

58-62 面向对象的需求可跟踪性支持工具的设计

雍信阳 蔡希尧 陈平

(西安电子科技大学软件工程研究所 西安 710071)

TP316.2

摘要 保证需求可跟踪性是实时系统开发中的一个重要方面,本文提出了一个实时系统的需求可跟踪性的模型。在此基础上,作者采用数据管理系统开发了一个支持基于使用实例分析的面向对象的系统开发过程的需求可跟踪性的辅助工具(OORTSI)。

关键词 需求可跟踪性,使用实例方法,面向对象技术

实时系统

操作手册

在大型的计算机密集的实时系统的开发中,人们所关心的一个首要问题就是开发系统的性能是否满足了所提出的要求。作为系统的开发和维护,往往要对许多影响系统的大部分组成部件进行决策或折衷考虑。系统的需求在开发过程中经常变化,必须在系统需求和系统设计阶段所产生的各种输出之间建立适当的需求可跟踪性,以此来保证所开发的系统满足了当前的需求集合。一个比较直观的保持需求可跟踪性的方法就是把系统开发的各个阶段所产生的系统部件与需求相联系^[1-3]。这种部件包括硬件、软件、人物(Humanware)、操作手册、各种政策以及工作流程等。为了达到这一目标,必须在系统开发的各个阶段(即从用户原始陈述经分析、设计、实现、测试直至最终产品)保持需求可跟踪性。

一、实时系统的需求可跟踪性模型

实时系统的开发中有许多影响因素,主要有项目倡导者、项目经理、分析员、设计人员、测试人员和最终用户等。因此,要建立一个合适的需求可跟踪性模型,首先要解决与这些因素有关的各种问题。

(1)全面的需求可跟踪性 一般分为四种。就方向而言,前向可跟踪性是我们系统在开发中经常能够想到的,但是,要使开发出来的系统不具有冗余的功能以及将来维护方便,则必须建立起反向可跟踪性,因此对需求可跟踪性的支持应该是双向的;就需求描述前后的可跟踪性而言,需求描述后的可跟踪性是要在需求描述和其后的设计、实现各阶段之间建立起可跟踪性,而需求描述前的可跟踪性则是要在用户原始需求陈述和需求描述之间建立可跟踪性,无论是在技术方面还是在工具方面,需求描述后的可跟踪性都受到了较多的重视。至于需求描述前

的可跟踪性,不仅研究较少,而且也基本上没有工具支持。笔者认为这种情况对于可跟踪性是极为不利的。对于需求描述后的可跟踪性的支持,一般都把需求描述假设成一个黑箱,却没有关于得到这个需求描述的过程的描述,只有最终输出的需求描述。需求描述往往是静态的、结构化的和综合的,无法适应需求来源的易变和动态等特点,当然也就无法支持需求的逐步精化过程和来自不同信息源的需求的集合。所以需求描述后的可跟踪性是依赖于需求描述前的可跟踪性的。另外,由于需求描述前的可跟踪性问题所特有的一些范式独立性,有些问题无法彻底解决,只能通过显式的方法手段加以辅助。很显然,用户的原始需求陈述是不规则和模糊的,自然难于在它和需求描述之间建立起可跟踪性。

需求描述前的可跟踪性目前已经逐渐为人们所认识。系统开发质量的好坏和抗变化能力取决于一个好的需求描述,而产生需求描述的过程则是决定需求描述质量好坏的首要因素,解决好这一问题将有利于提高系统的开发质量。

(2)关键需求 在系统开发过程中,保持所有需求信息的可跟踪性既不希望、也不必要,更不可能,因为这需要太多的开销。一个好的识别系统关键需求的方法就是将需求和系统的中心任务联系起来。通过识别产生需求的业务过程和任务,可以识别需求并加以评价和分类。例如,对于需求描述前的可跟踪性而言,我们应该能够解释为什么会有某个需求和如何得到这个需求,这就需要一种能够表示从产生任务的需求开始的需求逐步精化的过程,事实上,这就是一种以需求描述为目标的需求描述前的可跟踪性。

(3)设计说明 在系统开发过程中,建立设计说

明信息的可跟踪性,可以帮助捕获进行设计决策的理由。另外,跟踪不同的设计对象,理解其受变化的影响在系统维护中也是非常重要的。系统开发者需要利用可跟踪性来检查系统背后的逻辑原理,这在实时系统的设计中尤为重要。

