BEHAVIORAL SPECIFICATIONS OF BUSINESSES AND SYSTEMS

BEHAVIORAL SPECIFICATIONS OF BUSINESSES AND SYSTEMS

edited by

Haim Kilov
Genesis Development Corporation
U.S.A.

Bernhard Rumpe
Technische Universität München
Germany

Ian Simmonds
IBM T J Watson Research Center
U.S.A.

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

Electronic Services <http://www.wkap.nl>

Haim Kilov - *hkilov@gendev.com*
Bernhard Rumpe- *rumpe@informatik.tu-muenchen.de*
Ian Simmonds - *simmonds@us.ibm.com*

Library of Congress Cataloging-in-Publication Data

Behavioral specifications of businesses and systems / edited by Haim Kilov, Bernhard Rumpe, Ian Simmonds.
p. c.m -- (Kluwer international series in engineering and computer science ; SECS 523)
"The second volume of papers based on a series of workshops held alongside ACM's [Association for Computing Machinery] annual conference on Object Oriented Programming Systems Languages and Applications (OOPSLA) and European Conference on Object-Oriented Programming (ECOOP)"--Pref.
Includes bibliographical references and index.
ISBN 978-1-4613-7383-4 ISBN 978-1-4615-5229-1 (eBook)
DOI 10.1007/978-1-4615-5229-1

1. Object-oriented programming (Computer science) 2. System design. I. Kilov, Haim. II. Rumpe, Bernhard. III. Simmonds, Ian, 1966- . IV. Series.
QA76.64.B42 1999 99-40807
005.1'17--dc21 CIP

Printed on acid-free paper.

CONTENTS

PREFACE

'Thinking again?' the Duchess asked, with another dig of her sharp little chin.

'I've a right to think,' said Alice sharply, for she was beginning to feel a little worried.

'Just about as much right,' said the Duchess, 'as pigs have to fly; and the m—'

But here, to Alice's great surprise, the Duchess's voice died away, even in the middle of her favourite word 'moral,' and the arm that was linked into hers began to tremble. Alice looked up, and there stood the Queen in front of them, with her arms folded, frowning like a thunderstorm.

'A fine day, your Majesty!' the Duchess began in a low, weak voice.

'Now, I give you fair warning,' shouted the Queen, stamping on the ground as she spoke; 'either you or your head must be off, and that in about half no time! Take your choice!'

The Duchess took her choice, and was gone in a moment.

'Let's go on with the game,' the Queen said to Alice.

Lewis Carroll

WHY IS THIS BOOK OF INTEREST TO YOU?

This book is about thinking in specification reading, writing and understanding.

The teaching of specifications (in the same way as the teaching of programming) is the teaching of thinking. The teaching of programming has little to do with the teaching of a particular programming language; and in the same way, the teaching of specifications has nothing to do with the teaching of a particular methodology, notation, or tool. When features of a notation are emphasized, "[t]he student is made to believe that the more [language] ideosyncrasies he understands, the better a programmer he will be." [D76] However, these "powerful features belong more to the problem set than to the solution set", and as a result the student becomes "an expert coder of trivial [usually given] algorithms" [D76]. Thus, "the really tough problems are way beyond his mental horizon". Such tough problems, both in programming and in specifications, are about understanding and proper structuring of (often large amounts of) information. Clear thinking is needed to formulate and solve these problems.

WHAT TO SPECIFY, AND WHY

There is no need for contrived or toy examples to demonstrate that apparently complex business situations can be specified in an abstract, precise, clear and explicit manner. Consider Adam Smith's 1776 description [S1776], provided in *italics*, of the centuries old workings of a market economy:

> *The produce of industry is what it adds to the subject or materials upon which it is employed. In proportion as the value of this produce is great or small, so will likewise be the profits of the employer.*

A business or system specification describes the things and relationships of the domain. The structure of the domain — i.e., the shape of its relationships — is more fundamental than the particular things that participate in these relationships. The semantics of relationships ought to be clear from their specification; and it is much deeper than the cardinalities that we are so used to.

> *But it is only for the sake of profit that any man employs a capital in the support of industry; and he will always, therefore, endeavour to employ it in the support of that industry of which the produce is likely to be of the greatest value, or to exchange for the greatest quantity either of money or of other goods.*

Since information systems involve people, we must understand something of their motives both as individuals and as part of larger social systems. In more modern terms, the Enterprise Viewpoint of the RM-ODP systems specification standard "focuses on the purpose, scope and policies" of the system which may or may not be wholly an IT system [ISO95a].

