

## 基于 CAFISE Framework 的高适应性面向服务软件开发<sup>\*</sup>

李刚<sup>+</sup>, 赵卓峰, 韩燕波, 梁英

(中国科学院 计算技术研究所 网格与服务计算研究中心, 北京 100080)

### CAFISE Framework Based Development for Service Oriented Applications with High Adaptability

LI Gang<sup>+</sup>, ZHAO Zhuo-Feng, HAN Yan-Bo, LIANG Ying

(Grid and Service Oriented Computing Research Center, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-88449181 ext 620, Fax: +86-10-88455011, E-mail: ligang@ict.ac.cn, <http://www.ict.ac.cn>

Li G, Zhao ZF, Han YB, Liang Y. CAFISE framework based development for service oriented applications with high adaptability. *Journal of Software*, 2006,17(6):1372-1380. <http://www.jos.org.cn/1000-9825/17/1372.htm>

**Abstract:** In enterprise information systems and e-government systems, the problems caused by the changes of heterogeneous distributed resources and requirements are becoming more and more serious. How to make an application accommodate to these changes rapidly has become a hot research topic. This paper presents an adaptive service-oriented software framework, named as CAFISE (convergent approach for information system evolution) Framework for supporting business-end programming, and a CAFISE Framework based development method for service-oriented adaptive software. The CAFISE Framework enables to encapsulate the heterogeneous resources into business services and provides strong supports for open and dynamic service-oriented software architecture. Beginning with analyzing the environment factors that affect an application, the CAFISE Framework based development method supports to design software architecture according to the changes that the application will adapt to, with quality feature driven architecture analysis, and facilitates the just-in-time development and evolution of service oriented applications by architecture based business-end programming. The actual applications and experimental results show that the service-oriented enterprise information systems and e-government systems developed with the framework and the method of this paper can accommodate better to the changes of heterogeneous resource and business requirements, and can continuously evolve with lower costs.

**Key words:** software architecture based development; adaptability; service oriented architecture; software architecture; service composition

**摘要:** 在企业及电子政务应用中,由资源分布异构及需求动态变化而产生的问题越来越严重,如何让应用快速适应这些变化一直是人们关注的一个研究难题.首先给出了一个面向服务的支持业务端编程的适应性软件结构框架 CAFISE(convergent approach for information system evolution) Framework,然后给出了基于该框架的面

<sup>\*</sup> Supported by the National Natural Science Foundation of China under Grant Nos.90412005, 90412010 (国家自然科学基金)

Received 2006-01-08; Accepted 2006-03-28

向服务的适应性软件开发方法.CAFISE Framework 对异构资源的服务化、业务化及开放动态的面向服务软件体系结构提供了较好的支持.基于该结构框架的开发方法,从对影响软件的环境要素分析入手,通过质量属性驱动的体系结构分析,重点针对应用要适应的变化进行结构设计,最终通过以体系结构为基础的业务端编程,实现面向服务应用的即时开发与演化.在实际项目中的使用及实验证明:用该框架和方法开发的面向服务应用,能够较好地适应异构资源的动态变化及用户业务变更,能以较低的代价实现企业及电子政务面向服务应用的持续演化.

关键词: 基于软件体系结构的开发;适应性;面向服务的软件体系结构;软件体系结构;服务组合

中图法分类号: TP311 文献标识码: A

随着信息技术在各个应用领域的普及和信息化建设的深入,企业及政府部门信息系统中由于资源异构及需求变化而引起的问题越来越突出.经过多年的信息化建设,企事业单位内部已经存在着大量的遗留系统.由于这些系统当时的建设目标、面向的用户和应用范围都不尽相同,致使系统使用方式各异,所处理的数据、采用的流程及输出结果也都没有统一的标准.相互协作的不同企事业单位之间情况更加严重.不仅存在资源异构的问题,而且各异构资源分属于不同的拥有者,可以自主地动态演化.而随着业务的发展,迫切要实现企事业单位内部、外部各种数据、应用资源的协同共享.如何让应用能够适应这些异构、动态的资源,已经成为企事业单位信息系统中一个亟待解决的问题.另外,用户的业务也在不断发展变化.用户需要一种开发方法,该方法能够支持根据业务需要快速构建应用,实现从业务需求到软件应用的便捷过渡,以适应动态变化的新需求.同时,用户业务的变化也向现有信息系统提出了适应性要求,要求它们能够随需求动态变更,即时演化.另外,近年来网络计算环境越来越具动态性,也使这些问题更加严重.如何解决这种由异构资源和需求变化而产生的软件适应性问题,受到了各方面的广泛关注.