(4)项目管理 需求可跟踪性用于项目管理也非常有利,在系统的设计和随后的各个阶段,必须利用可跟踪性来保证所有的需求得到了满足。进一步地,在系统生命周期的各个阶段建立尽可能完善的需求可跟踪性,可以大大简化测试过程。

项目经理可以利用关于项目的状态、完成日期、保密权限等信息的可跟踪性进行系统的计划和安全等方面的考虑。而且,要想把项目管理纳入整个系统开发过程中,这种信息是必不可少的。

(5)审计 可跟踪性还可以用于系统的审计,例如,人们可以合理地使用可跟踪性与系统部件的原始设计者进行交流并理解其功能,但是,这些审计有时必须谨慎对待,因为用这种审计来代替对系统性能的评价是非常不合适的。使用可跟踪性可以捕获的审计信息包括系统开发人员在系统开发各阶段修改和证实的各个设计单元。很显然,这种信息对于系统维护也是必不可少的。

(6)人物 人是系统开发中最活跃的因素,因此在系统开发中捕获需求与人之间的可跟踪性也是非常重要的。这种可跟踪性主要是指在需求信息和人之间建立责任联系,更为一般的方法是,把人看作是系统的构成部件,在这种部件和其它部件之间建立可跟踪性,这样就可以保障需求的正确分配和实现。

(7)文档 文档可跟踪性主要是在不同的需求文档之间建立可跟踪性,能保障一个文档中的所有组成部件在另一个文档中都有反映,可以解决文档中存在的的不同一致性和不完整性。

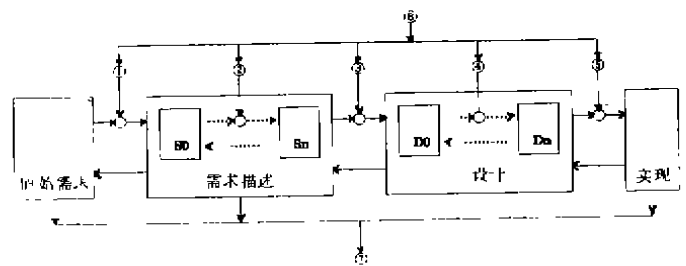
(8)依赖性 复杂系统往往是由大量的相互依赖的部件组成的,建立基于依赖关系的可跟踪性在这种系统的开发中是一个现实的问题。另外,在系统设计过程中的许多决策也是相互依赖的,如果不能建立关于这种信息的合适的可跟踪性,那么随着时间的推移,这种信息将会丢失,这样在进行系统维护时,对于一个部件的变更可

能导致系统中存在不一致(因为涉及了其它的部件),这也是系统维护中的一个重要问题。

(9)水平和垂直可跟踪性 垂直可跟踪性是指系统生命周期不同阶段之间的可跟踪性,一个典型的例子就是分析和设计之间的可跟踪性。水平可跟踪性是指同一系统开发阶段内部的可跟踪性,例如,对于 Yourdon 结构化方法,在进行 DFD 分层描述时,不同层次的 DFD 之间需要有可跟踪性。很显然,垂直可跟踪性比水平可跟踪性容易获得,因为垂直可跟踪性的建立已经存在一些规则。

(10)自动支持 在系统较复杂时,通过手工方式进行可跟踪性支持不仅费时易错,而且能力有限,对需求可跟踪性的自动化支持目前已成为一种迫切的愿望。

根据以上的讨论,我们得到如图1所示的需求可跟踪性模型,该模型通过在系统开发的各个阶段之间和各个阶段内部进行显式的需求可跟踪性描述来对系统开发的全过程进行双向的可跟踪性支持。通过对各个阶段之间和各个阶段内部的工作的责任分配和文档的形成来跟踪文档信息和人物信息。通过设计说明、依赖性描述、审计信息记录来支持有关信息的跟踪,通过关键需求的识别对需求信息进行有选择的跟踪,通过需求描述、设计和实现各个阶段的逐步精化过程来支持水平可跟踪性,通过对建立各个阶段之间和各个阶段内部的需求可跟踪性提供自动工具支持来提供一个自动的需求可跟踪性工具,具有可跟踪性特性的系统开发各阶段的信息被用于项目管理。由此可见,该模型较好地解决了建立需求可跟踪性时存在的问题,为系统开发提供了较好的



