一种软件集成开发系统的设计与实现

王晓晔[1]，张　林[2]

(1. 湛江师范学院信息科学与技术学院，湛江　524048；2. 西北大学信息科学与技术学院，西安　710069)

摘　要：在分析协同开发和软件复用在现代软件开发过程中出现的问题的基础上，提出分布式协同集成开发环境体系结构。详细论述该系统的组成和具有的特征，并阐述该系统的关键技术，包括模型设计控制器、源程序编辑控制器和构建协议、项目服务器、构建服务器及构件库的实现，并给出该系统的应用及效果。

关键词：分布式；协同；开发环境；DCIDE

0　引　言

随着计算机系统被广泛应用到各个行业，软件需求量剧增，“软件危机”的出现直接催生了软件工程这一门学科。1968 年，在 NATO 会议上首次提出了“软件工程”这一概念，使软件开发开始了从“艺术”、“技巧”和“个体行为”向“工程”和“群体协同工作”转化的历程。软件工程技术也从 20 世纪 60 年代末~70 年代中期的结构化程序设计，20 世纪 70 年代中期~80 年代的计算机辅助软件工程(CASE)，20 世纪 80 年代中期~90 年代的面向对象语言和方法发展到了现阶段的面向方面的软件开发方法学；软件的定义也从“数据结构+算法=程序“发展到了“设计模式+对象组件+开发工具=程序”。以上所述反映了这么一个事实：软件开发已经是一项团队工作，软件复用也达到了相当的高度，开发工具已经在构建软件中起着显著的作用，有着突出的地位，而集成开发环境在开发工具中又处于重中之重的地位。

1　问题的提出

协同开发和软件复用，在现代软件开发过程中大幅度提高了软件生产率的同时，也导致了软件开发过程中不少新问题的出现，在这些问题当中最主要的问题有：

(1)并行版本控制问题。这是由协同开发所带来的最主要问题，多人参与开发的系统必然会导致源程序和文档在管理上的混乱，必然会导致源程序和文档出现版本冲突。记开发者集合为 U，文件集合为 F，o(u,f)为用户 u 打开文件 f，w(u,f)为开发者 u 写文件 f，c(u,f)为开发者 u 关闭文件 f，假设在软件开发过程中，存在 o(u1,f)，o(u2,f)，w(u1,f)，w(u2,f)，c(u2,f)，c(u1,f)这样一个序列存在，因为最后关闭文件的操作由开发者 u1 来完成，那么显然开发者 u2 对文件 f 所作的修改没有发生任何作用。而集成开发环境不同于其他系统并发控制的地方就在于允许不同开发者对同一个文档进行修改。

(2)进度控制质量保障问题。在协同开发情形下，进度控制明显不同于传统的开发模式，进度应该集成到开发环境，只有开发环境本身有着这种强协同意识，软件开发的进度才能够得到保证。在协同开发情形下，可能有多个开发者在不同的历史时期维护同一片源程序，若集成开发环境缺乏一种机制来跟踪质量的话，那么软件的维护将变得极其艰难，同时软件质量也缺乏保障。

(3)构件维护及依赖冲突问题。这是由软件复用所带来的最主要问题，构件技术在现代软件开发当中被广泛使用，这样，每台开发机在构件维护上有着相当的工作量，另外构件之间存在一定依赖关系，这个关系，我们记为 d，d(A,B)表示软件 A 依赖于构件 B，假设我们有 d(A,B)，d(A,C)，d(B,D)，d(C,D')这样一组构件依赖关系，其中用软件 A 依赖于构件 B 和构件 C，软件 B 和软件 C 依赖于同一个构件 D，但是 B 和 C 所需要 D 的版本并不一致，那么上述构件冲突的存

收稿日期：2009-04-28　修稿日期：2009-07-16

作者简介：王晓晔(1979)，女，河南洛阳人，教师，研究方向为计算机网络及应用

在所带来的直接后果就是软件 A 开发上的失败。

目前，主流的集成开发环境有：Windows 平台上的 Visual Studio，Linux/Unix 平台上的 KDeveloper，Anjuta，Emacs，跨平台的有 Eclipse。上述集成开发环境仍然未能脱离传统的集成开发环境架构，即给开发者提供一个可视化软件开发环境，给出了一些应用软件向导以快速产生应用软件，并没有在根本上提供上述三大问题的解决方案。显然，这些集成开发环境不再适应于现代软件开发，现代软件开发对集成开发环境有了更多的要求，本文所提出的分布式协同集成开发环境是对传统的集成开发环境的继承和发扬，就如何在传统的集成开发环境架构的基础上融入协同开发思想做了研究并给出了设计思路并初步实现。

