软件原型系统在软件项目开发中的应用

姜 文,刘立康
(西安电子科技大学 通信工程学院,陕西 西安 710071)

摘 要:软件原型法是软件开发的一种工程方法。原型法通常是建立系统的基本结构,实现基础模型;然后采用修改与增加相关构件和功能来开发软件系统。依据软件开发的工作实践,结合软件原型和基线的概念提出了新的软件原型系统定义。软件原型系统包含四个要素:需求分析、软件架构设计、基础软件、软件运行环境。通常软件原型系统可以作为软件产品开发的初始基线。之后依据一个软件开发实例,详细叙述了软件原型系统在该软件开发过程中的应用。最后介绍在云环境中软件产品原型系统的开发实践,叙述了 IMS 产品的层级结构,包括 IMS 硬件设备的层级结构和基于云环境的 IMS 软件产品层级结构;叙述了云化 MRP 产品原型系统和云转码产品原型系统开发。软件项目开发实践表明,软件原型系统可以应用于复杂环境中的大型软件开发,且应用前景广阔。

关键词:软件原型系统;需求分析;版本;基线;迭代开发;云环境
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2019)04-0110-06
doi:10.3969/j.issn.1673-629X.2019.04.023

Application of Prototype System in Software Development

JIANG Wen, LIU Li-kang
(School of Telecommunication Engineering, Xidian University, Xi'an 710071, China)

Abstract: Software prototyping is a kind of project method to software development. Prototyping usually sets up the basic structure of the system and implements basic model. Then the software system is developed by modifying and adding related components and functions. According to the practice of software development, combined with the concept of software prototype and baseline, we put forward the new definition of software prototype system. Software prototype system contains four elements: requirement analysis, software architecture design, basic software, software operating environment. Usually software prototype system can be used as the initial baseline of software product development. Then according to a software development example, the application of the software prototype system in the software development process is described in detail. Finally the practice of software prototype system development in a cloud environment is introduced, description of the hierarchical structure of IMS including the hierarchical structure of the hardware of IMS and IMS software product hierarchy based on cloud environment, as well as the development of cloud MRP prototype system and cloud transcoding prototype system. The software project development practice shows that the software prototype system can be used for complex software development environment, with broad application prospect.

Key words: software prototype system; requirement analysis; version; baseline; iterative development; cloud environment

0 引 言

软件原型法[1-6]是软件开发过程中常用的一种软件工程方法。原型法的基本思想是根据用户提出的需求,由用户与开发者共同确定系统的基本要求和主要功能,并在较短时间内建立一个实验性的,简单的小型系统。确定需求的可行性之后,再在原型系统的基础上,进一步开发正式的软件产品。软件原型法符合人们认识事物的规律,软件原型法的灵活应用可以缩短开发周期、提高开发效率、降低开发成本。

依据软件开发的工作实践,结合软件原型和基线的概念提出了新的软件原型系统定义。软件原型系统包含四个要素:需求分析、软件架构设计、基础软件、软件运行环境。建立软件原型系统的过程中可以充分利用面向对象、软件复用[7-8]等技术。软件原型系统的

收稿日期:2018-04-30 **修回日期**:2018-09-05 **网络出版时间**:2018-12-20
基金项目:国家部委基础科研计划:国防预研基金项目(A1120110007)
作者简介:姜 文(1986-),女,工程师,硕士,CCF 会员(E200032324M),研究方向为图像处理、数据库应用和软件工程;刘立康,副教授,研究方向为数字通信、图像处理和软件工程。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20181219.1542.080.html

思想与方法可以用于复杂环境下的大型软件系统的开发。

1　原型法

原型法又称快速原型法(rapid prototyping),是近年来提出的一种开发软件系统的新方法。原型(prototype)即样品、模型的意思。在软件开发过程中,"原型"可以形象地表示系统的一个早期可运行版本,能够反映该系统的部分重要功能和特征。

1.1　原型法定义

原型法(prototyping)是指在获取一组基本的需求定义后,利用高级软件工具可视化的开发环境,快速地建立一个目标系统的最初版本。

软件原型的基本要素包括:体现软件主要的功能、建立软件的基本架构。

1.2　原型法分类

原型主要分为三种类型:

1.2.1　抛弃型原型(throw-it-away prototype)

这种原型用于评价软件系统的某个特性,以便更准确地确定需求,或更严格地验证需求。使用完后抛弃该原型系统,然后建立完整的、正式的原型系统。

1.2.2　进化型原型(evolutionary prototype)

