

基于语言转换工具的遗产系统移植方法

陈明 朱炜 潘金贵

(南京大学软件新技术国家重点实验室 南京大学计算机科学与技术系 南京210093)

摘要 解决遗产信息系统问题有三种策略:重新开发、封装和移植。由于前两者存在较大的缺陷,移植策略成为了重要的研究和应用领域。本文分析了目前已有的遗产系统移植方法的不足和未解决的问题,在此基础上提出了一种着重于实现代码自动移植的基于语言转换工具的移植方法,给出了一种具有灵活性和可操作性的通用移植过程模型。

关键词 移植,遗产信息系统,网关,数据移植,语言转换工具

Legacy System Migrating Method Based on Language Transformation Toolset

CHEN Ming ZHU Wei PAN Jin-Gui

(National Key Laboratory of New Software Technology, Computer Science of Nanjin University, Nanjing210093)

Abstract Redevelopment, wrapping and migrating are the three basic solutions to the problems of legacy information systems. Because the two formers have inherent shortcoming, migration becomes an important research and practice issue. This paper introduces the existing migrating methods, points out their shortage and the problems they did not resolved, then proposes a migrating method based on language transformation toolset, which emphasizes on implementing code migrating automatically, and provides a general flexible and operational model of migrating process.

Keywords Migration, Legacy systems, Data migrating, Gateway, Language transformation toolset

1 引言

在过去的几十年里,计算机技术在社会各行各业中得到广泛的使用。很多信息系统已经成为企业的重要资源。但随着时间的推移,这些规模庞大且功能复杂的系统已经到了无法继续维护的状态。这些系统存在很多问题,比如,脆弱、不灵活、不可扩展、无法满足不断变换的业务需求。这样的系统被称为遗产信息系统(Legacy Information System, LIS)^[1]。它们已经成为企业或组织进一步发展的障碍。但由于遗产系统是企业信息系统多年经营的结晶,往往意味着巨额的投资、成熟的应用模式和稳定的运行,并在一定范围内仍然可提供充足的服务,突然停止运行这些系统会给企业带来巨大的损失,因此必须想办法解决这个问题。

解决遗产信息系统的问题可以有三种策略:重新开发(Re-development),封装(Wrapping)和移植(Migrating)。重新开发策略由于失败风险太大和费用太高,开发周期长,现实中很少被采用。封装是目前采用较多的策略。它只能在短期内暂时和部分地解决问题,一般只是提供一个新的GUI,而不能彻底地解决问题。移植策略由于风险和费用相对较小,又能较彻底地解决问题,因此越来越受到重视。很多遗产系统正在被移植到新的软硬件环境中,以满足新的业务需求。GTE Lab的Brodie提出了数据库最先,数据库最后,复合数据库的移植方法和Chicken Little策略^[2]。该策略的基本思想是增量式的移植。爱尔兰都柏林大学的Bing Wu等提出了蝴蝶(Butterfly)方法^[3],致力于遗产数据的移植。这些方法有些仅限于理论上的探讨,现实中不可行,有些注重于解决某个特定的问

题,没有提出一个完整的移植过程框架。本文在介绍已有移植方法的基础上,结合已有的研究成果和实际情况,提出了一种新的移植方法——基于语言转换工具的移植方法。

2 移植策略和方法

移植是指把一个已存在的、运行中的大型系统通过一定的方法和技术转移到一个新的软硬件平台,并保留原有系统的功能,尽可能少地打断原有系统的操作和业务环境。移植涉及到软件工程的许多领域,如程序理解、数据库理解、系统开发、软件测试等。目前已提出了五种移植方法或策略,其中有的是面向数据库,有的是面向移植过程。

2.1 数据库最先方法

数据库最先方法也称正向移植方法^[2],基本思想是先把遗产数据移植到一个先进的数据库管理系统,然后逐步移植应用程序和接口。当目标应用程序和接口在开发时,遗产信息系统可保持原有的操作。这个性能的实现由一组允许在遗产信息系统和目标系统之间进行互操作的动作来完成。这种互操作动作由一个称为网关(Gateway)^[1]的模块提供。网关在移植过程中具有多个重要角色,比如,隔离单个组件使得不被其它组件改变,在组件之间转发请求和数据或者代理数据查询和更新。

