



**ADVANCES
IN
SOFTWARE
SCIENCE
AND
TECHNOLOGY**

Volume 3

**JAPAN
SOCIETY
FOR
SOFTWARE
SCIENCE
AND
TECHNOLOGY**

TP31
J35
V.3

9361763



ADVANCES IN SOFTWARE SCIENCE AND
TECHNOLOGY
VOLUME 3



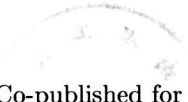
JAPAN SOCIETY FOR SOFTWARE SCIENCE AND
TECHNOLOGY



ACADEMIC PRESS, INC.

Harcourt Brace Jovanovich, Publishers

Boston San Diego New York
London Sydney Tokyo Toronto


Co-published for
Japan Society for Software Science and Technology
by Academic Press, Inc.
and
Iwanami Shoten, Publishers

This book is printed on acid-free paper. ∞

Copyright © 1992 by Academic Press, Inc. and Iwanami Shoten, Publishers

All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher.

ACADEMIC PRESS, INC.
1250 Sixth Avenue, San Diego, CA 92101

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24-28 Oval Road, London NW1 7DX

Library of Congress Catalog Card Number: 90-660056

ISBN 0-12-037103-0
ISSN 1044-7997

AP Exclusive Sales Territory: United States, its territories and dependencies, Canada, and Europe.

Iwanami Shoten Exclusive Sales Territory: Japan.

Nonexclusive sales rights throughout the rest of the world.

Printed in the United States of America

91 92 93

9 8 7 6 5 4 3 2 1

ADVANCES IN SOFTWARE SCIENCE AND
TECHNOLOGY

VOLUME 3

JAPAN SOCIETY FOR SOFTWARE SCIENCE AND TECHNOLOGY
World Trade Center Building 7F, 2-4-1 Hamamatsu-cho
Minato-ku, Tokyo, 105 Japan

Executive Editors for This Volume

Hiroyasu Kakuda, University of Electro-Communications, Chief Executive Editor
Yasushi Kuno, University of Tsukuba
Yoshio Ohno, Keio University
Hideki Sunahara, University of Electro-Communications
Jiro Tanaka, Fujitsu Laboratories Ltd.
Yoshikazu Yamamoto, Keio University
Nagatsugu Yamanouchi, IBM Research, Tokyo Research Laboratory

Editorial Board

Ikuo Nakata, University of Tsukuba, Editor-in-Chief
Hitoshi Aida, The University of Tokyo
Tsuneo Ajisaka, Kyoto University
Takeshi Chusho, Hitachi Ltd.
Norihisa Doi, Keio University
Ken-ichi Hagihara, Osaka University
Masami Hagiya, Kyoto University
Koiti Hasida, ICOT
Teruo Hikita, Meiji University
Yasuyoshi Inagaki, Nagoya University
Hiroyasu Kakuda, University of Electro-Communications
Yahiko Kambayashi, Kyoto University
Tsutomu Kamimura, IBM Research, Tokyo Research Laboratory
Hiroshi Kimijima, Fujitsu Ltd.
Yasushi Kuno, University of Tsukuba
Toshio Miyachi, NEC Corporation
Fumio Mizoguchi, Science University of Tokyo
Yoichi Muraoka, Waseda University
Yoshio Ohno, Keio University
Yasuki Saito, NTT
Masataka Sassa, University of Tsukuba
Masahiko Sato, Tohoku University
Masaaki Shimasaki, Kyushu University
Hideki Sunahara, University of Electro-Communications
Akihiko Takano, Hitachi Ltd.
Akikazu Takeuchi, Mitsubishi Electric Corporation
Hidehiko Tanaka, The University of Tokyo
Jiro Tanaka, Fujitsu Laboratories Ltd.
Hiroyuki Tarumi, NEC Corporation
Satoru Tomura, Electrotechnical Laboratory
Kazunori Ueda, ICOT
Yoshikazu Yamamoto, Keio University
Nagatsugu Yamanouchi, IBM Research, Tokyo Research Laboratory
Michiaki Yasumura, Keio University
Hiroto Yasuura, Kyoto University
Yasuhiko Yokote, Sony Computer Science Laboratory Inc.
Naoki Yonezaki, Tokyo Institute of Technology
Taiichi Yuasa, Toyohashi University of Technology

Contributors

Numbers in parentheses refer to the pages on which the authors' contribution begin.