> *But the annual revenue of every society is always precisely equal to the exchangeable value of the whole annual produce of its industry, or rather is precisely the same thing with that exchangeable value.*

When large things are composed of smaller things, some properties of the larger things are determined by those of the smaller. Such composition relationships are common. Properties of a composite may be determined by those of its components and the way that those components are combined [ISO95a]. Here the property determination is simple — it can be expressed as a simple formula.

> *As every individual, therefore, endeavours as much as he can both to employ his capital in the support of domestic industry, and so to direct that industry that its produce may be of the greatest value; every individual necessarily labours to render the annual revenue of the society as great as he can.*

The purpose and behavior of a component must also be understood and specified and within the context of the compositions of which it is a part. In this particular case, had Smith not included the word 'necessarily' he could have been accused of writing romantic nonsense. However, he chose his words very carefully. The word 'necessarily' is elaborated as 'the invisible hand' of his following sentences:

> *He generally, indeed, neither intends to promote the public interest, nor knows how much he is promoting it. By preferring the support of domestic to that of foreign industry, he intends only his own security; and by directing that industry in such a manner as its produce may be of the greatest value, he intends only his own gain, and he is in this, as in many other cases, led by an invisible hand to promote an end which was no part of his intention.*

The 'invisible hand' is — in the more modern terminology of RM-ODP — the way that the components are combined within the composition. The 'invisible hand' is not algorithmic, is not supposed to be, but nevertheless is the way that determines some properties of the composite. The "glue" between components is in part defined very carefully by contract law and, more generally, by business law, the main concepts and constructs of which have remained unchanged for centuries and perhaps millennia.

> *Nor is it always the worse for the society that it was no part of it. By pursuing his own interest he frequently promotes that of the society more effectually than when he really intends to promote it.*

Adam Smith was able to give a precise, elegant, concise and abstract specification of a market economy in 1776. Such specifications are even more important for computer-based information systems. This book contributes to that goal.

ONLY OO?

The papers presented here are not about traditional OO "Analysis and Design". You will often encounter useful and sometimes elegant concepts, good practices (in programming and in specifications), and solid underlying theory that is of interest and importance to all of us. At times, these ideas and concepts will be represented in (and flavored by) contexts that are there for various technical, political and other reasons; but we hope that you will be able to distinguish semantics issues from various representations.

PRACTICE

This book is not unique: developments of the kind presented here are encountered in practice, described in international standards (such as RM-ODP [ISO95a] and GRM [ISO95b]) and in some cases have even become new buzzwords. Whilst several years ago the concepts of pre- and postconditions and invariants were considered inappropriate for business specifications (we often heard that "nobody could ever understand them"), now they are often used in various specifications, standards, and even "CASE tools". Users, including business SMEs (subject matter experts), do understand and successfully apply such concepts. Perhaps now is the turn of applied concepts of category theory; some papers in this book describe how category theory can be used.

Having said that, we want to emphasize that many concepts shown in this book have been used, and successfully, in real industrial projects. In the same manner as mathematics and physics is the basis of traditional engineering knowledge and activities, (possibly different areas of) mathematics are the basis of software engineering — and this includes specification! — knowledge and activities.

WHERE DO THE PAPERS COME FROM?

This is the second volume of papers based on a series of workshops held alongside ACM's annual conference on Object Oriented Programming Systems Languages and Applications (OOPSLA) and European Conference on Object-Oriented Programming (ECOOP). The first volume — *Object-Oriented Behavioral Specifications*[1], edited by Haim Kilov and William Harvey [KH96] — was also published by Kluwer, in 1996. It was based on the first four workshops, held from 1992-1995. While several of this volume's authors contributed to the first volume, there are also many new names.

We have tried to ensure that the workshops on behavioral semantics represent and bring together a wide variety of voices. They range from the practitioner and business thinker through to the academic and (almost pure) mathematician. Some papers also use (and may even quote) foundational material like Wittgenstein's ideas (e.g., [W33]). We encouraged this kind of diversity, and therefore not all authors (including the editors) agree with everything included in this book.

FROM THE HIGHLY THEORETICAL TO THE EXTREMELY PRAGMATIC

Don't be scared by the mathematics that appears in some papers. In all cases the authors were asked to demonstrate the relevance of their work in the simplest possible terms. As E.W. Dijkstra suggested, mathematics is "the art and science of effective reasoning". So is the art of programming — "the art of organising complexity, of mastering multitude and avoiding its bastard chaos as effectively as possible" [D72], and the art of business specifications.

