维普资讯 http://www.cqvip.com

计算机科学1999Vol.26№ 8

基于软件体系结构的软件设计及构造*)

Software Designing and Compositing Based on Software Architecture

胡 华[1,2] 高 济[2] 何志均[2]

(杭州商学院计算机与信息工程系 杭州 310035)[1]

(浙江大学计算机科学与工程学系 杭州 310027)[2]

Abstract Based on the character and trend of current software developing, this paper proposed a new method for applied software system designing and composting—MSABAD method. People with computer applying background can use MSABAD at ease because MSABAD consorts to their thinking custom. Using ideas of Software Architecture as the core, MSABAD integrates effectly many techniques of Object-Oriented, Software Architecture and Software Design Pattern. MSABAD can not only hepl to smooth the transition of different software developing phases, but also make the action of software developing more effective, reusable, intelligible, maintainable and evolvable.

Keywords Object-oriented, Software architecture, Software design pattern, Integration, Method

1 引言

软件体系结构和软件设计模式是九十年代兴起的两种软件系统研究和设计新技术。这两种技术的典型特点就是将软件设计和开发的关注重点从传统软件设计开发方法的算法和数据结构转向了对整个软件系统的组织结构和系统性能表现[1,3,12,15,16,21]。其中，软件体系结构研究注重的是构成软件系统的各有机组成成分及其关联作用和语义模式，通过对软件系统的组成成分及其相互之间的关联作用和语义模式进行分析和研究，软件体系结构期望通过加深对系统构造的理解来提高有关的软件工作人员的系统设计和系统分析能力，从而在系统组织、结构重用、运行模式、系统分析和系统维护等方面降低软件设计和开发的成本，促进软件系统生产的效率提高。软件设计模式侧重的是描述软件设计表现的特定方面、设计的应用环境、相应的设计约束以及它们对整个软件系统所造成的影响。软件设计模式关注的不是具体的数据结构、对象实体，也不是关于一个复杂、完整的具体应用软件的设计，它是关于构成系统几个相互作用的构件的描述。这样的描述可以用来解决在不同应用环境下具有某种共性的软件设计问题。

我们针对大型复杂软件系统开发过程中所具有的本质困难和属性特点[25]，在综合集成当今软件工程领域内有关先进技术和方法的基础上，提出了一种基于软件体系结构的软件系统的分析和设计开发方法(MSABAD-Method of Software Architecture Based Analysis and Design)。从方法的应用对象角度看，MSABAD方法是一种面向有一定计算机应用开发基础的软件设计人员的软件集成设计开发方法，该方法从软件系统的整体性能需求入手，综合集成了面向对象技术、软件体系结构和软件设计模式等先进的软件开发技术，通过体系结构描述模型和软件设计模式管理，不但逐步实现了由问题领域向分析文档、由分析文档向系统设计实现的平稳过渡，而且使得软件系统的构造开发活动在开发效率、可重用性、可理解性、可维护性和易于演化等方面有了很大的提高。

2 关键问题

MSABAD有三个主要的特点：方法与概念的综

*)本研究得到国家自然科学基金资助。胡华 博士，副教授，主要研究领域为：人工智能，软件工程。

维普资讯 http://www.cqvip.com

合集成、开放精确的体系描述语言和基于领域模型的模式。对于现代软件系统大型化复杂化的发展趋势而言，仅仅局限于一种开发方法的概念与技术会给具体的开发活动和工作带来很多问题。MSABAD方法在对不同的软件开发技术与方法进行提炼与集成的基础上，通过开发和使用支持演化与改进的软件体系结构描述语言为进行大规模的软件开发提出了一条新的思路和方法。

2.1 阶段模型的选择

在MSABAD方法中，软件系统的设计开发活动划分为：软件系统特性分析、软件体系结构的设计建立与确认以及面向具体问题应用领域的系统应用框架构建等三个不同的设计开发阶段。

