

GSEIB 软件工程信息库 图形规范子系统的设计与实现^{*}

齐治昌 刘春林 宁 洪

(国防科技大学计算机系 长沙 410073)

摘 要 软件信息库是集成化软件工程环境的核心。本文给出了一种图形化的软件过程与产品 (software process and product) 规范语言及其支撑环境的设计与实现。采用这种规范语言可对给定的软件开发项目进行规划, 据此自动产生定做的软件信息库, 从而真正反映和支持该软件工程项目的开发。

关键词 软件工程环境; 软件信息库; 图形规范语言

分类号 TP311.5

在一个集成化的软件开发环境中, 软件信息库处于核心的地位。一方面, 软件信息库是软件开发各阶段的所有信息的存放和管理处, 各种软件工具从信息库中获得必要的输入信息, 并随时将输出信息放回信息库中, 信息库为软件文档的版本控制、配置管理、查询、追踪等提供支持; 另一方面, 软件信息库为环境提供统一的工具界面, 使各工具之间无需建立直接的接口。

由于软件工程不但随着环境 (工具、模型等) 的变化而变化, 而且, 即使在同一个环境下, 不同的软件项目之间也存在很大差异。采用固定数据模式的软件信息库, 必将有碍于自然、准确的描述信息, 有碍于合理、有效的存储和管理信息。因此, 我们设计的 GSEIB 软件信息库系统采用了一种产生式方法, 即根据用户用某种规范语言描述的软件开发过程和产品的形式规范自动产生定做的软件信息库, 从而真正反映和支持一个给定的软件开发项目。

我们认为, 一个大型软件项目的开发要经历规划、实施、学习与反馈三个阶段。规划阶段主要是为最终能生产高质量的产品而制定一个开发计划。首先要选定一个合适的过程模型 (process model) 和支持该模型的一系列特定方法; 其次, 针对项目特定的目标、环境和构成修正这些方法, 并为所选择的过程模型、方法和工具如何用于后续的执行阶段及学习反馈阶段订出计划。

实施阶段将主要依照开发计划具体执行各个开发过程, 在实施过程中将产生大量的信息, 它们包括:

- 1) 传统的工程文档 (如需求说明书、设计说明书、代码等);

* 1992年8月2日收稿

- 2) 开发计划中指定的各类信息（如测试结果、进度表、完成日期等）；
- 3) 从各个角度对开发过程及其导出产品的分析评估结果。

如此繁杂的信息必须通过信息库来高效地管理。

学习过程主要依据实施阶段导出的产品和各种分析结果,比较期望结果与实际结果,所得结论既可反馈给正在开发的项目（这可能导致重新规划该项目）又可为未来开发项目的规划提供参考。

基于上述观点,我们建立的软件过程支撑环境框架如图 1 所示。

该环境的用户系软件工程师和项目管理人员,它们在系统提供的用户界面上用图形化的规范语言对项目进行规划,规划的结果（即开发计划）存储在信息库中,并以此计划为依据,为实施阶段产生的各种信息确定信息库结构。

整个实施过程是在计划制导下进行的,实施过程导出的各种信息通过信息库界面存入信息库中,信息库界面是面向对象风格的,用户可以通过它进行查询、维护等。

在这个开发模型中,规划是否凑效,主要依赖于对过程模型、方法、工具的说明是否精确。下面介绍我们所采用的图形化的软件过程与产品规范语言 SPPSL (Software Process and Product Specification Language) 及其实现技术。

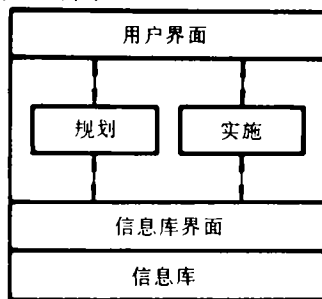


图 1

1 SPPSL 图形规范语言

因为“软件过程也是软件”，所以描述一个软件开发过程,既要描述软件开发过程本身的控制关系和结构关系,又要描述过程与产品的使用和生产关系,还要描述过程和产品须满足的条件。同时应能支持规范进行精化、描述不同程度的细节,规范应易于修改。

