

# 计算机软件开发方法、工具和环境

主编

唐稚松

郝克刚

李友仁

西北大学出版社

---

# 计 算 机 软 件 开 发 方 法、工 具 和 环 境

唐稚松 郝克刚 李友仁 主编

西 北 大 学 出 版 社

## **软件开发方法、工具和环境**

唐稚松 郝克刚 李友仁 主编

西北大学出版社出版

(西安市小南门外)

陕西省新华书店发行 西安市莲湖区友谊印刷厂印刷

开本 850×1168毫米 1/32 印张 13.5 字数 330千

1985年8月第一版 1985年8月第一次印刷

印数 1—3,500

统一书号: 13320·4 定价: 2.80元

## PREFACE

It is indeed a pleasure to write this preface for the proceedings of the workshop on Software Development Methods, Tools and Environments. My pleasure is derived from two sources. First of all, I have seen some excellent work contributed by my friends, many of whom have visited Maryland. And more importantly, I believe a budding base for advanced software technology has been formed in China.

Papers presented in this workshop concentrate primarily on tools and languages. Good papers on methods are also being presented. There is, however, no paper dealing with integrated environments, although some basis for such environments exists. For example, various methods and tools surrounding the xyz language could be integrated into a sophisticated development environment. Another example is the WPADT system. Although this system is perhaps less sophisticated than the xyz system, it is much more oriented towards the engineering of software in a productive environment.

The workshop papers can also be classified into evolutionary and revolutionary approaches. As in other workshops, most of the papers in this workshop take an evolutionary approach, whereas the papers on PAGE and PROLOG-based systems represent an initial move towards the 4th generation software technology. The set of papers based on xyz language is encouraging although I do not yet see a concerted effort to develop a truly

integrated software development environment with this base. However, I am impressed by the fact that the xyz language is a relatively independent piece of work that has the potential to absorb most of the modern software technology concepts to form the basis for the development of a sophisticated programming environment. I am happy to learn that two excellent papers on LR (K) grammar and structured editor are the work of two young students.

While there are many high quality papers, I have found a general lack of future direction. This is not surprising since software engineering is a relatively new field for researchers in China. Let me take this opportunity to express my belief that a key factor in future direction is "integration." One type is the integration of various tools in an environment. This integration can be achieved via a common information base where various knowledge accumulated through development, testing, and operational phases is stored and can be linked. A key objective of this kind of integration is the ability to answer the "what if" questions. For example, how many modules or expected lines of code need to be changed if a certain change in interface design is made? Or what would be the man-loading curve if incremental development is used instead of parallel development of the entire system? Less demanding integration consists of traceability among design/code levels, automatic regression test upon completion of a change, or the automatic generation of various reports from the information base. e. g., the generation of PSA type of reports from data-flow diagram information or the scheduling and budget

reports from implementation plan information, or the use of different representations utilizing the same information base. In order to achieve these objectives another level of integration is necessary, the integration of certain supporting technologies. In particular, the compiler technology developed in the 60's, the database technology developed in the 70's, and the artificial intelligence (A.I.) technology that is maturing during the 80's. Compiler technology is an important base for many tools such as syntax-directed editors, static analyzers, and program generators. Database technology is the key for tool integration. The software information base structure is basically a database design problem. Finally, the A. I. technology provides sophistication to the environment. The degree of "intelligence" in a programming environment to a large extent depends on the amount of knowledge and how it is structured in the software information base. For example, the synthesis of certain application domain knowledge could provide the system with some problem solving capability to aid the users. The accumulation of design/code knowledge would provide rapid prototyping capability. In general, in order to aid the user to answer "what if" questions, the system must be equipped with knowledge of various aspects from application domain to the system itself. Neither kind of integration mentioned above was represented by papers in this workshop. It is my hope that these problems will be tackled in future workshops.

At this point, I would like to issue a challenge to workshop participants. That is, to aim for China's supremacy in software technology in 10 years. Such a cha-

llenge might seem outlandish considering the current state-of-the-art in China. However, I do believe that China has sufficient talent for developing a group of world class software engineers and scientists in 10 years. But, you must aim at this objective now and you must create a suitable environment for such talent. If there were a Software Olympics, I would look forward to pleasant surprises similar to those made by the Chinese athletes in the 1984 Summer Olympics!

