文章编号: 1007- 144X(2005)01- 0102- 05

软件体系结构设计方法的研究及应用

周琼朔

(上海师范大学 机电学院, 上海 201418)

摘 要: 提出了一种体系结构设计方法, 把需求细分为功能需求和体系结构需求, 迭代增量的构建软件体系结构, 并采用开发者熟悉的 UML 机制来描述软件体系结构, 最后结合 C2 软件体系结构风格在 Microsoft 商务参考体系结构(B2C 电子商务)中进行了运用。

关键词: 体系结构; 构件; 设计方法

中图法分类号: TP311 文献标识码: A

1 引 言

所有软件开发方法都要解决从需求到实现之间的转换问题。为了提高软件需求和软件设计的质量, 软件工程界提出了需求分析工程技术和各种软件建模技术。但是, 在需求与设计之间仍存在一条很难逾越的鸿沟, 即缺乏能够反映做决策的中间过程, 从而很难有效地将需求转换为相应的设计。为此, 软件体系结构概念应运而生, 并试图在软件需求与软件设计之间架起一座桥梁, 着重解决软件系统的结构和需求向实现平坦地过渡的问题。本质上, 软件体系结构是对软件需求的一种抽象解决方案。这样, 在引入了体系结构之后, 软件系统的开发变为 "问题定义→软件需求→软件体系结构→软件设计→软件实现 "的过程。

2 已有软件体系结构设计方法的不足

为了获取对体系结构设计的抽象, 人们已经提出了许多方法。把这些体系结构设计方法分类为: 工件驱动的方法; 用例驱动的方法; 模式驱动的方法; 领域驱动的方法[1]。

工件驱动的体系结构设计方法从方法的工件描述中提取体系结构描述。工件驱动的体系结构设计方法的例子包括广为流行的面向对象分析和设计方法 OMT 和 OAD。在 OMT 中, 体系结构抽象被表示为从需求规格说明导出的分组类。在体系结构开发方面, 该方法存在着如下问题: ①文本形式的系统需求含混不清, 且不够精确和完整, 因此, 它作为提取体系结构抽象的来源作用不够。②子系统的语义过于简单, 难以作为体系结构构件。③对子系统的组合支持不足。

用例驱动的体系结构设计方法主要从用例导出体系结构抽象。一个用例, 是指系统进行的一个活动系列, 它为参与者提供一些结果值, 参与者通过用例使用系统。参与者和用例共同构成了用例模型。用例模型的目的是作为系统的系统预期功能及其环境的模型, 并在客户和开发者之间起到合约的作用。在使用这一方法标识体系结构抽象时, 必须处理下面几个问题: ①对于如何选择与体系结构相关的用例没有提供系统的支持。②用例没有为体系结构抽象提供坚实的基础。③包的语义过于简单, 难以作为体系结构构件。

在领域驱动的体系结构设计方法中, 体系结构抽象是从领域模型导出的。领域模型可以有多种不同的表示方法, 比如类、实体关系图、框架、语义网络和规则等。与此相应, 领域分析的方法也有多种。由于对 "领域 "这一术语有着不同的解释, 领域驱动的体系结构设计方法也有多种, 其中存在着一些问题, 如问题领域分析在导出体系结构抽象方面效果较差; 解决方案领域分析不够充分。

体系结构模式类似于设计模式, 但它关心的是更粗粒度的系统结构及其交互。实际上, 它也是

收稿日期: 2004- 10- 23.

作者简介: 周琼朔(1982-), 女, 河南许昌人, 上海师范大学机电学院本科生.