SPPSL 图形规范语言中定义了三种对象类型:过程类型、产品类型、约束类型。它们分别用矩形框,圆圈和菱形框表示,如图 2(a)。

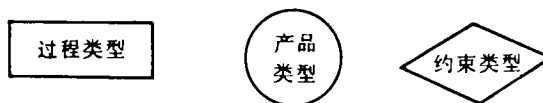


图 2(a)

在此,过程表示软件开发过程中的各种活动,包括软件开发中所采用的各种模型、方法、工具等。

产品表示软件中各种信息,这些信息包括:规范过程所作的计划,计划实施所产生的结果(如需求说明、设计说明、代码),以及测试数据,评估数据等。

约束表示过程开始或结束时刻以及所产生的产品需满足的条件。

SPPSL 图形规范语言可描述对象之间的各种关系,这些关系包括:

- 过程与产品类型之间的输入输出关系,用过程指向产品或产品指向过程的实线表示;
- 约束类型与过程或产品之间的关系,用约束指向过程或约束指向产品的双线表示。
- 过程类型之间的四种控制流关系,用过程指向过程的实线表示;
- 产品类型之间的三种结构关系,用产品到产品的实线表示;
- 过程间依赖关系,用过程指向过程的双线表示;

过程与产品类型之间的输入输出关系描述一个过程实施所需的各种信息和过程实施所产生的各种信息。例如,过程类型 P 使用产品类型 IP1 和 IP2 作为输入,产生 OP1,OP2 及 OP3 作为输出,则表示为图 2 (b)。

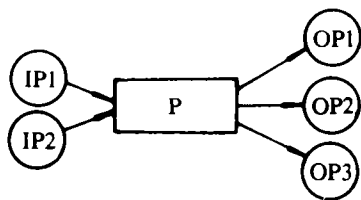


图 2(b)

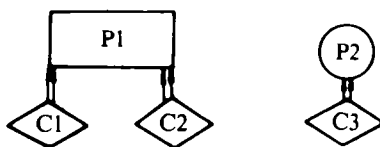


图 2(c)

输入信息一般包括:历史经验数据,由同一项目中其它过程所产生的产品,前期项目所产生的产品,及项目环境的特性等等。输出信息:既可以是可提交的产品(如设计编码文件等等),也可以是各种评估数据,甚至可以是基于学习和反馈所建立起来的新的过程和产品描述。

对过程的约束可分为前置约束和后置约束,前置约束用于描述过程开始前必须满足的条件,后置约束用于描述过程终止后必须满足的条件,产品的约束通常描述产品所应满足的要求,如程序的时空要求等。例如,图 2(c)表示约束 C1 是过程 P1 的前置约束,C2 是 P1 的后置约束,C3 是产品 P2 的约束。约束条件采用布尔表达式形式,在图 2(c)中,若 C3 对应的约束表达式为 $V(P2) < 20$,表示 P2 的时空开销应小于 20。

过程类型之间的控制流关系有四种:顺序关系、选择关系、迭代关系和并行关系。顺序关系表示过程的执行顺序;选择关系表示只能从若干候选中选择一个执行,选择结果根据各候选过程的前置约束决定;迭代关系表示某些过程的重复执行,迭代是否终止由迭代过程的前置约束决定;并行关系表示各过程可以同时进行,相互独立,连线上标记“||”表示并行关系。例如,图 2(d)表示 P0 与 P1 顺序执行,P2 被反复迭代执行,执行完 P3 后,P4 与 P5 基于两个互斥的条件 C1、C2 选择执行,P7 与 P8 可并行执行。

