M.O.O.T

META OBJECT ORIENTATED TOOL

A NOVEL META-CASE TOOL METHODOLOGY REPRESENTATION STRATEGY

A dissertation submitted in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Massey University, New Zealand

David Charles Page

1998

Abstract

This thesis presents an investigation into current meta-CASE technology. The research focuses on CASE tool support for the concept of methodology, the representation of methodology syntax and semantics, and the support for re-use of methodology descriptions and software artefacts. A novel methodology representation strategy for meta-CASE tools is proposed and implemented with the development of a new meta-CASE tool (MOOT – Meta Object Orientated Tool).

The novel strategy propounded in this thesis uses an object-orientated meta-model and views methodology descriptions as potentially re-usable components. The coupling between methodology syntax and semantic descriptions is minimised so they can be re-used independently.

Two new modelling languages have been derived, to support the definition of syntax (NDL – Notation Definition Language) and semantics (SSL – Semantic Specification Language) of software engineering methodologies. Semantic descriptions are compiled to a platform independent representation (SSL-BC), which is executed on a purpose built virtual machine (SSL-VM). Late binding of syntax and semantic methodology descriptions is implemented with the development of Notation Semantic Mapping (NSM) tables. Two libraries of re-usable methodology description components, the Core Knowledge Base (CKB) and the Generic Object Orientated Knowledge Base (GOOKB), have been derived during this research.

Empirical results gained from applying the MOOT prototype demonstrated the flexibility, extensibility and potential of the novel methodology representation strategy. This approach permitted the implementation and modelling of UML and patterns, two recent advances of object technology that did not exist when the research commenced.

The novel strategy presented in this thesis is more than an untried theory. It has been implemented, applied and is being evaluated. Simply, it is real and it works.

DEDICATION

This thesis is lovingly dedicated to my parents

Michael Julius Page and Susan Evelyn Page

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Colours

You reap what you sow. Put your face to the ground.
Here come the marching men. Your colours wrapped around.

The Sisters of Mercy

The research detailed in this thesis was supported by two PGSF funded research projects (MAU-503, MAU-807). Financial assistance was also received with a New Zealand postgraduate scholarship.

The following people deserve special mention.

My supervisors, who inspired me and taught me a great deal

Assoc. Prof. Daniela Mehandjiska-Stavreva, Prof. Mark Apperley

The Masters and Honours students who were also involved in this research

Paul Clark, Steven Adams, Hong Yu, Duane Griffin, Sarisha Dasari, Mi Duk Choi, Jonathan Ham

Two people that selflessly proof-read the thesis

Rachel Page, Wendy Browne

My family, without whom this would not have been possible

Michael Page, Susan Page, Audrey Isaac, Rachel Page, Jonathon Page, Ruth Page

My friends (you know who you are), who all helped without knowing it

Extra thanks to: Nick Earle, Paul Clark, Duane Griffin, Lisabeth Weston, Shamus Smith, Luke Usherwood, Marion Moore, Andrew Turvey, James Fulton, Steven Adams

GLOSSARY

The content of the glossary has been derived from a range of dictionaries (Collins, 1995; Nuttals, 1902; Readers Digest, 1988; Oxford, 1993; Mirriam-Webster, 1998), the Dictionary of Object Technology (Firesmith and Eykholt, 1995) and (D'Souza and Wills, 1998; Jacobson *et al.*, 1995; Pressman 1997; Schach, 1993, 1997; Somerville, 1996).

Abstraction. Any model that includes the most important, essential, or distinguishing aspects of something while suppressing or ignoring less important, immaterial, or diversionary details. The result of removing distinctions so as to emphasise commonalties.

Arity. The cardinality of something. For example the arity of a relation specifies the number of concepts that are involved in the relation.

Attribute. Any named property used as a data abstraction to describe its enclosing object, class or extent.

Behaviour. Anything that an organism does involving action and response to stimulation. The way in which someone behaves; also: an instance of such behaviour.

Bind. To place under certain constraints. To cohere or cause to cohere. To place under obligation; oblige.

Binding. Any selection of the appropriate method for an operation on receipt of a corresponding message.

Browser. Any view that allows you to access hierarchically organised and indexable information.

CASE Tool. A) Any computer based tool for software planning, development and evolution. This includes all examples of computer-based support for the managerial, administrative, or technical aspects of any part of a software development project. B) Products that assist the software engineer in developing and maintaining software.

CASE. An acronym that stands for Computer Assisted Software Engineering.

CKB. Core Knowledge Base. A library of methodology semantic components that implements a meta-model of methodology.

Class. Any uniquely identified abstraction (i.e. a model) of a *set* of logically related instances that share the same or similar characteristics. The combination of a type interface and associated type implementation.

Classification. The act of forming into a class or classes; a distribution into groups such as classes, orders, families, etc., according to some common relations or affinities.

Cohesion. The degree, to which something models a single abstraction, localising only features and responsibilities related to that abstraction.

