

# 软件工程 CASE 工具——RSL/RSA

许龙山 贾霖\*

(中国科学院软件研究所, 北京 100080)

## SOFTWARE ENGINEERING CASE TOOL—RSL/RSA

Xu Longshan and Jia Lin

(Institute of Software, Academia Sinica, Beijing 100080)

**Abstract** The development of software has experienced several periods, and it comes nowadays, to the age of software engineering. Conceptionally, the software engineering consists of a number of system development stages, and the major issue to be solved in the age of software engineering is technique and methodology to be applied to requirement definition and analysis. The key technique in requirement analysis will decide the quality of requirement definition and analysis. Requirement Specification Language/Requirement Specification Analyzer (RSL/RSA) is a computer aided requirement analyzer, which is structured upon specification as its kernel, based on the requirement specification data base. This paper, from the point of view of software engineering, proposes some key techniques in requirement analysis, introduces the concept of integrated software tool package, and advances designing objective for RSL/RSA. It also gives an explanation of functions, structure and applications for RSL/RSA as a computer aided requirement engineering tool, and makes a brief introduction on system development tendency, project planning and management, system design and other further work in the future.

**摘要** 软件的发展经历了若干阶段,到今天已进入了软件工程时代。软件工程将系统开发分为几个阶段,其中需求定义与分析阶段的技术和方法已成为软件工程时代所要解决的主要课题之一。需求分析的关键技术是需求定义与分析的质量问题。“需求规格说明语言/需求规格说明分析系统”(RSL/RSA)是一种以规格说明为核心,以需求规格说明数据库为基础的计算机辅助需求分析系统。本文从软件工程的角度的角度,提出了需求分析的关键技术、集

\* 本文1991年3月5日收到,1991年6月6日定稿。许龙山,1991年逝世,享年49岁,生前的主要研究成果有:主持过石油部“七·五”项目PSL/PSA系统汉化工作,国家“七·五”科技攻关项目RSL/RSA,北京菜蔬公司计算机管理系统等。贾霖,助研,主要研究领域为软件工程,CASE环境工具。

成化的软件工具包的概念和 RSL/RSA 系统的设计目标,同时介绍了计算机辅助需求工程工具 RSL/RSA 系统的功能、结构和作用,并对今后系统的发展方向,项目计划与管理、系统设计等进一步的工作进行了简要说明。

## § 0. 引 言

计算机系统分为软件与硬件两大部分,硬件作为一个产业已发展得具有相当的规模,软件方面却相对落后。近些年来,软件生产工业化的要求越来越迫切,各发达国家纷纷提出了各自的发展战略,我国也开始了这方面的工作。软件要真正实现产业化,软件开发的形式化、规范化和工程化是软件产业的基础和保证。软件的发展经历了以手工作坊式为代表的程序设计时代和以设计组为标志的系统设计时代,现已进入软件工程时代。软件工程从概念上将系统开发分为若干阶段,包括需求分析与定义阶段,逻辑设计阶段,物理设计阶段,系统测试和系统的运行与维护。其中需求分析的方法与技术已成为软件工程时代所要解决的主要问题之一。八十年代以来,计算机辅助软件工程(CASE)发展十分迅速,已成为软件工程研究的一个十分活跃的领域。一些发达国家在这方面起步较早,已有许多 CASE 工具产品问世。我国在这方面的发展相对来说比较落后,但许多软件研究人员和单位在这方面做了大量的工作。特别是在程序设计与系统生成方面取得了可喜的成就,然而在需求分析方面开展研究较少。这方面国外虽有一些成功的系统,如美国 ISDOS 的 PSL/PSA, TRW 公司的 RSL/SERM 等,但都是一些规模庞大,运行环境要求较高的系统。随着计算机行业的飞速发展,一些小型的工作站和高档微机具有了较高的性能/价格比。CASE 工具的宿主机型已瞄准了这些工作站和高档微机。

另一方面,软件工程在概念上将系统开发集成在一起的实用性的 CASE 工具仍然较少。需求规格说明语言/需求规格说明分析系统(Requirement Specification Language/Requirement Specification Analyzer, RSL/RSA)项目的研制开发,在这方面做了一些工作。RSL/RSA 是一个计算机辅助需求定义与分析工具包,目的在于提供系统开发人员一个交互式的从目标系统需求定义到系统逻辑设计的转换界面,以求提高系统开发质量和系统开发效率。RSL/RSA 系统与同类型的其它软件工具相比具有以下特点:1. 采用了核心信息库,提供了多种表达的一致性和完整性好的集成机制。2. 以需求描述的各个方面为基础,不以某一种方法为准,可支持不同的软件开发方法与模型。3. 采用了交互式的用户界面,使系统具有良好的用户接口,提高了整个系统的使用效率和准确性。4. 支持系统的增量式开发。5. 提供了完整的需求规格说明描述语言。6. 丰富的工具支持。7. 多种形式的辅助分析报告。

