

文章编号: 1000-1220(2001)07-0874-04

# XML 技术在 CBSE 中的应用

唐振云 齐克科 钱乐秋

(复旦大学 计算机系 上海 200437)

**摘 要:** 随着面向对象方法的日趋成熟, 构件的独立性、可重用性、在不同粒度上的可扩充性被人们所认识, 基于构件的软件工程 CBSE (Component-Based Software Engineering) 将形成未来开发软件系统的基础。而另一方面, 计算机网络发展迅速, 各项技术日新月异。其中扩展标志语言 XML (Extensible Markup Language) 这项最新技术以其简单化、多用途形成了强大的生命力, 本文详细论述了 CBSE 与 XML 的技术特性, 并结合 CBSE 与 XML 这两方面技术的发展讨论了 XML 在 CBSE 中可发挥的强大作用与应用前景。

**关键词:** 基于构件的软件工程; 扩展标志语言; 层叠样式单; 扩展样式语言

**中图分类号:** TP31

**文献标识码:** A

## 1 简介

### 1.1 CBSE 简介

软件工程学的整体目标是高质量、低投入的完成软件系统的开发、运行与维护, 而人们经过几十年的开发, 发现有大量软件具有相同的应用领域或其应用领域极其相似, 这些软件系统在应用框架上是相通的, 人们进行大量的重复开发是对经费、人力、时间等各项资源的极大浪费, 人们希望在整个系统, 某个具体应用模块直到一个独立的较小功能等各个层次, 各种粒度上都能提供对以前开发的软件的重用, 甚至做到象硬件一样组装以后, 即插即用, 大规模流水线般的生产。为满足这种需求, CBSE 应运而生, 它以软件构件化为基本思想, 以简化开发, 提高复用, 增强灵活性等各方面特性在软件工程领域提出了一种全新的模式, 下面将详细论述其特性。

谈到 CBSE, 首先要讨论构件与构架。构件与构架是 CBSE 的两个关键概念, 其中构件模型是基础, 它包含了一系列的建模概念, 各自的语义以及交互合作时的规则。具体来说, 构件是一个系统的功能性单元, 它可以是不同粒度上功能的抽象。它分原子构件与组合构件两种类型, 原子构件指功能单一、不可再分的构件, 组合构件指包含了其它原子构件和(或)组合构件的构件。因此, 一个完整系统的构架被描述成一系列原子与组合构件的组合。但是, 构件最大的优点是它是可重用与可共享的单元, 它可以在多个项目中被重用。

构件的功能是通过其接口定义的。接口分需求接口与服务接口两种类型, 一个构件允许有一个或多个需求接口与服务接口。从语法上讲, 两者均与类描述相似; 从语义上讲, 服务接口规定了一个构件提供给其它构件的服务, 需求接口规定了一个构件为执行它自己的功能需要从其它构件中接受的服务。在 CORBA 的接口定义语言 (IDL) 中, 接口可支持方法、属性、异常、多重继承和名字空间等诸多特性, 此外还可以加入前置、后置条件, 不变量和状态等特性。

例如:

```
Interface ImprovedPolicy: Policy{
    States { stInitial(init), stAuthorized };
    Void Authorize(in string name, in string password) raise
(U nAuthorized)
    precondition { stInitial; };
    postcondition { stAuthorized; };
    void IncreasePremium { in float percent }
    precondition { percent > 0 0; stAuthorized; };
};
```

在这个 ImprovedPolicy 的接口中, 可以看到它是从 Policy 接口继承, 并增加了两个方法: Authorize 与 IncreasePremium, IncreasePremium 的前置条件是需要 percent > 0, 同时状态是 stAuthorized。状态是用于规范执行接口中方法的次序, 在这个接口中, 它规定了必须经过授权才可以进行增加保险费的操作。同样, Authorize 也有前置条件, 它还有后置条件, 即在“授权”方法执行后, 状态应变为“已授权”, 同时如果验证密码未成功, 它还会产生一个异常 raise (U nAuthorized)。

构件通过互相连接形成组合构件, 一个连接是通过将一个构件的需求接口与另一构件的服务接口绑定, 如下:

```
component Client {
    requires ImprovedPolicy requiredPolicy;
};
component Server {
    provides ImprovedPolicy providedPolicy;
};
component Application {
    Client clientInstance;
    Server serverInstance;
    Bind clientInstance requiredPolicy
    To serverInstance providedPolicy;
};
```

