Software Metrics from Meta Definition Language (MDL)

Nakul Sharma
Student (M.E. (S.E.)): Computer Science and Engineering Department, Thapar University
Thapar University, (TU)
Patiala, India.
nakul777@gmail.com

Shantanu Pathak
Lecturer, IT Dept, Pune Inst. of Computer Technology
Pune Inst. of Computer Technology, (PICT)
Pune, India.
shantanuspathak@gmail.com

***Abstract*— Software metrics is a technique for assessing the software in qualitative and quantitative terms. This paper deals with using Meta Definition Language (MDL) to get useful software metrics. MDL is generated by Rational Rose software of IBM. In this paper we propose to analyse and present a new technique of getting metrics from this mdl format. The future research directions are also presented herein.**

***Keywords*— Software Engineering, Software metrics, UML, Meta Definition Language (MDL)**

I. INTRODUCTION

Software metric forms an important part of Software Engineering. It aids in accessing software form prospective of various stakeholders in software such as domain engineers, application engineers, managers etc. A measurement provides clue about the degree and extend. Metric is more precise in judging the measurement, along with the attributes of the work products being measured. Meta Definition Language (MDL) has a petal format. In this paper we strive to present various ways and means of figuring out metrics of all types of work products in software reuse.

II. BACKGROUND

MDL is employed by Rational Rose in its .ptl and .mdl file for storing UML diagrams. The file format is not documented. There is lack of version support provided by the rational rose. MDL's basic measurement entails use of techniques for quantifying a asset. These measurements can be applied to a set of processes, products or services. Upon assessment of such measurements, we can apply them to various paradigms of engineering. By undertaking various assessment programs it becomes possible to improve upon such assets. There are various motivations behind making use metrics some of them are as follows:-

- Estimate the cost & schedule of future projects.
- Evaluate the productivity impacts of new tools and techniques.
- Establish productivity trends over time.
- Improve software quality.

Metrics in Software Engineering

Depending on the scenarios various software engineering metrics have evolved. But generically they can be divided into following two categories:

[1] Structural metrics: These provide for the measurement of the asset. For example, cyclomatic complexity.
[2] Functional metrics: These focus on the functions provided by the asset. For example, Function Point (F.P.)

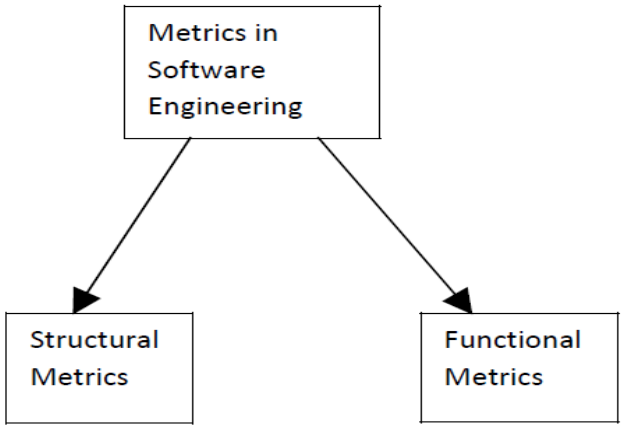

Figure 1 Metrics of Software Engineering

Rational Rose is a software product of IBM incorporation. Rational Rose is used in making UML diagrams. It makes use of simple ANSI format for storing MDL file[2].

III. LITERATURE REVIEW

The various classification of structural measures is discussed NE Fenton. They are classified as control flow, data flow, data structure [1]. These structures help in the metrics of MDL.

The various types of Object Oriented metrics that can be studied from MDL are as follows are discussed by Fenton[1].

These metrics can change depending upon the scenarios in which they are being studied.

Weighted Methods Per Class (WMC)
This metric provides us the notion of complexity. For a given class and methods, it is possible to compute the WMC by following equation:-

$$WMC= \sum c_{i.}$$

Where i=0...n

Depth of Inheritance Tree (DIT)
It is the length of maximum path from the node to the root of the tree. DIT provides the amount of ancestor classes that affect a given class.

Number of Children (NOC)
This metric gives the number of immediate successors of the class.

The data structures employed in representation of MDL file are objects [2]. MDL file has an object format. The UML diagrams of use-case and class diagram can be used in assessing various metrics of design model as mentioned above. The petal files format pertinent to discussion is as under[2]:

{petalfile}->{petal}
{design}->(object design"Logical view"
is_unit TRUE
is_loaded TRUE
Quid<ident>
defaults<Defaults>
root_usecase_package <UseCaseCatergory>
root_cateory<LogicalCategory>
root_subsystem<Subsystem>
Process_structure<Processes>
properties<Properties>)

Use case diagrams start with root_usecase_package.

IV. PROPOSED METHODOLOGY

The proposed methodology entails following different modules.

MDL Parser: MDL file will be scanned for getting the necessary data structure required for generating the metrics of an asset. In this section we propose use of a reference model for this module. We take the UML diagrams of use-case and class in getting metrics. The diagrammatic representation of this module is as follows:

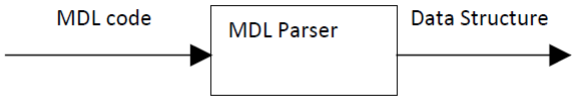

Figure 2. MDL Parser

A use case can divided into actors, its relationships. The actor and its relationship with the use cases can be defined using data structure.

A formal description of the data structure is as under:

- Class (Actor)= $\{a_1,a_2,...a_n\}$
- Use Case=$\{uc_1,uc_2, ...uc_n\}$
- Rel=$\{A_{i,} U_j\}$

We undertake scanning of file such that metrics can be generated using following algorithm.

We design an algorithm for this module named Dsgen. In this algorithm we scan through the MDL file for generating the necessary data structure.

Dsgen:

Begin:

[1]. Iterate till keyword!= "logical_presentation":

a. Initialize the data structure to NULL value.

b. Keywords Root_usecase_package, actors, operations to be scanned for getting the necessary information.

[2]. Saving of data structure in a file.

End.

Metrics Generator: This module scans through the data structure created by the parser for generating the metrics corresponding to a UML's use-case and class diagram. The diagrammatic representation of this module is as follows:-

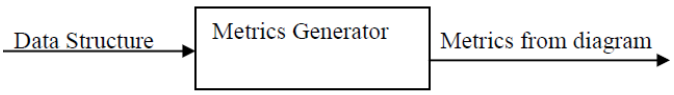

Figure 3. Metrics Generator.

This module uses the definition of various metrics as they are indicated by their definition. The metrics are derived from the definition of data structure.

V. CONCLUSION

Software metrics is new emerging field within software engineering paradigm. In this paper we have described a way of getting software metrics from Meta Definition Language. As MDL file makes use of object in storing information, we make use of available metrics in form of given paradigm. The future work entails making use of modules presented herein for studying its perspective from software reuse.

REFERENCES

[1]. Norman E. Fenton , Shari Lawrence Pfleeger , "Software Metrics: A Rigorous And Practical Approach", 2005,vol. 02.

[2]. M. Dahm,"Grammar and API for Rational Rose Petal files", July 19, 2001. version 1.1