual metrics to comment and identifier understanding that employs other variations of IR, and preferably industrial strength IR implementations, or possibly combinations of existing IR approaches.

Furthering this idea would be an examination of how previously defined semantic metrics and cognitive metrics would be defined differently when based on different kinds of IR systems. For example, semantic metrics are defined in the context of a knowledge-base loosely based on the conceptual graph representation format [7]. How would the definition of the semantic metrics change when defined on different knowledge representation formats, for example, a knowledge-base defined in the Resource Description Framework (RDF) knowledge representation format? Similar questions could potentially be raised for conceptual metrics if applied to different kinds of IR systems.

This question leads into another research area, that of metrics to measure ontology-based systems, otherwise known as ontology metrics, or ontology-based metrics [14,19]. In ontology-based systems, ontologies provide a common, shared interface definition between applications. One example application area is internet-based systems providing Business to Business or Business to Consumer services. These types of service-oriented architectures, as well as other web based systems, often employ ontology data in a description of web services. Such ontology data is often represented in Resource Description Format (RDF), and stored in XML [14].

The same kinds of software processes are used to develop ontology-based systems as are used for more traditional software systems that do not employ ontologies. However, an integral part of an ontology-based system is the development

of an ontology. Therefore, an important part of measuring the quality of an ontology-based system is measuring the quality of the ontologies themselves.

Recently, Orme, Yao, and Eitzkorn [14] and Yao, Orme, and Eitzkorn [19] examined ontology cohesion and coupling, using techniques similar to the calculation of the earlier semantic metrics within the knowledge base of the program understanding system, but in this case the calculations were performed directly on the ontologies themselves, which were stored in RDF format. The intent of this work was to examine ontology cohesion and coupling as they might affect a software system that employed those ontologies. Other work, by Burton-Jones, et al. [1], examined ontology quality of individual ontologies based on a semiotic framework derived from linguistics theory that includes general elements of quality based on linguistic-style concepts related to the analysis of signs and symbols. However, the work of Burton-Jones et al. was not directly targeted toward the use of these ontologies in a software system.

In addition to web services employed for internet based systems, bioinformatics and genomics researchers also make heavy use of ontologies. Orme, Yao, and Eitzkorn [14] performed a bioinformatics case study in which they examined the integration of existing ontologies to reduce failures that could occur in the runtime use of these ontologies.

Substantial work remains in ontology metrics, particularly validating the metrics in various application areas, but also determining how these metrics can best be employed in software maintenance on the new kinds of software applications that make heavy use of ontologies.

As mentioned earlier, the ontology metrics of Orme, Yao, and Eitzkorn [14] and Yao, Orme, and Eitzkorn [19] were calculated similarly to how semantic metrics are calculated within the knowledge base of a program understanding system. However, other more advanced semantic techniques are possible (in the case of these ontology metrics as currently defined, an IR-based program understanding system is not employed; rather, the metrics are calculated based on the relationships as specified in the ontologies themselves). Recently, Kotis, Vouros, and Stergiou [9] examined the automated mapping/merging of ontologies using the meaning of concepts by mapping them to WordNet senses using LSI. This kind of approach could lead to conceptual metrics to measure ontology qualities. Similarly, combinations of the original ontology metrics of Orme, Yao, and Eitzkorn [14] and new conceptual metrics for ontologies could potentially be useful.

## REFERENCES

- [1] A. Burton-Jones, V.C. Storey, V. Sugumaran, and P. Ahluwalia, “A semiotic metrics suite for assessing the quality of ontologies,” *Data and Knowledge Engineering*, Vol. 55, No. 1, pp. 84-102, October, 2005