在软件系统特性分析阶段、其分析处理工作特点与传统软件设计开发方法中的系统分析工作有着许多相似和密切联系之处。因此、我们借鉴和使用了许多传统软件开发方法中系统分析的技术与手段。尤其是面向对象的系统分析技术，因其对现实客观世界的自然精确描述特点而成为MSABAD方法中的系统特性分析阶段的主要参考和基础。但是由于MSABAD方法的核心又是软件体系结构的抽象性组织结构，为使MSABAD方法的其它开发设计阶段不受到面向对象技术中的许多独特技术特点的制约与限制，我们在采用面向对象的分析处理技术时又增加了一些针对软件体系结构方法特点的限制和要求，例如，在具体使用有关的面向对象方法进行处理时，工作人员应尽量避免使用继承等面向对象发表方法中容易受到指责和限制的特殊处理组织机制。

对于软件体系结构的设计建立与确认阶段的处理活动，其目的是要建立起与具体应用领域问题相对独立的抽象软件体系结构，作为一种与具体应用问题领域独立的抽象软件系统组织结构的描述方法、软件体系结构不仅要在忽略具体实现细节和使用背景的基础上，给出软件系统中的有关构件和连接件的刻画与描述，而且还要给出整个系统中有关实体的定义规则、处理机制和运行策略，现今的软件体系结构设计与描述从本质上讲还是一种典型的依赖于专家经验性质的软件处理过程和活动。对于有关体系结构或模式的如何选择和采用，即使是具体实施处理的有关工作人员也很难清晰地解释为何会有这样的决定或决策。从软件体系结构方法的这一特点和角度出发，我们对MSABAD方法中有关的建立应用独立软件体系结构的技术和方法综合考虑了以下几个方面的细节问题处理：

(1)数据组织和存取的方法与机制。对相应软件系统中有关数据的组织方法、存取策略和限制条件等。

(2)系统的控制策略。对相应软件系统中的各组成成分之间的交互协调处理策略给出描述和限制。

(3)系统中的任务调度策略。对软件系统中抽象功能实体的处理和活动进行设计与描述。

(4)系统中的时间控制策略。对软件系统中有关操作和控制的时间因素进行组织与描述。

(5)软件体系结构的应用独立策略。为了使所建立的软件体系结构具有与具体的应用问题领域相互独立的特性，MSABAD方法在提供以上有关结构和策略的同时，给出其相应的软件体系结构模型而不给出它所提供的有关结构与实体的具体应用细节与使用。

最后、对于面向具体问题应用领域的系统应用框架构建处理，由于软件体系结构关注和处理的重点是软件系统中与具体应用问题相独立的抽象组织概念，因此我们在MSABAD方法中，通过有效合理地使用软件设计模式的技术和概念，利用系统特性分析和与应用独立的软件体系结构工作结果来实现从与应用领域无关的抽象软件体系结构向面向问题处理领域的软件设计框架进行平滑的设计转换，并且在达到软件设计知识和模式的重用效果基础上，有效地提高有关软件生产的生产效率。

2.2 设计开发方法的集成与衔接

现代软件系统日渐大型化、复杂化和分布化的综合发展趋势使得现有的各种软件设计和开发方法在进行具体的实践和应用时都存在着不同程度的不足和局限。针对软件开发设计活动中的这一挑战，在进行实际的软件开发设计处理时，很多软件系统设计工作者采用了一种将不同的软件开发设计方法进行混合组织集成的解决方案[14,27,28]。

考虑到现有的各种软件开发集成方法所面临的问题和困难，在MSABAD方法中，我们采用了一种不同于传统方法集成的集成思路，即：不是象传统方法那样对软件开发方法在所有阶段和活动中所涉及的不同概念和术语的对等混合集成，而是试图在以一种较为抽象通用的与应用领域无关的方法为核心的基础上，通过在不同阶段采用具有不同优势的方法的衔接集成形式这一软件开发设计方法集成框架，将面向对象分析方法、软件体系结构和软件设计模式技术进行纵向的综合衔接集成。这种纵向上的