数据库最先方法中使用的网关称为正向网关,它使得遗产应用程序在移植过程中能够访问目标数据库环境。如图1所示,正向网关负责向新数据库服务转发来自旧系统的请求,同时新数据库服务返回的结果被网关转发到遗产应用程序。数据库最先方法只适用于可完全分解的遗产信息系统,遗产数

陈明,朱炜 硕士研究生,主要研究领域为软件工程。潘金贵 教授,博士生导师,从事中间件、Agent技术及多媒体远程教学系统等研究。

据服务必须有清晰的接口。

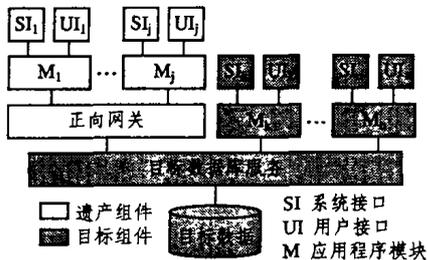


图1 数据库最先方法

K. Menhoudj 和 M. Ou-Halima^[4]对此方法提出了改进,移植被分为多个更小的移植步骤进行。在每一小步,一个或多个文件根据预定义的次序被移植。应用程序直到它所访问的所有文件被移植完成后才进行移植。因此,目标系统中永远不会有有一个模块需要访问遗产信息系统中的文件,遗产信息系统中的文件可以访问目标系统中的文件,因此需要一个正向的数据库网关。图2是此方法的一个例子,模块为 M_i , 数据文件 F_i , 假定数据的移植次序是, F_1, F_2, F_3 。 F_1 首先被移植为 T_1 。由于 M_1 只访问 F_1 , 因此 M_1 可以被移植, 见状态 b , 然后 F_2 被移植, 则 M_2 访问的数据全部被移植, 所以 M_2 可以被移植。

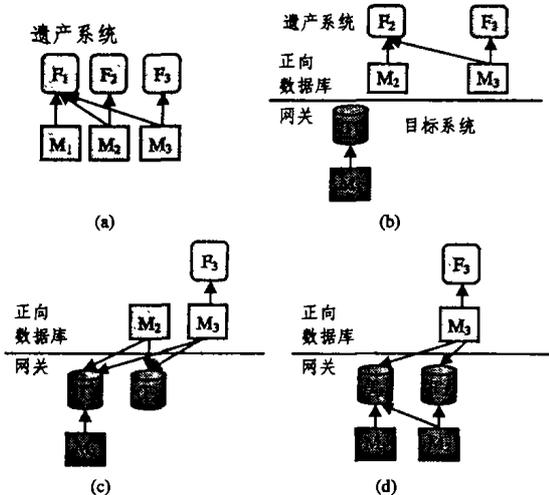


图2 改进的数据库最先方法

这个方法的关键是事先指定移植次序。它基于最小化改变遗产模块的原理,主要思想是分析遗产数据文件间的依赖关系来决定移植次序。整个过程的细节可以参考文[4]。此方法对遗产信息系统的移植领域做出了较大的贡献,因为它从实践的角度解决了问题,提出了实际可操作的步骤,而现有的其它方法都比较抽象,不能在移植过程中解决实际问题。

2.2 数据库最后方法

数据库最后方法也称逆向移植方法^[2],类似于数据库最先方法,也只适用于可完全分解的遗产信息系统。遗产程序逐步地被移植到目标平台,数据库保留在遗产平台。数据库的移植是移植过程的最后一步。和数据库最先方法一样,网关被用于进行遗产信息系统和目标系统之间的互操作。数据库最后方法所采用的逆向网关允许目标程序访问遗产数据,把目标系统的请求重定向到遗产数据库服务。

2.3 复合数据库方法

复合数据库方法适用于完全可分解、半可分解、不可分解的遗产系统。现实中,很少有系统的结构是单一的,大多数遗产信息系统包含有可分解的组件、半可分解的组件和不可分解的组件,这样的结构称之为复合信息系统结构。