此类原型的构造从软件系统的基本需求出发,通过修改和扩充建立原型;最后,在此基础上开发软件系统。

1.2.3　增量型原型(incremental prototype)

原型系统是一次一段地增量构造,逐渐叠加子系统的需求功能,形成软件开发不同阶段的原型系统;增量型原型是软件系统在迭代开发不同阶段建立的原型系统。

软件系统通常采用迭代开发,是逐步完成的,每次迭代都要对系统重新进行需求分析、规格说明、软件设计。根据不同阶段的需求分析,需要建立不同形式的原型系统。

2　软件原型与基线的关系

2.1　软件原型

软件原型是指软件系统开发初期,开发者采用原型法,快速形成一个能反映系统主要功能及特征的简单版本(如软件界面与布局、基本功能等);在后续开发过程中不断地修改、测试、完善,形成软件产品。软件原型与软件产品的初始基线版本有类似之处。

2.2　基线的定义

基线[9](baseline)是软件开发的一个重要的基本概念,有多种定义,这里介绍基于版本的基线定义。

基于版本的基线定义:基线是项目文档或者源代码等文件的一个稳定版本。基线有名称、版本、标识符、日期等属性。它是进一步开发的基础,其变更必须通过正式的变更程序。

如果某一产品版本被定为基线,那么它就被冻结,要想变更基线必须建立一个新的版本。

2.3　初始开发基线

软件产品由设计阶段转为迭代开发阶段,需要确定软件的初始开发基线。可以结合原型法和基线的方法建立软件产品开发的第一个版本,作为软件产品的初始开发基线。

3　软件原型系统

依据软件开发的工作实践,结合软件原型和基线的概念赋予软件原型系统新的内涵。

3.1　软件原型系统的定义

软件原型系统的定义:软件原型系统包含了四个要素:

(1)需求分析:软件产品需要实现的功能。

(2)软件架构设计:实现软件功能的软件架构(模块、界面和布局)。

(3)基础软件:软件开发的基础软件平台、软件模块。

(4)软件运行环境:操作系统、数据库、数据资源、其他辅助软件。

四个要素中的前两条构成通常软件原型的概念,通常只能用于小型软件产品(如游戏软件)的开发;整合后两个要素后可以用于复杂环境中大型软件开发。通常软件原型系统可以作为软件产品开发的初始开发基线。

3.2　原型系统应用实例

为了进一步理解软件原型系统的概念,下面介绍一个原型系统应用实例,该实例是手写体维吾尔文字符识别软件。

3.2.1　软件实例原型系统的四个要素

(1)需求分析。

输入的手写体维文字符,给出相似度最高的字符(1-5 个),从中选择一个字符作为该字符的识别结果。(这种方法类似于手机中的手写体汉字输入)

(2)软件架构设计。

软件界面选择对话框方式,采用 VC6.0 编程,生成对话框类 class CUrghurDlg:public CDialog,程序启动后显示对话框,如图 1 所示。

图 1 中的软件界面由七部分组成:选择数据库所在文件夹;选择测试参数;读取手写体字符数据库中字符图像文件时依次显示字符图像和字符序号;输出实验结果;KNN分类器参数选择;点击按钮提取训练组

图 1 手写体维文字符识别软件界面

样本特征,训练结束后点击测试按钮开始测试测试组样本,点击统计按钮给出测试结果;滑动条显示程序执行的进度。

(3)基础软件模块。

选择图像处理软件模块 class CDib2(包含 18 个图像处理函数成员和 14 个数据成员)作为软件开发的基础模块,在图 1 对应的对话框类 class CUrghurDlg 中加入数据成员 CDib2 Dib。

(4)软件开发环境与运行环境。

软件在 Windows 操作系统下运行,数据资源为 115 组手写体维文单字符图像,每组包含 128 个维吾尔文单字符。从字符数据库中选取 70 组样本进行训练,剩余 45 组进行测试。其中,测试的样本组不参与训练。

3.2.2 软件实例迭代开发过程

软件原型系统建立之后,可以作为软件初始基线,开展软件迭代开发工作。软件开发过程如表 1 所示。

表 1 软件开发过程

软件开发过程			实例(手写体维文字符识别)
建立软件原型系统	需求分析	手写体维文字符识	识别输入的手写体维文字符,给出相似度最高的(1-5 个)字符
	软件架构	软件界面和架构	图 1 对应的对话框类 class CUrghurDlg:public CDialog
	基础软件	图像处理函数包	图像处理软件模块 class CDib2(包含 18 个图像处理函数成员和 14 个数据成员)
	环境	软件运行环境	Windows 操作系统和手写体维文单字符图像库
迭代 1	特性 1	字符图像预处理	二值化、细化处理、倾斜校正处理、归一化处理
迭代 2	特性 2	特征提取算法	1. 基于方向线素的特征提取算法 2. 基于实值 Gabor 滤波器的特征提取算法
迭代 3	特性 3	分类器识别算法	KNN 分类器(K-最近邻域分类算法)
迭代 4	特性 4	对字符进行分类识别,给出实验结果	编程实现如下三个程序 1. 提取训练组样本特征 2. 开始测试(识别测试组样本) 3. 给出测试实验结果