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.n

体系结构风格的另一种名称。体系结构设计模式是体系结构层次的一种抽象表示。模式驱动的体系结构设计方法是从模式导出体系结构抽象。该方法存在的问题主要有: ①在处理范围广泛的体系结构问题时, 模式库可能不够充足。②对模式的选择仅依靠通用知识和软件工程师的经验。③对于模式的组合没有提供很好的支持。

3　软件体系结构设计方法

3. 1　体系结构需求和软件体系结构

按照 Bass L 等人的观点, 基于体系结构的开发就是将以软件体系结构为核心的方法应用于软件产品的开发中[2]。在以体系结构为中心的系统开发过程中, 除了开发出一组功能需求, 还要开发一组体系结构需求。体系结构需求由开发组织创建, 并受技术环境和体系结构设计师个人经验的影响。体系结构需求可能是对体系结构变更点的列举和对性能与可靠性需求的列举, 可以是系统可能要进行的更改的类型的列举, 还可以是一些性能需求。

设计过程的前提之一是认为: 系统的体系结构需求和行为需求是同样重要的。两种需求都用场景或用例的形式具体表示。有多种不同类型的场景可用。普通的用例被用于单个产品的行为需求, 抽象场景被用于产品线的行为需求。

3. 2　用例和软件体系结构

用例和软件体系结构之间存在着紧密的关系[3, 4]。一方面, 软件体系结构受到所希望系统支持的用例的影响, 用例驱动软件体系结构的建立, 软件体系结构受到体系结构需求的影响。另一方面, 用例也受到软件体系结构的影响。软件体系结构是在细化阶段的迭代过程中被创建的。开始先确定软件体系结构的高层设计, 例如一个分层体系结构, 然后在第 1 次迭代的几次构造中逐步确立该体系结构。在细化阶段的最后, 可以得到一个相对稳定的软件体系结构。当建立了一个稳定的软件体系结构后, 就可以通过在构造阶段实现其余的用例来实现系统的全部功能。在开发构造阶段中实现的用例, 主要是以客户和用户需求作为输入。但这些用例也会受到细化阶段所确定的体系结构的影响。在捕获新的用例时, 可以利用对已经存在的体系结构的知识来更好地完成工作。在评估每个所选用例的价值和成本时, 也是根据现存的体系结构进行的。通过使用例与体系结构保持一致, 便可以利用现存的资源有效地创建新的用例、子系统和类。

3. 3　软件体系结构风格和软件体系结构

软件体系结构风格是描述某一特定应用领域中系统组织方式的惯用模式。它反映了领域中众多系统所共有的结构和语义特性, 并指导如何将各个模块和子系统有效地组织成一个完整的系统。按这种方式理解, 软件体系结构风格定义了用于描述系统的术语表和一组指导构建系统的规则。

尽管可以把某一软件的体系结构看成是体系结构元素的集合, 而且构成系统的这些体系结构元素都是个体特殊的。但是好的体系结构师更倾向于复用已经构建好的一系列体系结构组织, 即体系结构风格, 使用体系结构风格能带来很多重要优势[5]: ①促进设计重用; ②带来巨大的代码重用; ③采用例行的结构, 将使系统组成更易于被其他人理解; ④使用标准化的风格有利于系统的互操作性; ⑤通过约束设计空间, 采用体系结构风格的设计, 通常允许专门的、此风格所特有的体系结构分析; ⑥一般可以提供特定风格的可视化。

3. 4　迭代开发和软件体系结构

体系结构团队将经历一个细化高层软件体系结构的过程。这一过程通常要经过若干次迭代才能完成。随着开发过程的进行, 将会找出必要的变化, 软件体系结构也会随之变化[3]。这将作为一种规格说明交付给开发团队, 供他们在自己的相关层次的设计中使用。尽管每个层次生产的产品类型相似, 但其内容和详细程度是不同的。此外, 一些来自软件体系结构描述的产品应该能够被后继的开发团队直接使用, 而不需要做进一步的详细解释。

迭代开发的生命周期是基于对一个系统进行连续的扩充和精化, 这个过程要经历若干个开发周期, 每个周期都要经历分析、设计、实现和测试阶段。在每个开发周期中通过增加新的功能使系统得以扩充。在经过一个初步的计划和细化阶段后, 开发进入了由一系列开发周期组成的系统构造阶段。每个开发周期只针对比较小的一部分需求, 它也要经历分析、设计、构造和测试等活动。每个开发周期完成后, 系统都获得一定的扩充。采用迭代的开发方法不会因为一个开发过程太复杂而使人无从下手, 另外, 在开发过程的早期就能够获得反馈信息, 因为每个周期只快速实现系统的一部分。