## § 1. 需求分析的关键技术

需求分析在系统开发中是整个开发工作的第一步,是系统开发人员与用户的接口。需求分析的目的在于完整、准确地了解现行系统的基础上,建立目标系统的逻辑模型。需求分析实际上是整个系统开发的基础,它的好坏直接关系到系统开发的成败。由于需求分析阶段在系统开发中所处的这种特殊地位,使得需求分析的质量和效率成为系统开发人员和用户都十分关心的问题。据统计,软件开发过程中的错误来源于需求分析阶段的占 65% 以上,如图 1. 需求分析阶段造成的一个错误,在测试阶段要改正这个错误,需要花费 100 倍于需求分析

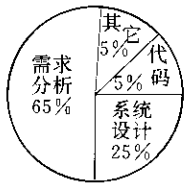


图1 软件错误来源比例

阶段改正这个错误所花的代价。

如果把系统开发过程看作是对问题的求解过程,那么需求分析过程就是对问题的认识过程.如何准确理解和表达目标系统的需求就成了需求分析的关键问题.传统的系统开发方法是使用数据流程图(DFD),数据字典(DD),小说明(MS),判断表和控制流图(CFD)等,这些方法和工具确实为系统开发人员提供了系统需求描述与分析的支持,但这些工具和方法从本质上

讲是对一个完整问题领域不同侧面的刻划.由于每一种方法不能单独把整个系统的所有方面都描述清楚,而各工具间又缺乏统一的集成机制,使得单个方法或工具的最优,而整体的优化程度却相对降低.造成这一问题的主要原因是单个工具、方法的描述方面存在交集,这种重复性在没有统一的集成机制的情况下,就引起了“质量”问题.在低质量的需求分析基础上很难开发出高质量的软件.因此,提高需求分析的质量,关键在于提供一种统一的、完整的表达和检测机制.另一方面,在传统的开发方法中,都是用手工方式对各工具表达上的不一致、不完整问题进行检查,引入计算机辅助软件工程(CASE)后,在以统一的需求规格说明的表达基础上,由计算机代替人来对整个系统描述的一致性和完整性进行检查,并提供一系列的辅助分析报告.

## § 2. 系统开发目标与设计思想

一个有效的需求工程辅助工具包,应能提供集成化的表达、分析与测试,而不是仅仅停留在单个工具的编辑与检查上.需求规格说明语言/需求规格说明分析系统(RSL/RSA)采用了概念集成的思想方法,以形式化的需求规格说明语言为工具,从提高系统需求描述的准确性、一致性和分析结果的可靠性出发,利用形式化技术和数据库技术开发的,系统的目标是:

### (1) 面向目标系统的问题描述语言

系统提供一种描述性的规格说明语言,按现实世界的 ERA 模型,对目标系统的每一个对象进行描述.

### (2) 自动进行一致性和完整性检查

系统可以对用需求规格说明语言描述的目标系统进行质量分析,即对系统描述的一致性和完整性两个方面进行检查.对于系统说明中存在的 inconsistence 和不完整性问题,以辅助分析报告和图形形式给出具体说明.

### (3) 支持多种软件开发方法学

需求分析的方法、工具有很多,如以数据流为中心,以数据结构为中心等开发方法,系统均能支持.

### (4) 支持系统增量式开发

需求分析、定义是一个对问题的认识过程.认识一个事物需要一个过程,同样开发一个系统也是一个逐步求精的过程,所以一个 CASE 工具应能支持这样一个不断完善的过程.

### (5) 自动文档生成系统

文档的编写是系统开发过程中一个十分繁杂的工作,常常是一些海量文字编辑工作.利

用计算机的文字处理能力,开发人员只需提供出一定的文档模式,计算机即可根据核心数据库内容按照文档模式自动生成需求规格说明文档.

(6)开放式系统结构

一个软件工具包的能力无论如何是十分有限的,如果能提供与其它工具联结的接口,将有利于发挥不同工具的作用.

(7)友善的用户界面

“交互式”的界面已成为 CASE 工具中越来越重要的目标.一个友善的人—机交互界面已成为软件系统的一个重要技术指标.交互式的系统宜于用户的使用、学习,减少了使用中带来的错误和一些不必要的重复.

§ 3. 系统概述

RSL/RSA 系统是计算机辅助需求定义,分析和文档生成系统,该系统以规格说明语言为工具,以数据库为核心,无论目标系统的需求描述以何种方式输入,最终都以需求规格说明语言的形式存储在核心数据库中.系统组成主要包括:

1. 描述工具—RSL

利用一种面向问题的需求规格描述语言,对目标系统进行说明定义.