目前,与该问题相关的研究工作已有不少,这些工作从不同角度对如何针对需求变化快速构建应用、如何使用应用动态演化、如何让应用适应资源变化等问题进行了研究,并在不同的方面都有所贡献,但问题远没有得到解决.

为了解决上述问题,本文给出了一个支持适应性面向服务软件开发的体系结构框架 CAFISE(convergent approach for information system evolution) Framework,并对基于该框架的开发方法进行了介绍.CAFISE Framework 对异构资源的服务化、业务化及开放动态的面向服务软件体系结构提供了较好的支持.基于该结构框架的开发方法从对影响软件的环境要素分析入手,通过质量属性驱动的体系结构分析,重点针对应用要适应的变化进行结构设计,最终通过以体系结构为基础的业务端编程实现面向服务应用的即时开发与演化.

## 1 相关工作

为了明确本文所要解决的主要问题及其意义,在此首先对现有的相关工作进行分析比较.

美国东北大学的 Lieberherr 等人提出了一种适应性软件设计方法——Demer 方法<sup>[1]</sup>.Demer 方法建立在面向对象思想的基础上,认为类结构也是一种可变的环境因素,在类结构变化时,应用程序也应进行相应的调整,其中用于实现适应的核心机制是可增殖模式中的可遍历小规约.Demer 方法是一种比较完整的适应性软件开发方法,尽管它偏重于结构风格程序设计,解决的问题主要针对软件层面上类结构的变化,但对解决由需求及资源变化引起的适应性问题确有借鉴之处.

软件体系结构是对软件系统静态和动态拓扑结构的描述,可以通过构件、构件连接和结构约束来刻画.软件体系结构为软件的适应性调整提供了一个基本框架.目前,对软件体系结构的研究集中在体系结构描述语言、体系结构形式化、体系结构分析、基于体系结构的开发、动态软件体系结构及体系结构的设计与选择等方面<sup>[2,6,8]</sup>.其中,动态软件体系结构是与软件适应性密切相关的一项工作.这部分工作主要集中在动态体系结构描述语言上,较具代表性的有 Rapide<sup>[9]</sup>,Darwin<sup>[10]</sup>等.在基于软件体系结构的开发上,较有代表性的是卡内基-梅隆大学提出的基于体系结构的开发方法<sup>[5]</sup>.此外,Peyman 等人对在软件体系结构基础上开发适应性软件进行了

探讨<sup>[11]</sup>,并将开发过程分为演化管理和适应管理两部分:前者对适应性软件生命周期进行了粗略描述;后者简要给出了软件适应性修改的方法.该方法的创新之处在于将软件体系结构、适应等思想集成到了一个综合性方法中,但该方法只是框架性的,能适应什么样的变化不明确.

软件体系结构的变更管理也是当前适应性软件体系结构研究的重要内容,它从变更管理角度出发,为适应性软件开发提供了新思路.软件变更可以归纳为以下几个阶段:变更需求获取、变更影响分析、变更验证、变更实施、文档记录,而变更影响分析是其中的关键阶段<sup>[20]</sup>.在文献<sup>[21]</sup>中,研究人员提出了一种软件体系结构变更影响分析方法,并期望通过该方法界定体系结构组件变更时涉及的影响范围,以解决软件体系结构变更的正确性及一致性等问题,从而更好地支持软件体系结构变更.

MDA(model driven architecture)是OMG(object management group)提出来的一种软件体系结构标准,包括一种软件系统设计和实现的方法<sup>[12]</sup>.MDA在开发中将软件功能与功能实现分离,支持创建类似系统时重用系统模型,使软件开发对新技术变化的适应能力更强,以解决由于新技术、新平台的出现,原结构设计不能再用的问题.但是,这种方法并不能解决由于软件运行后,资源、需求等环境变化产生的软件适应性问题.

SOA(service oriented architecture)是以Web服务为基础的,用于实现分布式应用系统的结构,它包括SOAP, UDDI, WSDL等规范.Web Service是组件技术在Internet中的延伸,从本质上讲是放置于网络上的可重用构件.面向服务的体系结构为如何描述、发现、调用Web服务,提供了最基本的支持.SOA的优势是它支持将异构资源服务化,通过标准的服务增强了软件构件的互操作性,为灵活地组装软件提供了基本支持.目前, Websphere, Weblogic等平台都提供了对Web服务的支持.但是,SOA目前还停留在抽象的高层概念模型层面上,还没有一个实用、具体的应用参考模型<sup>[13]</sup>.在对环境变化的适应性方面更是欠缺.

