Manfred Nagl (Ed.)

Building Tightly Integrated Software Development Environments: The IPSEN Approach

Springer

Volume Editor

Manfred Nagl
RWTH Aachen, Lehrstuhl für Informatik III
D-52056 Aachen, Germany
E-mail: nagl@i3.informatik.rwth-aachen.de

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Building tightly integrated software development environments: the IPSEN approach / Manfred Nagl (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1996
(Lecture notes in computer science ; Vol. 1170)
ISBN 3-540-61985-2
NE: Nagl, Manfred [Hrsg.]; GT

CR Subject Classification (1991): D.2, G.2.2, D.3.1, K.6.3,F.4.2

ISSN 0302-9743
ISBN 3-540-61985-2 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10549145 06/3142 - 5 4 3 2 1 0 Printed on acid-free paper

Preface

This book is a state-of-the-art *report* on the *IPSEN project* which aims to introduce a new quality in integration between tools devoted to the development and maintenance of large software systems. Although there are different subprojects for software development environments and also for environments outside the software context investigated in the group, and different grants have supported our work, we inaccurately speak about *the* IPSEN project in the following book. Indeed, it is more a group of related projects which produced a series of prototypes than one single environment demonstrating all we have achieved. IPSEN is an acronym for ***I***ntegrated Software ***P***roject ***S***upport ***En***vironment. The project is long lasting and has required considerable human effort. The results produced in this project are collected in this book.

We shall learn later in this book that integration has different facets which we shall carefully explain in chapters 1–4. To give an impression at the very beginning: By integration we mean *tight integration* on and of software documents which is achieved by specific tools. Integration is not only on coarse-grained document level but also on the level of syntactical and semantical units within these documents. Therefore, integration especially means *fine-grained* integration for different tools on one document and, even more important, between different documents by integration tools. We shall carefully argue in chapter 1 that it is especially this kind of integration which is needed for maintenance and other key activities of software engineering. In chapter 2 we demonstrate how this kind of integration appears from the user's perspective.

The specification of what is going on inside the IPSEN software development environment(s) is done by viewing any document/configuration the user is dealing with internally as a graph belonging to a graph class or a complex of graphs. The tool activities are mapped onto activities of graph processors. A formal specification is given by graph rewriting systems, in short graph grammars. A language for expressing such specifications and an environment for handling them, called PROGRES, has been developed, and a lot of methodology knowledge of how to write down specifications has been found. From a specification the realization of tools is mechanically derived. This overall approach was called *graph technology* and is the most *specific aspect* of IPSEN. It is this graph technology which distinguishes this project from most other software development environment projects (see chapter 3).

Another highlight of IPSEN is the results on *software engineering* level, discussed in detail in chapter 4, based on the IPSEN software architecture. The understanding of the term "architecture" is different from that of other authors: An architecture is not a rough draft but an exact plan across all different logical layers of the software system. For the IPSEN system we present its framework architecture, discuss its standard components, and describe a machinery for deriving its specific components. Furthermore, tools to support this machinery are also discussed. The software engineering results are given in chapter 4.

The main concern of this book is to show *uniformity* throughout different logical levels. These levels are language definition, behavior of tools, internal graph grammar modeling, architectural design, and implementation. The uniformity investigation and the traceability it implies through all these levels may be the key result of the IPSEN approach.

The book and its results are (hopefully) *interesting* to the software development environment community mainly for the following reasons: (a) The project has produced a lot of *practical results* in the form of available prototypes showing this new kind of integration, internal modeling, and tool production. These prototypes have demonstrated that it is worthwhile and possible to transfer the underlying ideas to industrial practice. The project also has (b) a *theoretical side* which demonstrates that theory is necessary for good practice, in this case the practice of tool building. This has two reasons: There is no way to get reliable complex tools without specifying formally the internal behavior of a software development environment. Furthermore, the process of developing software tools and environments can only be made more efficient if handcoding is replaced by more intelligent production mechanisms. This, however, is not possible without a theoretical background and basis of corresponding specifications.