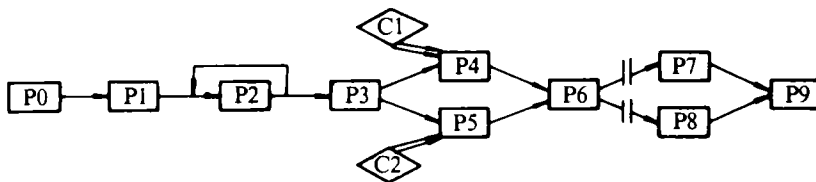


图 2(d)

产品类型之间的结构关系包括顺序关系、选择关系和迭代关系。其表示方式与过程间的控制流关系类似。

过程间的依赖关系定义了过程之间的主仆关系,其意义为:一旦主过程执行了仆过程也要执行。例如,我们要进行一次“设计”过程,同时希望统计设计过程的各种开销,则可将“设计”过程定义为主过程,把“开销评估”过程定义为仆过程,如图 2(e)所示。

它表示“设计”过程的每一次实施都将引发“开销评估”过程的实施。

软件开发过程是一个连续的抽象和精化过程。因此，SPPSL 图形规范语言允许描述过程或产品的分解/聚合关系（即 is-part-of 关系）和描述过程或产品的特殊化/一般化关系（即 is-a 关系）。is-part-of 关系用从过程指向过程或产品指向产品的虚线表示，is-a 关系用从过程指向过程或产品指向产品的带点虚线表示。



图 2(e)

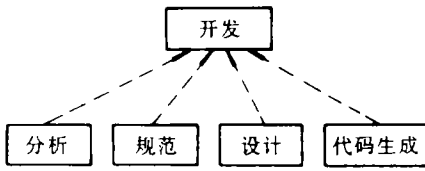


图 2(f)

is-part-of 关系把某些过程或产品精化为更确定的过程或产品，它支持用户自上而下进行规范设计。例如，开发过程分解为分析、规范、设计和代码生成四个子过程，如图 2(f) 表示。

is-a 关系描述过程或产品的特例化和一般化。通常，我们需对具有共性的过程、产品进行抽象，或者对某些抽象的过程、产品实例化。例如，按 yourdon 方法和按面向对象方法(ood)设计是一般设计 (design) 过程的特例，如图 2(g) 所示。

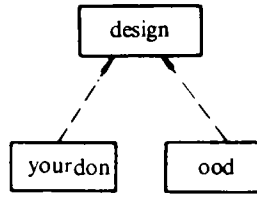


图 2(g)

SPPSL 中还包括其它一些辅助性的成分，如资源描述、角色、注释信息等等，在此不一一介绍。

作为一个例子我们来规划一个开发过程，该开发过程的输入是用户提交的非形式化的需求描述 R；输出是软件产品 S，包括源程序、目标程序、各种文档。用图 3(a) 表示。

开发过程分为分析、设计和代码产生三个子过程，对应地 S 也分解为



图 3(a)

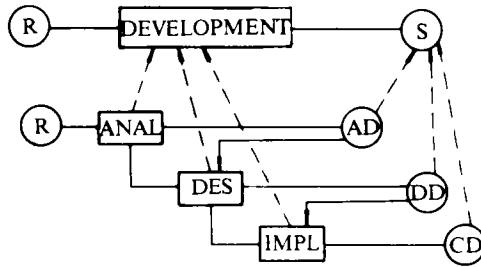


图 3(b)

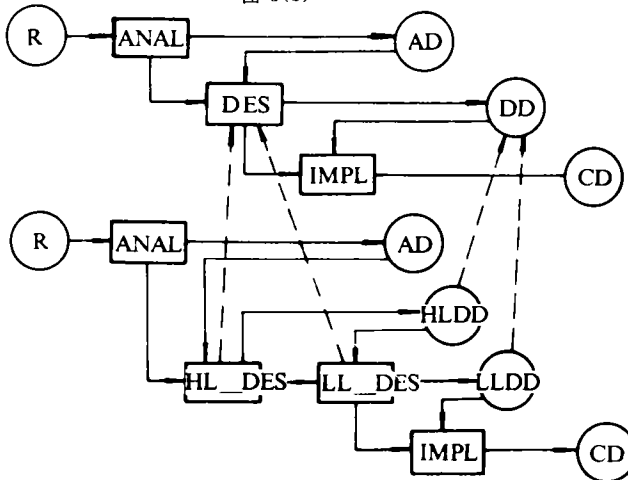


图 3(c)