Professor Tang is to be congratulated for collecting such an excellent set of papers. I am honored to be given the opportunity to write this preface.

Raymond T. Yeh  
University of Maryland  
U. S. A.

## 前 言

我非常高兴能为软件开发方法、工具和环境讨论会文集写这篇序言。这是由于以下两个原因：首先我看到了我的朋友们所做的许多卓越的工作，其中有很多人曾经访问过马里兰。而更重要的是，我相信中国已经建立了先进的软件技术的发展基础。

这次讨论会所提出的论文主要集中在工具和语言方面。也提出了有关方法方面的好论文。然而却没有有关整体化环境 (integrated environments) 方面的文章，虽然这方面在中国也是有基础的。例如，围绕着XYZ语言的许多方法和工具就可能结合成为一个高级的发展环境。另一个例子是WPADT系统，虽然这个系统或许比XYZ系统简单一些，但它更倾向于生产性环境的软件工程。

这次会议的论文也可以归纳为进化性方法和革命性方法两类。象其它讨论会一样，这次会议的大部分论文采用了进化性的方法，而基于PAGE和PROLOG的文章则代表着开始向第四代软件技术前进。与XYZ有关的一组论文是令人鼓舞的，虽然我还未看到以此为基础开发一个真正的整体化软件开发环境而进行的步调协同的努力。但是给我印象很深的是，XYZ语言是一个相对独立的工作，它有可能吸收大部分现代软件技术的概念以形成超开发高级程序环境的基础。我很高兴地看到关于LR(K)文法和结构编辑程序的两篇卓越的文章是两个年轻学生的工作。

在这次会上尽管有很多高质量的论文，但我感到还缺少面向未来的文章。这并不奇怪，因为软件工程对中国的许多研究人员来说还是一个相对新的领域。我想借此机会表示我的信念，即未来方向的关键是“整体化”

(integration)。整体化的一种类型就是在一个环境中各种不同工具的整体化。这种整体化可以通过公共信息库来加以完成，即把在开发，测试和操作各阶段所积累起来的各种知识都存贮在这个库中，并且把这些知识连结起来。这种整体化的主要目标是使其具有回答“如果怎样”(“what if”)

这类问解的能力。例如，如果在接口设计中做一些改变，会有多少模块或多少行代码需要改变？或者，如果对整个系统采用增量式开发来取代并行式开发，人力分配的曲线图将会是怎样的？要求较低的整体化包括设计/编码级间的跟踪能力，有关改变完成的自动回归测试或从信息库自动生成各种报告。例如，从数据流程图的信息生成PSA类的报告，或从实现计划的信息生成预算和日程表的报告，或使用不同的表示却利用同一信息库等。为了实现这些目标需要另一级的整体化，即某些支持技术的整体化。特别是指六十年代发展起来的编译技术，七十年代发展起来的数据库技术和八十年代正在成熟起来的人工智能（A.I.）技术。编译技术是许多工具的重要基础，比如语法引导的编辑程序、静态分析程序和程序生成器。数据库技术是工具整体化的关键。因为软件信息库的结构基本上是一个数据库的设计问题，而A.I.技术则给环境提供更高级的功能。在程序设计环境中，“智能”的水平在很大程度上依赖于软件信息库中知识的数量和如何构造它。例如，某一应用领域中的知识的综合将会使系统具有某些解决问题的给用户以帮助。设计/编码知识的积累将会提供快速建立软件模型的能力。一般说来，为了有助于用户解决“如果怎样”的问解，系统必须用从应用领域到系统本身各方面的知识来装备自己。在这次讨论会上没有提出有关上述任何一种整体化方法的论文，我希望这些问题将会在今后的讨论会上引起重视。