①关键需求识别、责任分配、文档形成;②责任分配、文档形成、审计信息;③责任分配、设计说明、文档形成、依赖性描述、审计信息;④责任分配、设计说明、依赖性描述、文档形成、审计信息;⑤责任分配、文档形成、审计信息;⑥自动工具支持;⑦项目管理。

图1 需求可跟踪性模型

可跟踪性支持。

二、OORTST 的设计

下面我们讨论一个支持需求可跟踪性的工具(OORTST)的设计,其应用对象是一个 C3I 系统中的雷达子系统(一个典型的实时系统),其支持对象是一种支持关键使用实例识别的面向对象方法,它可以支持开发方法的全过程。就这一点而言,OORTST 是一种支持环境类型的需求可跟踪性支持工具。在进行 OORTST 的设计时,除了以需求可跟踪性为主要目标外,还考虑了许多其它的需求工程问题。对工具的需求管理支持要求如下:

(1)支持实例、主题、对象、属性、服务的识别与描述;

(2)支持图1所描述的需求可跟踪性;

(3)支持来自不同信息源的需求的综合,主要包括对用户原始需求陈述、开发合同、用户访谈、领域专家访谈、相互系统的开发经验及其它有关材料的综合;

(4)支持关于需求描述的必要性、与稳定性的注释,必要性分为必须的、期望的和可选的三种,稳定性分为稳定的和易变的两种;

(5)支持关于接受测试的描述,及对于特定的需求在系统开发的各个阶段给出其测试方法和测试状态;

(6)支持需求描述的一致性证明;

显然,OORTST 不仅仅是一个需求可跟踪性的自动支持工具,在一定程度上可以说是一个辅助需求开发工具。就需求可跟踪性而言,通过与系统开发的全过程相结合,在一定程度上可以说是一个需求可跟踪性支持环境。鉴于 OORTST 的特定和具体的工作条件,我们将采用关系数据库系统进行数据存储,所开发的工具虽然基于使用实例的面向对象方法,但是,由于所有面向对象方法与基于对象的一致性,因此对其它面向对象方法也适用,其移植也相当方便。

2.1 OORTST 的设计考虑

工具的使用效果好坏与特定的系统开发环境和开发任务有紧密的联系。因此,在 OORTST 开发之前,我们对工具的所处环境进行了分析,图2是一个环境示意图。

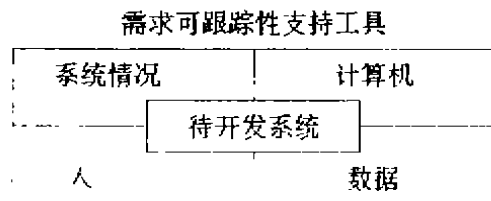


图2 需求可跟踪性支持工具所处环境

系统情况 (1)就系统开发时间和耗费的金钱和人力而言,目前的系统规模越来越大,较长的开发周期给开发管理带来了许多困难,C3I 系统的开发就是一个典型的例子;(2)系统开发的组织结构可能是分布的和庞大的;(3)系统开发过程中可能会有开发技术的变化,至少在局部极有可能;(4)财务、政策等问题可能对系统的开发产生影响;(5)系统开发中的变化是难以避免的,有些甚至是实质性的变动;(6)所开发的系统要受到军用标准的约束。

数据 (1)大型系统可能会有数以百计的对象,那么有关的属性和服务则更多,因此,总需求条目可能会数以万计;(2)需求可能是由文字、图形、表格等各种形式存在的;(3)大量的数据将会产生数据库的访问速度过慢的问题;(4)数据往往是易变的,对数据的理解需要文档的支持;(5)需求描述往往是非形式化的和不完整的;(6)需求是多种多样的,其顶层的描述往往比底层要容易得多;(7)需求来源是多种多样的,其描述需要服从军用标准。

人 (1)需求管理人员可能是有经验的人员,也可能是新手;(2)对待支持工具的态度经常不一,有人不喜欢支持工具,有人不喜欢某一种类型的工具;(3)工具使用人员的经验对使用效果有重要的影响;(4)人们对需求管理的重要性经常认识不足;(5)管理的短期性与系统开发的长期性之间存在矛盾。

