UNIVERSITI PUTRA MALAYSIA

USE OF AN ATTRIBUTE GRAMMAR FOR SOFTWARE PROCESS MEASUREMENT

RODZIAH BINTI ATAN.

FSKTM 2005 1

USE OF AN ATTRIBUTE GRAMMAR FOR SOFTWARE PROCESS MEASUREMENT

RODZIAH BINTI ATAN

DOCTOR OF PHILOSOPHY
UNIVERSITI PUTRA MALAYSIA

2005

USE OF AN ATTRIBUTE GRAMMAR FOR SOFTWARE PROCESS MEASUREMENT

By

RODZIAH BINTI ATAN

Thesis submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirements for the Degree of Doctor of Philosophy

November 2005

بسم الله الرحمن الرحيم

Alhamdulillah... segala puji bagi Allah kerana dengan limpah rahmatNya dapat saya menyiapkan tesis ini.

Ayahanda Hj. Atan bin Awang dan Bonda Hjh. Halijah binti Awang, jasa kalian berdua mengasuh dan membesarkan anakanda tidak akan sia-sia dan tidak akan dipersiakan.

Suami tercinta Mohd. Nazri bin Hj. Yusof, banyak yang boleh diperkatakan.. tapi biarlah ia menjadi pengalaman kita sehingga akhir hayat .. Kesabaran, pengorbanan, pengalaman, pahit getir hidup kita banyak menjadi pendorong.. The only thing I could say is, "Thank you for everything and I love you."

Anak-anak tersayang, Muhammad Naqiuddin dan Nur Dini Syaira.. dua insan kecil yang menyeri dan menyinari hidup ini. Semoga anak-anak Ibu kelak menjadi manusia berguna di dunia dan di akhirat.

Abang-abang yang dihormati, Abang Kaya, Abang Li dan Abang Nuar, dorongan dan nasihat sentiasa menjadi pegangan hidup.

Kakak-kakak yang dikagumi, Kak Long, Kak Ngah, Kak Uda dan Kak Cik, kelembutan serta kebijaksanaan selalu menjadi contoh peribadi yang mulia.

Ayah dan Ibu mertua, ipar duai dan anak-anak saudara, kalian adalah semangat dan penghibur di kala kesunyian.

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirements for the degree of Doctor of Philosophy

USE OF AN ATTRIBUTE GRAMMAR FOR SOFTWARE PROCESS MEASUREMENT

By

RODZIAH ATAN

November 2005

Chairman: Associate Professor Abdul Azim Abd. Ghani, PhD

Faculty: Computer Science and Information Technology

The creations of models are essential for many knowledge disciplines to explain expected results. Modelling concept is well accepted in software engineering discipline. Some software models were built either to control the development stages or to serve as a medium that gives better understanding of the actual software systems. However, there is still a lacking of software process measurement being discussed and explored by software engineers. An aim of this research is to integrate measurement in software process modelling, to show that measurement in modelling software processes is important whereby to reduce flaws in developing large software.

Software process modelling has reached the level that allows software designs to be transformed into programming languages. Examples of such approaches are architecture design language (ADL) and unified modelling language (UML). Selecting a modelling technique is essential to the designers and the selection

depends on the needs of the system. This research focused on IDEF3 Standard notation as its approach to design software process models. The language syntax provided by IDEF3 standard have been modified and enhanced to suit the aim of the research. This new version of IDEF3 context-free grammar is named IDEF3-SPMA language.

IDEF3-SPMA language constructs and measurement metric defined in this research has been verified using attribute grammar approach. A prototype tool for automatic process model metrics calculation namely Software Process Measurement Application (SPMA) is developed to realize the definitions defined. SPMA system counts the particulars of a process model design and output a list of measurement values. Inference metrics and appropriate advices are also stated to indicate the use of the metric values. Testing and verification results indicated that with a standard notation of performing a software process model designs, the structure and relationship between processes could be clearly seen. This research also shows that automatic calculation of process model design measurement is possible using a well-defined language specification and a specific calculation tool. Several measurement metrics produced by SPMA system are such as number of calling sub-processes in the design, number of sub-junction processes and the size of a process structure.