从表 1 中可以看到软件开发的完整过程,软件原型系统提供了软件的雏形和基本结构,也为后续的迭代开发提供了稳定的初始基线版本。

该软件主要用于手写体维文字符识别算法研究,其架构比较容易改造为维文字符识别的应用软件产品。从软件实例可以看出,构建合适的软件原型系统有助于提高软件开发效率和质量。

4　基于云环境的软件原型系统开发

随着计算机技术的发展,以及云计算技术在各行各业的应用普及,基于云计算的应用软件的种类也越来越多。许多传统环境下运行的软件、硬件产品需要升级推出云化版本。在通信网中部分硬件设备不断推出在云环境中具有同样功能的云化软件产品。

4.1　IMS 产品的层级结构

IMS[10-12](IP multimedia subsystem)是未来统一核心网的基础,是 IP 网上进行多媒体通信的可运营、可管理、可增值的一种完整解决方案。IMS 采用统一的 ATCA(advanced telecommunications computing architecture,先进的电信计算平台)硬件平台和 CGP(carrier grade platform,电信级平台)软件平台。

ATCA 硬件平台,是为下一代融合通信及数据网络应用提供的一个高性价比的,基于模块化结构的、兼容的、可扩展的硬件构架。

CGP 软件平台是建立在 ATCA 硬件平台上的高可用、高性能、易运维、易集成的软件平台。它可以提供硬件的管理能力,多网元的维护管理能力,支撑业务进程开发的底层功能组件,支撑业务进程开发的开发环境。

IMS 产品的云化过程是在云环境中实现多媒体通信运营。这需要将原来硬件设备转化为在云环境实现同样功能的软件产品,原来由硬件单板提供的进程功能改由云环境中虚拟机的软件模块来提供。

4.1.1　IMS 硬件设备的层级结构

IMS 硬件设备的层级结构如表 2 所示。

表 2　IMS 硬件设备层级结构

序号	类别	层级	配置项
1	业务软件	产品业务	IMS(产品业务软件)
2	软件平台	CGP 平台	后台 OMU
3	内存数据库	数据库	后台 Oracle
4	操作系统	操作系统	Linux 或者 Vxworks 操作系统
5	硬件	硬件 ATCA 单板	硬件单板

CGP 软件平台负责软件、硬件、网络的管理维护工作;业务软件通过硬件单板处理各种业务。

4.1.2　基于云环境的 IMS 软件产品层级结构

基于云环境[13-14]的 IMS 软件产品是 IMS 硬件设备的云化产品,其层级结构如表 3 所示。

表 3　基于云环境的 IMS 软件产品层级结构

序号	类别	层级	配置项
1	业务软件	产品业务	产品业务软件
2	软件平台	业务开发平台	业务功能开发的公共服务软件
3		管理平台	CGP 平台(OMU)
4	云环境	操作系统和数据库	Linux 和 Oracle
5		虚拟计算节点	虚拟机
6		虚拟网络设备	私有云(VPC)网络
7		基础网络	单位私有云或者企业云

与 IMS 硬件设备不同,表 3 中 1 ~ 3 构成 IMS 软件产品,4 ~ 7 为云环境;在云环境中安装软件产品后使用。CGP 平台的后台软件 OMU 由网络管理软件和客户端工具组成,提供人机接口,管理虚拟网络的各种事务,负责安装、管理、维护软件。

4.2　云化 MRP 产品原型系统开发

硬件 MRP 实现 MRFP(multimedia resource function processor)实体的功能,提供多媒体资源的承载功能,与 MRC6600 一起实现媒体资源的信令与承载控制分离。

4.2.1　云化 MRP 产品原型系统

云化 MRP(multimedia resource processor)产品是硬件 MRP 云化后,在云环境中运行实现同样功能的软件产品。云化 MRP 产品的原型系统如表 4 所示。

系统工程师(SE)根据客户需求进行需求分析之后,给出原型系统可行的设计方案,在云环境中开发新产品 MRP,选择非云化产品 SE2900 某个版本开发云化 MRP 软件原型系统中的基础软件。