2. 辅助分析系统—RSA

实现目标系统需求描述的分析,信息追踪,共享和转换.并提供多种形式的辅助分析报告.

3. 自动文档生成系统

按照用户定义的文档模式和系统需求规格描述,自动生成需求分析阶段的规格文档.

4. 系统配置管理

提供用户自定义工具,对系统的输入、输出进行重构.



图2 RSL/RSA系统组成

§ 4. 需求规格说明语言(RSL)

RSL 是一种面向问题的描述语言,包括形式化和非形式化描述.用于目标系统的需求规格说明.RSL 描述模式采用 ORA(对象—关系—属性)模式.例如,对信息系统可从三个方面进行描述:系统边界、外部项、目标系统.

RSL 语言可定义十七种对象类型(处理、数据存储、输入、实体、组、元素、关系、条件、事件、单位、处理器、资源、外部项、属性、系统参数、备忘录、输出)和七十多种对象之间的关系(包括对象之间的关系和对对象本身的属性).RSL 可以从九个方面对目标系统进行描述:系统边界与输入/输出流、系统属性、系统结构、系统控制、系统动态、数据结构、数据推导、系统资源描述、项目管理.

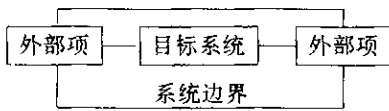


图3 系统描述模型

需求规格描述	需求规格描述文本
定义节	对对象进行定义
语句	对象关系描述
词	关键字和目标名
字符	RSL字符集

表1 RSL语法结构表

对于一个具体系统可以用层次结构和网状结构两方面进行描述。

RSL 的描述结构是以对象定义节为单位,每一节包括关语句,关系描述和属性说明。

### § 5. 需求规格说明分析系统(RSA)

RSA 是 RSL/RSA 系统的主要部分,它完成规格说明语言与内部存储结构的转换、描述信息的一致性和完整性检查与分析。

#### 5.1 信息转换

词法分析:对用户的形式化与非形式化的描述进行词法扫描,并将形式化描述中的保留字转换为内部代码形式。

语义扫描:RSL 是一种上下文相关的文法,RSA 对于形式化描述进行扫描,形成中间代码表达。

代码转换:将经过语义扫描过的语言表达经过压缩处理,按照统一的内部表达机制,生成内部存储代码并存入 RSA 专用数据库。

#### 5.2 信息的可靠性检查和分析

一致性检查:对于用户的 RSL 描述,按对象(对象类型),关系(对象与对象的关系)和对象自身的属性三个方面进行检查。

完整性检查:系统描述的完整性检查包括对象定义的完整性以及系统结构、数据结构、系统控制、系统动态和对象关系表达的完整性分析、检查。

错误处理:对于检查到的一致性和完整性的错误按类型进行不同的说明,提示有可能产生这类错误的原因,同时对错误进行定位。

### § 6. 支撑系统

RSL/RSA 系统的支撑环境是由若干工具组成,这些工具完成用户界面与分析系统的转换,并为用户界面实现支撑手段。

#### 6.1 RSL 编辑/分析

- 语法制导编辑器 提供用户一种解释型的语法制导编辑工具,用户输入一句,系统接受并处理一句,同时按 RSL 语义关系显示下一句输入提示,用户可在此提示引导下逐节、逐句输入。

- 结构化编辑器 提供一种语法结构框架,用户只需在框架中填入需要的描述,系统可自动对框架中所填信息进行分析。

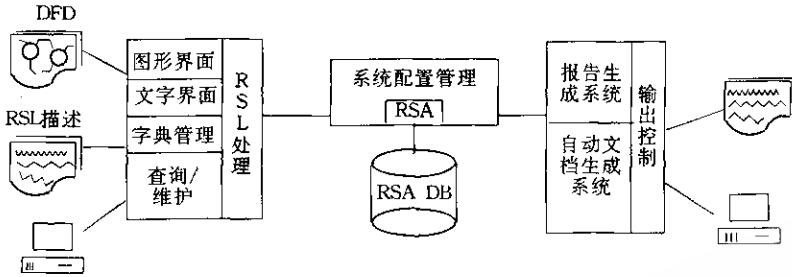


图4 RSL/RSA 系统结构

- **全屏编辑器** 为用户提供一种全屏的正文编辑工具, 具有删、插、改、重写、标记、移动等功能。

- **字典编辑器** 提供用户数据字典编辑录入手段, 该编辑器根据用户在系统配置中设置的字典模式, 按照模式中的条目进行数据录入工作。

### 6.2 查询/维护系统

查询/维护提供用户对系统的信息查询、追踪和维护手段。

- **字典查询/维护** 根据配置管理中提供的数据字典模式, 用户可直接在终端上查询到对象类型的相应数据形式表达。系统将字典分为两类, 数据字典和活动字典。修改后的信息采用增量式的开发方法, 存入数据库。该工具还为用户提供了字典的自动生成。