通过将构件 Client 的需求接口与构件 Server 接口的服务接口绑定, 实现了 server 向 client 提供所需功能。这种描述

构件绑定的方式是对一个大型软件系统建模的基础,它代表了一个系统的构架,系统构架的一个重要用途就是用于实现系统资源的优化分配

然而,为实现一个系统的构件模型以及构件的即插即用,还有必要考虑系统的配置、运行环境、接口的规范化分析、构件的提取与集成等等。因此我们需要利用多种技术实现对 CBSE 技术的完善

而另一方面,从现今计算机的发展情况看,网络化已是计算机发展不可逆转的潮流,软件工程的发展越来越强调计算机互连时所需的一些工程技术;反过来讲,研究网络技术对软件工程发展的促进作用也很有必要。在诸多网络技术中,XML 是一种全新且具有强大生命力的技术,它的发展越来越受到世人所瞩目,下面将对此进行介绍

## 1.2 XML 简介

网络技术的发展已渗透到社会生活的每一个角落,但单靠 TCP/IP 协议使网络简单化,大众化是远远不够的,因此导致超文本标志语言 HTML 迅猛发展,它依靠极强的通用性与易学性,大大丰富了网络信息。但随着 WEB 文件的复杂化、多样化、智能化,人们发现 HTML 缺乏一种自描述能力,不能根据不同用户的不同需求以不同方式表达,即缺乏适应性和互操作性,并且其维护性也较差。例如网上拍卖,拍卖者、竞拍者及一般的浏览者他们关心的拍卖重点肯定不一样,但他们看到的网页却是一样,服务器虽然可定制几种信息表达方式,但终究无法满足所有未知用户的需要。实际上这是由 HTML 本身的特性所导致的,HTML 是以 `< tag> content < /tag>` 的形式描述文本,其标记(tag)集合是固定的,用户无法增加供他人使用的标记,信息表达方式受到限制。另外 HTML 的文本内容与格式混在一起,要想更改表达方式极为困难

为解决 HTML 的这种局限性,XML 诞生了。XML 有 HTML 所欠缺的巨大的伸缩性与灵活性,它实际上是一种定义其他类似 HTML 的标记语言的元语言,比如微软公司支持的“推技术”所用的频道定义格式 CDF (Channel Definition Format) 就是 XML 定义的一种语言,使用者通过写文档类型定义 DTD (Document Type Definition) 可定义无穷无尽的元素、属性、内容模式及实体来描述 XML 文件中的任何数据,使 XML 文件的内容更丰富,更复杂并组成了一个完整的信息体系,保证了用户的互操作性

具体来说,XML 实际上是将文件的结构、内容与表现形式分离。它的内容放在以 xml 为后缀的文件中,其形式与 HTML 文件相似,也是以 `< tag> content < /tag>` 的形式表示文档中一个元素,例如 `< Book> XML < /Book>`

```
< Page> 450 < /Page>
```

```
< Author> John < /Author>
```

这每一个元素相当于数据库中的字段。整个 XML 相当于一个数据源,它只负责存放数据。而每一个元素的 `< tag>` 标记,可用属性,以及与其它元素的组合形式(在 XML 中称为内容模型)是在另一个被称为 DFD 的文件中定义 DTD。是为了表示某个特定类别的文档而定义的一组规则,它确定了

文档的结构。例如以上 `< Book>`, `< Page>`, `< Author>` 等标记语法,它们在 XML 文件中出现顺序,组合方式等均可在一个 DFD 中定义

此外,XML 文件的表现形式是利用层叠样式单 CSS 或扩展样式语言 XSL 来描述,相同内容的 XML 文件,只要附以不同的 CSS 或 XSL 文件描述,其表现形式就会大相径庭。这种将内容与表现分离的机制,在表现层上具有极大的灵活性,能较容易地对内容进行维护

除了以上 XML 的一些基本特征外,XML 的最重要特性在于其内容的结构化,其内容不再是毫无规则,只为了表现而杂糅在一起的数据集合,而是完全分层的,具有自己一套完成描述体系的数据。这正好对 CBSE 的发展完善提供了一个契机。下面将开始重点讨论 XML 对软件工程构件化发展的优点