表 4　云化 MRP 原型系统

原型系统要素	配置项
需求分析(基本功能)	实现硬件 MRP 的功能:提供多媒体资源的承载功能。支持放音收号、传真、多媒体播放和录制、音/视频会议、彩铃或彩影多媒体音/视频业务
软件架构	通过硬件 MRP 的业务软件模块搭建云化 MRP 软件业务软件模块架构:媒体资源处理模块、H.248 消息处理模块、连接管理和呼叫控制模块、承载处理及转发模块、IP 资源承载模块
基础软件	选择 SE2900 某个版本的代码,搭建业务开发平台和 CGP 平台(OMU)
运行环境	表 3 中的云环境

SE2900 是一个全功能的、高性能的、表现稳健的运营商级会话边界控制器[15](session border control,SBC)。主要提供呼叫接入控制、协议互通、网络间安

全、NAT 穿越、QoS、接入安全、媒体防火墙、媒体代理、媒体编解码转换和计费等功能。

选择 SE2900,主要基于以下原因:

(1) SE2900 硬件设备中采用 Linux 操作系统,CGP 平台方便移植到云环境中。硬件 MRP 中采用 Vxworks 操作系统,移植到云环境中不方便。

(2)云化产品具有统一的源代码结构,SE2900 涉及到的业务比较多,可以对云化 MRP 软件产品开发提供更多支持。

4.2.2 *云化 MRP 软件原型系统开发*

1. 软件原型系统的开发构建。

软件工程师负责云化 MRP 产品软件原型系统的开发构建工作:

(1)获取 SE2900 某个版本的代码,将其中的 CGP 软件平台移植到云环境的虚拟机上(虚拟机操作系统为 Linux);

(2)通过 SE2900 版本的代码构建基本软件业务开发平台;

(3)将硬件 MRP 的业务软件模块的源代码拷贝到 SE2900 源代码的 host 文件夹下,构建初始的业务软件模块架构;

(4)针对各模块编写 makefile 文件,启动业务软件模块代码、基本软件业务开发平台与 CGP 平台接口的编译,修正编译过程中的错误,实现对 CGP 平台接口的适配开发;

(5)软件工程师开发相关的 MML 命令,为安装、编译、调试软件版本包做准备;

(6)所有模块都能够编译通过后,将代码提交 SVN 版本库。

2. 持续集成。

持续集成工程师负责集成构建:

(1)从 SVN 版本库下载相关代码;在持续集成工具 ICP-CI 的页面上搭建编译出包工程,统计各模块与全部代码量、对所有模块执行静态检查;执行各模块编译;LMT 客户端的安装包与产品版本包的出包与归档。

(2)在后续的工作中及时完成代码更新与版本包的编译出包,为原型系统的调试验证提供版本包。

3. 软件原型系统测试。

测试工程师完成如下工作:

(1)准备测试环境:搭建私有云平台,在云环境中安装 CGP 平台,然后再安装持续集成工程师提交的版本包。

(2)根据软件原型系统开发的需求,编写软件原型系统测试的测试用例。

(3)开展测试工作,将测试结果反馈给云化 MRP 产品的系统工程师和软件工程师。

4. 软件原型系统调试。

在测试环境上启动调试工作,根据存在的问题,定位问题,修改代码。反复进行出包、测试、调试工作,直到所有问题都已解决。

软件原型系统开发完成后,可以作为云化 MRP 产品的正式启动开发的初始基线,开展云化 MRP 产品的迭代开发工作。逐步实现云化 MRP 产品的虚拟机弹性扩缩容等云化基本功能以及在云化平台上实现硬件设备 MRP 产品的所有功能。

需要同时开发业务软件和业务软件开发平台,在云环境中实现原来由硬件单板实现的功能。在开发过程中经常出现持续集成构建失败,在业务软件、软件平台和云环境三者的磨合过程中处理了大量的技术上的问题。

4.3 云转码软件产品原型系统开发

云转码(cloud video engine,CVE)产品是以云化 MRP 产品为基础开发视频文件点播与直播转码视频文件的产品。可以看作 IMS 在云环境中应用层的软件产品。

4.3.1 *云转码软件产品原型系统*

云转码软件产品的原型系统如表 5 所示。

表 5 *云转码软件产品原型系统*

原型系统要素	配置项
需求分析(基本功能)	视频文件点播与直播转码视频文件,视频文件采用 HLS+H264、HLS+H265、DASH+H264、DASH+H265、HLS+H264+H265、DASH+H264+H265 等方式,进行转码
软件架构	搭建云转码软件的业务软件模块架构:业务调度;转发处理;媒体处理
基础软件	选择云化 MRP 的某个版本的代码,搭建业务开发平台和 CGP 平台(OMU)
运行环境	表 3 中的云环境,数据资源:转码片源文件(视频文件)库