在这里，我愿意向这次讨论会的参加者提出一个挑战性的期望，这就是十年后在软件技术方面取得中国的领先地位。考虑到中国当前的技术状况，提出这个目标似乎是有点不够现实。然而，我的确相信中国有足够的人才，十年中，一定会造就出一大批具有世界水平的软件工程师和科学家。但目前你们必须面向这个目标，为造就这些人才创造合适的环境。如果有软件奥林匹克运动会的话，我盼望着类似于在1984年夏季奥林匹克运动会上中国运动员取得的那种令人惊喜的成就。

祝贺唐教授编辑了这本卓越的论文集，我为有机会写这个序言而感到十分荣幸。

美国马里兰大学 叶祖尧  
(李友仁译，卞雷、郝克刚校)

# 目 录

前言 (附译文) .....叶祖尧 (1)

形式技术在软件开发中的作用与形式软件开发方法导引

.... Dines Bjorner Ole Oest Soren Prehn (1)

报表处理语言及其实现.....卢慧琼 (61)

建于UNIX上的汉字文书处理系统.....沈海祥 (74)

交互图形编辑器的设计和实现...董士海 郭 宏 刘君强 (84)

PAGE——一个支持程序自动生成的软件环境

.....居德华 周善琼 (93)

程序的多维结构和结构的 Petri 网.....郝克刚 (99)

Ada程序的死锁检测原理.....周伯生(117)

一个新的软件实现工具系统——WPADT .....母国庆

陶培基 李宗荣 梁 英 黄水松 何克清(130)

一个构造增量式 LR(1)句法分析器的新方法 .....叶大兴(146)

数据库辅助设计系统.....计立奇(155)

网络数据库通用查询语言.....赵玉成(165)

一个软件维护工具 .....陈 敏(176)

Pascal到Modula—2的自动转换工具...郑国梁 叶晓风(184)

PROLOG 语言的变换语义学及其在软件开发中的应用

.....孙怀民 王立国(194)

FORTTRAN测试系统 .....周之英 刘爱武 王 静(204)

PROLOG描述的独立于机器的代码生成程序 .....刘椿年(217)

关于TUILI的设计.....陆汝钤(229)

System C与软件开发环境 .....张寿萱(239)

|                                      |             |       |
|--------------------------------------|-------------|-------|
| XYZ系统的逻辑基础.....                      | 唐稚松         | (251) |
| 时序逻辑程序的证明系统.....                     | 冯玉琳 林惠民 唐稚松 | (260) |
| 形成语义与编译形式化.....                      | 周盛宗 龚 淳 唐稚松 | (269) |
| 转换文法与程序转换.....                       | 郑茂松 李 新 唐稚松 | (280) |
| SFDT: 一个结构化流图的设计工具.....              | 印平旦 唐稚松     | (291) |
| 模块图式管理与版本控制系统XYZ/M.....              | 倪惜珍         | (300) |
| 多功能LR语法分析器构造系统XYZ/PG的理论和实现<br>.....  | 董 春         | (309) |
| XYZ/CGG——一个实用的代码生成程序的自动生成系统<br>..... | 汪健白 郑茂松     | (319) |
| 二级文法及其实现.....                        | 李友仁 顾元祥 刘秀芳 | (325) |
| 编译程序的测试方法与实现.....                    | 李友仁 顾元祥 华庆一 | (335) |
| 程序开发与程序错误分类.....                     | 顾元祥 李友仁     | (346) |
| XYZ/G <sub>2</sub> 的类型等价问题.....      | 华庆一 李友仁 顾元祥 | (362) |
| XYZ/G语言的树型中间语言的设计、表示和实现<br>.....     | 顾元祥 李友仁 华庆一 | (373) |
| 机器定理证明与程序综合.....                     | 孙永强 陆汝占 毕 华 | (386) |
| FORTTRAN验证条件产生器 .....                | 孙永强 宋国新     | (394) |
| 开设“软件工具”课的体会.....                    | 董士海         | (404) |
| 编后记.....                             | 唐稚松 郝克刚 李友仁 | (411) |