- [2] S. Chidamber, C. Kemerer, "A metrics suite for object oriented design," *IEEE Trans. Softw. Eng.*, Vol. 20 pp4 76-493, June 1994.
- [3] Glenn Cox, Letha Etkorn, and William E. Hughes, "Cohesion metric for object-oriented systems Based on semantic closeness from disambiguity," *Applied Artificial Intelligence*, Vol. 20, No. 5, pp. 419-436, May-June, 2006..
- [4] S. Deerwester, S., S.T. Dumais, G. W. Furnas, T.K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, 41(6), pp.391-407, 1990.
- [5] Letha Etkorn, Lisa Bowen, and Carl Davis, "An approach to program understanding by natural language understanding," *Natural Language Engineering*, Vol. 5, Issue 5, , pp. 219-236, March, 1999.
- [6] L.H. Etkorn, and C.G. Davis, "Automatically identifying reusable OO legacy code," *IEEE Computer*, Vol. 30, No. 10, , pp. 66-71, Oct. 1997.
- [7] L. Etkorn, and H. Delugach, H., "Towards a semantic metrics suite for object-oriented design," *Proceedings of the Thirty-Fourth International Conference on Technology of Object-Oriented Languages and Systems*, July 30, 2000, pp. 71-80.
- [8] Etkorn, Letha, Gholston, Sampson, and Hughes, William E., Jr., "A Semantic Entropy Metric," *Journal of Software Maintenance and Evolution* , Vol. 14, Issue 4, July/August, 2002, pp. 293-310.
- [9] K. Kotis, G.A. Vouros, G.A., and K. Stergiou, "Towards automating merging of domain ontologies: the HCONE-merge approach," *Web Semantics*, Vol. 4, Issue 1, pp. 60-79, Jan. 2006.
- [10] A. Marcus, A., and J.I. Maletic, J.I., "Identification of high-level concept clones in source code," *Proceedings of Automated Software Engineering*, (ASE '01), San Diego, CA, Nov. 26-29, 2001, pp. 107-114.
- [11] A. Marcus, and J.I. Maletic, J.I., "Recovering documentation-to-source-code traceability links using latent semantic indexing," *Proceedings of the Twenty-Fifth IEEE/ACM International Conference on Software Engineering*, Portland, OR, May 3-10, 2003, pp. 125-137.
- [12] A. Marcus, and D. Poshyvanyk,, "The conceptual cohesion of classes," *Proceedings of the Twenty-First International Conference on Software Maintenance*, 25-30 September 2005, Budapest, Hungary, pp. 133-142.
- [13] A. Marcus, A. Sergeev,, V. Rajlich,, and J.I. Maletic, J.I., "An information retrieval approach to concept location in source code," *Proceedings of the Eleventh Working Conference on Reverse Engineering*, Nov. 8-12, 2004, pp.8-12.
- [14] A.M. Orme, H. Yao, , and L. Etkorn, "Coupling Metrics for Ontology-Based Systems," *IEEE Software*, Vol. 23, No. 2, pp. 102-108, March/April 2006.
- [15] R. Ozcan,, and Y.A. Aslandogan, Y.A., "Concept based information access using ontologies and latent semantic analysis," *Technical Report CSE-2004-8*, Department of Computer Science and Engineering, University of Texas at Arlington.
- [16] Cara Stein, Letha Etkorn, Glenn Cox, Philip Farrington, Sampson Gholston, Dawn Utley, and Julie Fortune, "A new suite of semantic metrics for object-oriented software." *Proceedings of the 1st International Workshop on Software Audits and Metrics (SAM 2004)*, Porto, Portugal, April 13-14, 2004, pp. 49-58.
- [17] Cara Stein, Letha Etkorn, Sampson Gholston, Philip Farrington, and Julie Fortune, "A knowledge-based cohesion metric for object-oriented software," *INFOCOMP*, accepted, unpublished.
- [18] Cara Stein, Letha Etkorn, Dawn Utley, Philip Farrington, Glenn Cox, Julie Fortune, and Sampson Gholston, "Computing software metrics from design documents," *Proceedings of the 42nd Annual ACM Southeast Conference*, Huntsville, AL, April 2-3, 2004, pp. 146-151.
- [19] H. Yao, A.M. Orme, L. Etkorn, "Cohesion metrics for ontology design and application," *Journal of Computer Science*, 1(1), 2005, pp. 107-113.
- [20] D. Poshyvanyk,, and A. Marcus, "The conceptual coupling metrics for object-oriented systems," *Proceedings of the Twenty-Second International Conference on Software Maintenance*, Sept. 24-27, 2006.