密歇根州立大学的Zhang等人针对软件的适应性调整问题,提出了一种保证在运行时行为调整之后软件正确性的方法.该方法将软件的适应性行为说明和非适应性行为说明分开,使对软件的适应性调整与分析更加方便、安全<sup>[3]</sup>.与该项工作相比,本文所述成果能够在体系结构上提高面向服务软件对用户需求及服务资源变化的适应性,不仅可以增强软件的行为适应性,而且可以增强软件的结构适应性.

基于资源需求说明和行为适应的思想,CASA Framework使基于该框架开发的软件能够根据执行环境的变化,动态调整自己,以便在运行时改变应用的功能性和非功能性属性<sup>[4]</sup>.与CASA Framework相比,本文所提出的软件框架及基于该框架的开发方法,能够在业务层面上支持面向服务应用的开发和适应性调整,不仅可以支持应用的行为调整,而且可以通过服务的动态增加支持应用的结构调整.

此外,人们还在基于模型的程序自动生成<sup>[3]</sup>、适应性工作流<sup>[7]</sup>等方面,对如何提高软件对需求及环境变化的适应性做了不少工作.其中,软件自动生成及工作流动态调整的思想对通过快速自动重构软件,解决由于需求变化及资源动态变化而产生的软件适应性问题很有帮助.

通过对现有工作的分析可以看出,以下两点仍然是解决由异构资源及需求变化引起的适应性问题的关键:

- (1) 如何实现动态业务需求到软件应用的即时转换?
- (2) 如何适应异构服务资源的动态变化?

为了解决这些问题,本文提出了CAFISE Framework及基于该框架的、面向服务的适应性软件开发方法.

## 2 支持面向服务应用动态演化的软件框架CAFISE Framework

### 2.1 CAFISE Framework体系结构设计思想

CAFISE Framework是一个具有特定质量属性、面向特定领域的可复用、可扩展的软件框架.它基于Oasarm(open adaptive software architecture reference model)模型<sup>[8]</sup>,主要针对企事业信息系统中由异构资源和需求动态变化引起的适应性问题,为灵活地面向服务应用开发在体系结构层面提供大粒度的复用支持,并通过基于该框架的开发方法,扩展、定制体系结构实例,构造可动态扩展可持续演化的面向服务应用.

为此,在体系结构的设计上,我们将用户的功能性需求和系统的适应性需求分开考虑,分别将其设计为计算结构和控制结构.计算结构用于直接实现功能性需求;控制结构能够根据环境变化及计算结构内部状态变化,调

整计算结构.其中:控制结构主要包括用于感知外部变化的感知部件和变化参考实体、控制结构调整的控制部件、管理构件增添等的构件管理部件几部分组成.构件管理部件能够向体系结构中引进新构件,使软件具有结构上的开放性,使软件不仅在行为上具有适应性,在结构上也具有一定的适应能力<sup>[8]</sup>.

### 2.2 框架结构

在上述体系结构设计的基础上,我们设计了如图 1 所示的 CAFISE Framework.它包括应用本身、构建应用的业务端编程工具 VINCA Workshop、VINCA 解释器、运行时对象库、服务访问代理、监控调整工具、context 控制台、连接控制器和服务社区几部分.其中:应用实例、VINCA Workshop、解析器及服务访问代理构成计算结构;监控调整工具、context 控制台、连接控制器和服务社区等实现控制结构的变化感知、结构调整及构件引入等功能.

CAFISE Framework 下面向最终用户的应用包括可重用的服务组合模板和可执行的应用系统.这些应用由服务组合而成,用业务级服务组合语言 VINCA<sup>[14]</sup>描述,直接体现了用户业务层上的需求.应用的执行由解释器、服务访问代理组成的运行基础设施完成.应用实例在控制结构的操纵下可动态扩展、变更.控制结构及计算结构中的运行基础设施和服务组合模板可以由用户定制、重用.