## 2 XML 对发展 CBSE 的作用

### 2.1 对象文档化与构件提取

从形式上看,XML 实际上将数据文档化,结构化,进而可做到文档对象化,例如一个产品在 XML 文档中可表示为:

```
< name> 玻璃杯 < /name>
< model> 600 毫升 < /model>
< fill> 200 毫升 < /fill>
< price> 25 元 < /price>
```

它表示一个盛了 200 毫升水的玻璃杯,这与传统的关系数据库很相似,但对于对象来说还缺乏方法的描述,我们可以定义新标记来实现,例如:

```
< addwater> 加水量 < /addwater>
```

这样就与一个对象数据库很相似。我们可以使用这种结构化标记来表示在传统概念中无法用文档表示,但对于传统数据库来说又过于复杂的数据,例如脱氧核糖核酸分子模型,各种化学元素分子模型,虚拟现实世界等等。同样类层次关系,也可以在 XML 中定义,例如:

```
< elementType id= "编号">
  < string/>
< /elementType>
```

```
< elementType id= "交通工具">
  < element type= "# 编号" />
< /elementType>
```

```
< elementType id= "火车">
  < superType type= "# 交通工具" />
< /elementType>
```

```
< elementType id= "飞机">
  < superType type= "# 交通工具" />
< /elementType>
```

上面定义了“火车”与“飞机”为超类“交通工具”的子类,它们都有一个“编号”属性。实际上利用定义的通用标记,我们可以将各种各样数据及关系集成在一起。文档的数据库化和数据的文档化是 XML 走向成功的主要推动力之一。反过来想,现实世界中的对象也是通过一定的数据表达的,通过对象

文档化,我们就极其容易地找到了一种定义和描述对象的方法,而如果能从不同 XML 文档的对象数据中提取出组件,将会为 CBSE 提取组件开辟一种新的思路,在这方面的研究中,"组件化"已作为 XML 进行文档内容管理的一个新概念而被提出。文档组件是文档中逻辑性的和层次性的片段,通过语法分析,XML 文档能自动分解为一个组件对象结构 W 3C (World Wide Web Consortium) 在这方面已规定了 XML 在 WEB 上与之交互的文档对象模型 DOM (Document Object Model),DOM 定义了访问文档中数据的机制的界面。使用 DOM 可以用标准的方法访问 XML。它将 XML 文档表示成树形结构,并提取任意特定部分的数据进行分析,然后按所需方式进行表达。而 CBSE 正是需要这样一种从应用中提取对象并泛化为构件的功能。可以说,XML 的形式实现了对对象的文档化,而 DOM 提供了我们实现对象提取为构件的机制。

## 2.2 对象交互与异质构件的互连

在 CBSE 中,系统的功能必须通过集成的构件之间的交互来实现,如前所述,构件的交互是需要定义接口,接口屏蔽了构件实现的细节,现在已有 COBRA 的 DL, 构架规范语言 ASL 等多种定义方式,定义一个通用的接口形式有利于实现不同构件提供商构件的互连,并对不同环境下构件的互连也是不可或缺的,而 XML 语言作为一种元语言,不但可实现对接口的描述,并在通用性、灵活性方面都有极大的优点,它还通过属性(或者说"元数据")使接口信息增值,接口定义者通过增加"有关信息的信息",进一步对有可能重新设计的信息进行描述,使接口可随特殊领域的应用进行演变。

另一方面,从 XML 的对象文档化中可简单地找到一种对象数据交换格式,而让构件只与与之相同系统环境下的构件交互是不实用的,我们必须支持在不同运行环境下(包括不同的程序设计语言,连接协议,硬件平台,操作系统等)异质构件互连。这就需要一种强有力的机制保证在分布式系统中实现构件之间信息的交互。利用 XML 定义的格式及其强大的灵活性,我们可以方便地做到在不同应用领域中数据信息的格式化、压缩、传递与转换。举个简单的例子来说,现在的 WEB 应用均采用三层构架,即(图 1)