维普资讯 http://www.cqvip.com

衔接集成方法不仅避免了大量不同方法的不同概念的选择和冲突消解工作之困难，而且可以比较自然地利用和发挥有关方法的处理优势。

此外，通过对软件设计开发方法的组织结构和特点进行抽象和分析，可以将软件设计开发方法抽象成由多个构件相互作用组成的实体关系模型，并将该模型用如图1所示的简化的实体关系图表达出来。

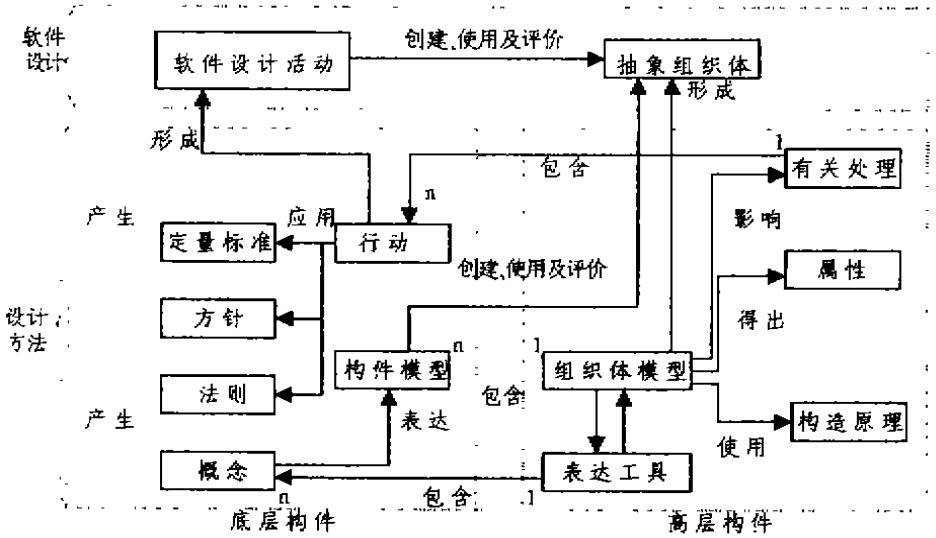

图1 软件设计开发方法模型

在图1中，上层虚线框代表由软件设计开发方法指导并得出的软件设计活动和抽象的开发组织体；下层两个虚线框合起来代表具体的软件设计开发方法结构。软件设计开发活动代表具体的操作活动，开发抽象组织体则代表了一组较大粒度的可以独立开发活动的原理、规范及步骤组成的构件体等。例如，系统分析和系统设计等都是包含可以被独立描述与使用的原理与规范组织指导的相对独立一组抽象的大粒度的软件开发活动，而软件体系结构和软件设计模式以及它们中的一些中间部件则包含可以被独立描述与使用的原理与规范组织指导的一些中等粒度的开发活动，开发活动与开发组织体之间的关系是设计开发活动可以建立、使用和评价软件开发抽象组织体。对于下层图所代表的软件开发方法，又由两个虚线框将其划分成高层构件和低层构件两大部分。其中，高层构件由一些与抽象组织体有关的抽象构件组成；而低层构件与高层构件不同，它们并不代表一些完整的实体，其存在的目的是对具体的开发方法中的抽象组织体的组成及其构造的具体操作步骤进行说明与指导。

根据图1可以看出：不同的软件开发方法的集成可以被看成是在以抽象组织体为主的高层构件指导下的集成。这种集成包含了开发模型、开发原理、开发概念、开发术语以及开发处理过程等各个不同方面的全方位的复杂集成。在MSABAD方法中我们所考虑的是以软件体系结构为核心在各个阶段采用不同方法针对软件体系结构的特点来增加衔接抽象组织体形式的不同方法这一包含衔接集成，这种包含衔接集成方法的简洁性特点使得我们在MSABAD方法中既能够充分利用各种不同开发方法在不同开发阶段的独特优势，同时又避免了对等混合集成各种不同方法的有关概念、原理和处理时所需要的巨大工作量。