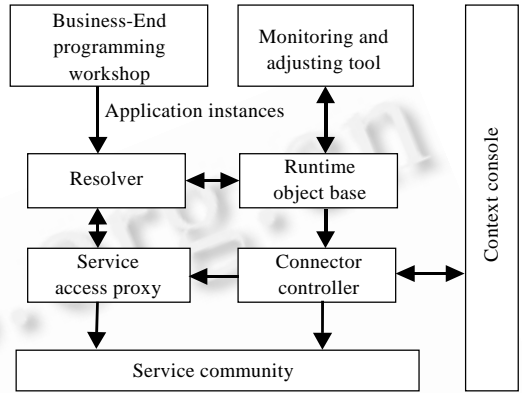


Fig.1 CAFISE Framework

图 1 CAFISE Framework 结构

CAFISE Framework 对应用适应异构资源及需求变化的支持主要有以下几方面:

#### (1) 异构资源服务化、虚拟化



Fig.2 Service community

图 2 服务社区

服务是 CAFISE Framework 下应用的基本运算单元,CAFISE Framework 通过资源服务化、虚拟化解解决资源异构产生的问题.首先,通过标准化的 Web 服务,封装各类异构资源;然后,将多个功能相似的标准服务聚类,形成抽象的服务;最后,基于 OWL-S 对服务进行语义和非功能属性描述,形成业务层面的业务服务.CAFISE Framework 中服务的注册、虚拟化、业务化由如图 2 所示的服务社区统一管理.

#### (2) 软件层与业务层的聚合

CAFISE Framework 通过业务层与软件层的聚合实现了业务需求到可执行软件的高效转换.聚合以 CAFISE 模型为基础<sup>[14]</sup>,在业务服务与基于反演的业务端服务组合基础上实现.CAFISE Framework 对聚合的支持使变化的需求能够快速反映到软件层,从而实现应用的即时构造.

#### (3) 动态开放的结构

在控制结构的支持下,CAFISE Framework 下的应用可以动态配置、动态扩展.这主要体现在服务动态选择、应用构件动态加入和应用实例的适应性调整几方面.CAFISE Framework 中的连接控制器采用服务连接器模型<sup>[15]</sup>解决由于服务资源自主变动而引起的适应性问题.此外,组成应用的服务由服务社区管理,服务可以自由加入、退出社区.因此,组成应用的构件可以动态扩展,应用配置可以动态改变.对应用的适应性调整主要通过监控调整工具实现,在该工具支持下,用户可以通过对运行时对象库中对象的操作,改变应用运行状态,以人工的方式增加新应用构件.

CAFISE Framework 的开放性还体现在与其他开发平台的兼容性上.目前,用 Weblogic8.1, Jbuilder 9.0 开发的 Web 服务及 JSP 应用能够方便地与 CAFISE Framework 集成.这种集成可以使该框架能够充分利用现有商业

开发平台增强自己的功能.

#### (4) 业务端动态演化

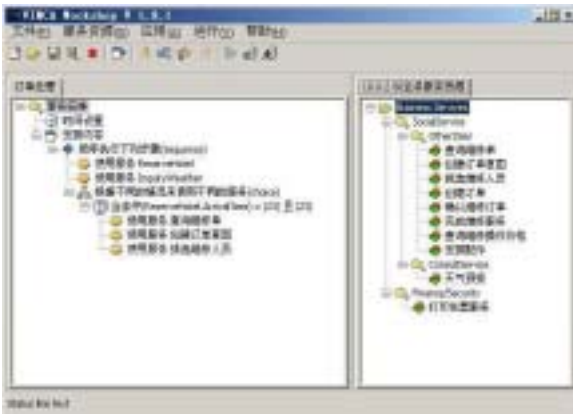


Fig.3

图 3

CAFISE Framework 支持用户在业务层通过服务组合即时构造应用,并即时运行.业务层应用以 VINCA 描述.VINCA 语言是一种业务端服务组合描述语言.利用该语言,用户可以通过组合具有业务领域特征的业务服务满足自己的需求.VINCA Workshop(如图 3 所示)以一种所见即所得的方式,让用户以可视化过程的形式从业务角度构造应用.在此过程中,用户只需从业务角度表达他们的需求,编写应用,无须了解过多的软件编程细节.这种在业务端支持用户编程的能力使得应用不仅可以随需即时构造,而且可以随时调整以适应需求变化.

对应用的调整可以在运行时也可以在非运行时进行.非运行时应用调整通过 VINCA Workshop 由业务端编程实现.运行时调整由监控调整工具完成<sup>[16]</sup>.