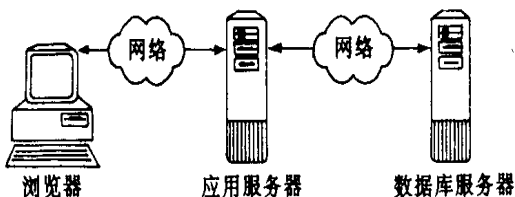


图 1

浏览器实现用户界面,应用服务器处理逻辑,数据库管理数据,但在处理逻辑中,业务规范常常与用户界面上的表现形式混在一起,一旦用户要求改变界面形式,就不得不再次访问应用服务器与数据库,这样将极大影响应用的效率,而且如果数据是分布在网络的不同数据库服务器上,其运行将更加复杂与耗时。而利用 XML 可以用以下一种方式处理(图 2)。

图 2 中从硬件角度讲,实际上仍为三层结构,但从逻辑上

看,图中浏览器与应用服务器中分别增加了两个 XML 文档层,数据被一次性从分布在网络的不同数据库中取回,这些数据可以通过 ODBC 或其它类似数据库连接方法取得,或者也可是一个 XML 文档数据源,当数据返回服务器后,应用服务器通过各种数据库数据与 XML 文档之间的转换,形成一个完整的 XML 文档,并根据需要将其一部分或全部发送给浏览器,浏览器得到 XML 文档后,当用户要求简单地改变界面形式时,XML 文档可与 JavaScript 等脚本交互或在浏览器端按不同的 CSS/XSL 显示,而不必再访问应用服务器。当用户要进行一些逻辑处理时,可再次访问应用服务器,由于应用的数据已在应用服务器中,应用服务器可按某种业务逻辑进行处理并返回 XML 文档而不再需要访问数据库服务器。当然,如果用户要求查询还未取到的数据或对数据作增改等操作,应用服务器还得访问数据库服务器。这种方式使界面与业务逻辑尽量分离,客户端访问应用服务器与数据库服务器的次数减少,网络性能大大提高。

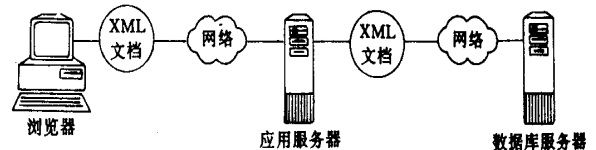


图 2

从 XML 文档返回时表现形式的多样性看,我们在 CBSE 的研究中,还可以考虑构件获取后,其功能表现的多样性、灵活性,这种多样性将会极大地丰富构件使用的形式。

## 2.3 内容模型与构件集成

CBSE 强调构件的集成,只有构件按某种形式集成到构架中,才能发挥它们功能上的互补性。除了定义接口以外,在整体上,我们还需要一种方式来表示在一个组合构件中包含的原子构件或组合构件的种类、数量及组合关系,这对于以后构件组装的市场化、商业化有极大的推动作用。在这方面,XML 文档的内容模型提供了我们这样一种集成机制。一个内容模型是一个模式,用来描述一个元素类型中包含的子元素以及它们之间的顺序,最简单的内容模型是一个元素中不再包含任何其它元素,这就象原子构件一样,例如:一本书包括书名,作者,各章与结束语,每一章中又包含标题和许多段落,而标题就可以作为一个最简单的元素(如果不再需要细分到词或字),其内容模型可定义为:

```
< ! E L E M E N T Title ( # P C D A T A ) >
```

而复杂一点的元素,如组合构件一样,里面还包含其它元素,并且出现的顺序与次数也有规定,例如前面提到的一本书与整个章节可用 XML 定义为:

```
< ! E L E M E N T Book ( Book N a m e , A u t h o r , C h a p t e r + ,
    T a g ) >
< ! E L E M E N T Chapter ( T i t l e , P a r a g r a p h + )
>
```

其中+ 号表示该元素可重现一次或多次,如改为\* 号,表示可重复零次或多次,此外还可以在内容模型中表示选择关

系,例如:图片可以是图形或代码,用 | 线表示选择:

< ! E L E M E N T Figure (Graphic | Code)>

由于 XML 文档中的元素相互嵌套,因此整个文档的结构是在文档元素的内容模型中进行描述的,所谓文档元素是指包含整个 XML 文档内容的一个元素,与 HTML 文档中的 < HTML > 标记类似,通过内容模型的层层解析,一个文档可看成一个组件表。在 CBSE 中的组件集成就可以用这种方式进行描述,用户可以一目了然地知道实现该构件需要哪些构件的组装。