2.3 软件体系结构的描述与表达

作为MSABAD软件开发和设计方法的核心，软件体系结构的抽象表达和描述是整个方法成功的关键所在。

对于软件体系结构的表达和描述，已经存在很多的表现形式和方法。如：图文法理论、模块内连接语言、软构件描述法和软件体系结构描述语言等[2,10,30~34]。由于软件体系结构系统描述语言在吸收了传统程序设计中的语义严格精确的特点基础上，针对软件体系结构的整体性和抽象性特点，定义和确定适合于软件体系结构表达与描述的有关抽象元素，因此，软件体系结构描述语言是当前软件开发和设计方法学中一种发展很快的软件体系结构描述方法。按照Mary Shaw和David Garlan的观点，典型的软件体系结构语言在充分继承和吸收传统程序

维普资讯 http://www.cqvip.com

设计语言的精确性和严格性特点的同时，还应该具有：构造、抽象、重用、组合、异构和分析推理等各种能力和特性。根据这个特点，软件体系结构描述语言的构成元素既与传统的程序设计语言相同，又使得各元素的含义与传统的程序设计语言中的有很大的不同。

当前，已经开发和使用的典型软件体系结构描述语言有：Aesop，C2，Rapide，Unicon 和 Wright 等[2,26~38,41,42]。由于这些语言大都是有关的设计研究者在进行某种特殊应用的研究开发中而设计和发展出来的结果，从而使得它们的运用都有各自所特有的侧重与特点，也在一定程度上各有自己的不足。此外，还由于每种软件体系结构设计描述语言都各自有着适合于其本身特点的开发环境与工具，这些工具和环境虽然在很多的文献中已有描述和介绍，但是出于知识产权和商业技术专利方面的考虑，目前还不是很容易得到。即使通过较高成本得到了，有关的用户亦必须花费相当长的时间和精力来调试和掌握它们。

也正是由于对以上因素和一些其他有关情况的考虑，在 MSABAD 软件设计和开发方法中，我们综合考虑了当前各种软件体系结构描述语言的特点和我们的实际条件，开发和使用了一种开放式的软件体系结构描述语言（A Core of Opening Architectural Description Language——COADL）。为了保证 COADL 的开放性和可扩展性，对 COADL 设计性能的一个重要要求是：它在提供描述和表达软件体系结构设计和开发术语及概念的同时，还应提供一种可扩展的软件体系结构描述框架机制。基于如上考虑，在 COADL 中，引入了用于设计和描述的以下八种标准术语和概念元素。

（1）构件。描述和表达系统中的主要计算和数据存储元素。

（2）连接件。描述和表达构件之间的相互作用连接机制。

（3）系统。代表软件系统中的构件和连接件之间的组合配置的拓扑结构和情况。一个合理的系统从某种意义上讲反映和代表了一种不包含语义性能的软件体系结构风格特点。

（4）端口。是一种描述和表达构件与其环境进行交互的界面或接口元素。在软件体系结构描述处理中，每个构件可以有一个或多个不同类型和不同表现形式的端口，每个端口代表了该构件与其环境进行交互作用的一个作用点。

（5）角色。是一种描述和表达连接件的界面或接口的元素。每个连接件的角色定义了该连接件由该连接件参与或表达的某组构件间的作用和连接的一个接口，一个连接件至少包含两个角色。其中只有两个角色的连接件我们称之为二元连接件，其他的连接件则称为多元连接件。不管是二元连接件还是多元连接件，都至少各有一个或一个以上信息发送角色和信息接受角色。

