A Survey on Base Measures of Software Metrics in Business Process Model

Ramesh Kumar, Dr.Rajesh Verma
Research Scholar, Department of Computer Science, Singhania University, Rajasthan.
Asst. Professor, Department of Computer Science , Govt. College, Indri, Karnal
ramesh.banwala@yahoo.co.in

Abstract: There is a set of metrics for the evaluation of conceptual models of business processes. FMESP includes a set of metrics, which provide the quantitative information to find out the maintainability of the software process models. There is a set of metrics for the evaluation of the complexity of business process models defined by BPMN (Business Process Modeling Notation). In this article we define a set of metrics for conceptual model of software process and also define base measure of elements in business process model.
Keywords: Metrics, Business process, Conceptual model, BPMN.

1. Introduction: - Software process improvement (SPI) has emerged as a critical area for an organization involved in software development. Various organizations have reported benefits from software process improvement programs [1]. For SPI, the most comprehensive framework is the capability maturity model (CMM) for software [2]. CMM does not specify which metrics should be used or how they should be used. So, there is a possibility that different organizations may employ different metrics and use them differently. However, as the basic objectives of high quality, high productivity, and small cycle time are same in all organizations [3]. One of the main problems in software process improvement initiatives do not know clearly what to do, as was revealed by an SEI survey. At high maturity organizations, metrics is expected to play a key role. If nature of metrics programs and similarities between them are determined, then it can help other organizations in building or improving their own metrics program and in their quest for high maturity. Software metrics and measurements have been an area of active interest for a long time. Though metrics can be used in many ways, in a software organization, the three main uses of metrics data are: for project planning, for monitoring and controlling a project, and for overall process management and improvement. To support these uses, some metrics infrastructure is needed. This study also considered the metrics infrastructure in high maturity organizations. The study found that most of the organizations studied collect similar metrics and have similar metrics infrastructure in place, though the details of the procedures for use of metrics differ.

1.2. Software Metrics

Software metrics are an integral part of the software development in software engineering. The customers are specifying software metrics reporting in their contractual requirements. Software metrics is defined as, "The continuous application of measurement-based techniques to the software development process and its products to supply meaningful and timely management information". [4] Software metrics can provide the information needed by engineers for technical decisions as well as information required by management.

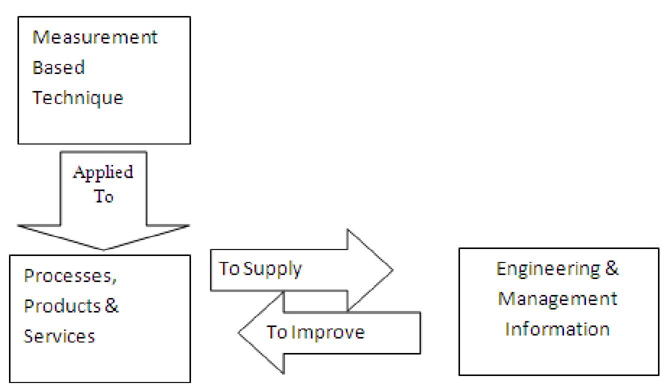


Figure1 software Metrics

1.2.1. Product Metrics: - Product metrics describe the attributes of the software product at any phase of its development. Product metrics may measure the size of the program, complexity of the software design, performance, portability, maintainability. Product metrics are used to presume and invent the quality of the product. Lines of code (LOC) is a software metric used to measure the size of a software program. SLOC is typically used to predict the amount of effort that will be required to develop a program [7].

1.2.2. Process Metrics: - Process metrics are used to measure the status and progress of the system design and to predict future effects. This metrics is used for estimation, monitoring and improvement of quality and reliability of software.

1.3. Objectives of Software Metrics:

Followings are the objectives of software metrics:

- Software metrics help software engineers to better understand the attributes of software and assess the quality of the software.
- Software metrics help software engineers to study of software design and development of the software.
- Software metrics focus on specific attributes of software engineering work products resulting from analysis, design, coding, and testing.
- Software metrics provide a systematic way to assess quality & product.

1.4. Characteristics of Software Metrics

Following are the characteristics of Software Metrics

- A metric should have desirable mathematical properties like a meaningful range (e.g., zero to ten). It should not be set on a rational scale, if it is composed of components measured on an ordinal scale.
- If a metric represents a software characteristic that increases when positive traits occur or decreases when undesirable traits are encountered, the value of the metric should increase or decrease in the same manner
- Each metric should be validated. It should scale up to large systems.
- Software metrics it should work in a variety of programming languages and system domains

1.5. Attributes of Software Metrics

Following are the attributes of software metrics:

- Simple and computable
- Consistent in the use of units and dimensions
- Programming language independent
- An effective mechanism for high-quality feedback

In software development process various models are used for different phases. For example analysis model, design model, source model, testing and maintenance model. Software metrics play an important role in there model to maintain the software quality.