3. 5　软件体系结构设计过程

根据所述软件体系结构和体系结构需求、用

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.n

例及软件体系结构风格的相互关系, 可以通过迭代增量的方式来构建软件体系结构。图 1 给出了笔者提出的软件体系结构设计方法的概念模型, 该方法主要由分析、描述、选择、构造和组合 5 个阶段组成。

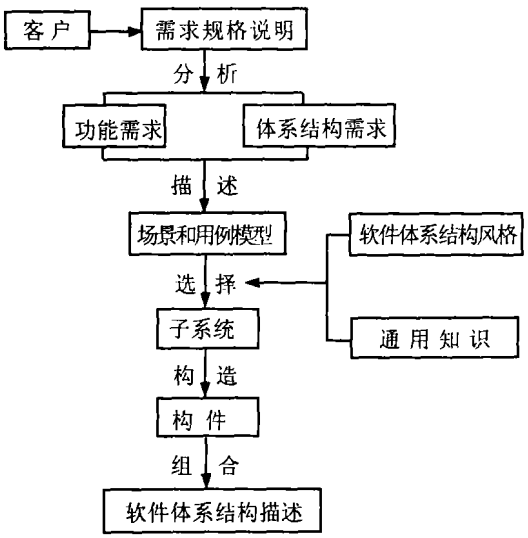

图 1 软件体系结构设计方法的概念模型

分析表示在需求规格说明书的基础上, 对文本形式的需求规格说明书进行分析, 开发出一组功能需求和一组体系结构需求。体系结构需求和功能需求用用例和场景的形式来描述。

因此, 在接下来的描述中, 列出系统的功能后, 就定义系统的边界, 以便识别出参与者和用例, 并用高层格式写出所有用例, 根据它们对体系结构的影响程度, 将它们分成主要的、次要的和可任选的 3 类。

选择就是从体系结构需求和功能需求开发子系统。标识系统功能的用例自动成为侯选的子系统, 并从体系结构需求导出其他的侯选子系统。对于每个体系结构需求, 可以列举能够满足该需求的可能的体系结构选项。例如, 如果需求允许改变操作系统, 那么满足该需求的体系结构选项之一是一个虚拟操作系统适配器。一些体系结构需求可能有多个可能的选项, 而另一些可能只有一个选项。这一列举来自于体系结构风格和体系结构设计师的经验。在这一可能的选项列表中进行选择, 使得所有的体系结构需求都能够被满足, 而且列表的选项要相互一致并在总量上最小。这里总量最小指的是: 如果一个选项能够满足 2 个体系结构需求, 那么就应当选择这个选项, 而不是选择 2 个分别满足一个需求的选项。选项越少, 构件也就越少, 在实现和维护时所要做的工作也就越少, 而且一般来说, 出错的机会也就越少。选择受到软件体结构风格和通用知识(指软件工程师的一般背景和经验) 的支持。

在迭代的第一个周期选取与最关键、影响最大和最具有开发风险的用例相对应的子系统, 对这些子系统进行分析。它们可能被归类为实际的子系统, 或更大子系统的构件。然后对它们进行构造, 它表示组建并发构造。组建并发构造是在参考子系统的情况下对分布单元和并行单元进行分析来完成的。每个子系统可能被分布到多个物理接点上, 在这种情况下, 分布单元被标识为子系统的构件。通过分析子系统的线程和线程的同步来对并行单元进行标识。如果所有的子系统驻留在单个处理器上, 线程被称作是 “虚拟线程 ”, 因为它们标识并行元素。当考虑物理构造时, 有可能确定虚拟线程转换为物理线程的位置, 以及从这一转换派生出的必要的网络消息。

在这一设计过程结束时, 子系统和构件已经被确定下来。然后对此步骤进行验证, 并对子系统进行组合。当然, 验证的步骤可能导致重新做出决定, 所以决定和验证之间的实际过程可能是高度迭代的。质量场景通常是首选的验证机制, 通过质量场景对所提交的构造进行检查, 来判定当前的设计层次是否达到了场景的要求。如果答案是肯定的, 那么可以继续进行设计的下一步求精; 否则, 就要重新考虑当前求精层次的设计。