2 分布式协同集成开发环境体系结构

分布式协同集成开发环境（如无特殊说明，以下简称系统）实际上是一个四元组 DCIDE（Project Server–项目服务器，Construct Servers–构建服务器，Network–网络，Participators–项目参与者），这反映了分布式协同集成开发环境具有集成性、协同性、分布性和编译系统无关性等功能特征。

角色不同的参与者分布于网络，通过浏览器和项目服务器直接交互，项目服务器根据参与者的请求需要请求构建服务器构建项目，系统体系架构见图 1。

（1）项目服务器

项目服务器是一个基于 B/S 架构的富互联网（Rich–Internet–Application）应用，项目参与者通过浏览器与项目模型服务器交互完成各项软件开发任务：需求分析、概要设计、详细设计、实现、测试等。项目服务器逻辑模型见图 2。

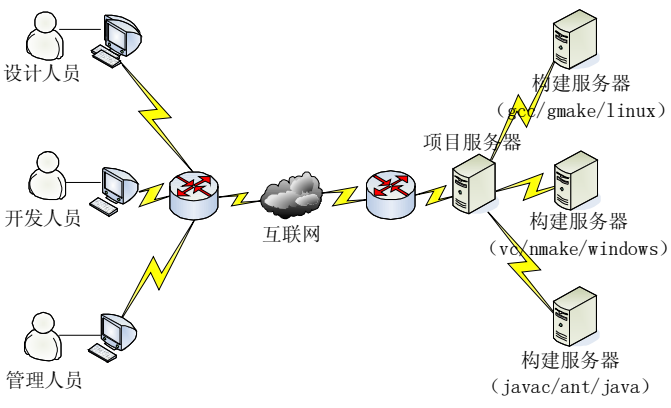


图 1 分布式协同集成开发环境体系结构

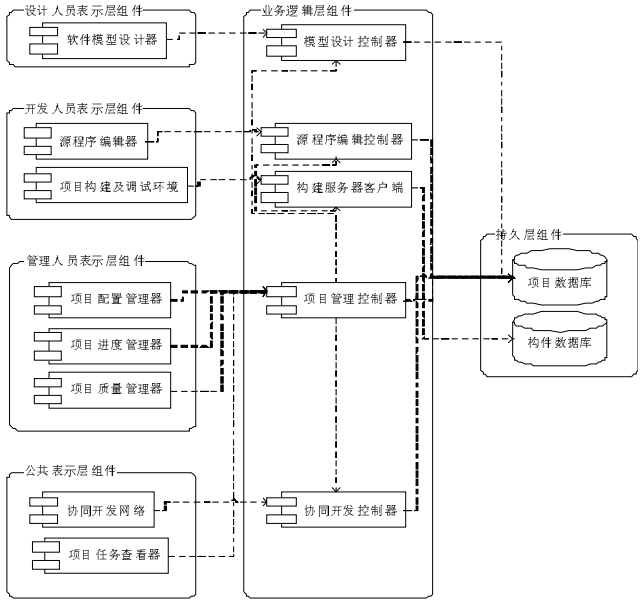


图 2 项目服务器逻辑模型

项目服务器采用 AJAX 作为表示层技术，其表示层包含有软件模型设计器、源程序编辑器、项目构建及调试环境、项目配置管理器、项目进度管理器、项目质量管理器、协同开发网络、项目任务查看器；采用 J2EE 作为业务层技术，其业务层包含有模型设计控制器，源程序编辑控制器，构建服务器客户端；其持久层组件包含有项目数据库以及构件数据库。

（2）构建服务器

构建服务器是一个传统的 C/S 架构应用，系统通过构建服务器来完成项目的编译，构建以及调试。构成构建服务器的核心组件有构建协议、协议预处理器、协议处理器等，其逻辑模型见图 3。