2. Conceptual Models of Software Process

Wedemeijer defines the conceptual process model as an abstracted model of the business process, whose purpose is to provide the actions that are required in producing quality based product [5]. Conceptual process models show what a system does or must do. They are independent of programming language. In the field of software process modeling, the SPEM (Software Process Engineering Metamodel) [6] specification is a generic metamodel used to describe a specific software development process. It is structured in five packages: Basic Elements, Dependences, Process Structure, Process Components and Process Lifecycle. SPEM is based on the UML metamodel.

In the business process field, there are many different proposals for the modeling of business processes. Nevertheless, all these proposals coincide on one point that is the affirmation of the importance of having a graphic notation. Among the methodologies mentioned in the literature, the following deserve special attention in relation to the modeling of business processes: (Integration Definition for Function Modeling) IDEF 0 [12], IDEF 3 [13], UML [11], and BPMN [8]. BPMN is the new standard for modeling business processes and Web services processes, proposed by the Business Process Management Initiative (BPMI). BPMI tries to identity the diversity of proposals and terminology related to business process modeling.

BPMN provides a graphic notation for expressing business processes in a Business Process Diagram (BPD), based on a flowcharting technique. At the same time, it is able to handle the complexity of business processes [9]. Another important characteristic of BPMN is that the XML languages designed for the execution of processes of business, such as BPEL and BPML, can be expressed visually with a common notation. The first goal of BPMN is to provide a notation that can be easily understood by all business users, business analysts, technical developers and business people [10]. To achieve this, BPMN facilitates the modeling of high-level business process through the BPD, which is composed of two basic categories. The first one composed of core elements with which is possible to develop simple process models. The second one contains a complete list of elements that allows the creation of complex or high-level business process models. The four basic categories of elements are Flow Objects, Connecting Objects, Swimlanes and Artifacts.

3. Metrics for Conceptual Models of Software Process

Software process has studied on huge dimensions over recent years, due to the growing complexity of software systems. Therefore the processes' need to continuously undergo changes and refinements in order to increase their ability to deal with the requirements and expectations of the customers. Hence, the processes need to be continuously assessed and improved [14]. Our study is based on the FMESP (Framework for the Modeling and Evaluation of Software Processes) proposal [15, 16], which consists of a framework for the modeling and measurement of the software process. FMESP starts from the idea that a good administration of the software processes must be carried out. For the evaluation of the software process, FMESP includes a set of metrics. This measures the structural complexity of software process models. FMESP is used to evaluate the influence of the structural complexity of the software process models on their maintainability. The FMESP metrics have been defined within two different scopes, one is model scope, to evaluate the overall structural complexity of the model and another is core element scope, to evaluate the specific complexity of the fundamental elements of the model, namely activities, roles and work products. The model scope metrics are shown in the Table 1.

Table 1. Model Scope Metrics

Metric	Definition	Formula
NA	Number of Activities of the software process model	-
NWP	Number of Work Products of the software process model	-
NPR	Number of Roles which participate in the process	-
NDWPIn	Number of input dependences of the Work Products with the Activities in the process	-
NDWPOut	Number of output dependences of the Work Products with the Activities in the process	-
NWDP	Number of dependences between Work Products and Activities	NDWP(PM)=NDWPIn(MP)+ NDWPOut(MP)
NDA	Number of precedence dependences between Activities	-
NCA	Activity Coupling in the process model	$NCA(PM) = \frac{NA(PM)}{NDA(PM)}$
RDWPIn	Ratio between input dependences of Work Products with Activities and total number of dependences of Work Products with Activities.	$RDWPIn(PM)= \frac{NDWPIn(PM)}{NDWP(PM)}$
RDWPOut	Ratio between output dependences of Work Products with Activities and total number of dependences of Work Products with Activities.	$RDWPOut(PM)= \frac{NDWPOut(PM)}{NDWP(PM)}$
RWPA	Ratio of Work Products and Activities. Average of the work products and the activities of the process model	$RWPA(PM) = \frac{NWP(PM)}{NA(PM)}$
RRPA	Ratio of Process Roles and Activities	$RRPA(PM) = \frac{NPR(PM)}{NA(PM)}$