Kokichi Futatsugi (127), Electrotechnical Laboratory, 1-1-4 Umezono, Tsukuba, Ibaraki, 305 Japan

Shigeki Goto (71), NTT Basic Research Laboratories, 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180 Japan

Takashi Hattori (53), Research Institute for Mathematical Sciences, Kyoto University, Kita-Shirakawa Oiwake-cho, Sakyo-ku, Kyoto, 606 Japan

Junji Hinami (1), Department of Computer Science and Communication Engineering, Faculty of Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812 Japan

Hiroshi Horiguchi (151), Department of Mathematics, Tokyo Denki University, 2-2 Nishiki-cho Kanda, Chiyoda-ku, Tokyo, 101 Japan

Yutaka Ishikawa (127), Electrotechnical Laboratory, 1-1-4 Umezono, Tsukuba, Ibaraki, 305 Japan

Takuya Katayama(13), Department of Computer Science, Tokyo Institute of Technology, 2-12-1 Oookayama, Meguro-ku, Tokyo, 152 Japan

Yasushi Kuno (109), Graduate School of Systems Management, Univ. of Tsukuba, Tokyo, 3-29-1, Otsuka, Bunkyo-ku, Tokyo, 112 Japan

Hiroshi Maruyama (157), IBM Research, Tokyo Research Laboratory, IBM Japan, 5-11 Sambancho, Chiyoda-ku, Tokyo, 102 Japan

Ken-ichiro Murakami (37), NTT Software Laboratories, 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180 Japan

Reiji Nakajima (53), Research Institute for Mathematical Sciences, Kyoto University, Kita-Shirakawa Oiwake-cho, Sakyo-ku, Kyoto, 606 Japan

Ikuo Nakata (93), Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba, Ibaraki, 305 Japan

Naoyuki Niide (53), Research Institute for Mathematical Sciences, Kyoto University, Kita-Shirakawa Oiwake-cho, Sakyo-ku, Kyoto, 606 Japan

Takashi Sakuragawa (53), Research Institute for Mathematical Sciences, Kyoto University, Kita-Shirakawa Oiwake-cho, Sakyo-ku, Kyoto, 606 Japan

Masataka Sassa (93), Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba, Ibaraki, 305 Japan

Toshihisa Takagi (1), Educational Center for Information Processing, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812 Japan

Shin Takakura(13), Advanced Software Development, IBM Japan Ltd., 1623-14 Shimo Tsuruma, Yamato, 242 Japan

Kazuki Takenaka (53), Control & OR Department, Sumitomo Metal Industries, 1-3 Nishi-Nagasu-Hondori, Amagasaki, 660 Japan

Naoyoshi Tamura(13), Department of Electrical and Computer Engineering, Yokohama National University, 156 Tokiwadai, Hodogaya-ku, Yokohama, 240 Japan

Tsutomu Tayama (151), Department of Mathematics, Tokyo Denki University, 2-2 Nishiki-cho Kanda, Chiyoda-ku, Tokyo, 101 Japan

Satoru Tomura (127), Electrotechnical Laboratory, 1-1-4 Umezono, Tsukuba, Ibaraki, 305 Japan

Kazuo Ushijima (1), Department of Computer Science and Communication Engineering, Faculty of Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812 Japan

Jeehee Yoon (1), Department of Computer Science and Communication Engineering, Faculty of Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812 Japan

Editor's Message

Ikuro Nakata
Editor-in-Chief

This is the third issue of *Advances in Software Science and Technology*, an annual journal published by the Japan Society for Software Science and Technology (JSSST). The Society was founded in 1983 as a professional society dedicated to the advancement of the science and technology of computer software.