AD、DD 和 CD 三个子产品，如图 3(b)所示。

其中设计子过程又进一步分解为高级设计和低级设计两个过程，如图 3(c)所示。
该规范还可以进一步精化和说明。

2 图形规范子系统的实现

GSEIB 软件信息库图形规范子系统的功能是实现 SPPSL 图形规范语言，支持用户对软件过程与产品进行规划。

该系统可划分为图形规范生成和转换两大部分，系统结构如图 4 所示。

图形规范生成部分的功能是支持用户在图形编辑窗口下用 SPPSL 图形规范语言对软件项目的开发进行规划，它包括图形界面和图形编辑、语法语义分析、规范生成、信息处理、浏览等功能模块。

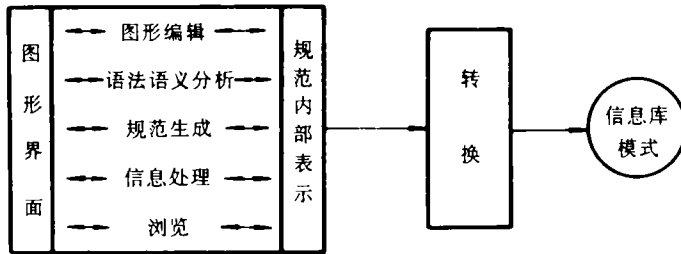


图 4

图形界面控制用户与计算机的交互，用户可以根据需要选择不同菜单，从而调用相应的功能。该界面采用多窗口技术，集固定菜单、下拉菜单、弹出菜单于一体，热键丰富，支持鼠标，使用灵活、方便。该界面还实现了求助功能、色彩的选择功能、图形大小控制功能等。

图形编辑模块支持用户进行修改、删除、移动等编辑功能，实现图形的显示和布局，并把它们表示为广义图段表形式。本系统采用语法制导编辑方法支持作图和编辑，保证图形规范的语法正确性。图形的有效存储是设计图形编辑功能必须解决的问题，为此，我们采用了一种高效的压缩机制节省图形的存储开销，并设计了一种有效的图形滚动算法实现了图形的快速滚动，可使用扩展内存或磁盘作为后援存储，使图形规范大小不受限制。

语法语义分析模块完成图形规范的语法和语义合法性检查。当编辑过程遇到语法错误（如对象重叠，连线不合法等）或语义错误（如对象重名等），系统立即报错，并拒绝该动作的执行。

规范生成模块，根据广义图段表的信息和语法语义分析的结果，产生软件过程与产品规范的内部逻辑表示。图形规范子系统不同于一般的图形编辑器。一般的图形编辑器，只注重图本身，但图形并不存在语法语义上的关系，而图形规范中各图元有着复杂的逻辑关系。这些逻辑关系将表示成规范的内部逻辑形式，它们由一组表格组成，主要包括：对象信息总表、分解关系表、特殊化关系表、残根表、树形结构表等等。

为了增加图形编辑器的通用性，我们把图形编辑界面、广义段表与图的内部逻辑表

示分离开来，这样即可提高效率又易于移植。在此图形编辑器的基础上构造图形分析设计工具，只需对逻辑表进行适当修改。

一个完整的过程与产品规范往往需要一些附加信息，如对象的注解、约束表达式等。对这些信息，我们通过专门的对话框进行编辑和处理；另外，为了支持自上而下设计方法，我们还允许用户把一个图形规范分解到若干图上，这种分解也通过信息处理模块处理。