## CONTENTS

|                                                                                                                                            |                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| Preface.....                                                                                                                               | <i>Raymond T. Yeh</i> (1)                                                                      |
| The Role of Formal Techniques in Software<br>Development and a Management Introduction to<br>a Formal Software Development Method<br>..... | <i>Dines Bjorner, Ole Oest, Soren Prehn</i> (1)                                                |
| A Form Manipulation Language and its<br>Implementation System.....                                                                         | <i>Lu Hui-qiong.</i> (61)                                                                      |
| Chinese Document Processing Based on UNIX<br>.....                                                                                         | <i>Shen Hai-Xiang.</i> (74)                                                                    |
| The Design and Implementation of An Interactive<br>Graphics Editor<br>.....                                                                | <i>Dong Shi-hai, Guo Hong, Liu Jun-qiang</i> (84)                                              |
| PAGE—a Software Environment for Supporting<br>Program Generation<br>.....                                                                  | <i>Ju De-hua, Zhou Shan-qiong</i> (93)                                                         |
| The Multidimensional Construction Specification<br>of Programs and Structured Petri Nets<br>.....                                          | <i>Hao Ke-gang</i> (99,                                                                        |
| Deadlock Detecting Principle for Ada Programs<br>.....                                                                                     | <i>Zhou Bo-sheng</i> (117)                                                                     |
| A New Software Implementation Tool System-WPADT<br>.....                                                                                   | <i>Mu Guo-qing, Tao Pei-ji, Li<br/>Zong-rong, Liang Yin, Huang Shui-song, He Ke-qing</i> (130) |
| A New Method for Constructing Incremental<br>LR (1) Parsers.....                                                                           | <i>Ye Da-xing</i> (146)                                                                        |
| Data Base Aided Design System.....                                                                                                         | <i>Ji Li-qi</i> (155)                                                                          |
| A Generalized Query Language for Network                                                                                                   |                                                                                                |

|                                                      |                                                 |
|------------------------------------------------------|-------------------------------------------------|
| Database System.....                                 | Zhao Yu-cheng(165)                              |
| A Software Maintenance Tool .....                    | Chen Min(176)                                   |
| A Pascal to Modula-2 Translator                      |                                                 |
| .....                                                | Zheng Guo-liang, Ye Xiao-feng(184)              |
| A Transformation Semantics of Prolog                 |                                                 |
| .....                                                | Sun Huai-min, Wang Li-guo(194)                  |
| An Automated Test System for FORTRAN Program         |                                                 |
| .....                                                | Zhou Zhi-ying, Liu Ai-wu, Wang Jing(204)        |
| Machine-Independent Code Generators Specified        |                                                 |
| in Prolog .....                                      | Liu Chun-nian(217)                              |
| The Design on TUILI .....                            | Lu Ru-qian(229)                                 |
| System C and Software Development Environment        |                                                 |
| .....                                                | Zhang Shou-xuan(239)                            |
| The Logic Foundation of XYZ System                   |                                                 |
| .....                                                | Tang Chih-sung(251)                             |
| A proof System for Temporal Logic Programs           |                                                 |
| .....                                                | Feng Yu-lin, Lin Hui-min, Tang Chih-sung(260)   |
| Formal Semantics and Compile Formalizing             |                                                 |
| .....                                                | Zhou Sheng-zong, Gong Chun, Tang Chih-sung(269) |
| Transformation Grammar and program Translation       |                                                 |
| .....                                                | Zheng Mao-song, Li Xin, Tang Chih-sung(280)     |
| SFDE, A Tool for Designing Structured Conventional   |                                                 |
| Flowchart.....                                       | Yin Wei-ping, Tang Chih-sung(291)               |
| The Management of Module Configuration and Version   |                                                 |
| Control System XYZ/M.....                            | Ni Xi-zhen(300)                                 |
| The Theory and Implementation of the Multifunctional |                                                 |
| Parser Generator-XYZ/PG .....                        | Dong Chun(309)                                  |
| A Practical Code Generator XYZ/CGG                   |                                                 |
| .....                                                | Wang Jian-bai, Zheng Mao-song(319)              |
| Two Level Grammar and its Implementation             |                                                 |
| .....                                                | Li You-ren, Gu Yuan-xiang, Liu Xiu-fang(325)    |