计算机 (1)计算机的能力在不断增加;(2)支持工具经常与特定的操作系统或软件平台有关;(3)分布式计算机系统和分布式数据库目前正在获得广泛应用;(4)对图形用户界面的支持正在增加;(5)计算机桌面印刷系统已经获得了成功的应用;(6)人工智能技术在转机系统中的应用产生了很大的影响。

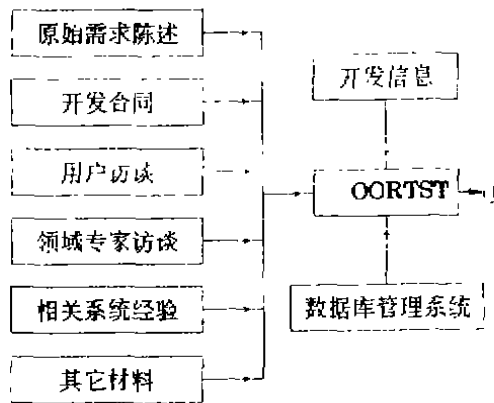
由分析可知,类似 OORTST 的一个需求工具要想成功地应用于特定的应用环境,需要协调好许多影响因素,首先必须适应待开发的系统和要操纵的数据情况,其次,必须有适当的计算机支持和用户的正确使用。另外,需要说明的是,工具 OORTST 是针

对特定的系统开发方法的,虽然限制了工具的灵活性,但对于提高工具的性能是有益的。

2.2 基于数据库管理系统的 OORTST 的设计

根据 OORTST 的具体特点,我们选择数据库系统来作为实现手段,为了支持保证系统的开放特性,采用支持 SQL 应用的实现方式。

在 OORTST 中,输入来自不同信息源的各种需求,通过系统的处理,为系统开发提供各种有用的信息。图3是一个反映 OORTST 中的输入与输出情况的数据流图。



①具有可跟踪性的与系统开发有关的经过一致性检查的各种需求信息和开发信息

图3 OORTST 的数据流情况

OORTST 的工作过程与系统开发过程相结合,从综合来自不同信息源的需求开始,它不仅把使用实例分析作为一种需求描述手段,同时还作为一种需求完善的手段,通过使用实例分析可以使分析员对待开发系统有一个更深的了解,有利于澄清一些模糊的问题。在完成原始需求收集后,对原始需求进行划分,由此为需求可跟踪性的建立打下基础,接下来的工作就是系统的分析、设计与实现。在设计的过程中,我们将重点放在需求描述前的可跟踪性问题上,对于需求描述后的可跟踪性,将在实例、对象、属性、服务四个层次上进行。根据数据库设计的情况,我们对输入和输出分别进行了设计。

OORTST 具有循环和递归的能力,在系统的设计或实现阶段,可能需要增加某些出于设计或实现考虑的对象等,为了保持需求可跟踪性,工具允许返回分析阶段增加关于该工作的考虑,这样在分析、设计、实现阶段的可跟踪性和平滑过渡问题都得到了完美的解决。同时,随着系统开发的过程深入,工具

允许对对象进行分解,被分解的对象则上升为主题。系统所提供的查询手段是很丰富的,既有较详细的查询,也有一般性的浏览。系统的输出内容非常丰富,有对象描述表、对象交叉参照表、设计说明表、对象依赖性表、对象关系表、测试情况表、管理信息表和实例、对象等层次的跟踪关系表。

在 OORTST 的设计过程中,我们发现,对于需求描述前的可跟踪性,其粒度和准确性经常是有限的,例如,对于一个需求描述中的对象,在原始需求中可能有很多地方会涉及到,我们不可能去掉其它地方的描述而只保留一个地方的描述,这样一来,我们就在粒度和准确性之间进行折衷考虑。如果需求划分的粒度较大,那么准确性可能就较小,如果需求划分的粒度较小,那么一种可能就是根据分析员的判断,将需求描述中的对象与特定的原始需求对应起来,或者干脆将该对象与需求描述中所有涉及该对象的原始需求相对应。对于后者来讲,显然影响了发自原始需求的前向可跟踪性。这个问题是由需求描述前的可跟踪的独特的范式所致。在 OORTST 的实现过程中,我们采用了小粒度与分析员判断相关的方法,即对于对象整体而言,选择原始需求与之关系最为密切的部分建立联系,之所以说与分析员判断相关,是因为实现中采用了一种辅助需求来源的注释方式,即在数据库中不仅标识与对象关系最密切的原始需求,还标识涉及该对象的所有原始需求,以此,我们成功地解决了上述问题。对于属性和服务,出现上述情况的可能性较小,即使出现,分析员的判断一般也不会出现误差。对于需求描述后的可跟踪性,并不是所有的信息的在任何阶段都是或都必须都是双向可跟踪的,例如,对于对象、属性、服务,双向可跟踪是必要和可行的,但对于使用实例,前向可跟踪性总是必要和可行的,而发自源代码的反向可跟踪性则基本是不可行的。在前面的几个阶段,双向可跟踪则是可行和必要的,因为分析、设计等各个阶段的工作是按照使用实例为单元来进行的。