组合就是定义构件的端口, 并将构件通过连接件组装起来形成软件体系结构。构件作为一个封装的实体, 隐藏了具体实现, 只通过接口提供服务, 而构件的接口由一组端口组成, 每个端口表示了构件和外部环境的交互点。这些端口的开发主要采用对相应用例的分析技术。构件之间是相对独立的, 对于构件而言, 连接件是构件的粘合剂, 是构件交互的实现。连接件作为建模软件体系结构的主要实体, 同样也有接口。连接件的接口由一组角色组成, 连接的每个角色定义了该连接表示的交互的参与者。连接件的接口由它与所连构件之间的一组交互点构成, 这些交互点被称为角色。角色代表了参与连接的构件的作用和地位, 并体现了连接所具有的方向性。

在接下来的迭代周期中, 从剩余的子系统中挑出那些对体系结构影响最大的, 依次进行上述的 5 个步骤, 逐步建立更加完美的体系结构。依次类推, 在每一次迭代中, 都选取并实现一组用例和场景来确认体系结构。如果必要, 再对体系结构进

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.n

行改进。随着每次迭代的进行, 还在所选用例和场景的基础上, 进一步实现体系结构的专门应用部分, 逐渐完善体系结构。

4　在 B2C 电子商务中的应用

笔者采用上述体系结构设计方法, 结合 C2 软件体系结构风格, 对 Microsoft 商务参考体系结构(B2C 电子商务) 进行了运用, 并采用类和对象策略描述体系结构, 用 UML 类表示构件类型, 用 UML 对象表示构件实例。使用 UML 接口来描述端口。把连接件类型也建模为类, 并把连接件实例建模为对象体。其接口也采用 UML 接口来描述。构件和连接件可以采用不同的版型加以区分(《omponent》表示构件、《onnector》表示连接件)。限于篇幅笔者只给出最后的软件体系结构图(为图形简单起见省略了具体属性) [6~8], 如图 2 所示。

5　结束语

笔者提出的体系结构设计方法把需求细分为与系统功能相关的功能需求和与体系结构相关的体系结构需求。采用迭代增量的软件体系结构构建方法, 对于软件体系结构的构建具有很好的指导作用, 它符合人们的认识和思维方式。并采用开发者熟悉的 UML 机制来描述软件体系结构, 使具体事项之间的映射更为接近, 并能得到商业工具的支持。实践证明, 这种方法相对于传统的体系结构构造方法, 具有更高的效率。