Unparalleled progress in hardware technology has been a driving force in modern computer technology. It has dramatically improved the performance and reliability, increased the level of complexity and sophistication, and created numerous new applications for computer systems. Progress in software technology, on the other hand, has been much more conservative. By and large, the volume and the quality of current software production depend on the skills and dedicated craftsmanship of programmers. With ever-increasing demand for software production, our ability to build and use computer systems is now limited mainly by our ability to produce software.

Advancing software technology requires active research efforts toward scientific understanding of software systems, organized efforts to improve the current practice of software production, and drastic improvement of software education and training programs. The JSSST was founded to provide leadership, to promote and exchange ideas and experience, and to develop and organize concerted efforts in this direction.

The society has published a domestic bimonthly journal, *Computer Software*, since 1984. This contains original technical contributions that are refereed by the normal scientific review process. In addition, it contains survey papers, tutorials, conference reports, and miscellaneous articles. The journal covers a broad range of computer software. Topics featured in recent issues include algorithms, theory of programs, programming languages and methodology, operating systems, computer architecture, software engineering, artificial intelligence, and natural language processing.

Advances in Software Science and Technology is the second journal published by the JSSST. It is an annual publication with the same scope as *Computer Software*, and is intended to give international exposure to the activities of JSSST and to promote exchange of ideas and information among professionals and the public world-wide. Each issue of the journal contains original technical contributions as well as contributions that have appeared in previous issues of *Computer Software* in Japanese. The JSSST forms a special editorial committee for each issue of this journal; members of the committee for the third issue are listed in the front page together with those of *Computer Software*.

Like the previous issue of the journal, this issue describes a variety of activities, primarily in Japan. Software problems, however, are something we must all face

today; and international collaboration and exchange are absolutely necessary. We very much look forward to publishing contributions from a wide variety of authors in future issues.



Contents

Editor's Message <i>Ikuo Nakata</i>	ix
Contents of Previous Volumes	xi

Research Contributions

Prototyping a System of Writing Tools for Japanese Documents <i>Kazuo Ushijima, Junji Hinami, Jeehee Yoon and Toshihisa Takagi</i>	1
Attribute Grammar Evaluation System for Natural-Language Processing <i>Naoyoshi Tamura, Shin Takakura and Takuya Katayama</i>	13
An Object-Oriented Implementation of TCP/IP Network Protocols <i>Ken-ichiro Murakami</i>	37
RACCO: A Modal-Logic Programming Language for Writing Models of Real-Time Process-Control Systems <i>Takashi Hattori, Reiji Nakajima, Takashi Sakuragawa, Naoyuki Niide and Kazuki Takenaka</i>	53
Formalizing Computer Program Traces <i>Shigeki Goto</i>	71
Regular Expressions with Semantic Rules and Their Application to Data Structure Directed Programs <i>Ikuo Nakata and Masataka Sassa</i>	93
Misty: An Object-Oriented Programming Language with Multiple Inheritance and Strict Type Checking <i>Yasushi Kuno</i>	109
The Semantics of an Object-Oriented Concurrent Programming Language: Process Algebraic Approach <i>Yutaka Ishikawa, Satoru Tomura and Kokichi Futatsugi</i>	127
Floating-Point Numbers and Real Numbers II <i>Hiroshi Horiguchi and Tsutomu Tayama</i>	151
Constraint Dependency Grammar and Its Weak Generative Capacity <i>Hiroshi Maruyama</i>	157

Society News

Rules for Submission of English Papers and English Paper Style Guidelines	175
Author's Guide	183
Japan Society for Software Science and Technology Membership Application Form	185