Component. A) Any standard, reusable, previously implemented unit that is used to enhance the programming language constructs and to develop applications. B) An independently deliverable unit of software that encapsulates its design and implementation and offers interfaces to the out-side, by which it may be composed with other components to form a larger whole.

Coupling. The degree to which one thing depends on another. Low coupling is desirable because it produces better encapsulation, maintainability, and extendibility with fewer objects needlessly affected during iteration.

Encapsulation. To enclose in or as if in a capsule; the act of enclosing in a capsule. The physical localisation of features.

Engineering. The application of scientific principles to such practical ends as the design, construction, and operation of efficient, economical structures, equipment and systems. The application of science to the design, building, and use of machines, constructions etc.

GOOKB. Generic Object Orientated Knowledge Base. A library of methodology semantic components that implements a meta-model of concepts germane to all object-orientated methodologies.

Identity. Individuality.

Information Hiding. The deliberate and enforced hiding of information (e.g. design decisions, implementation details) from clients. The limiting of scope so that some information is invisible outside of the boundary of the scope.

Inheritance. The incremental construction of a new definition in terms of existing definitions without disturbing the original definitions and their clients.

Instance. Anything created from or corresponding to a definition.

Interface. The visible outside, user view of something.

Language. Any method of communicating ideas, as by a stream of signs, symbols, gestures or the like. The special vocabulary and usage of a scientific professional or other group. The speech or expression of ideas.

MDT. An acronym that stands for Methodology Description Table. The Methodology Description Table provides an index of the methodologies supported by MOOT.

Message Send. The sending of a message to an object.

Message. Any communication sent or received by an object.

Meta. A Greek prefix signifying beyond, after, with, among and frequently expressing change. Going beyond or transcending. Used with the name of a discipline to designate a

new but related discipline designed to deal critically with the original one. Of a higher or second-order kind.

Meta-CASE Tool. *A)* A meta-CASE tool is any tool that provides automated or semi-automated support for developing CASE tools. *B)* ... are CASE tools which are used to generate other CASE tools. *C)* A CASE tool that operates on CASE tools.

Meta-language. The natural language, formal language, or logical system used to discuss or analyse another system. A form of language used to discuss a language.

Method. *A)* Mode of procedure, logical arrangement, orderly arrangement, system of classification. A means or manner of procedure, especially a regular and systematic way of accomplishing anything. The procedures and techniques characteristic of a particular discipline or field of knowledge. A special form of procedure esp. in any branch of mental activity. *B)* A way of carrying out a complete phase such as such as design or integration. *C)* The hidden implementation of an associated operation.

Methodology CASE Tool. A CASE tool that supports one or more software development methodologies and attempts to span most of the software development life-cycle.

Methodology. The science of scientific method of classification. From the Greek method and logis (science). The system of principles, practices, and procedures applied to any specific branch of knowledge. The science of method; a body of methods used in a particular branch or activity.

Model. *A)* Archetype; a description or analogy used to help visualise something that cannot be directly observed; a system of postulates, data, and inferences presented as a mathematical description of an entity or state of affairs. A preliminary pattern or representation of an item not yet constructed. A tentative framework of ideas describing something intangible and used as a testing device. *B)* A model clarifies – for a person or group of people – some aspect or perspective on a thing or event.

MOOT. Meta Object Orientated Tool. A new meta-CASE tool developed as a result of this research.

NDL. Notation Definition Language. A new language used to define the syntax of a methodology in MOOT.

Notation. A system of characters, symbols, or abbreviated expressions used in an art or science or in mathematics or logic to express technical facts or quantities.

NSM. An acronym that stands for Notation-Semantic Mapping. NSM tables are used to implement late binding of NDL and SSL methodology descriptions.

Object. Any abstraction that models a single thing.

Operation. Any service that may be requested.

Polymorphism. The ability of a single name to refer to different things having different forms.

Process. A) A system of operations in the production of something. A series of actions, changes or functions that bring about an end or result. A course of action or proceeding, esp. a series of stages in manufacture or some other operation. B) ... the way we produce software. It starts with concept exploration and ends when the product is finally retired. C) ... the set of activities and associated results which produce a software product.

Relation. Connection by consanguinity or affinity; kinship; relationship; as, the relation of parents and children; an abstraction belonging to, or characteristic of, two entities or parts together.

Semantic. Of, or relating to, meaning in language.

Software Development Life-cycle (SDLC). A process by which software engineers build computer applications.

Software Engineering. A) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is the application of engineering to software. B) ... is concerned with the theories, methods and tools that are needed to develop software for computers. C) A discipline whose aim is the production of quality software that satisfies the user's needs, and is delivered on time and within budget.

Software Project. A software project consists of a set of models (built using a particular methodology) which collectively define the software being constructed.

SSL. Semantic Specification Language. A new object-orientated language used to define the semantics of a methodology in MOOT.