### 3 基于 CAFISE Framework 的软件开发

#### 3.1 开发方法

CAFISE Framework 的主要目的是在框架重用基础上,在特定领域内解决由企事业信息系统资源变化和用户需求变化而产生的软件适应性问题.与已有的软件开发方法相比,基于 CAFISE Framework 的软件开发除了包括传统意义上的需求分析、设计等活动外,还包括软件生存环境分析、质量属性驱动的体系结构分析与设计以及基于开放适应性框架的业务端动态演化 3 个有特色的关键活动.软件生存环境分析在需求分析的基础上进行,用于确定影响软件适应的主要因素及应对这些因素变化的基本策略.其结果被用在质量属性驱动的体系结构分析与设计中,以 CAFISE Framework 为基础设计、定制软件开发所用的软件体系结构.基于开放适应性框架的业务端动态演化以质量属性驱动的体系结构分析与设计的结果为基础完成软件的开发与动态调整.

##### 3.1.1 软件生存环境分析

环境变化是引起适应性问题的根源.要解决该问题,首先要明确软件要对什么样的变化进行适应.只有这样,才能将 CAFISE Framework 中支持软件适应性的特征具体化.为此,我们在对影响企事业信息系统的环境因素进行长期统计、分析和研究的基础上<sup>[17]</sup>,提出了软件生存环境分析的方法.

在此,将影响软件存在的一系列相关因素及其相互作用称为软件生存环境.生存环境中影响因素的形式多种多样,可以是运行环境、用户需求,也可以是数据结构、类结构或数据模型等.生存环境分析的目的是确定影响软件的主要变化及其规律,它主要包括以下任务:(1) 确定生存环境范围,明确影响软件的环境所包括的内容,确定生存环境边界.生存环境范围将与软件无关的因素排除出去,界定了生存环境要素的种类.对复杂系统,可首先用 UML 对环境及软件涉及的业务过程进行建模,进一步分析、获得软件、环境及其相互作用的概要描述;(2) 通过分析影响软件功能/非功能属性的环境要素,确定环境要素的变化规律及对软件的影响方式,并考虑环境要素变化的持久性、对环境要素变化的应对策略,为在 CAFISE Framework 基础上进行软件体系结构设计提供支持.

软件生存环境分析结束后,将获得系统的适应性需求.

##### 3.1.2 质量属性驱动的体系结构分析与设计

在清楚了系统的适应性需求之后,如何在 CAFISE Framework 基础上设计软件体系结构、落实这些需求是实现软件适应性的关键.为此,我们提出了质量属性驱动的结构分析与设计,其前提是完成了软件生存环境分析

和传统意义上的软件需求分析。

质量属性驱动的结构分析与设计综合考虑系统的适应性需求、功能性需求及其他需求。在 CAFISE Framework 的基础上,将需求落实在系统的计算结构和控制结构上,使获得的结构在整体上满足用户需求。它包括以下任务:

(1) 确定软件为满足用户需求及适应环境变化应达到的质量目标,即在功能性、修改性、重用性、性能、安全性、适应性等方面的各项要求;

(2) 确定各质量目标的优先级(这里的优先级只是定性描述质量属性的相对重要性,而非严格定量描述)。某些质量目标之间存在冲突,用户和开发者需要根据需求确定其优先级,以便折衷,并分析在重用 CAFISE Framework 基础上设计的软件体系结构应具备什么样的条件或能力,就能满足上述需求。

(3) 通过对各类需求的分析,派生出软件的体系结构。首先分析功能需求,获取用户的业务流程,从用户业务角度设计实现用户业务需求的计算结构,明确用户要进行哪些业务活动,参与这些业务活动的角色及业务活动之间的关系是什么,从而明确应用中要包含的业务服务。业务服务与用户的业务活动相对应,是 CASIFE Framework 下实现计算结构的基本功能单元。这些服务将是服务社区可管理和连接控制器可调整的应用构件。其次,根据系统的适应性需求,在 CAFISE Framework 基础上确定应用的控制结构,包括服务的动态选择策略、对目标服务的质量要求、服务社区的定制等。在该过程中,可以参考已有结构片段、体系结构模式选择或设计相关结构,对当前结构进行完善和补充,最终获得满足需求的体系结构。

不同的质量属性对结构的要求不同,可能会存在冲突。为此,需要综合分析各项需求,消除各结构需求间的冲突,获得完整的系统结构。通过质量属性驱动的结构分析获得完整的结构之后,可以对该结构进行分析、评价,以确认该结构是否满足用户需求。

质量属性驱动的结构分析解决了将适应性需求转化为系统结构的问题。众所周知,从功能需求获得结构相对容易,比如:通过提取、分析领域公共功能需求或基本功能需求获得软件核心结构,将附加或专有需求转化为扩展结构、微内核结构,就是这种需求-结构关系的典型代表。而非功能性需求规定了功能执行时应具有的属性(如响应时间、内存消耗、易用性等)和软件本身应具有的性质(如适应性等),为使目标软件较好地满足非功能性需求,必须综合考虑各种因素,使其在结构中得到充分体现。质量属性驱动的结构分析恰好解决了这一问题。