CONTENTS OF PREVIOUS VOLUMES

VOLUME 1

Research Contributions

JPSG—A Phrase Structure Grammar for Japanese <i>Yasunari Harada, Takao Gunji, Hideo Miyoshi, Hidetosi Sirai and Kôiti Hasida</i>	1
Logical Semantics of Multiple World Mechanism for Knowledge Representation in Logic Programming <i>Hideyuki Nakashima</i>	17
Adaptive Optimal Segmentation Schemes for Multiattribute Files <i>Yuzuru Tanaka</i>	29
Methods for Transforming Attribute Grammars into Efficient Action Routines <i>Takehiro Tokuda</i>	55
Program Transformation of CCFG Programs <i>Ikuo Nakata and Yoshiyuki Yamashita</i>	71
Attribute Graph Grammars with Applications to Hichart Program Chart Editors <i>Tetsuro Nishino</i>	89
A Type Inference System for FP Programs <i>Junhui Luo and Takuya Katayama</i>	105
Overview of GMW+Wnn System <i>Masami Hagiya, Takashi Hattori, Akitoshi Morishima, Reiji Nakajima, Naoyuki Niide, Takashi Sakuragawa, Takashi Suzuki, Hideki Tsuiki and Taiichi Yuasa</i>	133
Floating-Point Numbers and Real Numbers <i>Hiroshi Horiguchi</i>	157
Tutorial	
Category Theory for Algebraic Specification <i>Susumu Hayashi</i>	169
On Concurrent Programming in Smalltalk-80 <i>Norihisa Doi and Kiyoshi Segawa</i>	187

VOLUME 2

Research Contributions

Attachment of a First-Order Data Constructor and Its Application <i>Hiroyuki Sato</i>	1
A Smoothing Algorithm of Polygonal Curves and Polyhedral Surfaces <i>Yoshio Ohno</i>	13
A Technique for Prolog Program Synthesis from Natural Language Specification <i>Kunikazu Fujii, Kuniaki Uehara and Jun'ichi Toyoda</i>	49
Processing Left Extraposition in a Bottom-Up Parsing System <i>Satoshi Kinoshita and Hozumi Tanaka</i>	65
On Learning Smullyan's Elementary Formal Systems: Towards an Efficient Learning Method for Context-Sensitive Languages <i>Yasubumi Sakakibara</i>	83
A Data Model in the Software Project Database KyotoDB <i>Yoshihiro Matsumoto and Tsuneo Ajisaka</i>	107
Hamada Floating-Point Numbers and Real Numbers <i>Hiroshi Horiguchi, Tsutomu Tayama and Kazuaki Kajitori</i>	129
An Efficient Multiple String Replacing Algorithm Using Patterns with Pictures <i>Masayuki Takeda</i>	137
Kamui88: A Parallel Computation Model with Fields and Events <i>Shin-ya Watanabe, Yasunori Harada, Kazufumi Mitani and Eiichi Miyamoto</i>	159

Tutorial

Formation and Development of the Concept of Algorithm <i>Ken Hirose</i>	183
--	-----

Software Critique

WEB System and Its Processor <i>Tsunetoshi Hayashi</i>	203
---	-----

Prototyping a System of Writing Tools for Japanese Documents

Kazuo Ushijima Junji Hinami Jeehee Yoon Toshihisa Takagi

Summary. We have developed a system of writing tools called SUIKOU that analyzes a machine-readable Japanese document textually and provides the writers with useful information for polishing it. The development methods for writing tools for English could not be applied to the development of tools for a Japanese document because of their features. A prototyping method was adopted to develop SUIKOU. SUIKOU was developed through the following four phases: (1) groping, (2) addition and determination of functions, (3) improvement of efficiency, and (4) modification of the user interface. This paper evaluates the prototyping method by describing the development process, functions, and user interface of SUIKOU.

1 Introduction

The rapid progress in computer technologies has produced inexpensive word processors for Japanese. Many Japanese documents are now being written with them. However, they are used only for inputting, editing, storing, and outputting of documents; they do not support polishing these documents.

We have developed a system of writing tools called SUIKOU that analyzes a machine-readable Japanese document textually and provides the writer with useful information for polishing it. SUIKOU is a Japanese word that means the polishing of writing.

Writing tools for English documents such as IBM's CRITIQUE [4, 5] and UNIXTM Writer's Workbench [2, 3] are well known. The design goals of these tools might be expected to be useful for designing and implementing Japanese writing tools. But the functional specifications of those tools could not easily be applied to SUIKOU because Japanese documents have some features that differ from English documents. For example, Japanese has a large character set and in Japanese sentences there are no spaces between words. We were not able to give the precise specification of SUIKOU beforehand. Since it was difficult to develop it using conventional methods based on the life-cycle model of software development, we applied the prototyping method [6] to develop SUIKOU.