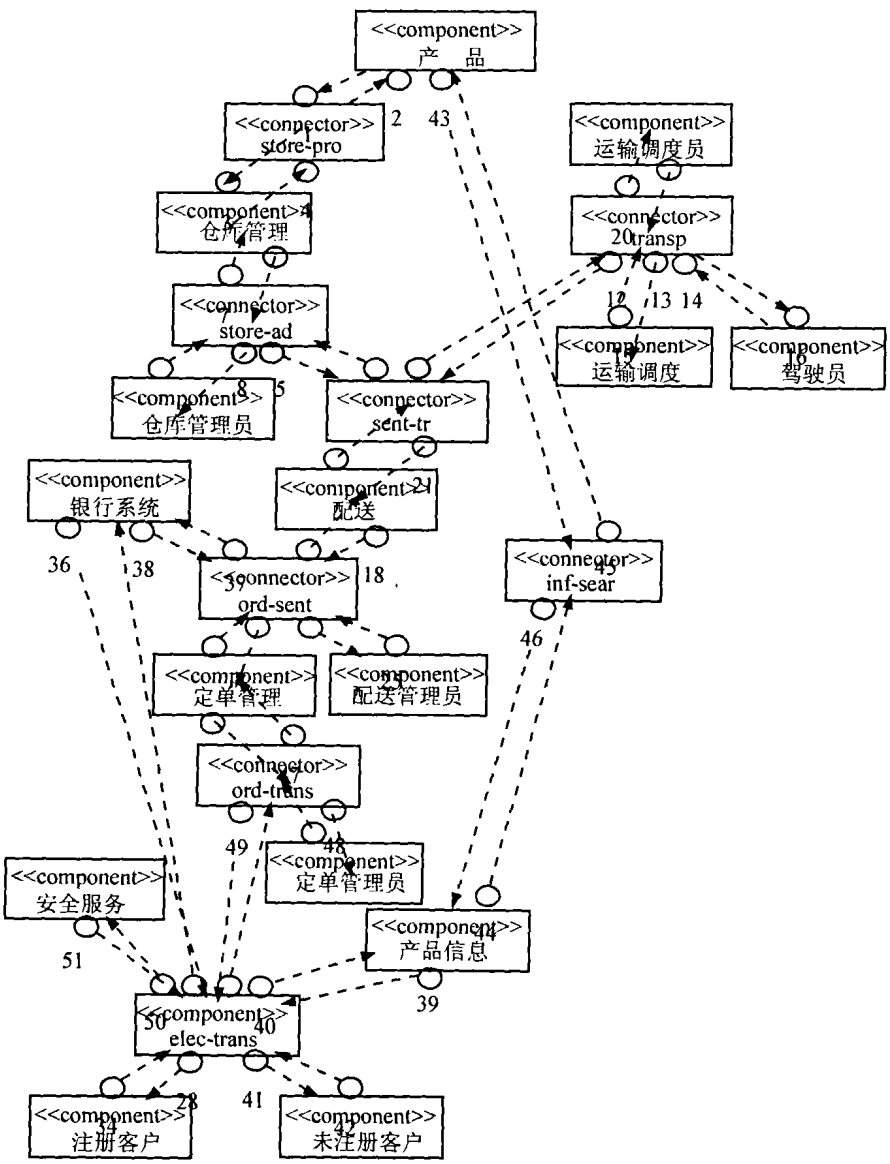

图 2　B2C 电子商务软件体系结构图

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.n

但是,这种开发方法需要一种新的管理项目的方式。另外,尽管使用类和对象是在UML中表示体系结构构件和实例的一个合理方法,一些重要的语义差异仍然存在。在UML中,类和类型应当进行同样的计算,并具有同样的内部结构。但在ADL中,构件的实例可以有不同的行为和结构。由于构件的类型/实例被表示成类/对象,热连接件的类型/实例也被表示成类/对象,这样,二者在图形表示上难以区别(虽然可以用不同版型进行区别,但效果并不明显)。

参考文献

[1] 冯 冲,江 贺,冯静芳. 软件体系结构理论与实践[M]. 北京: 人民邮电出版社,2004.

[2] Bass L, Kazman R. Architecture- Based Development[R]. Technical Report CMU/SEI- 99- TR- 007, CMU: 1999.

[3] Anthony J G R. 大型软件体系结构: 使用UML时间指南[M]. 叶俊民,汪望珠译. 北京: 电子工业出版,2004.

[4] Jacobson I, Booch G, Rumbaugh J. The Unified Software Development Process[M]. Addison- Wesley, 1999.

[5] Buschmann F, Meunier R, Rohnert H, Sommerlad P, Stal M. Pattern- Oriented Software Architecture: A System of Patterns[M]. New York: John Wiley & Sons, 1999.

[6] Medvidovic N, Rosenblum D S. Assessing the Suitability of a Standard Design Method for Modeling Software Architectures[A]. Proceedings of the First IFIP Working Conference on Sotware Architecture San Antonio, TX, Feb[C]. 1999.

[7] Medvidovic O, Robbins T. Using Object- oriented Typing to Support Architectural Design in the C2 Style[A]. In Proceedings of ACM SIGSOFT '96: Fourth Symposium on the Foundations of Software Engineering (FSE4)[C]. San Francisco: 1996. 24- 32.

[8] Garlan D, Shang W C, Kompanek A J. Reconciling the Needs of Architectural Description with Object- Modeling Notations[J]. Sci. of Computer Programming, 2002(44): 23- 49.

A Designing Method of Software Architecture and Its Application

Zhou Qiongshuo

Abstract: A software architecture designing method is put forward. Demands are classified as the function demand and the architecture demand. The UML mechanism that developers are familiar with is used to describe this architecture. Incorporating with C2 software architecture style, this method is put into application in the Microsoft commercial reference architecture.

Key words: architecture; component; designing method

Zhou Qiongshuo: Undergraduate; School of Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 201418, China.

[编辑: 王志全]

© 1994-2011 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.n