- **系统查询/维护** RSA 提供一个实现对象系统描述信息的追踪与维护的操作界面。用户可以对任意对象或类型进行查询, 查询到的信息可以进行任意修改、输入和删除。查询到的信息是相应对象的全部描述。

### 6.3 辅助分析报告生成系统

辅助分析报告是RSA系统的主要分析信息输出形式。报告生成系统为用户提供了有关系统描述各个方面的不同信息, 包括表格、矩阵、正文、图形和摘要等类型。这些报告主要以不同形式完整反映一个系统的各种关系, 并分析、检查、判断它们中间存在和可能存在的逻辑错误, 以帮助用户提高系统描述的质量。这些报告的内容、组成形式等可以由用户灵活选择, 这是RSL/RSA具有实际使用价值的主要部分。

### 6.4 规格文档生成系统

规格文档生成系统是RSL/RSA系统的重要部分。规格文档生成系统提供了完整、丰富的需求文档。规格文档是以用户目标系统的需求规格描述为基础, 根据用户定义的文档模式生成的。

- **规格文档模式处理** 输入规格文档模式, 对文档模式文件处理后按照不同的操作命令转向不同的处理模块, 规格文档模式处理包括段定义处理、正文描述处理、注释处理和文档工具命令处理。

- **文档工具** 规格文档生成系统提供了十五种文档工具, 包括表格、矩阵、正文、摘要和图形生成工具, 文档工具操作参数由文档模式定义时提供。

- **文档排版处理** 文档排版处理分为文档源文件处理、分页处理和目录索引处理, 结果送到文档输出控制。

- **文档输出控制** 文档输出有三种方式: 屏幕、文件和打印输出。行宽可以为80列或

125 列,可在系统配置管理中自行设定.

### 6.5 系统配置管理

系统配置管理提供了物理资源设置,用户自定义系统和不同项目的开发标准定义.系统配置包括:时间设置、日期设置、数据库初始化、字典模式定义、报告参数设置、文档模式定义.

**结束语:**RSL/RSA 系统于 1989 年研制成功.目前在该系统基础上,进一步研制有关项目计划管理,如计划网络控制图;系统设计方面,如 E-R 图的编辑、处理和转换,模块结构图的编辑、修改和转换,以及模块说明书和设计文档的自动生成等方面的项目.同时将在该系统单机版的基础上完成网络环境上的系统.

### 参考文献

- 1 《RSL/RSA 系统说明书语言参考手册》,中国科学院软件研究所,1990.10.
- 2 《RSL/RSA 系统说明书使用参考手册》,中国科学院软件研究所,1990.10.
- 3 《RSL/RSA 系统说明书范例参考手册》,中国科学院软件研究所,1990.10.
- 4 Alanm, Davis, A Strategy for Comparing Alternative Software Development Life Cycle Life Cycle Models, IEEE TRAN. on Software Eng. , Vol. 14, No. 7, 1988.
- 5 Herbert Weber, Specification of Modular Systems, IEEE TRAN. on Software Eng. , Vol. 30, No. 6, 1988.
- 6 Macdonald, I G, Automating Information Engineering, Inf. & Soft. Technology, Vol. 30, No. 6, 1988.
- 7 Lvngchun Liu, A Formal Model for Software Project Management, IEEE TRAN. on Software Eng. , Vol. 15, No. 10, 1989.
- 8 John R. Cameron, An Overview of JSD, IEEE TRAN. on Software Engineering, Vol. SE-12, No. 2, 1988.
- 9 H. Ehrig, Programming in—the Large with Algebraic Modular Specification. Information Processing 88, IFIP, 1986.
- 10 Ellis Horowitz, SODOS: a Software Documentation Support Environment — Its Definition, IEEE TRAN. on Software Engineering, Vol. SE-12, No. 8, 1986.
- 11 Dennis R Moreau, Object — Oriented Graphical Information Systems: Research Plan and Evaluation Metrics, The Journal of Systems and Software, 10, 23-28(1989).
- 12 Willis Stinson, Views of Software Development Environments: Automation of Engineering and Engineering of Automation. ACM SIGSOFT, Oct. 1989.
- 13 Sidney C. Bailin, An Object — Oriented Requirements Specification Method, Communications of the ACM, Vol. 32, No. 5, May, 1989.
- 14 Wilson, D N, CASE: Guidelines for Success, Information & Software Technology, 1989.