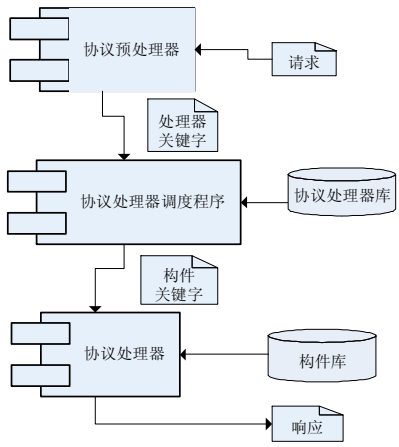


图 3 构建服务器逻辑模型

3 关键技术实现

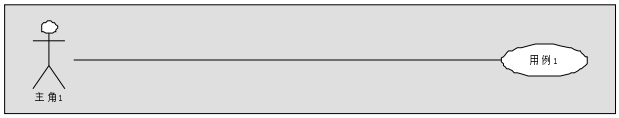
3.1 模型设计控制器实现

项目服务器提供了两个基于 AJAX 的模型设计工具箱:①传统的基于 UML 的面向对象设计工具箱;②基于 BPEL 的工作流工具箱以帮助设计人员对软件进行建模。模型设计控制器被定义成 MDC(Designer–设计器,Converter–转换器,Model–模型,Program Framework–程序框架)这样一个四元组。设计器采用文献[3]所述算法,运用 XML 和 SVG 技术来实现 UML 以及 BPEL 模型图表的产生;转换器采用文献[4]所述算法,实现了模型/源程序框架之间的正向/逆向转换,也就是能够根据模型产生源程序框架,能够根据源程序框架产生模型。

项目服务器采用 XML 语言来描述模型,因为 BPEL 本身已经采用了规范的 XML 语法表示形式,故系统只需对 UML 进行 XML 扩展。系统对文献[5,6]中所提出的 UML/XML 映射模型进行了修订和扩展以适应系统需求,图 4 给出了用 XML 表示 UML 用例角色的模式以及实例的例子。

```
<xs:complexType name="actor">
  <xs:sequence>
    <xs:element name="name" type="xs:string" minOccurs="1" />
    <xs:element name="description" type="xs:string" minOccurs="0
" />
    <xs:element name="initiates" type="xs:IDREF" minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required" />
</xs:complexType>
```

角色的 XML 模式



主角1在用例1中的UML表示

```
<actor id="customer">
  <name>主角1</name>
  <description>参与到某个用例的主角</description>
  <initiates>用例1</initiates>
</actor>
```

主角1在用例1中的XML表示

图 4 角色的 UML/XML 映射

此外,项目服务器将模型控制器与项目管理器关联起来,使得模型控制器所产生的代码框架可直接应用于项目任务分配。

3.2 源程序编辑控制器

项目服务器的源程序编辑控制器继承了传统的集成开发环境的源程序编辑器的语法高亮显示,基于块的编辑等功能特征,同时也有自己的扩展,主要体现在:①采用了 Ajax 技术,实现了类似 CodeIDE 所给出的源程序编辑环境;②源程序最小编辑粒度不再是文件,而是源程序块(具体到语言则为宏,函数或者方法等),源程序最小编辑粒度更小一些;③源程序编辑过程中,开发人员可查阅代码修订历史,保存了源程序修改记录,有利于源程序版本切换;④多个开发人员可协同编辑同一个源程序块。

源程序编辑控制器给出了 3 个接口:①语法高亮显示控制接口,接受源程序,语法高亮显示模板,源程序语法规范作为输入,以 XML 方式输出源程序编辑控制语言;②源程序版本控制接口,接受源程序,项目管理控制器作为输入,以 XML 方式输出源程序编辑控制语言,版本控制请求;③协同编辑控制接口,接受源程序编辑器作为输入(主要是编辑事件),以 XML 方式输出源程序编辑控制语言。源程序编辑控制器逻辑模型见图 5。

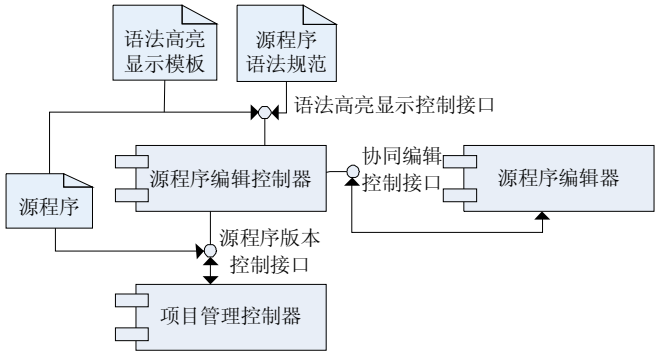


图 5 源程序编辑控制器逻辑模型

源程序编辑控制器在采用 GOT 算法 “在保证一致性的同时维护了用户操作意愿”的同时,提出了自己的协同编辑并发控制模型:焦点锁定模型和仲裁模型。

焦点锁定模型是严格的基于队列的控制模型:源程序编辑控制器在一段不长的时间 $\triangle t$ 内接收到 m 个编辑控制命令 $U=\{ecc_1, ecc_2, \cdots, ecc_m\}$,其中有 n 个编辑控制命令存在冲突集 $C=\{ecc_{i1}, ecc_{i2}, \cdots, ecc_{in}\}$,那么源程序编辑控制器将这 n 个编辑控制命令排入冲突队列,将编辑焦点锁定,序列执行 C 中的编辑控制命令。

