R. C. Veltkamp and
E. H. Blake (eds.)

Programming Paradigms in Graphics

Proceedings of the Eurographics Workshop
in Maastricht, The Netherlands,
September 2-3, 1995

Eurographics

SpringerWienNewYork

Dr. Remco C. Veltkamp
Department of Computer Science, University of Utrecht, The Netherlands

Dr. Edwin H. Blake
Department of Computer Science, University of Cape Town,
Rondebosch, South Africa

Typesetting: Camera ready by editors

Graphic design: Ecke Bonk

Printed on acid-free and chlorine-free bleached paper

With 41 partly coloured Figures

ISSN 0946-2767

ISBN-13:978-3-211-82788-8 e-ISBN-13:978-3-7091-9457-7
DOI: 10.1007/978-3-7091-9457-7

Preface

The papers in this volume are a good sampling and overview of current solutions to the problems of creating graphically based systems. This breadth of scope comes out of the closing discussion at the Fourth Eurographics Workshop on Object-Oriented Graphics. The fifth workshop, on Programming Paradigms in Graphics, set out to provide answers and alternatives to the shortcomings of object-oriented graphics. The presentations investigated the applicability, merits and problems of various programming paradigms in computer graphics for design, modelling and implementation.

This book contains a revised selection of the best papers from the Fifth Eurographics Workshop on Programming Paradigms in Graphics, held 2–3 September 1995 in Maastricht, The Netherlands. All papers at the workshop were subjected to a thorough review by at least three members of the international programme committee. The selection for this book was based on further review and the papers also incorporate the relevant aspects of the discussions at the workshop.

In past Eurographics workshops on Object-Oriented Graphics the prominent trend has been a discovery of the limits of object-orientation in graphics. The limitations of object-orientation were felt to lie in such areas as the expression of relationships between objects. This is an area of particular strength for the declarative languages, such as constraint-based languages. On the other hand, a notion of state has long been a problem in declarative languages and yet it is often seen as an essential aspect of graphical modelling, particularly in simulation and animation.

The lively discussions at the workshop are difficult to summarize. Reading these papers will give an idea of the issues addressed. The workshop was exciting because it brought together researchers whose common interest was graphics but not a particular programming 'ideology'. Some felt that one should not make a "theology" out of object-oriented programming. The need by constraint solvers for global information and the fact that relations cannot be encapsulated in single objects could simply be addressed by relaxing the rules of information hiding and being a bit less strict about encapsulation. The purist camp felt that if violations of good practice were allowed on an *ad hoc* basis then the benefits conferred by object-oriented practice would simultaneously evaporate. In this connection there was some concern expressed about potential directions of the new distributed virtual reality standard for the world wide web — VRML. There might be a danger of VRML unlearning a number of the lessons learnt in developing graphics standards that have culminated in the formulation of object-oriented standards such as PREMO.

The discussion sessions seemed to be good at issuing challenges. Here are a few:

- Is there one paradigm for graphics or are there several sub-domains each with their own? Tie up subdomains with their corresponding paradigms.

- Conceptually speaking, what are the limitations of object-oriented and functional programming paradigms? Is it as crude as that one cannot deal with relationships while the other leads to rather contorted ways of regarding persistent actors changing over time?

- What, if anything, is special about the requirements that graphics (in the broad sense) imposes on programming methodologies?
- Specifically, please present, at the next workshop,

  a useful functional animation system.

  a well designed VRML.

  with object-orientation as a base, an integration of other paradigms that works "well". *Well* means that the integration confers the sum of the benefits of the individual paradigms and not the lowest common factor. Start by integrating a functional paradigm.

  a working prototype for every claim; one that effectively demonstrates the solution. This is a call for constructed proofs.

One of the conclusions of the workshop was that the object-oriented methodology is still the most effective approach that is widely used to construct real computer graphics applications. Leaving its role as an implementation tool aside, practitioners have realized the shortcomings of the object-oriented methodology as a conceptual problem solving paradigm for the design and development of large computer graphics applications. The availability of powerful parallel systems and high-speed networks, the phenomenal popularity of World Wide Web, the advent of global computing, and the promise of information super-highways all allow graphics applications to extend the limits of how computers can impact our professional, personal, and social lives. In this context, concurrency and distribution have been proposed as one of the themes of the next workshop in this series. Deferred processing (related to laziness) is important in large animated systems and it is still an open question. The need for extensibility of graphical systems, particularly by third parties without access to source code, should also be addressed more adequately than at present.

The papers are grouped into four parts: Object-Oriented, Constraints, Functional, and Multi-Paradigm. This essentially takes the reader on a tour of the issues we have addressed. Firstly there are the impressive benefits and power of object-oriented approaches, both as conceptually tools for design and as programming frameworks. In constraints, though, the limitations are exposed, and addressed. Functional programming provides a clear alternative approach with some impressive recent breakthroughs conceptually (e.g., the use of monads to express state) and in practice. Finally we present some approaches to resolving (or at least enumerating) the multiplicity of approaches encountered.

The ERCIM Graphics Network supported the workshop. The workshop was organized by CWI, Amsterdam. We thank Ms. Marja Hegt for assistance and organizing local arrangements.

October 1995

Edwin Blake
Remco Veltkamp

Contents

Part I: Object-Oriented

Part II: Constraints

Part III: Functional

Part IV: Multi-Paradigm