（6）表达。在实际的软件系统所对应的软件体系结构中，有关的构件既可以是原子的也可以是复合的，其中原子构件是显然不能再进行分解描述的可用某种具体的软件开发实现工具（如程序设计语言）直接实现的软件单元，而复合的构件则是根据其抽象和复杂的程度由多个原子构件和连接件进行组合构造而成的抽象计算实体。在 COADL 中，我们采用一种叫做表达的层次描述概念和机制来实现有关构件和连接件的不同层次上的视图表达和描述。从本质意义上讲，表达是根据要求实现对同一实体的多个层次观察、分析和表达的多角度方法。为了保证同一实体的各层表达之间能够有效关联衔接，我们在提供表达机制的同时，还提供一张描述表达各层次之间关系连接描述表——表达连接图。

（7）扩展属性。是为方便 COADL 使用其他软件体系结构语言而设置的一种可选机制。从本质上讲，扩展属性是一种类似于 OLE 方式的嵌入接口机制，在 COADL 中，我们既不提供对扩展属性中的内容进行描述、解释和执行的处理功能，也不限制用户描述内容的详细格式，而是把这些功能的处理任务和具体的描述格式交给具体的使用者去决定。

（8）宏。是 COADL 中提供的一种可选的带参数重用机制。在实际的使用中，工作人员可以通过采用不同的参数进行调用的方式来使用宏所定义的句法结构体，以达到重用某一抽象的结构体的目的，这个特点使得宏与传统的程序设计语言中为减少程序员的代码量而采用的宏机制类似。

2.4 软件设计模式的产生、描述与存储

在 MSABAD 方法中，我们将软件设计模式作为一种与具体应用问题领域较为接近的微体系结构来指导和协助软件开发设计过程从抽象的软件体系结构向与具体应用问题领域相关的系统实现进行转换。虽然对于使用软件设计模式进行一个新的软件系统的设计和构造而言，一个主要关注的部分应该是软件设计模式的选择处理和操作，但是软件设计模式的发现（或产生）、描述、存储处理等其他几个方面的处理和操作结果却是软件设计模式的选择处理操作的基础和前提条件，这里必须指出的是，根据软

维普资讯 http://www.cqvip.com

件设计模式的知识与经验重用性特点，软件设计模式的发现(或产生)、描述、存储处理工作并不局限于某一次具体且实际的软件设计和开发活动，它们主要取决于软件设计和开发工作者对其所接触和了解的成功软件系统中存在的软件设计模式关注和提取。软件设计模式的产生和发现(Pattern Mining)从本质上讲是一件需要有关工作人员综合运用自己的知识和经验的复杂的高智力、高技能的知识处理过程，这个过程的处理，不仅需要有关的工作人员具有一定的软件设计工作基础，而且处理活动本身也可能是一个反复的过程。我们根据软件设计模式的特点与表现在 MSABAD 中给出以下一些生成和发现软件设计模式的参考建议准则。这些准则是产生和发现软件设计模式的必要条件。

1. 重复性准测(或普遍性)。软件设计模式所处理和针对的问题，在其他的软件系统的设计和开发活动中经常出现。

2. 完整性准则。软件设计模式对其所处理和针对的问题的解决方案应该是完整的。

3. 独立性准则。软件设计模式的解决方案不依赖于该软件系统中的其他设计而独立存在。

4. 抽象性准则。软件设计模式的解决方案是其抽象针对某一类问题而不是某一具体问题的。

5. 证明性准则。软件设计模式对其所处理和针对的问题的解决方案是经过实际的运行经历证明的。

6. 关联性准则。软件设计模式的解决方案描述了软件系统中有关组织成分之间的关联与协调关系。

对于软件设计模式的描述，采用以下描述格式来描述软件设计模式(带下划线的标识符为保留字)：

```
Pattern Name
SystemID Identifer
Problem Description{
   Keywords Keyword1,…;
   Describing systans.
}
Application Score{
   Keywords Keyword1,…;
   Describing systans.
}
Application Score{
   Keywords Keyword1,…,;
   Describing systans.
}
Solution{
   Keywords Keyword1,…;
   Describing systans.
}
```