SSL-BC. SSL Byte Code. A platform independent, binary, representation of SSL, which is generated by the SSL compiler.

SSLC. SSL compiler.

SSL-VM. SSL Virtual Machine. A new virtual machine which supports efficient processing of SSL.

State. Any status, situation, condition, mode, or life-cycle phase of an object or class during which certain rules of overall behaviour (e.g. response to messages) apply.

Syntax. The way in which linguistic elements (as words) are put together to form constituents (as phrases or clauses).

Tool. A thing used in an occupation or pursuit. Any instrument of use or service.

Type. A lower taxonomic category selected as a standard of reference for a higher category. The declaration of the interface of any set of instances (e.g. objects) that conform to this common protocol.

PUBLICATIONS

The results of this research have been presented in the following refereed publications.

Phillips, C.H.E., Adams, S., **Page, D.** and Mehandjiska, D. (1998) The Design of the Client User Interface for a Meta Object-Oriented CASE tool, Proceedings of TOOLS Pacific'98, Monash Printing Services, Victoria, pp145-157

Page, D., Mehandjiska, D. and Phillips, C.H.E. (1998) Methodology Independent OO CASE: Supporting Methodology Engineering, Proceedings of Software Engineering: Education and Practise (SE:E&P'98), IEEE Computer Society Press, Dunedin, New Zealand, pp373-380

Phillips, C.H.E., Adams, S., **Page, D.** and Mehandjiska, D. (1998) Design of the User Interface for a Methodology Independent OO CASE Tool, Proceedings of OZCHI'98, IEEE Computer Society Press, Los Alamitos, California, pp106-114

Phillips, C.H.E., Mehandjiska, D. and **Page, D.** (1998) The Usability Component of a New Framework for the Evaluation of Object-Oriented CASE tools, Proceedings of Software Engineering: Education and Practise (SE:E&P'98), IEEE Computer Society Press, Dunedin, New Zealand, pp131-141

Page, D., Griffin, D., Usherwood, L. and Mehandjiska, D. (1997) Implementation of a Semantic Specification Language Interpreter for a Methodology Independent OO CASE Tool, Proceedings of IASTED International Cenference on Software Engineering (SE'97), ACTA Press, San Francisco, USA, pp239-242

Mehandjiska, D., **Page, D.**, Griffin, D. and Usherwood, L. (1997) Methodology Knowledge Representation and Interpretation for a Methodology Independent OO CASE Tool, Proceedings of IASTED International Cenference on Software Engineering (SE'97), ACTA Press, San Francisco, USA, pp243-247

Mehandjiska, D., **Page, D.** and Choi, M. D. (1996) Meta-Modelling and Methodology Support in Object-Oriented CASE Tools, Proceedings of 3rd International Conference on Object-Oriented Information Systems (OOIS'96), Eds. Patel, D., Sun, Y. and Patel, S., Springer-Verlag, London, pp370-386

Mehandjiska, D., **Page, D.** and Dasari, S. (1996) Generic Knowledge Base for a Methodology Independent Object-Oriented CASE Tool, Proceedings of the IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks, Ed. Hamza, M., IASTED/Acta Press, Honolulu, Hawaii, pp23-26

Mehandjiska, D., Apperley, M. D., Phillips, C.H.E., Dasari, S. and **Page, D.** (1996) Advancing information technologies through CASE, Proceedings of the 19th Australasian Computer Science Conference (ACSC'96), Ed. Ramamohanarao, K., Melbourne, Australia, pp213-222.

Dasari, S., Mehandjiska, D. and **Page, D.** (1995) Construction of a Generic Knowledge Base for a Methodology Independent CASE Tool, Addendum to the Proceedings of The Second NZ International Two-Stream Conference on Artificial Neural Networks and Expert Systems (ANNES'95), Dunedin, pp466-473

Mehandjiska, D., Apperley, M.D., Phillips, C., **Page, D.** and Clark, P. (1995) A Methodology independent object oriented CASE tool, New Zealand Journal of Computing, Vol. 6, pp95-105

Mehandjiska, D., **Page, D.** and Ham, J. (1995) Template generator for methodology independent object oriented CASE tool, Proceedings of 2nd International Conference on Object-Oriented Information Systems (OOIS'95), Eds. Murphy, J. and Stone, B., Springer-Verlag, Dublin, Ireland, pp431-440

Page, D., Clark, P. and Mehandjiska, D. (1994) An Abstract Definition of Graphical Notations for Object Orientated Information Systems, Proceedings of 1st International Conference on Object-Oriented Information Systems (OOIS'94), Eds. Patel, D., Sun, Y. and Patel, S., Springer-Verlag, London, pp266-276

Mehandjiska, D., **Page, D.** and Clark, P. (1994) An Intelligent Object Oriented CASE Tool, Proceedings of 1st International Conference on Object-Oriented Information Systems (OOIS'94), Eds. Patel, D., Sun, Y. and Patel, S., Springer-Verlag, London, pp168-172