三、小结

OORTST 对需求描述前后的可跟踪性都是较全面的,提供了丰富的用户界面和灵活的查询方式。在工具的实现过程中,充分考虑了前述四方面的因素。但是,由于关系数据库在复杂数据类型和图形存储方面的局限性,使得工具对可跟踪性的支持也受到了限制。

软件开发

面向对象模型

软件工程

计算机科学1996 Vol. 23No. 1

62-65

软件开发与面向对象模型*

王斌君 郝克刚

(西北大学计算机科学系 西安 710069)

TP311.52

摘要 From software development views, this paper discusses some important object-oriented models in current, and give an ideal 3-DOOM model by myself.

关键词 Model, Object-oriented model, Three-dimensional object-oriented model.

一、引言

70、80年代,以结构化方法为代表的软件工程方法使得人们开发软件可按步骤、有章可循地组织完成,但它并没有从根本上彻底解决60年代末出现的软件危机所带来的问题,其原因是计算机可接受的软件模型(程序设计语言的结构)与其要解决现实世界中的问题结构大相径庭。面向对象的程序设计语言(OOPL)可直接地表达问题域中的对象(如人、马达、栈等)和结构(如分类结构、各种关系等),使得

软件的模型与问题空间模型非常地相近和相似,从而保证了软件易于构造。

近几年,人们为了充分地发挥OOPL的优势,将研究的精力集中在面向对象的设计(OOD)^[1-2]和面向对象的分析(OOA)^[3-6],试图从面向对象方法学的高度,探索出一条面向对象的软件开发(Object-Oriented Software Development,简称OOSD)方法,即通过OOA、OOD和OOP将现实世界的空间模型平滑而自然地过渡到面向对象的软件模型,使得软件开发过程与人们认识问题空间的过程保持最大限

OORTST的使用比较简单。尽管OORTST的实用性已经得到了证明,但只是试验性的,作为产品其考虑仍不完全。例如,采用一个面向对象的数据库系统将有利于复杂信息的查询与存储,可以提高OORTST的支持能力。就OORTST的用户而言,对使用实例的面向对象方法和需求可跟踪性问题必须有一定的了解。另外,在系统开发中由专人负责工具的使用也是比较有益的。

参考文献

- [1] A. M. Davis, Software requirements: Analysis and specification, Prentice-Hall Inc., 1990
- [2] Richard Jordan 等, Streamlining the project cycle with object-oriented requirements, OOPSLA'94
- [3] J. D. Palmer 等, An integrated environment for requirements engineering, IEEE Software, May 1992
- [4] Kenneth S. Rubin 等, Object behavior analysis, CACM, Vol. 35, No. 9, 1992
- [5] Klaus Pohl, The three dimension of requirements engineering: A framework and its application, Information Systems, Vol. 19, No. 3, 1994
- [6] R. G. Mays 等, PDM: A requirements methodology for software system enhancements, IBM System J. Vol. 24, No. 2
- [7] B. Ramesh 等, Issues in the development of a requirements traceability model, Proc. of the IEEE Intl Symposium on Requirements Engineering, San Diego, California, Jan. 4-6, 1993
- [8] J. M. Drake 等, Approach and case study of requirement analysis where end users take an active role, IEEE Software, 1993
- [9] C3ISE-XD-93计划, 西安电子科技大学软件工程研究所

* 1) 本文受863/CIMS高技术项目支持。王斌君 讲师;郝克刚,教授,主要研究方向为软件工程、面向对象等