对于软件设计模式的存储处理工作来说，我们根据软件设计模式产生和使用所具有的知识与经验的渐进和积累的知识性特点，采用一种基于文档数据管理的设计模式字典存储管理系统来支持和实现对软件设计模式文档的存储、查询和进化修改等活动。在这样一个设计模式的字典管理系统中，我们不仅提供了将新设计或产生的软件设计模式文档存放到字典中的组织管理功能，而且也提供了对字典中所存放的软件设计模式进行灵活方便的查询和进化修改处理的功能。模式字典方式的存储管理见图2所示。

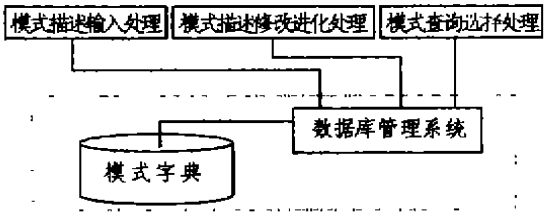

图2 模式字典管理系统示意图

3 方法实施步骤

根据阶段模型与集成方案我们把 MSABAD 中的有关操作活动进一步划分为：软件系统特性分析展望、软件概念实体的定义、实体操作转换理论的定义、软件体系结构的建立、系统应用实例数据的收集与组织、软件设计模式的选择、生成与构造软件系统的应用框架等七个主要处理。其中，各操作处理并不是严格地按照所列顺序进行，它们之间的很多处理实际上是并发进行的，而且都允许一定的反馈和回溯总结过程。

3.1 软件系统特性分析展望

软件系统特性分析展望的处理方式主要采用对软件系统组织结构及表现特性方面的问题以表格问答的形式进行。其处理目标是帮助工作人员综合考虑并协调许多有关系统软件体系结构方面的复杂因素以及有关软件系统规模、存储空间、数据存取效率、数据处理吞吐量、关键线程的规划定义以及系统处理响应时间等一系列问题。尽管许多问题和因素与传统上属于具体应用和其它的域知识相关连，但是，由于我们这里强调的是关于软件体系结构设计的抽象，因此从本质上讲更要利用工作人员的以往的具体经验进行估算，这种估算的结果随着以后有关处理的进行将允许调整和修正。

3.2 软件概念实体定义

软件概念实体定义处理主要关注和帮助系统设计工作人员对软件系统中将要涉及和处理的软件体

维普资讯 http://www.cqvip.com

系结构元素以及它们之间的交互关系进行确认和定义。本步骤中的有关术语和概念虽然有许多与软件体系结构中的相同，但是意义并不完全一样，在此阶段中，这些术语具有更多对象的属性和特点。

1. 构件实体。指的是系统中一个单一且可标识的抽象实体或概念。

2. 连接件实体。描述和表达系统中构件实体间的一种逻辑连接。

3. 构件类型。按照某一逻辑理由，把一些构件实体分为若干个集合，其中的任意一个集合中的构件实体的抽象属性及结构成为一个抽象的构件类。

4. 连接件类型。是一组连接件集合，该集合中的每一连接件都具有相同的结构和语义。

连接件类和构件类结合构成的抽象拓扑结构从语义上理解就是软件体系结构风格(Style)的雏型。在这样一个风格的雏形中，构件类指定了参与连接的构件的源。连接件类指定了构件实体间连接的方式和语义。

5. 约束语义。是对系统中的构件类和连接件类增加的一些约束定义以指出它们所具有的一些独特性质。如：基本约束、参与约束、并发约束、特殊约束和一般约束等。其中，一个基本约束定义了一个构件类中构件的最大数目；参与约束则定义了构件类中可以参与连接到一个给定的连接件的构件的数目；并发约束定义了一个构件类中有多少个不同构件可以与其它构件类中的特定的一个(或一组)构件一起出现；一般约束则是一个一般的陈述语句，用于进一步限制有关的构件类和连接件类，系统分析员可以按自己希望的形式给出一般约束。

6. 实体交互模型。通过刻画构件实体和连接件实体之间的相互作用情况来实现从宏观上刻画软件系统的行为和功能。在 MSABAD 方法中提供了如图3所示的四种描述实体交互的格式：