SUIKOU was eventually developed through the following four phases:

(1) groping, (2) addition and determination of functions, (3) improvement of efficiency, and (4) modification of the user interface.

In this paper, we evaluate the prototyping method by describing the development process, functions, and user interface of SUIKOU. Section 2 introduces briefly

1 はじめに

日本語文章推敲支援ツール〔推敲〕〔12, 13〕は機械可読な形で存在する日本語文章を解析してその推敲に役立つ情報を提供することを目的としたシステムである。その開発に際して、仕様をあらかじめ明確に与えることができないため、最近注目されているプロトタイピングの手法を採用した〔11〕。

プロトタイピングは使い捨て型と改良発展型の2種類に大別できる〔8〕。どちらかといえば、前者は従来の要求仕様定義の代わりに用いられることが多く、後者は研究的色彩が強いソフトウェアの開発に利用されることが多い。〔推敲〕におけるプロトタイピングは仕様の確定という観点からは前者とみなせるが結果的には後者に属するものであり、徐々に改良して実用システムに近づけるという方式によりシステムを開発した。このような手法を使ってソフトウェアの設計や開発を行った事例は既にいくつか報告されているけれども〔8〕、今回、我々が行った〔推敲〕のプロトタイピングでは従来あまり指摘されなかった過程を経験した。そこで、本論文ではプロトタイピングの観点から〔推敲〕の開発経過を順を追って述べることにより、プロトタイピング手法の意義とその問題点を明らかにする。

Fig. 1 An example of a Japanese document.

the textual features of Japanese documents. Section 3 describes the policy adopted at the start of the development. Sections 4 and 5 give some functions of the first and second versions of SUIKOU by using examples. Sections 6 and 7 describe the improvement of efficiency and the modification of the user interface brought about by the improvement. Section 8 presents the environment of the implementation. Section 9 discusses the effectiveness of our policy and method.

2 Features of Japanese Documents

There are many differences between Japanese and European languages such as English. In this section, we point out principal features of Japanese documents from the viewpoint of designing and implementing SUIKOU.

(1) Japanese has a large character set:

More than 2,000 characters are needed just for daily use. Therefore, at least two bytes are needed to encode a Japanese character. Although for a Japanese document file, coding methods and file formats differ from one computer system to another, a Japanese document is generally composed of a free mixture of passages in two kinds of character sets: One is a set of traditional alphabetic characters coded in one byte, and the other consists of *Kanji*, *Hiragana*, *Katakana*, and other Japanese characters that are coded in two bytes. To solve the coexistence problem of these two character sets, several methods are proposed: some of them use shift codes, and the others use special coding methods to distinguish them.

(2) A Japanese sentence is composed of a nonsegmented sequence of characters: That is, Japanese sentences have no spaces between words as English does. Therefore, it is much harder to recognize words and analyze a document morphologically in Japanese than in English. Figure 1 shows an example of a Japanese document. KUTEN, a small circle, is ordinarily used as the end mark of a sentence. In some documents “.” is used instead of KUTEN.

3 Development Policy

We adopted the following policy while developing SUIKOU.

(1) SUIKOU processes an ordinary complete scientific or technical paper.

We aimed to develop writing tools that can process an ordinary complete technical paper within a reasonable time. The size of a Japanese complete paper including figures and tables, which is printed in seven to eight pages of a technical journal, is usually about 10,000 Japanese characters (that is, about 20,000 bytes). Our tools must handle a paper of this size within a short response time of, say, a few seconds.

(2) SUIKOU analyzes a document only textually.

It is generally regarded as natural that writing tools analyze sentences both grammatically and semantically and then provide the writer with useful information for rethinking them. This means that writing tools are required to have natural language processing facilities. Actually, IBM's CRITIQUE [5] uses them to analyze an English document. But natural language processing is a time-consuming task, and the result of analysis may have ambiguities. To introduce natural language processing techniques would probably violate the development policy (1). Thus, we only adopted a textual analysis method that extracts information from a Japanese document with no machine dictionaries or grammatical analyses.