4.3.2 *云转码软件原型系统开发*

云转码软件原型系统开发与 4.2.2 中的云化 MRP 软件原型系统开发基本相同。软件工程师负责构建软件原型系统,持续集成工程师负责集成构建。测试工程师负责软件原型系统测试。

在测试过程中不同之处在于测试工程师在完成搭建测试环境之后;需要准备视频素材库:准备测试的视频文件,使用工具 MediaInfo[16] 完成视频文件参数检测,使用 FFmpeg 工具完成视频文件典型场景截取等处理;然后开展测试工作。测试结果反馈给系统工程师,软件开发工程师,根据存在的问题进行原型系统的调试。

软件原型系统开发完成后,可以获得软件产品开

发的初始基线,下一步开展云转码软件的迭代开发工作。在开发云转码产品的同时,希望同时开发一个通用的基础软件平台(包括软件业务开发平台和 CGP 平台)。该平台除了为云转码产品各软件业务模块提供公用函数功能支持外,还能为更多的软件产品提供开发平台。在后期软件迭代开发的过程中需要进行软件联调。软件联调,是指软件通用开发平台和软件产品上线后同时进行相关功能点和性能调试。云转码产品的软件开发工程师、软件平台的开发工程师与测试工程师共同参与联调工作。

基于云环境的软件项目开发实践表明,采用软件原型系统的方法,有助于规范软件开发流程和软件产品层次结构,提高软件开发效率和软件质量。

5　结束语

软件原型系统法是一种有着巨大潜在价值的软件工程方法,建立软件原型系统的过程中可以充分利用面向对象、软件复用等技术。软件原型系统法中的许多问题需要进一步进行深入的研究和探讨。软件项目开发实践表明,应用软件原型系统法,可以提高软件质量,缩短软件开发周期,降低企业的开发成本。软件原型系统可以用于复杂环境和大型软件系统的开发,有着广泛的应用前景。

参考文献:

[1] STEPHENS R. Beginning software engineering[M]. New Jersey, U. S: John Wiley & Sons, Inc., 2015.

[2] GIBSON J. Introduction to game design, prototyping, and development: from concept to playable game with unity and C# [M]. New Jersey, U. S: Pearson Education, Inc., 2015.

[3] OGEDEBE P M, JACOB B P. Software prototyping: a strategy to use when user lacks data processing experience[J]. ARPN Journal of Systems and Software, 2012, 2(6): 219-224.

[4] SABALE R G. Comparative study of prototype model for software engineering with system development life cycle[J]. IOSR Journal of Engineering, 2012, 2(7): 21-24.

[5] 刘天白,朱冯喆. 原型法在软件项目中的运用[J]. 信息化研究,2016,42(1):68-71.

[6] 张希奇. 原型法在企业管理信息系统开发中的应用[J]. 安徽电子信息职业技术学院学报,2014,13(1):31-34.

[7] JACOBSON I, BOOCH G, RUMBAUGH J. 统一软件开发过程[M]. 周伯生,译. 北京:机械工业出版社,2002.

[8] JACOBSON I, GRISS M, JONSSON P. 软件复用结构、过程和组织[M]. 韩　柯,译. 北京:机械工业出版社,2003.

[9] 姜　文,刘立康. 软件配置管理中的基线问题研究[J]. 计算机技术与发展,2016,26(6):6-10.

[10] 梁雪梅,方晓农,杨　硕,等. IMS 行业专网应用[M]. 北京:人民邮电出版社,2016.

[11] 韩　妮. IMS_SIP 系统关键技术及其视频会议原型软件研究[D]. 西安:西安电子科技大学,2008.

[12] 李广野. IMS_SIP 会话边界控制器的设计与实现[D]. 沈阳:中国科学院沈阳计算技术研究所,2009.

[13] 陈国良,明　仲,冯禹洪. 云计算工程[M]. 北京:人民邮电出版社,2016.

[14] 姜　文,刘立康. 基于云环境的大型应用软件联调[J]. 计算机技术与发展,2018,28(2):14-18.

[15] 齐幸辉,张　庚,刘　革. 会话边缘控制器关键技术实现[J]. 无线电通信技术,2014,40(1):75-78.

[16] 姜　文,刘立康. MediaInfo 工具在视频文件管理中的应用[J]. 计算机技术与发展,2018,28(4):36-41.