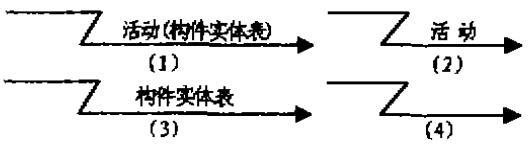

图3 MSABAD 方法中描述交互的基本格式

其中，格式(1)表示既有交互行为又有信息交换；格式(2)表示只有交互行为；格式(3)表示只有交换信息；格式(4)表示交互的情况不言自明或分析员不能用简单的标识清晰地描述之。通过这四种格式，我们在 MSABAD 方法中可以描述：同步交互、异步交互、特定交互、多点交互、双向交互、黑板交互和不间断交互等多种形式的交互模型。

3.3 实体操作转换的定义

系统操作转换的定义用于描述软件系统中有关实体的动态结构特征，及记录实体从可察觉的一种状态向另一种状态进行转换的条件、事件和有关的动作。在 MSABAD 方法中，我们采用有限自动机来定义和描述系统中实体的操作转换。

对于一个实体 E，设 S_E 是由该实体的所有状态组成的集合，P_E 是由所有触发该实体状态进行变化的触发事件组成的集合，C_E 是由所有允许该实体状态进行变化的条件组成的集合，A_E 是促使该实体状态进行变化的操作组成的集合，则实体 E 的一个转换可以定义成一个三元组(p,c,a)，其中，$p\in P_E$，$c\in C_E$，$a\in A_E$。若再设所有 E 的转换组成的集合为 V_E，则实体 E 的一个状态的状态语义可以看成一个三元组(E_id,s,v)，其中，E_id 是实体 E 在系统中的标识，$s\in S_E$，$v\in V_E$。由此，我们可以将软件系统中有关实体状态的操作转换定义成由三元组描述的状态转换自动机。

3.4 软件体系结构的建立

本步骤的处理在实体概念定义工作的基础上，采用 COADL 语言对系统中软件体系结构进行设计与描述。在设计和描述过程中，方法的使用人员必须根据 COADL 语言和软件体系结构的特点要求，对有关的实体概念进行综合与抽象，以使之符合软件体系结构的设计描述标准。例如，对于一个事件广播系统机制，实体概念定义中的定义有从事件产生者到多个事件接受者之间的多个连接件，而在采用 COADL 的软件体系结构设计与描述中，这些连接件可能被抽象综合成一个多对一的抽象连接件。

3.5 应用实例数据的收集与组织

应用实例数据的收集与组织主要是在体系结构的指导下对软件系统将进行的有关应用与处理的实际数据与信息进行采集和归总，其目的是为下一步的软件设计模式生成或选取做准备，以开始实现从抽象特性的软件体系结构向实际问题领域处理的软件系统进行转化。由于将涉及到系统有关构件、连接件以及有关关联的具体实现细节，因此本步骤的数据和信息不仅应该比较详细和具体，而且还应该考虑和借鉴一些已有的成功系统应用实例数据与信息。例如，对于财务电算化处理软件系统，不仅要考虑收集有关的凭证及报表的格式和计算公式等数据信息，而且最好能够收集和考察现已有的一些这方面的应用系统的应用情况数据和信息。考虑到这一

维普资讯 http://www.cqvip.com

步骤的特殊性，我们在MSABAD方法中并不严格固定地给出需要采集和组织的数据信息格式和项目，只是以建议方式给出一些比较典型的数据和信息：(1)系统中将要处理的各种凭证、报表和文档等实体的内容与格式；(2)对有关实体信息进行的处理原理模型或公式；(3)各种信息与数据处理的功能分配情况，以及它们与软件体系结构的匹配情况；(4)有关的相似系统的规模以及使用情况的数据与信息等。