4. Metrics for Base Measures of Elements in Business Process

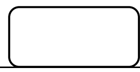
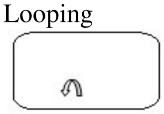
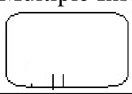
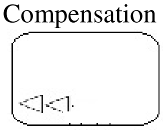
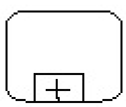
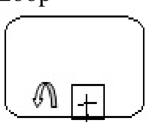
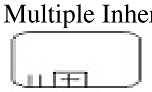
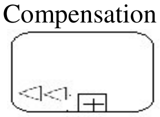
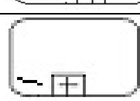
The objective of the validation of the metrics in FMESP is to determine a group of useful indicators of the maintainability of software process models. It can be done by evaluating their structural complexity. The first step in achieving this goal is to define a set of suitable metrics for the evaluation of the structural complexity of business models. These metrics have been grouped into two main categories: Base and Derived Measures [17]. Here we focus on base measures. The base measures have been defined by counting the different kind of elements that a business process model is composed of represented with BPMN. In Table 2, the base measures defined for the constructor "Event" in the BPMN metamodel are shown.

Table 2 Base Measures for the element Event in BPD Object Flow

Core Event	**Notation**	**Metric Name**	**Base Measure**
Start Event	Start	NSNE	Number of Start None Events
	Timer	NSTE	Number of Start Timer Events
	Message	NSMsE	Number of Start Message Events
	Rule	NSRE	Number of Start Rule Events
	Link	NSLE	Number of Start Link Events
Intermediate Event	Intermediate	NINE	Number of Intermediate None Events
	Timer	NITE	Number of Intermediate Timer Event
	Message	NIMsE	Number of Intermediate Message Events
	Error	NIEE	Number of Intermediate Error Events
	Cancel	NICaE	Number of Intermediate Cancel Events
	Compensation	NICoE	Number of Intermediate Compensation Events
	Rule	NIRE	Number of Intermediate Rule Events
	Link	NILE	Number of Intermediate Link Events
End Event	End	NENE	Number of End None Events
	Message	NEMsE	Number of End Message Events
	Error	NEEE	Number of End Error Events
	Cancel	NECaE	Number of End Cancel Events
	Compensation	NECoE	Number of End Compensation Events
	Link	NELE	Number of End Link Events
	Multiple	NEMuE	Number of End Multiple Events
	Terminate	NETE	Number of End Terminate Events

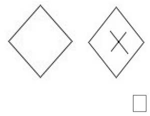
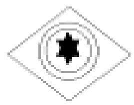
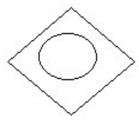
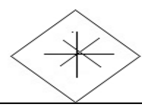
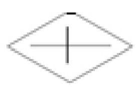
As we can observe in Table 2, the base measures defined for all the triggers of events. They belong to the BPD "Flow Objects" category. The base measures shown in Table 2 were determined according to each of the different types of events that may be contained in a Business Process. Next, in Table 3 below, the base measures for the BPMN metamodel element "activity" are shown.

Table 3. Base Measures for the element Activity of the BPD Flow Objects.

Core Element	**Notation**	**Metric Name**	**Base Measure**	**Definition**
Task	Task 	NT	Number of Task	Indicates the total number of tasks in the model
	Looping 	NTL	Number of Task Multiple Instances	Indicates the total number of task looping in the model
	Multiple Instances 	NTMI	Number of Task Multiple Instances	Indicates the total number of task multiple instances in the model
	Compensation 	NTC	Number of Task Compensation	Indicates the total number of task compensation in the model
Collapsed Sub-process	Collapsed Sub Process 	NCS	Number of Collapsed Sub-Process	Indicates the total number of Collapsed Sub-Processes in the mode
	Loop 	NCSL	Number of Collapsed Sub-Process Looping	Indicates the total number of Collapsed Sub-Process Looping in the model
	Multiple Inheritance 	NCSMI	Number of Collapsed Sub-Process Multiple Instance	Indicates the total number of Collapsed Sub-Process Multiple Instance in the model
	Compensation 	NCSC	Number of Collapsed Sub-Process Compensation	Indicates the total number of Collapsed Sub-Process Compensation in the model
	 Ad-Hoc	NCSA	Number of Collapsed Sub-Process Ad-Ho	Indicates the total number of Collapsed Sub-Process Ad-Hoc in the model

Within Flow Objects, the activity element of the BPD can be made up of atomic activities (tasks) and of compound activities (collapsed sub-processes) and within each category different classes can be observed. In the same category of "Flow Objects", the "Gateways" are the elements used to control the divergence and convergence of Sequence Flow. In the BPD, there are five types of Gateways, and we have defined metrics for each type (Table 4).