We all deal with the increased complexity of businesses and systems. This complexity should never be(come) artificial: the basics of many businesses were perfectly specified centuries ago, and these specifications can be succesfully (re)used now. At the same time, it only recently became possible to articulate the common mathematical ideas underlying different branches of mathematics and probably engineering: to quote a recently published introduction to category theory used (among others) by high school classes, "the ideas we'll study have been employed for thousands of years, but they first appeared only as dimly perceived analogies between subjects" [LS97]. These fundamental insights, both in category theory and in run-of-the-mill specification and programming areas, lead to better understanding and thus to delivering products that are what all concerned want them to be (rather than what was rapidly hacked together and often hated by the customer).

By bringing these papers together in one volume we hope to show you that these seemingly different papers address different aspects of a single problem. In our workshops we had a similar mixture, which turned out to be very fruitful for inspring discussions during and after the workshops. The papers are all about better understanding of business enterprises and of the software systems that they rely upon.

While it is always tempting to determine a classification scheme for a collection of papers, once again we admit defeat. We used a lexical rather than semantic classification: the papers are in alphabetical order of their first author's last name.

INVITATION

On behalf of our fellow authors, we invite and encourage you to tell us what you think of this book and its contents.

Firstly, each paper includes the e-mail addresses of all authors. Please send any comments or questions to the relevant authors.

Secondly, we invite you to submit papers to future workshops in the OOPSLA and ECOOP series. As we mentioned above, we encourage a wide variety of contributions, from academics and practitioners. Perhaps you will contribute a paper to the next book in *this* series? You can find details of OOPSLA and its workshops at: http://www.acm.org/sigplan/oopsla or by contacting one of the editors of this book.

ACKNOWLEDGEMENTS

As editors and organizers of the above mentioned workshops, we would like to thank the organizers of the conferences ECOOP'97, OOPSLA'97, ECOOP'98 and OOPSLA'98 for their wonderful assistance in running the workshops, as well as Samer AlHunaty and the authors of the book's articles for their help in putting the work together (despite layout-problems).

We would like to thank our loved ones for their help, support, encouragement and tolerance of hours and days of virtual (e-mail) and physical (workshop) absence, and partly also for their active assitance in putting the workshops and the book together.

We also want to express our sincerest thanks to the Technische Universität München for allowing and encouraging the reuse of parts of the articles and of full articles of the following four workshop proceedings in enhanced form:

- TUM-I9725, ECOOP'97 Workshop on Precise Semantics for Object-Oriented Modeling Techniques,
- TUM-I9737, OOPSLA'97 Workshop on Object-Oriented Behavioral Semantics,
- TUM-I9813, Second ECOOP Workshop on Precise Behavioral Semantics, and

- TUM-I9820, Seventh OOPSLA Workshop on Behavioral Semantics of OO Business and System Specifications.

The work was partially supported by the Bayerische Forschungsstiftung under the FORSOFT research consortium and by the DFG under the Leibnizpreis program.

June, 1999

Haim, Bernhard, Ian

Haim Kilov, Genesis Development Corporation
Bernhard Rumpe, Technische Universität München
Ian Simmonds, IBM T J Watson Research Center

References

[D72] E.W.Dijkstra. Notes on structured programming. In: O.-J.Dahl, E.W.Dijkstra, C.A.R.Hoare. *Structured programming*. Academic Press, 1972, pp. 1-82.

[D76] E.W.Dijkstra. The teaching of programming, i.e. the teaching of thinking. In: *Language hierarchies and interfaces* (ed. by F.L.Bauer and K.Samelson), Lecture Notes in Computer Science, Vol. 46 (1976), pp. 1-10, Springer Verlag.

[ISO95a] ISO/IEC JTC1/SC21, Open Distributed Processing - Reference Model: Part 2: Foundations (IS 10746-2 / ITU-T Recommendation X.902, 1995).

[ISO95b] ISO/IEC JTC1/SC21, Information Technology. Open Systems Interconnection - Management Information Services - Structure of Management Information - Part 7: General Relationship Model, 1995. ISO/IEC 10165-7.2.

[KH96] H.Kilov and W.Harvey (Eds.). *Object-oriented behavioral specifications*. Kluwer Academic Publishers, 1996.

[LS97] F.William Lawvere, Stephen H. Schanuel. *Conceptual mathematics: A first introduction to categories*. Cambridge University Press, 1997.

[S1776] Adam Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations*. 1776.

[W33] Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. 2nd corrected reprint. New York: Harcourt, Brace and Company; London: Kegan Paul, Trench, Trubner & Co. Ltd., 1933.

Endnotes

1. Behavior is defined in RM-ODP [ISO95a] as "[a] collection of actions with a set of constraints on when they may occur." Many papers deal with these constraints expressed (in a precise and abstract manner) in different ways.