The IPSEN project is a university or *research project*. Therefore, the aim was not to produce industrial tools, although prototypes are efficient and stable. The aim, furthermore, was not to gain completeness, i.e. to support all activities of development and maintenance of software, although a remarkable set of tools has been realized. Being researchers, our interest centered more on learning about how new tools have to behave, to be modeled internally, to be realized, and how realization effort can be reduced, than implementing one tool after the other. Therefore, a new tool was only built if we expected to get new insight in one or more of the above questions. This, however, does not mean that the tools we have built are only valuable for research. In the numerous demonstrations we have given, application programmers were especially enthusiastic in their responses, telling us that they would like to have IPSEN-like tools for their practical work. Nevertheless, the process of transforming results to industrial practice is still in its infancy. However, it has started.

Although the IPSEN project mainly deals with software development environments, the *approach* taken can also be *applied* to any *interactive system* on *complex*, structured, and tightly *connected objects* developed in a group, as is the case with software documents and software projects. We have some results proving this claim.

The IPSEN *approach* or project has *different facets* which are shown in the respective chapters of this book. We summarize them in order so as to clarify the range of discussion. IPSEN is:

(a) a certain idea of how tools of an integrated (software) development environment should behave,

(b) an approach using one key idea, namely that of complex interrelated objects (graphs) at all levels of (software) development environments: language definition, tool behavior, conceptual internal modeling, and realization,

(c) the key modeling idea of separation (separate structures) and integration (fine-grained links between structures) being applicable to different technical grains as documents, subconfigurations, configurations, but also for management information and the interrelation of technical and management information,

(d) a uniform conceptual modeling approach, which makes the internal specification clearer, easier to build up, and maintain and which is the prerequisite for any "intelligence" in the tool production process,

(e) a specification language for the internal specification of tools together with corresponding tools (PROGRES and its environment),

(f) a set of environments built according to these ideas which have been demonstrated at many sites,

(g) an approach to configure integrated environments which all share the same architectural framework and corresponding basic blocks,

(h) an approach to generate specific components within this framework and to realize the corresponding generator tools, i.e. IPSEN is a meta-SDE environment,

(i) an endeavor to build a universal administration component which can be taken to cover the coordination aspects of any (software) project,

(j) a platform project for integrated and distributed environments for data, control, and representation integration, especially an underlying object storage,

(k) an approach which can be transferred to other disciplines outside the software development environment world (e.g. CIM, classification and retrieval systems, intelligent text systems, etc.),

(l) an effort to make languages or methods used in software engineering clear and precise, to learn how to use them, and to build tools for them.

The detailed description of these facets (besides the language and method work) for the scientific world and, especially, for the software development environments community is the main goal of this book.

This book is an effort for the scientific community. It presents material in order to *learn,* to a certain extent, *about software development environments* in general *and IPSEN* in particular. This book should help to spread the essential ideas of IPSEN within the (software) development environments community but also in areas outside of environments.

Besides these outside goals the book also has some *internal goals* for the IPSEN group itself. Up to now we have had no compendium for new group members at the Ph.D. candidate or Master's level to enable them to familiarize themselves with the established results. Furthermore, it was discovered that the preparation of this book, with the knowledge gained during that process, was very stimulating.

After having introduced the aims of this book let me list a number of items the *book* does *not handle*:

(1) It is *not* a textbook on software development environments *in general.* It is a monograph devoted to the IPSEN approach and it discusses (some of the numerous) other approaches only to the extent necessary for characterizing IPSEN. There are also monographs on special topics of the project (/8. Na 79, 4. Schä 86, 5. Eng 86, 4. Lew 88c, 2. ES 89, 5. Schü 91a, 4. Wes 91a, 7. Jan 92, 7. Bör 94, 4. Lef 95, 7. Bee 95, 5. Zün 96, 4. Koh 96/).