|                                                                                               |                                                      |
|-----------------------------------------------------------------------------------------------|------------------------------------------------------|
| Testing Methods and its Implementations for<br>Compiler.....                                  | <i>Li You-ren, Gu Yuan-xiang, Hua Qing-yi</i> (335)  |
| Program Development and Classification of Program<br>Errors.....                              | <i>Gu Yuan-xiang, Li You-ren</i> (346)               |
| The Issue of Type Equivalence for XYZ/G2 language<br>.....                                    | <i>Hua Qing-yi, Li You-ren, Gu Yung-xiang</i> (362)  |
| Design, Representation And Implementation of the<br>Tree Intermediate language for XYZ/G..... | <i>Gu Yuan-xiang, Li You-ren, Hua Qing-yi</i> (373)  |
| Mechanical Theorem Proving and Program Synthesis<br>.....                                     | <i>Sun Yong-qiang, Lu Ru-zhan, Bi Hua</i> (386)      |
| A Verifiction Condition Generator for FORTRAN<br>.....                                        | <i>Sun Yong-qiang, Song Guo-xing</i> (394)           |
| About new offerings——A" Software Tools" Course<br>.....                                       | <i>Dong Shi-hai</i> (404)                            |
| Postscript.....                                                                               | <i>Tang Chin-sung, Hao Ke-gang, Li You-ren</i> (411) |

# 形式技术在软件开发中的作用与 形式软件开发方法导引

Dines Bjorner Ole Oest, Soren Prehn

(丹麦技术大学)

(丹麦数据处理中心)

## 摘 要

本文给出了在软件开发中为什么必须使用形式的、基于数学的技术的理由。扼要地叙述了若干辅助规范和实现技术，并说明了它们的作用。在许多工业化的大型软件产品的开发中使用VDM (Vienna Development Method, 维也纳开发方法) 所积累的非常丰富的经验基础上——我们讨论了形式方法在现今技术和工程上的潜在能力、缺点、应用领域、困难和进一步发展的可能性，以及采用这些形式方法的软件公司可能得到的好处。

## 目 录

### 部分 I：在软件开发中形式技术的作用

- 0. 导引
- 1. 系统规范的要求
- 2. 规范的风格
- 3. 为什么有不同的规范风格？什么情况下选择哪一种风格？

### 部分 II：形式软件开发方法导引

- 4. VDM的软件工程方面
- 5. 开发生存周期的规则
- 6. 应用方法分类
- 7. 可重复使用的开发部分
- 8. VDM概貌
- 9. VDM在程序设计语言中的应用
- 10. 研究的需要
- 11. VDM支持工具
- 12. VDM的管理方面
- 13. 使用VDM的Ada编译程序的形式规范和开发
- 14. VDM的教育与培训
- 15. VDM的社会学
- 16. 丹麦数据中心 (DDC)

### 部分 III：书目与参考文献

一般的与基本的文献目录

## 部分 I 在软件开发中形式技术的作用

### 0 导 引

软件系统的各种不同的数学语义定义愈来愈引起人们广泛的兴趣。不论是在国际标准化工作中,在体系结构设计中,以及作为开发正确性实现的基础,还是在验证软件系统应用正确性方面,这些定义都日益被接受。

本文的目的是要使更多的软件工程部门的研究人员和专业人员了解这些事实,认识到构造和使用这些定义应该做些什么,以及这些定义的目的是什么。

#### 软件开发方法

在本节我们给出一个很特殊的软件开发方法的核心部分。接着概述一般的软件开发方法。最后轮廓地说明这个主要方法的特点。我们知道对任何软件的开发可以由图 1 所示重复进行的九个阶段组成。图中我们把每个阶段的名字放在左边。并对各阶段的内容和结果加以简要的说明。

图 1 中我们所以把 7—9 阶段合并一框,是因为我们主要关心的是 2—4 阶段。我们把设计阶段与编码阶段分开;把系统规范(我们称其为体系结构)阶段与设计阶段分开。

在上述简图中,我们没有列出文档编制以及对操作员和用户的培训方面。我们认为文档编制是每一阶段中的一个完整部分,而对用户进行培训则与上述各阶段无关;如果文档编制很差,那么培训就不可能。只有好的文档化了的和受人欢迎的软件,才能通过培训成功地被传送到实际使用这个软件的人的手中。