仲裁模型则是基于级联式 RBAC 的控制模型:源程序编辑控制器在一段不长的时间 $\triangle t$ 内接收到 m 个编辑控制命令 $U=\{ecc_1, ecc_2, \cdots, ecc_m\}$,其中有 n 个编辑控制命令存在冲突集 $C=\{ecc_{i1}, ecc_{i2}, \cdots, ecc_{in}\}$,对

任意的编辑控制命令 $ecc_j \in C$，均有一个权值 $w(ecc_j)$ 与之关联，其大小取决于命令主体的在角色层次中的位置，源程序编辑器在冲突集 C 中选举出权值最高的命令主体作为仲裁者从 C 中选择一个编辑控制命令执行。

3.3 构建协议，项目服务器，构建服务器及构件库

(1)构建协议是沟通项目服务器和构建服务器的桥梁，项目服务器根据开发人员需要向构建服务器发出编译，构建，调试等请求，而构建服务器在执行该请求并且构造响应返回项目服务器。构建协议采用 HTTP 协议同样的消息格式，也是基于请求-响应模型的协议。

构建协议：:= 请求行|响应行
*(通用消息头 CRLF)
CRLF
[消息体]

构建协议的请求由请求行，请求头，请求体三部分构成。请求方法有 4 个：①configure，用于检查/配置项目构建环境；②compile，用于编译单个源文件；③make，用于构建编译单个项目所有文件并且连接；④debug，用于调试单个项目，构建协议关于调试项目部分的子命令与 gdb 命令句法兼容。图 6 给出了上述请求方法的常见例子。

构建协议的主要消息头有：①Date，日期；②Platform，平台，其值表示形式限定在构建协议规范所规定的操作系统平台集合；③Language，语言，其值表示形式限定在构建协议规范所规定的程序设计语言集合；④Set-Environment-Varaible，设置环境变量；⑤Include，项目所依赖库的包含目录；⑥Lib，项目所依赖库的库目录等。

```
configure -default project1 BP/0.1      //配置项目 project1 的构建环境。
configure -cc project1 BP/0.1           //检查项目 project1 的构建环境的一致性。
make project1 BP/0.1                    //构建项目 project1。
debug project1 BP/0.1                   //调试项目 project1。
```

图 6 常见的请求方法示例

构建协议的响应由响应行，响应头，响应体三部分构成，构建响应是对构建结构的反映，构建协议设计了一组 RDP 兼容的子协议以支持远程桌面，便于开发人员观看编译、连接、调试等各种构建任务的输出。

(2)构建服务器对项目服务器的项目构建请求分两阶段处理：①协议预处理阶段，器通过对原始请求进行预处理，提取出构建项目所依赖的具体协议处理器的关键字 K；②由协议处理器调度程序查找关键字 K 所对应的协议处理器来处理原始请求。第一阶段的任务是确定构建系统(vc/nmake/windows，javac/ant/java，还是 gcc/gmake/linux)，第二阶段的任务用第一阶段选定好的构建系统对项目进行构建。协议处理器被设计成插件的形式，这样构建服务器可根据需要扩展其功能。

(3)构件库是一个软件包的库，在现代项目开发过程中广泛使用到各种软件包作为其内部构件，而这些软件包在安装配置上存在多种问题：①安装配置文档不全；②并不是所有的开发人员对多种构件都懂得安装配置；③构件依赖冲突严重；某些构建配置环境要求苛刻等。显然，传统的软件开发环境要求每台装有集成开发环境的主机都安装配置多个构件，这个工作在软件项目开发过程当中重复性高，工作量需求大。而系统在项目服务器集中存储，安装，配置所有构件，构建服务器则根据项目服务器的构建请求按需从项目服务器读取构件，在本地缓存一份构件拷贝，这样，原来需要在多个主机安装配置构件的工作量，减少为仅在一台主机的工作量，同时，也由多个失效点减少至单个失效点。为解决构件依赖冲突问题，系统在构件库建立构件依赖树，在项目设计时给出一个构件依赖冲突报告以阻止因为构件依赖冲突带来的软件开发返工问题。构件分布于项目服务器和构建服务器，项目服务器维护有一个全局的构件库，构建服务器仅维护有它所构建项目所需要的构件。