(2) This is *not* a *method book.* Some of the existing methods (better termed notations) for the different working areas are introduced because they are the basis for tools. However, they are only introduced by examples and not given formally, and only to the extent needed for the corresponding tool discussions.

All *persons* on the list on the next page have *contributed* considerably to the success of the IPSEN project. They have contributed by finished or ongoing Ph.D. thesis work, by Master's theses, and as programmers. The *commitment* of all these members as developers of ideas, concepts, languages, mechanisms, components, and prototypes is gratefully *acknowledged.* Especially, the contributions of P. Klein, Dr. A. Schürr, and Dr. B. Westfechtel made it possible to edit this book. Furthermore, the careful text preparation work done by A. Fleck, M. Schürr, and R.P. Rössel is responsible for the layout of this volume. Finally, A. Johnston was responsible for improving our English.

The following *grants* given by national and international research organizations have directly or indirectly given considerable support to the IPSEN project and, therefore, have layed the financial basis for the results of this book:

- German Research Council (DFG): project 'Software Workbench'; project 'Graph Grammar Specifications'; project 'Software Architectures for Interactive Systems'; project 'SUKITS', subproject 'CIM Manager'; project 'Graduate College Computer Science and Engineering', subproject 'Administration of Development',
- European Community: ESPRIT-project 'COMPLEMENT', subproject 'Requirements Engineering and Programming in the Large', together with CAP debis Aachen,
- German Ministry for Research and Technology: project 'European Software Factory', subproject 'Requirements Engineering Tools', together with CAP debis Aachen; project 'Reengineering of Client/Server Applications', together with AMI and GEZ,
- Stiftung Volkswagenwerk: project 'Intelligent Editors for Technical Documentation',
- Ministry for Science and Research of North-Rhine Westphalia: project 'Tool Generators'; participation in 'Client/Server Applications'; project 'Software Reuse'.

Finally, the basic support given by the state of Lower Saxony, when the group was at the University of Osnabrück, and by the state of North-Rhine Westphalia, the group since 1986 being located at Aachen University of Technology, is gratefully acknowledged.

Aachen, September 1996 *Manfred Nagl*

Persons Engaged in the IPSEN Project

D. Adamcyk
G. v.Amerongen
Dr. V. Bacvanski
R. Baumann
J. Bausch
J. Beckers
Dr. M. von der Beek
C. Beer
N. Beermann
A. Behle
U. Belten
E. Berens
R. Biermanns
Dr. J. Börstler
Dr. Th. Brandes
H. Brandt
M. Breuer
R. Breuer
M. Broekmans
T. Bruckhaus
B. Bücken
St. Coors
K. Cremer
D. Däberitz
A. Deparade
J. Derissen
O. Dickoph
H. Docquier
R. Dömges
H.J. Dorka
M. Eichstädt
U. Eisenmenger
Prof. G. Engels
F. Erdtmann
A. Feye
R. Gombert
Th. Gröger
M. Guhl
H. Haverkamp
St. Hardt
D. Haßl
H. Hassoun
H. Heil
P. Heimann
R. Herbrecht
M. Heyde
F. Höfer
F. Høgberg
C. Horbach
P. Hormanns
Dr. Th. Janning
A. Joereßen
G. Joeris
N. Kiesel
P. Klein
J. Kloth
R. Koether
Dr. Ch. Kohring
P. Kossing
W. Kothes
C.–A. Krapp
H. Kreten
S. Krüppel
J. Lacour
Dr. M. Lefering
Prof. C. Lewerentz
T. Liese
B. Lücken
S. Mau
R. Melchiesedech
G. Metzen
M. Meuser
P. Möckel
E. Nadarzinski
Prof. M. Nagl
E. Nourbakhsh
M. Pandey
F. Phillip
W. Pickartz
A. Poensgen
B. Pohlmann
M. Rademacher
A. Radermacher
C. Rövenich
M. Rövenich
A. Rossow
A. Sandbrink
Prof. W. Schäfer
H. Scharrel
U. Schleef
H. Schlüper
V. Schmidt
J. Schmitz–Lenders
O. Scholz
Dr. A. Schürr
J. Schwartz
A. Speulmanns
R. Spielmann
T. Spellerberg
G. Sobbe
G. Starke
O. Steffens
J. Theis
M. Thiele
P. Tillmann
B. Tophoven
J. Turek
S. Vaillant
A. Welbers
A. Winter
Dr. B. Westfechtel
A. Winter
St. Witt
H. Zinnen
S. Zohren
Dr. A. Zündorf