3.6 软件设计模式的选择

本步骤主要在软件体系结构的抽象系统组织结构基础上，利用我们在关键问题中所描述的模式字典管理系统以及有关的原理与法则实施软件设计从通用的抽象组织结构语义向具体应用问题领域的应用系统进化。在本步骤的处理工作中，有关的软件设计工作人员可以根据具体的应用问题和体系结构特征，在设计模式字典中搜索和查找相应的软件设计模式。由于应用问题和需求本质上是千变万化，因此模式字典中可能并不存在与所需条件完全一致的设计模式，对于这种情况，设计人员可以在有关的软件体系结构的指导下，在字典中找一个与所需要求相近的模式，并按照有关的原理和准则进行相应的修改，以生成一个满足要求的新的软件设计模式。为此，我们在设计模式的查询选择支持处理机制中，不仅考虑到了支持根据设计模式有关特征的关键字查询处理，而且还可以通过设置相近问题、方案和应用关联形式等特征关键字的方式来提供对应用问题或方案的相近模式的查询和处理。最后，需要指出的是，本步骤的有效应用，取决于对大量的应用设计模式的产生收集和存储。这一特点上，本质上亦是MSABAD方法的知识和经验重用性的一个显著表现。

3.7 生成与构造软件系统的应用框架

在MSABAD方法中，生成与构造软件系统的应用框架步骤的处理是作为一种参考为软件系统的开发人员提供的一个可选处理步骤，原因有两个：

首先，软件设计模式作为一种软件微体系结构，已经是一种与具体应用处理问题关联较为密切的设计结构结果，这种结果从本质上来讲，可以直接指导具体的软件开发活动；

其次，具体的软件系统应用框架往往与具体的应用开发语言和工具有着十分密切的关系，而从软件开发的发展历史来看，用于软件开发的程序设计语言和支持工具实际上是软件系统设计和开发各组成成分中最为活跃且发展较为迅速的部分。

虽然有着以上两个原因使得我们没有把软件系统应用框架的生成处理作为MSABAD方法的必须处理，但是考虑到毕竟在每个历史时期，软件开发处理活动中都存在着若干个广为流传和使用的主流语言和支持工具。因此，系统应用框架作为一种较为具体的代码重用手段，对于提高软件系统的开发和生成效率，保证软件系统的可靠性仍然具有一定的积极作用。

在MSABAD方法中，我们约定一个系统应用框架应该是包含以下几个组成部分的一种较为通用的程序库机制：1. 语言和环境的注解说明；2. 基于特定语言和环境的用户界面成分(如菜单、窗口和对话框等)；3. 系统工作和调度的核心机制；

从以上，我们可以看出，本质上讲，框架也可以看成是一种与具体语言和环境相关联的软件设计模式。

应用与结论 为了验证MSABAD方法的实际使用效果，我们在作者参加的"中国温州正泰集团领导决策支持系统"(以下简称为"决策系统")和"浙江好来西集团生产流程管理控制系统"(以下简称为"管理控制系统")的设计研制过程中，采用MSABAD方法来指导有关项目的设计开发活动。首先，在"决策系统"的设计开发过程中，由于我们坚持以与应用功能独立的抽象软件体系结构为核心并辅以以知识和经验重用的软件设计模式来指导软件的设计与开发活动，因此，尽管在设计开发过程中，企业因业务发展和市场变化的需要而多次对其部门和有关功能进行重组与调整，但软件系统的设计开发过程与结果却能够很快地适应相应的变化，并且提前完成了有关的开发任务。目前，根据有关设计结果而开发的实际系统已经正常投入运行并获得用户的好评；其次，在"管理控制系统"的设计开发活动中，由于"管理控制系统"的分布式流程控制与协调机制与"决策系统"中的分布信息查询与决策机制在本质上有许多设计知识和经验可以借鉴和重用，因此相应的设计与开发周期比预期提前了近一半的时间，设计结果也使用户十分满意。总之根据实际的使用情况和效果证明，MSABAD方法确实达到了有效集成与利用面向对象、软件体系结构和软件设计模式等先进的软件开发设计技术来支持大型软件系统的设计开发活动的研究目标。(参考文献共42篇，略)