It is concluded that this research has produced an environment consisting of design notation, language constructs, a tool that enabled measurement metrics to be

calculated automatically, metric inferences and appropriate advices. The specification of measurement metrics definition using attribute grammar, the SPMA tool, metric inference and metric advices are contributions of this research.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah

PENGGUNAAN NAHU ATRIBUT BAGI PENGUKURAN PROSES PERISIAN

Oleh

RODZIAH ATAN

November 2005

Pengerusi: Profesor Madya Abdul Azim Abd. Ghani, PhD

Fakulti: Sains Komputer dan Teknologi Maklumat

Mencipta model bagi kebanyakan disiplin pengetahuan adalah penting bagi menerangkan jangkaan keputusan. Konsep pemodelan diterima dengan baik di dalam disiplin kejuruteraan perisian. Beberapa model perisian dibina samada untuk mengawal tahap pembangunan atau untuk berkhidmat sebagai perantara bagi memberi kefahaman yang lebih jelas tentang sistem perisian yang sebenar. Walaubagaimanapun, masih terdapat kekurangan bagi pengukuran proses perisian yang dibincang dan diteroka oleh jurutera perisian. Satu sasaran kajian ini adalah untuk menyuaipadan pengukuran ke dalam pemodelan proses perisian, bagi menunjukkan bahawa pengukuran di dalam pemodelan proses perisian adalah penting dalam mana bagi mengurangkan kesilapan semasa membangunkan perisian yang besar.

Pemodelan proses perisian sekarang telah mencapai satu tahap yang membenarkan rekabentuk perisian diubah ke dalam bentuk bahasa pengaturcaraan. Contoh bagi

pendekatan tersebut adalah bahasa senibina rekabentuk (ADL) dan bahasa pemodelan gabungan (UML). Memilih teknik pemodelan adalah penting bagi pereka dan pilihan tersebut bergantung kepada keperluan sistem. Kajian ini memfokus kepada notasi piawaian IDEF3 sebagai pendekatannya untuk merekabentuk proses perisian. Sintaksis bahasa yang disediakan oleh piawaian IDEF3 telah diubahsuai dan dipertingkatkan bagi disesuaikan dengan sasaran kajian ini. Versi baru nahu kontek bebas IDEF3 ini dinamakan bahasa IDEF3-SPMA.

Binaan bahasa IDEF3-SPMA dan definisi metric pengukuran dalam kajian ini telah ditentusahkan menggunakan pendekatan nahu atribut. Sebuah peralatan prototaip bagi pengiraan metrik pemodelan proses secara automatik bernama *Software Process Measurement Application* (SPMA) dibina bagi merealisasi takrifan definisi-definisi tersebut. Sistem SPMA menghitung perincian sesebuah model rekabentuk proses dan mengeluarkan satu senarai nilai pengukuran. Metrik inferens dan nasihat yang berkaitan juga dicatat untuk menunjukkan kepenggunaan nilai-niali metrik tersebut. Keputusan pengujian dan verifikasi menunjukkan bahawa dengan penggunaan notasi piawai bagi membentuk rekabentuk model proses perisian, struktur dan hubungan antara proses dapat dilihat dengan jelas. Kajian ini juga menunjukkan pengiraan pengukuran rekabentuk model proses secara automatik berkemungkinan dilaksanakan dengan menggunakan spesifikasi bahasa dan peralatan yang spesifik. Beberapa metrik pengukuran yang dihasilkan oleh sistem SPMA adalah seperti bilangan subproses pemanggil di dalam rekabentuk, bilangan proses subhubungan dan saiz sesuatu struktur proses.

Adalah disimpulkan bahawa kajian ini telah menghasilkan sebuah persekitaran mengandungi notasi rekabentuk, binaan bahasa, peralatan yang membenarkan pengukuran dilaksanakan secara automatik, metrik inferens dan nasihat yang berkaitan. Spesifikasi definisi metrik pengukuran menggunakan nahu atribut, peralatan SPMA, metrik inferens dan nasihat metrik adalah merupakan sumbangan kajian ini.

ACKNOWLEDGEMENTS

Bismillah…

I would like to take this opportunity to record my gratitude towards many people that contributed much throughout my study in Faculty of Computer Science and Information Technology. I wish to express extreme gratitude to my supervisor, Associate Professor Dr. Abdul Azim bin Abd. Ghani for his highly and invaluable guidance, urges and stimulating suggestions that were enabled me to go through the tough time during my studies and enables me to complete the research and thesis. A thousand thanks goes to my co-supervisors, Associate Professor Dr. Ramlan bin Mahmod and Associate Professor Hj. Mohd. Hasan bin Selamat for their full commitment and constructive ideas, which had given me strength to accomplish these hard duties.

I would like to convey my gratitude to Universiti Putra Malaysia for providing me a place to do my research and studies and also a job where I can earn my living all these years. I sincerely would like to thank Jabatan Perkhidmatan Awam (JPA) for sponsoring me with SLAB scholarship which funding me with a large amount of money. I would also like to thank the dean's Secretary, Puan Norhaidah for her help and patience.