(3) It is not a computer but, rather, the writer who polishes sentences.

This policy means that SUIKOU itself does not improve a Japanese document. It only extracts information from a document so that the information can help the writer find the sentences and words whose usage is improper. The information extracted by SUIKOU may contain unrequested information. But SUIKOU must not miss requested information, because a user tends to examine only the information provided by SUIKOU. For example, when finding sentences in the passive voice, SUIKOU may extract nonpassive voice sentences, but must not miss any passive voice sentences. The user can decide whether or not the information is useful.

4 First Version of SUIKOU

To begin with, we tried to make some prototypes of writing tools intended to be useful. At first we developed these prototype on a mainframe FACOM M382, whose architecture is compatible with that of the IBM370 and whose performance is almost equivalent to that of the IBM 3090-200. These prototypes were written in SNOBOL4 with Japanese text processing facilities, since SNOBOL4 is suitable for developing prototypes of programs for text processing [8].

The first version of SUIKOU consists of the following four tools:

(1) SENTENCE

This tool analyzes a Japanese document, separates it into sentences by textually searching occurrences of KUTEN, “.”, or carriage return code, and displays the head part, end part, and length of each sentence in a line in the order of occurrence. The end part of a Japanese sentence is important, because most Japanese sentences end with a predicate. This tool is useful for discovering the monotonicity of the end parts of sentences and indicating sentences that are too long.

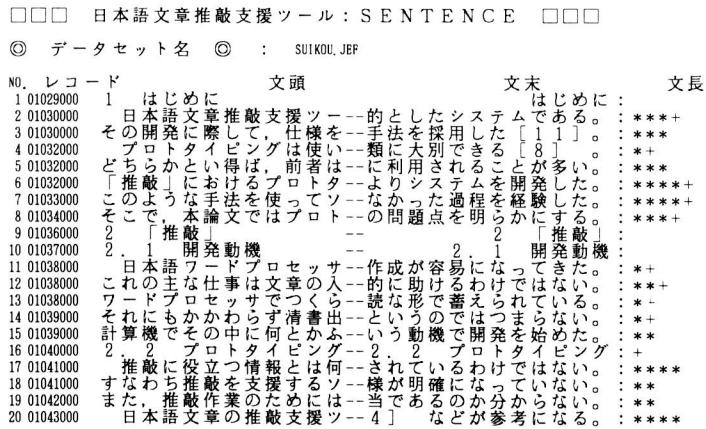


Fig. 2 An example of an output of SENTENCE.

Figure 2 is an example of an output of SENTENCE. In this figure, “*” and “+” mean 20 and 10 characters, respectively. For example, “****+” indicates that the length of the sentence is 70.

(2) KWIC

This tool makes KWIC (Key Word In Context) lists. As mentioned in Section 2, written Japanese consists of three kinds of characters: *Kanji*, *Katakana*, and *Hiragana*. This tool recognizes a keyword as a string of the same kind characters. A KWIC list is useful for detecting misspelled words. Figure 3 shows an example of an output of KWIC for *Katakana* strings. This tool supports checking the uniformity of the expressions in *Katakana* character standing for words of foreign origin. *Katakana* strings are mainly used for expressing words of foreign origin. Since it is impossible to write pronunciation of foreign words exactly in *Katakana*, we use several *Katakana* strings to express one foreign word. It is not desirable to use multiple expressions for one foreign word in the same document.

(3) XREF

This tool cross-references keywords and the record numbers where they appear.

(4) PAREN

This tool points out the correspondence of parentheses and multi-nested parentheses.

5 Addition and Determination of Functions

The first version of SUIKOU revealed that some prototypes of writing tools provide a writer with useful information for improving a document by analyzing it only textually, although these prototypes have some drawbacks. For example, the tools KWIC and XREF produce a lot of output, making it hard for users to find the