#### 2.4 构件查找与信息反馈

CBSE 的未来方向是市场化集成,在网上寻找组件并到构件供应商网站下载组件都需要一种机制帮助用户方便可靠地找到所需的组件。另外,当出现问题时还需要通知构件供应商,用户使用构件后还可以交流经验。为满足这些要求,从 XML 中发展得来的 X-link 应该是一种较好的解决方式。

X-link 的基础是 HTML 的超链接方式,但 HTML 中的链接只能从链接所在的地方开始遍历,而不能从被链接的内容开始,如一个链接为 A -> B,则链接只能从 A 开始,而 X-link 提供了称为“扩展链接”的方式,它允许链接开始定位点与链接本身不同,可以从该链接的任何一个定位点开始,遍历到其它任何一个定位点,而不用知道链接在何处,比如以上链接,就可从 B 开始遍历。更重要的是,扩展连接不仅仅是两个定位点的链接,它还可以将多个元素链接起来,用户可以从多方向检索链接内容,实际上链接本身的信息已从 XML 文件中独立出来,保存在一个单独的文件中。

对于构件的查找,我们在网上寻找构件时,与一个构件合作的相关构件可以用一个扩展链接表示,我们只要找到一组构件的任何一个,就可以很容易找到这组构件中的其它组件;此外,用户还可以为使用的构件写批注,虽然他无权修改供应商的网页上,但他可以将自己的批注保存在一个网页上,利用

扩展链接的方式链接到构件供应商的相应网页上,任何人到达这个网页,通过网页就可以看到相应的批注,这样就实现了与其他用户交流信息的目的,对于构件供应商来说,可以及时了解用户信息,并可以统计网页上扩展链接的密集程度而知道哪一类构件人们最感兴趣或可能急需改进。

### 3 总结

XML 的实质是提供了一种标准化、可独立发布、有级别操作的领域,它的诸多优点及先进性在当今网络发展技术中倍受关注,这种关注不仅在于它对网页开发的应用,更重要的是它提出的一系列概念和机制对于数据库,软件工程,分布式系统等诸多方向都有极大的影响,本文提到 XML 对 CBSE 发展的作用只是其中的一方面。而除了以上提到的几点,XML 实际上在接口继承及其简化,自动化生成适配器模版,构件包装等诸多方面对 CBSE 的发展也有一定的研究价值。W3C 正号召全世界的开发人员共同对 XML 进行定义和开发,这必将使 XML 的各项技术日趋成熟,同时也将促进软件事业在各方面的加速发展。

### 参 考 文 献

- 1 D. Bryan, F. Bronsard. Toward Software Plug-and-Play. Software Engineering Notes, May 1997, Volume 22(3), 19~ 29
- 2 Mark A. Linton, John M. Vlissides, and Paul R. Calder. Composing User Interfaces With Interviews. IEEE Computer, February 1989, Volume 22(2), 8~ 22
- 3 T. Bray, J. Paoli, and C. M. Sperberg-MacQueen. Extensible markup language (XML) 1.0, w3c recommendation, <http://www.w3.org/TR/REC-xml>, feb 1998
- 4 J. Suzuki and Y. Yamamoto. Managing the Software Design Documents with XML. ACM SIGDOC '98, September 1998, 127~ 136
- 5 Charles F. Goldfarb, P. Prescod 著,张利等译,XML 实用技术.北京:清华大学出版社,1999. 9

## APPLICATION OF XML IN CBSE

TANG Zhen-yun QI Ke-ke QIAN Le-qiu

(Computer Department of Fudan University, Shanghai 200437, China)

**Abstract** With the maturation of Object Oriented method, the independency, reuseability and expandability of component at different granularity have been realized by people. CBSE will become the fundament of the software development in the future. On the other hand, computer is developing very fast in the field of network, various kinds of technology change with each passing day. Among them, XML has great vitality because of its simplicity and a large variety of usage. This article will first talk about the technological speciality of CBSE & XML in detail. Then, with the blending of CBSE & XML, something will be talked about the important role XML plays in CBSE and its prospects for practice.

**Key words** CBSE; XML; CSS; XSL