Adresses of Contributors

Dipl.-Inform. R. Bauman, Lehrstuhl für Informatik III,
E-Mail: roland@i3.informatik.rwth–aachen.de

Dipl.-Inform. A. Behle, Lehrstuhl für Informatik III,
E-Mail: behle@i3.informatik.rwth–aachen.de

Dipl.-Inform. K. Cremer, Lehrstuhl für Informatik III,
E-Mail: katja@i3.informatik.rwth–aachen.de

Dipl.-Inform. A. Deparade, Lehrstuhl für Informatik III,
E-Mail: deparade@i3.informatik.rwth–aachen.de

Prof. Dr. G. Engels, Dept. of Computer Science, Leiden University, P.O.Box 9512,
NL–2300 RA Leiden, Netherlands, E-Mail: engels@wi.leidenuniv.nl

Dipl.-Inform. P. Heimann, Lehrstuhl für Informatik III,
E-Mail: peter@i3.informatik.rwth–aachen.de

Dr. Th. Janning, Deutsche Krankenversicherung DKV, Aachener Str. 300, D–50933 Köln

Dipl.-Inform. N. Kiesel, Software–Ley GmbH, Venloer Str. 83–85, D–50259 Pulheim,
E-Mail: nk@col.sw–ley.de

Dipl.-Inform. P. Klein, Lehrstuhl für Informatik III,
E-Mail: pk@i3.informatik.rwth–aachen.de

Dr. Ch. Kohring, Burgstr. 13, D–38272 Burgdorf

Dipl.-Inform. C.A. Krapp, Lehrstuhl für Informatik III,
E-Mail: krapp@i3.informatik.rwth–aachen.de

Dr. M. Lefering, Frankenberger Str. 25, D–52066 Aachen,
E-Mail: lef@gs–ac.aachen.cap–debis.de

Prof. Dr. M. Nagl, Lehrstuhl für Informatik III, Aachen University of Technology,
Ahornstr. 55, D–52074 Aachen, Germany, E-Mail: nagl@i3.informatik.rwth–aachen.de

Dipl.-Inform. A. Radermacher, Lehrstuhl für Informatik III,
E-Mail: ansgar@i3.informatik.rwth–aachen.de

Dipl.-Math. R.P. Rössel, Sudetenstr. 27, D–52477 Alsdorf,
E-Mail: Rudolf.Roessel@aachen.netsurf.de

Dipl.-Inform. C. Rövenich, Aachener+Münchener Versicherung AG,
Aureliusstr. 2, D–52064 Aachen

Prof. Dr. W. Schäfer, Universität–GH Paderborn, FB 17, Warburger Str. 100,
D–33098 Paderborn, E-Mail: wilhelm@uni–paderborn.de

Dr. A. Schürr, Lehrstuhl für Informatik III,
E-Mail: andy@i3.informatik.rwth–aachen.de

Dr. B. Westfechtel, Lehrstuhl für Informatik III,
E-Mail: bernhard@i3.informatik.rwth–aachen.de

Dipl.-Inform. A. Winter, Lehrstuhl für Informatik III,
E-Mail: winter@i3.informatik.rwth–aachen.de

Dr. A. Zündorf, Universität–GH Paderborn, FB 17, Warburger Str. 100,
D–33098 Paderborn, E-Mail: zuendorf@uni–paderborn.de

Contents