### 3.1.3 基于开放适应性框架的业务端动态演化

经过软件生存环境分析和质量属性驱动的体系结构分析与设计之后,将获得一个由 CAFISE Framework 定制而来的、开放的、具有适应性的体系结构实例。在此基础上,可以重用或设计所需的服务,利用 VINCA Workshop 将服务组合起来,实现用户所需的应用。基于开放适应性框架的业务端动态演化主要是通过动态服务组合的方式构造业务应用,并在业务层面实现应用的动态演化。该部分的主要活动包括以下几点:

#### (1) 服务重用与业务服务构造

服务是 CAFISE Framework 下软件开发、运行和演化的基础,在获得应用的体系结构实例之后,需要根据该结构的要求选择、修改可重用的服务,或设计新服务。构造领域本体,并将设计好的服务封装成业务服务,注册到服务社区中,完成服务社区的搭建。异构资源的服务化解决了由于资源异构而产生的适应性问题。

#### (2) 业务端编程

业务端编程就是从用户业务角度出发,在业务层面组合服务,构造应用。业务端编程通过对用户业务领域概念、业务活动、工作流程等的支持,为实现用户业务需求到软件应用的平滑过渡,提高在新需求出现的情况下高效、低代价地构造应用提供了保证。在 CAFISE Framework 下,业务端编程工具 VINCA Workshop 提供了逻辑模板、领域模板、用户模板 3 类模板,并通过领域模板推荐与定制实现半自动的应用构造,利用这些支持,可以方便地将服务社区中的服务组合起来,满足用户需要。

#### (3) 业务端动态调整

在投入使用后,为了能够快速适应动态变化的需求,应用需要有持续演化的能力。很多情况下,应用在投入使用后不久就会因需求变化,需要进行另一次开发或扩展性维护。业务端动态调整就是在用户业务层面对应用

进行动态重配置和即时修改.CAFISE Framework 通过监控与调整工具,对业务端动态调整提供支持.用户可以进行的调整操作包括:业务过程结构修改、服务连接关系修改、业务服务属性调整、应用状态改变等.这些调整操作将在业务层面进行,并通过 CAFISE Framework 支持的聚合关系,直接映射到软件层,减少了软件专业人员的参与,缩短了软件适应性调整的时间,降低了软件演化的代价.

基于开放适应性框架的业务端动态演化将软件的开发、使用和维护作为一个由可交替执行的活动组成的整体来考虑,并将生存环境分析和质量属性驱动的体系结构分析的结果落实到软件实现中,从方法学的角度为软件适应异构资源和需求变化提供了支持,并使软件具有可持续演化的能力.

### 3.2 方法特点

基于 CAFISE Framework 的软件开发吸收了现有相关工作的一些优点,如不同抽象层面的概念映射、动态应用调整、基于体系结构的思想等,但在此基础上,又有自己的特点和贡献:

- 从影响应用的环境要素分析入手,给出了一个包括可复用软件框架、支撑工具、开发过程在内的可提高面向服务软件适应性的开发方法;
- 通过面向服务的开放、适应性软件结构框架,在业务层面提供了对应用动态调整的支持;
- 在不同抽象层面概念映射的基础上,实现了业务层与软件层的聚合,提高了应用对异构资源及需求变化的应对能力.

CAFISE Framework 下的软件开发、使用和适应性调整是一个持续的交替进行的过程,在这种开发模式下的应用结构与以往也有所不同.可执行的应用不仅仅包含了满足用户功能需求的部分,而且包括支持它即时构造、运行和动态调整、扩展的基础设施.这种结构一方面将可复用的基础设施与用户需求分离,使用户可以更多地关注于自身的业务需求;另一方面又通过提高体系结构的适应性,使应用具有持续演化的能力.

## 4 案例分析

CAFISE Framework 及基于该框架的开发方法在多个项目中得到了应用,在此以其中的一个案例为主,结合其他案例对它的实际应用效果进行分析.

网络科技资源整合示范项目是一个国家科技基础条件平台项目,目的是以现有的网络科技信息资源为基础,实现从中央部委到地方省市网络科技信息的系统共享和业务流程的动态协作.在系统构建过程中,我们遇到了以下问题:

- 由于多年的信息化建设,各部门已经存在大量异构的数据资源和应用系统.如何高效地整合这些资源并使之得到充分利用是一个亟待解决的问题;
- 随着业务、部门职能、组织结构等的变化,不仅各部门发布的信息在发生变化,而且信息处理、审核的流程也需要及时调整.在流程执行过程中也需要根据参与工作流程的部门情况对流程进行动态定制,以使系统能够随需求变化动态演化,持续地满足用户需求.