Table 4.Base Measures for the Gateway Control Types in the BPD Flow Objects.

Core Element	Notation	Metric Name	Base Measure	Definition
Exclusive Decision Data-Based XOR Decision		NEDDB	Number of Exclusive Decision/Merge Data-Based	Indicates the number of points of exclusive decision and merging based on data of the model
Exclusive Decision Data-Event XOR Decision		NEDEB	Number of Exclusive Decision/Merge Event- Based	Indicates the number of points of exclusive decision and merging based on events of the model
Inclusive (OR)		NID	Number of Inclusive Decision/Merge	Indicates the number of points of inclusive decision and merging of the model
Complex		NCD	Number of Complex Decision/Merge	Indicates the number of points of complex decision merging of the mode
Parallel (AND)		NPF	Number of Parallel Fork/Join	Indicates the number of points of parallel forking and joining of the process

5. Conclusion

In this work it has been proved that metrics for software process models can be applied to business process models. There may be a more refined framework for evaluating business process models. This gives support to Business Process Management. Model metrics can be very useful when selecting the models which are most easily maintained. It can help to facilitate the evolution of business processes in companies by assessing the process improvement at a conceptual level. The metrics for business process models provide companies with objective information about the maintainability of these models. More maintainable models can benefit the management of the business processes in two main ways: i) guaranteeing the understanding and the diffusion of the processes as they evolve, without affecting their successful execution; ii) reducing the effort needed to change the models, with a consequent reduction of maintenance. At present we have survey the measure of elements in business process model. A study can be done on derived measures of elements in business process.

References:-

1. L. J. Arther, "Quantum improvements in software system quality", Commn. Of the ACM, 40:6, June 1997, pp. 47-52.
2. M. Paulk, et. al., "The Capability Maturity Model – Guidelines for Improving the Software Process", Addison Wesley, 1995.
3. M. C. Paulk, "Practices of high maturity organizations", 1999 SEPG Conference, Atlanta, Georgia, March 1999.
4. Paul Goodman, 1993, Practical Implementation of Software Metrics, McGraw Hill, London.

5. d.L. Wedemeijer and d.i.E. de Bruin. "Conceptual Process Models: Using Process Architecture in Practice". In Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA´04). Zaragoza, Spain: IEEE Computer Society. pp. 2004.
6. OMG, "Software Process Engineering Metamodel Specification", adopted specification, version 1.0. Object Management Group, Inc., November, 2002.
7. Park, Robert E., et al.. "Software Size Measurement: A Framework for Counting Source Statements.
8. BPMI, "Business Process Modeling Notation", Specification Version 1.0. Business Process Management Initiative, May 3, 2004. available on www.bpmi.org.
9. M. Owen and J. Raj, "BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard", White Paper. Popkin Software, September, 2003. http://www.popkin.com.
10. S.A. White, Introduction to BPMN. bpmn.org (pub), May 2004 www.bpmn.org.
11. H.-E. Erickson and M. Penker, "Business Modeling with UML- Business Patterns at Work", ed. I. John Wiley & Sons. USA: Robert Ipsen. 2000.
12. FIPS, "Integration Definition for Function Modeling (IDEF0)", Standard. National Institute of Standards and Technology, December, 1993.
13. R.J. Mayer, C.P. Menzel, M.K. Painter, P.S. de White, T. Blinn, and B. Perakath, "Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report", Interim Technical Report. September, 1995.
14. A. Fuggetta. "Software Process: A Roadmap". In Proceedings of the 22th. International Conference on Software Engineering (ICSE-2000). Limerick, Ireland. pp. 25-34, 2000.
15. F. García, F. Ruiz, M. Piattini, G. Canfora, and C.A. Visaggio, "Framework for the Modeling and Evaluation of Software Processes". Journal of Systems Architecture, (accepted to appear). 2006
16. F.O. García, "FMESP. Marco de Trabajo Integrado para el Modelado y la Medición de los Procesos Software", in Departamento de Informática, Escuela Superior de Informática. Universidad de Castilla La Mancha: Ciudad Real. 2004.
17. B.-J. Hommes and V. van Reijswoud. "Assessing the Quality of Business Process Modelling Techniques". In Proceedings of the 33rd Hawaii International Conference on Systems Sciences (HICSS 2000). Maui, Hawaii, USA: IEEE. pp. 1007-1016, 2000.