

Assessing Modeling Languages, metrics and tools

No Author Given

Department of Informatics, University of Minho
Campus de Gualtar, 4710-057 Braga, Portugal

Abstract. Any traditional engineering field has metrics to rigorously assess the quality of their products. From a long time ago, engineers know that the production process is not all; the output must comply with the rules and good-practices, must satisfy the requirements and must be competitive.

Professionals in the new field of software engineering started a few years ago to define metrics to appraise their product: individual programs and software systems.

This concern motivates the need to assess not only the outcome but also the process and tools employed in its development. In this context, assessing the quality of programming languages is a legitimate objective; in a similar way, it makes sense to be concerned with models and modeling approaches, as more and more people starts the software development process by a modeling phase.

In the paper we discuss the quality of modeling languages, introducing and motivating the topic, presenting metrics, and comparing tools.

Keywords: : Modeling Languages, Software/Language Quality, Software/Language Metrics, UML

1 Introduction

* Falar da importância das linguagens de modelação como especificação formal de um projecto e também para ter uma visão global do projecto.

*Falar da qualidade do SW e das Linguagens em geral e introduzir o tema de "aferição de qualidade em Linguagens de Modelação"; relacionar com o tema das "métricas".

* Importância do uso de métricas num projecto: o que medem, o que ajudam a melhorar, etc.

* Especificar o tipo de métricas sobre o qual nos vamos focar (UML).

** estrutura do artigo

2 Metrics Assessment

—Entra o assunto da Secção 2 do artigo do Azevedo, JJ e Tiago.

* Explicar aqui que as métricas não podem ser observadas, por si só, fora do contexto. Este problema prende-se essencialmente com o facto deste tipo de medições ser empírica, ou baseada em métodos empíricos.

* Explicar em maior detalhe o que se entende como validação de métricas (Kaner e Walter Bond).

3 Applying Metrics To UML Models

–Junção da Sec.1 do artigo dos Pedros e Ulisses com Sec.3 do artigo do Ismael e Daniela.

* Explicar que as métricas sobre UML derivam directamente do OO.

* Falar em CK metrics e falar apenas nos tipos de CK metrics mais importantes.

* Relacionar as tabelas do paper do Ismael e Daniela com os tipos de CK metrics.

4 Case Study

–Apresentação de um Exemplo a ser analisado por cada uma das ferramentas: encontrar um projecto (idealmente open source) que tivesse disponível os diagramas UML (não apenas diagramas de classes, para os programas que conseguem processar mais que estes)

5 Tools

–Surgiram para já estes 3 nomes, o ideal seria caso aparecessem mais escolhermos os 3 ou 4 principais.

Referir para cada ferramenta:

* o que são capazes de fazer (algumas métricas que calculam),

* se é proprietário, open source, licença académica,

* que input recebem (XML, formato próprio, etc...)

* o que devolvem (se fazem análise apenas das métricas em separado ou se tentam ir mais longe e dão resultados sobre qualidade, tamanho do projecto, etc...)

One of the tools that we are going to adress is SDMetrics. SDMetrics is a design measurement tool for the UML¹.

SDMetric is not free so we required an academic license from the staff, wich was quickly provided. It's core is open source and is available under the GNU Affero General Public License.

The core functionalities of SDMetrics include:

¹ SDMetrics can be found at <http://www.sdmetrics.com/>

- the configurable XMI parser for XMI1.0/1.1/1.2/2.0/2.1 input files,
- the metrics engine to calculate the user-defined design metrics,
- the rule engine to check the user-defined design rules.

5.1 SDMetrics

SDMetrics is a very complete design measurement tool, analysing a wide range of UML diagrams, including Class, Usecase, Activity and Statemachine diagrams. For each type of diagram, this tool generates several metrics.

For example, **NumAttr** metric, one of the metrics that has already been addressed in this paper, is calculated from Class diagrams. Other one is **ExtPts**, which is calculated from Usecase diagrams, and gives us the number of extension points of a given use case.

SDMetrics is written in java, and provides us a graphical user interface. The type of source files it receives to analyse are *XMI*² files, most modeling tools support project exportation in XML.

This tool allows us to access the results from different views. We will approach the ones that seem the most important:

- **Metric Data Tables** provides a table that matches each UML model element analysed (table line) to its value for each metric (table column);
- **Histograms** provides a graphical distribution for each design metric;
- **Design Comparison** provides us a mean to compare the structural properties of two *UML* designs. It is very useful to compare the same design in different iterations of the development, or to compare an alternative design to the current one.
- **Rule Checker** design rules and heuristics detect potential problems in the UML design such as:
 - incomplete design such as unnamed classes, states without transitions;
 - violation of naming conventions for classes, attributes, operations, packages;
 - etc.
- **Catalog** this view provides us with the definitions of the metrics, design rules, and relation matrices for the current data set, and provides literature references and a glossary for them.

Not a view, but one of the most advanced features in this software is the possibility of defining Custom Design Metrics and Rules. The new metrics are defined in a XML file, with a very particular format, the *SDMetricsML* (SDMetrics Markup Language).

The SDMetrics tool doesn't provide a direct notion of good/bad quality of the design model. Despite that, on the SDMetrics website we can find several indications of how to interpret each output.

² *XMI* (**X**ML **M**etadata **I**nterchange) is an *OMG* (Object Management Group) standard to generate XML-based representations of UML and other OO data models.

Results

5.2 Sparkx Systems Enterprise Architect

–Permite modelação + aplicação de métricas sobre Use Cases

Results

5.3 IBM Rational

–Permite modelação + aplicação de métricas

Results

6 Conclusion

Comparação entre as diversas ferramentas.

References