在移植过程中,遗产信息系统和目标系统形成了一个复合信息系统。如图3所示,复合数据库方法使用了正向网关和逆向网关。此方法允许数据在遗产信息系统和目标系统之间进行复制。为了维护数据的一致性,一个事务协调器被用于截获所有到遗产信息系统或目标系统的更新请求,判断是否同时影响到两个数据库中的重复数据,如果是,通过两阶段提交协议,更新操作将会在两个数据库中都生效。

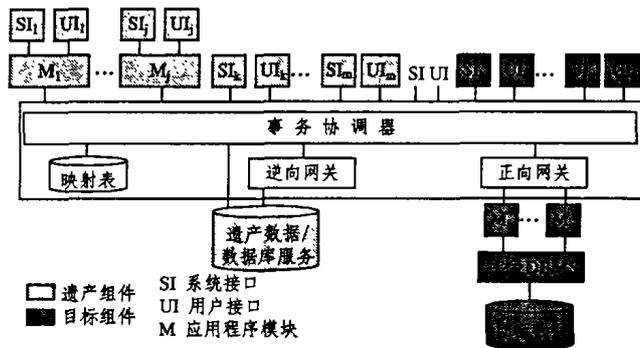


图3 复合数据库方法

复合数据库方法取消了数据库最先和数据库最后方法中单一的、大规模的遗产数据移植,这在任务紧急的环境下是很重要的。但是,这个方法不仅有前两种方法的问题,而且还增加了事务协调器,同时使用两个网关也带来了更大的复杂性。

2.4 Chicken Little 策略

Chicken Little 策略是复合数据库方法的优化^[1]。此方法提出了通过使用多种网关来处理可分解、半可分解和不可分解遗产信息系统的解决方案。这些不同种类网关之间的区别取决于它们在系统中的位置和它们提供的功能。但是,它们有一个相同的目标,就是协调软件组件之间的互相操作。Chicken Little 策略提出了一个适用于任何移植项目的11步的移植过程,每一步处理移植的一个方面,可以调整此方法的步骤来满足某个具体的遗产信息系统的移植需求。这些步骤不必按次序执行,有些步骤还可以并行执行。

- Step1: 增量分析遗产信息系统;
- Step2: 增量分解遗产信息系统的结构;
- Step3: 增量设计目标接口;
- Step4: 增量设计目标应用程序;
- Step5: 增量设计目标数据库;
- Step6: 增量安装目标环境;
- Step7: 增量创建和安装必要的网关;
- Step8: 增量移植遗产数据库;
- Step9: 增量移植遗产应用程序;
- Step10: 增量移植遗产接口;
- Step11: 增量交割到目标信息系统。

Chicken Little 策略的核心思想是实现增量移植,比如一次移植一个组件。网关用于遗产信息系统和目标系统的交互操作。使用 Chicken Little 策略,数据被存储在遗产信息系统和目标系统中。在多数情况下,网关协调器不得不用来维护数据的一致性。正如 Brodie 和 Stonebraker 指出,异构信息系统之间的更新一致性是一个复杂的技术问题,还没有通用的解决方法,仍然是一个开放的研究课题。因此,使用 Chicken Little 方法对任何移植工程师来说是一个巨大的挑战,另外,此策略并不包含测试步骤,而测试是移植过程中一个非常重要

的部分。

2.5 蝴蝶方法(Butterfly Methodology)

以上四种都是基于网关的移植方法,但是网关的使用给移植带来了复杂性。蝴蝶方法是一种无网关的移植方法^[3],它的出发点是质疑移植过程中遗产信息系统和目标系统需要并行操作的必要性。蝴蝶方法的目标是指导任务紧急的遗产信息系统的移植。这个方法不需要系统用户同时访问遗产信息系统和目标系统来保持两个异构信息系统的数据库一致性。蝴蝶方法基于如下2个假设:

(1) 遗产信息系统的数据库是系统最重要的部分。

(2) 从目标系统开发的角度看,不是持续变化的遗产数据而是数据的语义或者模式是至关重要的。

因此,蝴蝶方法分离了目标系统开发和数据的移植,淘汰了网关。

蝴蝶方法把遗产信息系统的移植分为六个主要阶段。每个阶段由许多单个的、独立的移植活动组成。蝴蝶方法的侧重点是数据移植。

阶段0:移植的准备工作。

阶段1:理解遗产信息系统的语义,开发目标数据模式。

阶段2:在目标系统中,构造一个基于目标取样数据的取样数据库。遗产取样数据被数据转化器转化成目标取样数据,用于目标系统的开发和测试。

阶段3:增量地移植除数据外的所有遗产组件到目标体系结构。

阶段4:逐步移植遗产数据到目标系统,并且培训用户使用目标系统。对数据移植进行更加详细的讨论和分析参考文献[6]。

阶段5:交割到目标系统。蝴蝶方法的最后一个阶段是交割阶段,一旦目标系统被建立和所有的遗产数据被移植,新的系统就准备运行。

3 基于语言转换工具的移植

3.1 基本思想

前面介绍的一些移植方法和策略有一个共同特征就是遗产信息系统的理解 and 目标系统的开发是两个相互依赖的过程,目标系统的开发必须建立在对遗产信息系统的理解之上。在理解遗产信息系统的过程中,有许多工具可以支持^[5],但是系统专家的作用比工具更重要,目标系统的开发也要依赖于优秀的系统设计师和程序员,可见人的因素在这两个过程中占有重要的份量。同时,除去测试的时间以外,理解遗产信息系统和开发目标系统占去了几乎所有时间。如果能够减少理解和开发这2个过程的时间,那么将能大幅地缩短移植项目的工期,节省大量的人力物力资源,把更多的精力放在测试工作上。

因此我们提出了一种基于语言转换工具的移植方法。移植过程包括三个主要步骤:理解,开发和测试,其中测试工作是必须的,不可能大幅度地缩短时间。但是,理解和开发是有紧密联系的,理解是通过分析某种程序语言来抽取功能,开发是根据抽取的功能用程序设计语言来实现。在以前的方法中,理解和开发是独立且按次序人工进行的。如果能够找出语言之间的对应关系,就可以通过工具自动化地进行语言转换,从而节省大量的移植时间和费用,这就是本文提出的基于语言转换工具的移植方法的基本思想。

从理论上讲,只要有顺序、转移和循环三种基本控制结构

就能表达任何可计算功能,这个结论在上世纪50年代就被证明过,这就从理论上证明了在程序设计语言之间进行转换的可行性。因为功能包含在代码中,代码是功能的载体,自动转换了代码也就自动转换了功能。这就是基于语言转换工具的移植方法的基本思想,其中语言转换工具是此方法的核心。

目前市场上也出现了很多语言转换产品,能有效地辅助移植,但是这些产品不是很完善,还有许多问题没有解决好。

3.2 基于语言转换工具的移植方法

基于语言转换工具的移植方法侧重于代码的自动化移植,同时吸取了蝴蝶方法中数据移植的优点,并且对其做了改进。整个过程分为5个阶段,每个阶段由多个相对独立的活动组成,最终目标是目标系统的正常运行。

阶段1. 可行性论证和准备工作

1.1 明确移植要求,明确用户需求,明确移植的目标。

1.2 决定目标系统的硬件平台、软件平台和体系结构。

1.3 选择语言转换工具(由第三方提供或者自行开发),以及其它辅助工具。

阶段2. 遗产信息系统的代码移植 此阶段致力于遗产信息系统的代码移植,使用语言转换工具将遗产系统理解和目标系统的开发同时进行,实现自动化代码移植,这是此方法的核心阶段。此阶段由如下具体活动构成:

2.1 预处理: 遗产系统的代码,经过多年的维护、修改等,可能已经变得面目全非,代码风格的一致性遭到破坏。对于原编程语言中某些特定的语法,转换工具可能不能处理,需要先改写遗产代码。另外,每个遗产信息系统的语言有其自身的特点,已有的转换工具未必能直接使用,需要调整来满足当前语言的特点。基于上述原因,为了得到高质量的目标代码,需要对遗产代码和工具进行预处理。