4 系统的应用及效果

DVCS 是一个用于分布式环境检测漏洞的软件项目，系统作为 DVCS 的一个子项目为 DVCS 提供了集成开发环境。参加项目的 6 个人通过浏览器访问项目服务器开展各项软件开发任务，项目文档，源代码，以及构件都存储在一台项目服务器上，另外，我们选择了两台主机作为构建服务器，一台是基于 Linux 的 gcc/gmake 系统，一台是基于 Windows 的 vc/nmake 系统。具体开发流程见图 7。

第一步：管理人员利用系统提供的需求分析工具输出需求分析文档。

第二步：设计人员利用系统提供的模型设计器针对第一步输出的需求分析文档进行设计并且输出软件模型文档。

第三步:①管理人员利用系统提供的项目配置管理器配置项目开发所需构件。②管理人员利用系统提供的项目进度管理器针对第二步输出的软件模型文档分配任务并且输出开发任务书。

第四部:开发人员利用系统提供的源程序编辑器针对第三步输出的开发任务书输出源程序。

第五步:①管理人员利用系统提供的项目质量管理器针对第四步输出的源程序进行测试并且输出质量报告书,转第四步。②管理人员利用系统提供的项目配置管理器针对第四步输出的源程序进行版本控制。

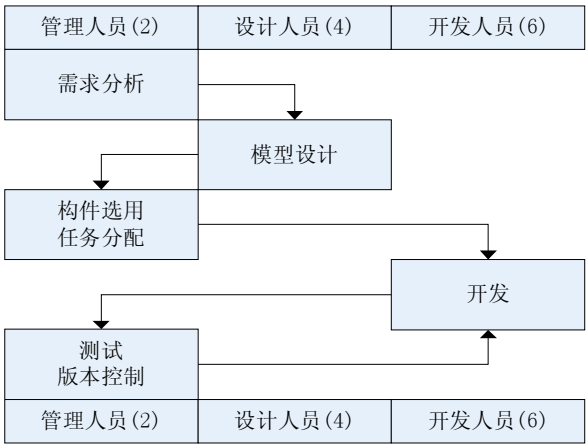


图 7 DVCS 开发流程

5 结 语

在 DVCS项目实施过程中,①项目开发环境配置需求从传统的每人一个集成开发环境减少至每项目组一个开发环境;②项目组成员通过协同工作网络增强了沟通,加快了开发进度。但是仍然存在着一些问题,①远程调试项目时,因为使用 RDP 协议作为远程桌面协议,因此调试效率比较低,相比本地调试,速度要慢一些; ②系统在配置管理上因为开放了 CVS 和 Subversion 接口,因此版本控制入口不再单一,导致版本控制上出现了混乱。

参考文献

[1]杨芙清. 软件工程技术发展思索[J]. 软件学报.2005,16(1)

[2]伍恒,张卫民,王靖. 软件的分布式协同开发环境[J]. 吉首大学学报,2003,24(1)

[3]Justin Elsberry,Nicholas Elsberry. Using XML and SVG to Generate Dynamic UML Diagrams[EB/OL]. http://www.cwu.edu/~gellenbe/docs/xmltouml/xmltechnicalreport.html.

[4]田丽从,张莉,周伯生. 基于 UML 的集成化软件开发环境的研究与实现[J]. 北京航空航天大学学报,2003,29(10)

[5]王明文,朱清新. 基于 UML 的 XML Schema 设计[J]. 电子科技大学学报,2006,35(3)

[6]龙鹏飞. 基于 UML 与 XML 的软件输出文档模型设计[J]. 计算机应用与软件,2008,25(3)

[7]Mihail Ionescu,Ivan Marsic.Tree-Based Concurrency Control Indistributed Groupware[J]. ACM,Computer Supported Co-operative Work, 2003,12(3): 329~350

[8]景栋盛,杨季文,朱晓旭. 协同编辑器中并发算法的研究与实现[J]. 计算机工程与设计,2007,28(2)

Design and Implementation of an Integrated Software Development System

WANG Xiao-ye[1] , ZHANG Lin[2]

(1. School of Information Science and Technology, Zhanjiang Normal University, Zhanjiang 524048;
2. College of Information Science and Technology , Northwest University, Xi´an 710069)

Abstract: Based on the analysis of the problems which brought in the process of modern software development, proposes a distribution cooperation integrated development environment system. Discusses the composition and the characteristics of the system, and also the key technologies which includes the implementation of model design controller, source code editing controller and construct protocol, project servers, construct servers and component liberary, and gives the application and effect.

Keywords: Distribution; Cooperation; Development Environment; DCIDE