Unity-inspired object-oriented concurrent system development

by

Marlene Maria Ross

Thesis submitted in fulfillment of the requirements for the degree
Doctor of Philosophy in Computer Science
in the Faculty of Natural and Agricultural Sciences
University of Pretoria
Pretoria

January 2001

Unity-inspired object-oriented concurrent system development

by

Marlene Maria Ross

Thesis submitted in fulfillment of the requirements for the degree
Doctor of Philosophy in Computer Science
in the Faculty of Natural and Agricultural Sciences
Department of Computer Science
University of Pretoria
Pretoria

January 2001

Supervisor: Professor D.G. Kourie
Co-supervisor: Professor R.J. van den Heever

ABSTRACT

The design of correct software remains difficult, especially when dealing with concurrency. The primary goal of the research presented here is to devise a **pragmatic** software development method which

- aids the software designer in producing **reliable** software,
- is **scalable**,
- is **understandable**,
- follows a **unified approach** towards software development (is applicable to different implementation architectures),
- promotes **reuse**,
- has **seamless** transitions between the software development phases,
- guarantees **general availability** and **minimises developmental resources**.

The two main characteristics of the proposed new development method are captured in its name, viz. **Single Location Object-Oriented Programming (SLOOP)**. It is an **object-oriented** method, but its **computational model** is that of a set of statements that execute infinitely often and in any order. A program with such a computational model is called a Single Location Program (SLP). A UNITY program can also be classified as a Single Location Program. In the UNITY theory of programming it was demonstrated how this computational model could **simplify correctness reasoning**, particularly for concurrent systems.

It is this simplification, together with the structuring and reuse features of object-orientation, that is leveraged in the SLOOP method to produce a mechanism whereby **ordinary software practitioners** can take advantage of the benefits of a more rigorous approach towards software development without requiring an in-depth understanding of the underlying mathematics.

The following features of the SLOOP method contribute towards achieving the above goals:

- its **computational model** (it simplifies correctness reasoning, thereby promoting understandability and scalability, and also facilitates designs that are independent of the target implementation architectures),

- its **object-oriented** nature (apart from promoting reuse of frameworks, design patterns and classes, the SLOOP method provides the necessary mechanisms to facilitate reuse of correctness properties, correctness arguments as well as mappings to implementation architectures),
- its **emphasis on correctness reasoning** throughout the software development life cycle (its "constructive approach" aids reliability and seamlessness),
- the **unique** way in which the correctness properties can be specified, reused and reasoned about (this contributes towards understandability and scalability),
- the **checklist** of useful correctess properties that is provided (this promotes reliability),
- the **incorporation of existing notations** into the SLOOP syntax (this guarantees general availability, minimises developmental resources and aids understandability).

The main **contribution** of this thesis is that it presents a **unique** way of incorporating the SLP computational model into an object-oriented method with the specific aim of **simplifying informal correctness reasoning** and **promoting reuse**. The notation used for the specification of correctness properties facilitates reuse of correctness properties, ensures the integrity of these specifications and allows one to specify correctness properties at a higher level of abstraction.

The SLOOP method offers a **unique way of modelling concurrency in object-oriented systems** (via its parallel methods), which takes full advantage of the **encapsulation** and **inheritance** features of object-orientation. The issues surrounding **mappings** to implementation architectures are addressed, showing how even mappings can be **reused.** Finally, the **general applicability** of the SLOOP method is demonstrated.

SAMEVATTING

Die ontwerp van korrekte programmatuur bly moeilik, veral wanneer gelyktydigheid ter sprake is. Die primêre doel van hierdie navorsing is om 'n **pragmatiese** programmatuur ontwikkelingsmetode te ontwikkel wat

- die ontwerper sal help om **betroubare** programmatuur te ontwikkel,
- **skaaleerbaar** is,
- **verstaanbaar** is,
- 'n **eenvormige benadering** volg ten opsigte van programmatuurontwikkeling,
- **hergebruik** aanmoedig,
- 'n **gladde oorgang** tussen ontwikkelingsfases teweegbring,
- **algemene beskikbaarheid** waarborg en die **gebruik van ontwikkelingsbronne minimeer**.