2.2 语言转换: 使用语言转换工具进行语言代码之间的转换,这阶段的主要工作就是配置和操作语言转换工具。

2.3 编译目标代码: 转换生成的目标代码,不一定可以直接运行,甚至可能会存在语法错误,这取决于转换工具的质量。如果在转换生成的目标代码中发现语法错误就需要调整和改进工具,重新进行转换,直到目标代码在语法级上是正确的。如果编译通过,则转到活动2.5。

2.4 改进工具: 在前一阶段发现的错误,如果是转换工具不完善导致的,那么就需要改进工具,而不是修改目标代码,尽可能地让工具来解决问题。修改完成后,转到活动2.2,重新进行语言转换。

2.5 后处理和部署目标系统: 对于通过编译的目标代码,如果还存在问题,并且这些问题通过改进工具无法解决,那就只能人工修改目标代码,这就是后处理工作。如果上述工作都已经完成,那么就可以把目标系统部署到目标平台上,准备下一阶段的测试工作。

阶段2对应于其它方法中的理解遗产信息系统和开发目标系统两个阶段,通过使用转换工具,可以较大地提高移植效率,并且用自动化的方法保证移植质量。

阶段3. 测试和培训 程序移植工作完成后,使用取样数据库就可以进行测试和培训工作了。测试可以使用一般软件工程的测试策略和方法进行,培训的目的是使用户熟练操作目标系统。

阶段4. 遗产数据的移植 此阶段借鉴蝴蝶方法数据移植的思想,并对其进行了优化。准备工作是开发一个数据转化器(Data Transfer, DT),用于把遗产数据库的数据复制到目

标数据库中,开发一个操作分配器(Manipulation Allocator, MA),用于记录在数据移植过程中遗产信息系统产生的新操作,并保存到临时数据库中。此阶段的活动主要如下:

4.1 把操作分配器 MA,数据转化器 DT 植入到遗产信息系统。

4.2 产生临时数据库(TempStore, TS)TS₁,然后设置遗产数据库 TS₀为只读。

4.3 移植 TS₀中所有的数据到目标数据库,此过程中的操作结果被 MA 重定向到 TS₁,直到 TS₀移植完成。

4.4 产生临时数据库 TS₂,然后设置 TS₁为只读。

4.5 移植 TS₁中所有的数据到目标数据库,此过程中的操作结果被 MA 重定向到 TS₂,直到 TS₁移植完成。

4.6 对于 TS_n和 TS_{n+1},重复4.4和4.5,直到满足终止条件,即 SIZE(TS_{n+1}) < ε。

4.7 把 TS_{n+1}中的数据移植到目标系统。

整个过程如图4所示。

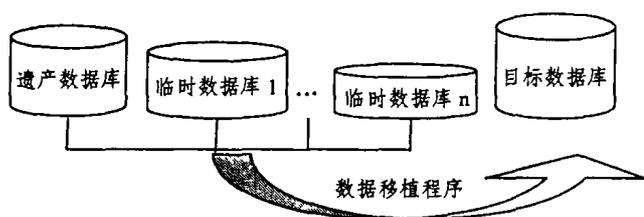


图4 遗产数据的移植

遗产数据移植的基本思想同蝴蝶方法,但是蝴蝶方法在移植每个临时数据库时,把新的操作结果保存到新的临时数据库中,而这里我们只保存新的操作而不是操作结果。一方面,可以减少临时数据库的数据量,加快移植速度,另一方面,可以充分利用遗产数据库的事务和日志功能(如果有的话),保证数据移植的稳定。

阶段5.交割 此阶段是方法的最后一个阶段,代码、数据移植和测试、用户培训工作完成后,遗产信息系统就可以被关闭,新的系统投入正式使用。

3.3 方法的总结和讨论

蝴蝶方法中没有给出计算数据移植时间的具体方法。本文给出计算公式如下,假定 u 是数据转化器转化数据的速率, v 是操作分配器构造新临时数据库的速率, D 是遗产数据库的初始大小,所需的数据移植时间 T ,则

$$T = \frac{D}{u-v} \quad (1)$$

如果 $v=0$,所有的数据将被一次性地移植。如果 $v \geq u$,那么移植过程就永远不会结束。为此,本文提出以下措施,可以保证 $u > v$,确保数据移植能顺利完成。