A special thank goes to Technical Department of FSKTM, which provides me with technical knowledge and hardware support that enables me to accomplish my study. Special thanks also to a colleague who was formerly FSKTM member whom did the set up to my Linux operating system, and also good friends of mine who are at first floor for all the kindness, help and support.

Tokens of gratitude to my family who had and always give great critiques and support for me in my journey to persuade the success of accomplishing my Ph.D. My father and mother, in-laws, brothers and sisters, nieces and nephews, and my beautiful son and daughter… without them, my life will never been great and colourful. They have been contributing bits in several different ways and I really appreciated them all. For my husband, the one who supports, who critiques, who comments, who gives ideas, who prayed, who willingly took over my daily chores, who smiled, cared, loves… My greatest appreciation is for you. Thank you all for many things that had happened. I love you all and may GOD bless all of you. AMIN.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Developing software within scheduled time and estimated cost is a difficult task for many software development companies. Any flaws or late delivery of a system gives a great impact for many individuals or companies involved. It is indeed vital to produce usable software within time scheduled to avoid inconveniences for the developers, vendors and users in terms of costs and punctualities. One of the steps taken in order to satisfy promptness issue is planning. Planning to develop a software system requires several steps to be taken into account, such as gathering requirement information, estimating cost and effort and also sketching the layout of the system model.

The creation of model is essential for many knowledge disciplines to explain predicted results. Modelling concept is well accepted in software engineering discipline. Some software models were built either to control the development stages, to measure program quality or to serve as a medium that gives better understanding of the actual software systems. Software process modelling is one category of modelling activity.

Software process modelling is a challenging task from software development perspective. It requires creative analysis and design for problem solving, and for a strategic co-ordination of the development team. In order to establish process modelling as a unique area, researchers must identify its conceptual boundaries that distinguish the works from modelling techniques in other technological sciences. The main difference between software process modelling and other modelling activities is that it should be particularly and explicitly focus on phenomena that occur during software creation and evolution.

The software community places great hope on software modelling notations and techniques to ease various software development challenges. Software process modelling is one of the techniques used to creatively define and analyse significant aspects of an enhanced application and to structure a strategic co-ordination of the development team. The intellectual tool set available for software developers has steadily been enriched with more powerful and comprehensive models. There have been many approaches introduced to this particular field of software engineering. It started from the basic structure of software designing model and evolved throughout the time. Software process modelling nowadays has reached a level that allow software designs to be transformed into languages, such as architecture design language (ADL) and unified modelling language (UML).

These two types of process modelling languages proved that people in software development team are able to execute their designs. There are other existing software

process notations and enactions that give much more choices of method for software developers to improve their process models. Above all the benefits offered by these known techniques, one factor that needs to be emphasized more into is, measurement.

In this research, there are eight measurement metrics that has been defined. The measurement metrics are defined and proven structure and specification-wise using attribute grammar (AG) approach. AG is selected to prove definitions for the language specification for its renowned efficiency in specifying language-based rules and definitions. The aim of this research is to integrate measurement in software process modelling, to show that measurement in modelling software processes is important at crucial moment to reduce flaws especially during the development of large software systems. In order to achieve the aim, we utilize a process definition standard IDEF3, which is mainly used in business process modelling, create a corresponding language constructs for model designs, define the measurement metrics and prove the metrics accordingly.

To realize the automatic calculation of the measurement metrics, we develop a **S**oftware **P**rocess **M**easurement **A**pplication **(SPMA)** system tool that executes the counting of process models' measurement metrics. SPMA is developed to automate assistance from the implementation of particular measurement metrics defined in the study, for determining the decomposition and structure of a process model design of a *to be built* software systems.

This thesis will describe the measurement metrics definition and specification language, discussing the architecture and elements put together to produce the prototype and validation of the SPMA system. It also illustrates and demonstrates the usage of SPMA by testing it with multiple software process model designs supported with actual users testing the prototype themselves.

1.2 Statement of Problem

Developing software systems is not an easy task. Many software systems face the risks of having flaws and malfunctions. Errors found during delivering the software system is highly potential been caused by the failure while coding the system, or it could happen while designing the product. Repairing a 'completed' software system costs a lot. The best opportunity for short-term software cost reduction is to eliminate rework or fixing defects that is more than 33 percent of new development (Grady, 1997).

The problem of reworking a software system is highly potential be avoided by tackling the predicted problems far before the system is developed or implemented, i.e. at stages of requirement gathering and system analysis and design (Sommerville, 2004). Grasp from the readings we found out that measurement in process modelling can aid in determining characteristics of a software system. By determining characteristics of a particular software system, predicted problems