为了解决上述问题,我们在该项目中采用了 CAFISE Framework 及相关的开发方法.

通过软件生存环境分析,我们将异构的科技信息资源及信息审批流程的变化确定为影响软件的主要环境因素,并通过业务过程建模及面向对象的分析确定了用户的业务需求.为了适应上述两类变化,并满足用户的业务需求,我们首先针对各地方及行业科技部门拥有的网络科技信息资源,确定了 3 类主要的待整合资源:

- 网页信息资源,即存在于各级部门不同站点网页上的科技信息资源.该类科技信息资源具有实时性高但结构化程度低的特点;
- 后台数据库资源,即以数据库形成保存的科技信息资源,它们往往是各种科技信息网站的主要内容来源.该类科技信息资源具有结构化程度高但开放性不好的特点;
- 网站服务系统资源,即在网络上以各种应用系统形式提供的科技信息服务资源,如番茄种植咨询系统等内容.该类科技信息资源具有针对性强但共享效果差的特点.

面对上述 3 类待整合的科技信息资源,我们将它们封装为开放、标准的软件服务,以解决在整合中各自存

在的问题,为资源整合奠定基础.此外,我们从资源整合及共享服务的需求角度出发,建立相应的科技信息资源整合元数据规范,主要包括科技信息资源描述元数据规范、科技信息资源分类规范、科技信息资源服务主题词等共享的语义基础.在此基础上,通过语义标注、服务虚拟化等手段将这些软件服务进一步封装为业务服务,以对用户屏蔽底层技术内容的变化,方便用户直接利用它们进行资源整合,适应不同、多变的资源整合需求;在进行质量属性驱动的结构分析与设计时,充分考虑前述 3 类待整合资源对系统的适应性需求,以系统的适应性属性为目标,利用服务社区实现对业务服务的统一管理;将 context 控制台定制为用户语境管理工具,并在连接控制器中配置语境敏感的服务动态选择策略,实现服务的个性化推荐,以满足面向不同需求的科技信息资源整合目的;定制监控调整工具,用于实现对业务流程的动态调整;将 VINCA Workshop 定制为业务流程定义工具,用于实现科技信息资源审核等业务流程的即时构造与非运行时调整.

在定制重用 CAFISE Framework 的基础上,我们采用业务服务组合的方式构造了适合科技信息管理的应用,并将服务化、业务化的异构数据资源在资源目录中发布出来,供用户使用.

除了网络科技资源整合示范项目之外,我们还在奥运综合信息服务系统、先进制造网格项目 AmGrid(如图 4 所示)、高要市低成本电子政务网格平台(如图 5 所示)项目中使用了本文所述的框架和开发方法.



Fig.4 Advanced manufacture grid  
图 4 先进制造网格 AmGrid



Fig.5 A cost effective grid-enabled eGovernment system  
图 5 高要市低成本电子政务网格平台

在实际项目中的使用情况表明:CAFISE Framework 及基于该框架的面向服务的开发方法能够较好地解决由资源及需求动态变化引起的软件适应性问题,支持面向服务应用的快速构建.

## 5 结 论

为了解决企事业单位信息系统中由服务资源及需求变化引起的软件适应性问题,本文给出了一个开放、适应性软件结构框架 CAFISE Framework,并提出了一种基于该框架的面向服务的适应性软件开发方法.在实际项目中的应用及实验证明,本文所述研究成果具有的创新之处在于能够较好地使应用适应异构资源的动态变化及用户业务变更;能够以较低的代价实现应用的持续演化.

在今后的工作中,我们将继续收集采用本文所述框架及方法的大型应用项目的软件过程管理数据及应用配置管理数据,并在此基础上对本文所述成果进行进一步的定量分析与完善.

致谢 我们的研究工作得到了中国国家自然科学基金委员会的支持,在此表示诚挚的感谢.

## References:

- [1] Licherherr K. Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns. PWS Publishing Company, 1996.
- [2] Gacek C, Abd-Allah A, Clark BK, Boehm BW. On the definition of software system architecture. In: Garlan D, ed. Proc. of the 1st Int'l Workshop on Architectures for Software Systems. New York: ACM Press, 1995. 85-95.



