Contents

Figures

Tables

Preface

FIELD, the Friendly Integrated Environment for Learning and Development, is the research project that demonstrated that practical integrated graphical programming environments are possible. It provides user-friendly graphical interfaces to the multitude of UNIX programming tools, integrating these separate tools into a unified whole.

This book describes the FIELD environment. It discusses the history and evolution of the environment, concentrating on the development of ideas that both worked and didn’t work. It discusses the inner workings of the environment, showing how each of the programming tools works and how the various tools interact with each other. It discusses the user interfaces provided by the various tools, how they are used, why they were chosen, and their strengths and weaknesses.

FIELD has been a remarkably successful research project. The ideas first exhibited in the environment now form the basis for most of the current generation of programming environments including Hewitt-Packard’s Softbench, DEC’s FUSE, Sun’s SPARCworks, Lucid’s Energize, and SGI’s CodeVision. FIELD pioneered the notion of broadcast messaging as a basis for tool integration. Moreover, many of the other tool concepts we introduced in FIELD have found their way into these environments. Thus, in discussing the FIELD environment, this book actually explains the inner workings of today’s programming environments.

The concepts presented here — the message passing framework, the various graphical user interfaces, and the integration of a wide variety of tools to form a single application — are applicable to domains other than programming environments. Many of the lessons learned from FIELD can be applied to general distributed object systems as well as a variety of new applications that can better be structured as loosely coupled processes rather than a single massive entity. The work on program visualization can be applied to visual database query interfaces and to visually browse the ever-expanding information highway.

The primary audience for this book are those interested in the development of programming tools and environments. The book will also be valuable to serious users of programming environments. The book should also be of interest to anyone undertaking a large software project, both by introducing the software tools needed to work on such a project and by demonstrating the concepts of message-based integration which can be applied to a variety of domains.

This book can be divided into three parts. The first part, Chapters 1 through 3, details the message-based integration mechanism at the core of the environment. The second part, Chapters 4 through 6, describes the underlying services provided by FIELD through wrappers around traditional programming tools. The third part, Chapters 7 through 17, describes the tools and user interfaces FIELD offers to the user.

Chapter 1 provides an overview of integrated programming environments, providing context for the use of message-based integration by briefly reviewing the history of such environments and the alternative technologies that were proposed and considered.

Chapters 2 and 3 detail the message-based integration mechanism, describing the concepts behind it, how it is implemented, and the simple interface it offers the various tools. Chapter 2 provides the basics, while Chapter 3 describes extensions to this mechanism that make it more flexible.

Chapter 4 describes the debugger monitor provided by FIELD. This service consists of a complex wrapper around the system debugger, either *dbx* or *gdb*, that interacts with the rest of the environment through the message server.

Chapter 5 details the cross-reference database service, a new tool that was developed to support the environment. It is used extensively for visualization and as a service available to other tools and, through an appropriate interface, to the user.

Chapter 6 describes the remaining services provided by the environment. Two of these are wrappers around existing programming tools, one for configuration management and one for profiling, while the third is an interface for execution monitoring.

Chapter 7 describes the user interface tools that were developed to support FIELD and other programming projects, emphasizing the use of these tools in the environment.

Chapter 8 details the internals and the user interface of the annotation editor . Source code must be a central focus of any programming environment, since it is the programmer's concrete input and the program representation. The annotation editor provides FIELD with a clean interface for tying the source to the rest of the environment.

Chapters 9 through 17 then describe the user interfaces provided by the other FIELD tools, showing both how the tools are used and the tradeoffs made to use graphics effectively . Especially important here are the techniques used in the various tools for managing complexity , allowing graphical interfaces to be used on relatively large programs.

The final chapter provides an overview of the lessons we learned from the system, what we felt were its successes and failures, and some sense of future research directions for programming environments.

This book was written using FrameMaker . All the FIELD images in the book are screen dumps taken from the current working version of the system. Because we have been developing almost exclusively on color workstations over the last five years, the user interface has become color-oriented and the various images are taken from a color display . Since they are printed in black and white, the actual colors are shown using gray scale or dithered images. Where it is appropriate throughout the text, we describe the use of color in the images and leave it to the reader's imagination to fill in the colors.

The FIELD environment is available in source form without charge via ftp on the internet or on various media at a nominal cost, although its use is restricted to non-commercial purposes. Documentation is available in the form of *man* pages and a new tutorial and reference manual. Persons interested should contact the Brown University Computer Science Department Software Librarian through email at **brusd@cs.brown.edu** or at:

Software Librarian
Department of Computer Science
Box 1910

Brown University
Providence, RI 02912

The current version is known to run on Sun workstations with the current operating system. It has been ported to ULTRIX on DECstations, HPUX, IBM's AIX, and other systems, but we do not test these ports on a regular basis. For more information on the availability and current status of the environment, or questions regarding the book or any of the tools, contact the author at **spr@cs.brown.edu**.