(1)从软件和硬件两方面,提高数据转化工具的速度,确保转化速度大于操作增长速度。

(2)选择在操作数量比较少的时段,比如深夜,进行数据移植。

(3)在某段时间内,通过组织内的协调工作或行政干预,减少操作数量。

一般来说,目标系统的硬件和软件性能比遗产信息系统强很多,数据转化速度比新操作增长速度慢的情况很少会发生。

基于语言转换工具的移植方法的最大优点就是将遗产信

息系统的理解和新系统的开发由转换工具自动化地完成,不需要理解复杂的遗产信息系统,也不需要设计、编码生成目标系统,并且用自动化的方式保证了移植质量。对于缺少文档的遗产信息系统,此方法尤其适用,在生成目标系统的同时,还能自动生成相应的文档和程序注释。

虽然方法有以上优点,但是也存在一些未解决的问题,比如,自动化的转换增加了测试的难度,因为目标代码是工具自动生成的,不是程序员手工编写的。测试中发现的 Bug,将很难定位和修改。当用户有新的需求时,需要增加新的功能,新功能与原有功能的集成也会变得困难,新功能需要程序员来实现,原有功能则是工具生成的,可能会导致设计和代码风格不一致而难于集成和维护。这些问题的解决有待于进一步的研究。

最后,在实际的工作中,应该根据遗产信息系统的特点和实际情况来选择最合适的移植方法。

结束语 迄今为止,研究者提出的5种移植方法各有其自身的特点和侧重点,比如 Chicken Little 策略提出了增量移植思想,蝴蝶方法主要解决了数据移植的问题。本文总结了这些方法的优点,提出了基于语言转换工具的移植方法,给出了较完整的移植过程模型,侧重于遗产信息系统的代码移植,并讨论工具的选择标准和移植质量的评价标准,在数据移植方面,优化了蝴蝶方法。我们基于该方法将日本某 IT 公司的物流系统入库管理子系统从 Cobol 语言移植到 Java,比使用其他移植方法缩短了约一半工期,减少了成本,同时也取得较好的效果。

遗产信息系统移植研究直到上世纪末才受到重视,目前的研究工作还是很不够的。移植是个很复杂的过程,涉及到软件工程的各个方面,尚需要进行更深入的研究。例如,提出更多的完善的移植过程模型;继续细化底层的移植活动和工作流;开发支持移植工作的各种辅助工具;开发高质量的语言转换工具等。

参考文献

- 1 Brodie M, Stonebraker M. *Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach*. Morgan Kaufmann Publishers, Inc. USA, 1995
- 2 Brodie M, Stonebraker M. DARWIN: On the Incremental Migration of Legacy Information Systems. [Technical Report TR-022-10-92-165 GTE Labs Inc]. 1993
- 3 Wu B, Lawless D, Bisbal J, et al. The Butterfly Methodology: A Gateway-free Approach for Migrating Legacy Information Systems. In: Proc. 3rd IEEE Conf. on Engineering of Complex Computer Systems (ICECCS '97), Sep. 1997. 200~205
- 4 Menhoudj K, Ou-Halima M. Migrating Data-Oriented Applications to a Relational Database Management System. In: Proc. of the Intl. Workshop on Advances in Databases and Information Systems (ADBIS'96), Moscow, Sept. 1996
- 5 Schmidt J. A Practical Implementation with Migration Tools. Chapter 10 of 'Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach'. By M. Brodie and M. Stonebraker, Morgan Kaufmann Publishers Inc. 1995
- 6 Wu B, Lawless D, Bisbal J, et al. Migrating Legacy Systems: From a Caterpillar to a Butterfly. Trinity process of legacy system migration with the main focus. [College Technical report]. Jan. 1997