- [3] Zhang J, Cheng BHC, Yang ZX, McKinley PK. Enabling safe dynamic component-based software adaptation. In: de Lemos R, Gacek C, Romanovsky AB, eds. Proc. of the Workshops on Software Architectures for Dependable Systems 2004. Berlin: Springer-Verlag, 2005. 194–211.
- [4] Mukhija A, Glinz M. CASA—A contract-based adaptive software architecture framework. In: Castelluccia C, ed. Proc. of the 3rd IEEE Workshop on Applications and Services in Wireless Networks. Piscataway: IEEE Press, 2003. 275–286.
- [5] Bass L, Kazman R. Architecture-Based development. Technical Report, CMU/SEI-99-TR-007, 1999.
- [6] Allen RJ. A formal approach to software architecture [Ph.D. Thesis]. Pittsburgh: Carnegie Mellon University, 1997.
- [7] Buhler P, Vidal JM. Towards adaptive workflow enactment using multiagent systems. Information Technology and Management Journal, 2005,6(1):61–87.
- [8] Li G. Adaptive software architecture [Ph.D. Thesis]. Beijing: Beihang University, 2002 (in Chinese with English abstract).
- [9] Luckham DC, Vera J. An event-based architectural definition language. IEEE Trans. on Software Engineering, 1995,21(9):717–734.
- [10] Kramer J, Magee J. Analysing dynamic change in software architectures: A case study. In: Cole R, Schlichting R, eds. Proc. of the 4th IEEE Int'l Conf. on Configurable Distributed Systems. Piscataway: IEEE Press, 1998. 91–100.
- [11] Oreizy P, Gorlick MM, Taylor RN, Heimbigner D, Johnson G, Medvidovic N, Quilici A, Rosenblum DS, Wolf AL. An architecture-based approach to self-adaptive software. IEEE Intelligent Systems, 1999,14(3):54–62.
- [12] Soley R, OMG Staff Strategy Group. Model driven architecture. White Paper, OMG, 2000.
- [13] Nadhan EG. Service-Oriented architecture: Implementation challenges. The Architecture Journal, 2004,2(4):24–32. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmaj/html/aj2soaimpc.asp>
- [14] Han YB, Geng H, Li HF, Xiong JH, Li G, Holtkamp B, Gartmann R, Wagner R, Weissenberg N. VINCA—A visual and personalized business-level composition language for chaining web-based services. In: Orłowska ME, Weerawarana S, Papazoglou MP, Yang J, eds. Proc. of the 1st Int'l Conf. on Service-Oriented Computing. Berlin: Springer-Verlag, 2003. 165–177.
- [15] Li G, Han YB, Zhao ZF, Wang J, Wagner RM. Facilitating dynamic service compositions by adaptable service connectors. Int'l Journal of Web Services Research, 2006,3(1):68–84.
- [16] Li G, Wang JW, Wang J, Han YB, Zhao ZF, Wagner RM, Hu HT. MASON: A model for adapting service-oriented grid applications. In: Li ML, et al. eds. Proc. of the GCC 2003. Berlin: Springer-Verlag, 2004. 99–107.
- [17] Li G, Han YB, Jin MZ. Factors affecting enterprise application evolution and their function. Journal of Computer Engineering, 2004,30(16):70–71 (in Chinese with English abstract).
- [18] Bohner SA, Arnold RS. Software Change Impact Analysis. Los Alamitos: IEEE Computer Society Press, 1996.
- [19] Zhao JJ, Yang HJ, Xiang LM, Xu BW. Change impact analysis to support architectural evolution. Journal of Software Maintenance and Evolution: Research and Practice, 2002,14(5):317–333.

#### 附中文参考文献:

- [8] 李刚. 适应性软件体系结构研究[博士学位论文]. 北京: 北京航空航天大学, 2002.
- [17] 李刚, 韩燕波, 金茂忠. 影响企业应用演化的因素及其作用. 计算机工程, 2004, 30(16): 70–71.



李刚(1972 - ),男,山东潍坊人,博士,副研究员,CCF 高级会员,主要研究领域为面向服务的软件体系结构,面向服务的网格计算,动态自组织无线网络中间件.



韩燕波(1962 - ),男,博士,研究员,CCF 高级会员,主要研究领域为服务网格,软件集成.



赵卓峰(1977 - ),男,博士,助理研究员,主要研究领域为面向服务计算,网格 workflow.



梁英(1962 - ),女,高级工程师,CCF 高级会员,主要研究领域为服务网格,软件集成.