Die twee hoofeienskappe van die voorgestelde nuwe ontwikkelingsmetode word weerspieël in die naam - **Enkel Posisie Objek-georiënteerde Programmering (EPOP)**. Dit is 'n **objek-georiënteerde** metode, maar die **verwerkingsmodel** is die van 'n groep stellings wat oneindig gereeld en in enige volgorde uitvoer. 'n Program met so 'n verwerkingsmodel word 'n Enkel Posisie Program (EPP) genoem. 'n UNITY program kan ook geklassifiseer word as 'n EPP. In die UNITY teorie van programmering is gedemonstreer hoe hierdie verwerkingsmodel die **beredenering van korrektheid** kan **vereenvoudig**, spesifiek vir gelyktydige stelsels.

Dit is hierdie vereenvoudiging, tesame met die strukturering en hergebruikseienskappe van objek-oriëntasie, wat aangewend word in die EPOP metode om 'n meganisme te produseer waar **gewone programontwerpers** die voordele van 'n meer formele benadering teenoor programmatuurontwikkeling kan benut, sonder om noodwendig die beginsels van die onderliggende wiskunde te hoef te bemeester.

Die volgende eienskappe van die EPOP metode dra by om bogenoemde doelstellings te bevredig:

- die **verwerkingsmodel** (dit vereenvoudig korrektheidsberedenering, dus verhoog dit ook verstaanbaarheid en skaleerbaarheid, en maak ook ontwerpe moontlik wat onafhanklik is van die implementasie argitektuur),
- die **objek-georiënteerde** aard (bo en behalwe die feit dat dit hergebruik van raamwerke, patrone en klasse aanmoedig, voorsien die EPOP metode die meganismes wat die hergebruik van korrektheidseienskappe, korrektheidsargumente, asook afbeeldings na die implementasie argitektuur moontlik maak),
- die deurgaande **klem op korrektheidsberedenering** tydens die progammatuur= ontwikkelingsfases (die 'konstruktiewe benadering' moedig betroubaarheid en 'n gladde oorgang tussen ontwikkelingsfases aan),
- die **unieke** manier waarop die korrektheidseienskappe gespesifiseer, hergebruik en beredeneer kan word (dit dra by tot verstaanbaarheid en skaleerbaarheid),
- die **lys van nuttige korrektheidseienskaptipes** wat verskaf word (dit moedig betroubaarheid aan),
- die **insluiting van bestaande notasies** in die EPOP sintaks (dit waarborg algemene beskikbaarheid, minimeer die gebruik van ontwikkelingsbronne en verhoog verstaanbaarheid).

Die **hoofbydrae** van hierdie verhandeling is die feit dat dit 'n **unieke** manier bied om die EPP verwerkingsmodel in 'n objek-georiënteerde metode te inkorporeer, met die spesifieke doel om **informele korrektheidsberedenering te vereenvoudig** en **hergebruik aan te moedig**. Die notasie wat gebruik word vir die spesifikasie van korrektheidseienskappe maak die hergebruik

van korrektheidseienskappe moontlik, waarborg die integriteit van hierdie spesifikasies en laat die spesifikasie van korrektheidseienskappe op 'n hoër vlak van abstraksie toe.

Die EPOP metode bied 'n **unieke manier om gelyktydigheid in objek-georiënteerde stelsels te modelleer** (by wyse van parallelle metodes), wat ten volle voordeel trek uit die **enkapsulasie** en **oorerwingseienskappe** van objek-oriëntering. Die kwessies rondom die **afbeeldings** op implementasie argitekture word geadresseer, en wys uit hoe selfs hierdie afbeeldings **hergebruik** kan word. Laastens word die **algemene toepaslikheid** van die EPOP metode gedemonstreer.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to the following people:

Professor D. Kourie, my supervisor, for his stimulating discussions, excellent advice and attention to detail. His encouragement and support have been invaluable.

Professor R. J. van den Heever, the co-supervisor, for his helpful suggestions, many stimulating discussions and careful reading of this thesis. His vision and insight regarding the various aspects of Computer Science have been a continued source of inspiration and motivation.

The management team at Azisa, who allowed me to work flexible hours in order to provide me with the opportunity to devote time to my studies.

My husband and parents, for their continued encouragement and all the opportunities that they have given me.

DECLARATION

I hereby declare that, unless otherwise indicated, all the work in this thesis was done by myself and that this thesis has not been submitted to this or any other institution of learning in support of an application for any other degree or qualification.

M. M. Ross
January 2001

DEDICATION

To Victor, for his love, support and understanding

and to Pieter Thomas and Emul, for being two such lovable boys.

TABLE OF CONTENTS

LIST OF FIGURES