浏览模块支持用户对图形规范及各种信息的浏览。浏览内容包括：浏览分解/聚合关系，浏览特殊化/一般化关系，浏览约束表达式，浏览各种注释信息等。

转换部分的功能主要是根据用户设计的图形规范，从内部逻辑表示中搜索和提取信息，自动生成与规范相适应的支持该项目开发的信息库模式。在转换过程中首先要把它转换为一种归一化的中间形式，然后对图形规范的完整性和一致性进行检查，最终转换成信息库界面上定义的信息库模式语言表示。

归一化的数据结构是信息转换的中间形式，它主要包括：对象文件 Objfile、产品的输入输出关系文件 I-O-file，过程的聚集关系文件 Aggcfile，产品的聚集关系文件 Aggdfile，过程的 is-a 关系文件 Genfile，产品的 is-a 关系文件 Gendfile，过程的顺序关系文件 Cfrsfile，过程的依赖关系文件 Cfrdfile，过程的分支关系文件 Cfrafile，过程的循环关系文件 Cfrifile，过程的并行关系文件 Cfrpfile，对象的约束关系文件 Condfile 等。其结构简明，用元级形式表示出复杂的关系。以 I-O-file 为例，其元级框架为

agname	procname	prodname	tag
--------	----------	----------	-----

其中 aggeame 为聚集名，procname 为过程名，Prodname 为产品名，tag 为标志位（“i”表示输入，“o”表示输出）。基于此结构更便于模式转换和一致性与完整性检查。

为了从内部逻辑表示中搜索和提取信息，我们采用了线性遍历与局部树遍历相结合的方法，提高了转换速度。

目前，GSEIB 软件信息库图形规范子系统及其相应的软件项目支撑环境 SPSE 已在 COMPAQ—386 机上实现并通过技术鉴定。

参 考 文 献

- 1 P A Bernstein. Database System Support for Software Engineering. Proceedings of the Ninth International Conference on Software Engineering, Monterey, CA, March 30—April 2, 1987, 166~178
- 2 H D Rombach. L Mark. Softrare Process and Product Specifications; A Basis for Generating Costumized Information Basis, University of Maryiand
- 3 Leo Mark, H Diezer Rombach. Generating Cuetomized Software Engineering Information Basis from Software Prooess and Product Specifications, University of Maryland
- 4 K Dittrich. Object—Oriented Database Sstems; The Notion and the Issues. Proceedings International Workshop on Object—Oriented Database Systems, Pacific Grove, CA, Sept. 1986
- 5 L Osterweil. Software Processes are Software Too. Proc. Ninth International Conference on Software Engineering, Monterey, Ca, March 1987, 13

The Design and Implementation of Graphical Specification Language in GSEIB

Qi Zhichang Liu Chunlin Ning Hong

(The Department of Computer Science and Engineering, NUDT, Changsha, 410073)

Abstract

SE information base is the kernel of an integrated SE environment. This paper presents a graphical software process and product specification language, a framework for SE project support, and the actual efforts at implementing the graphical specification subsystem. From the formal specification of a given SE project, a customized SE information base can be generated, which truly reflects and benefits the SE project.

Key words software engineering environment; software information base; graphical specification language

(上接第 60 页)

The Failure Detection System of Liquid Rocket Engine Based on Neural Network

Huang Minchao Zhang Yulin Feng Xin

(The Department of Aerospace Technology, NUDT, Changsha, 410073)

Abstract

A neural network system used for failure detection of a liquid rocket engine (LRE) is described and presented in this paper. The system includes two layers; the first processing layer consists of the winner-take-all (WTA) neural network used for LRE fault diagnosis; the second layer consists of the back-propagation (BP) neural network used for displaying the failure value with the output result of the first layer. In the paper, the failure detection simulation of a liquid rocket engine is given to show great advantages for the failure detection system of the neural network.

Key words liquid rocket engine; WTA neural network; BP neural network; failure detection