

Liquid——构件化嵌入式操作系统

江 峰, 陈文智, 吴朝晖

(浙江大学计算机学院, 杭州 310027)

摘 要: 构件化的软件开发方法是一种将一个完整系统看作是若干独立部分(构件)的装配的软件开发方法。该文介绍了Liquid构件化嵌入式系统, 分析了其他的体系结构及采用的各种构件技术。

关键词: RMI; CORBA构件模型; 普适计算; 构件化; 嵌入式操作系统; 中间件; 分布式系统

Liquid——A Component-based Embedded Operating System

JIANG Feng, CHEN Wenzhi, WU Zhaohui

(College of Computer Science, Zhejiang University, Hangzhou 310027)

【Abstract】 This paper introduces a new component-based embedded system——Liquid, describes the whole system architecture and the techniques that the system adopts.

【Key words】 RMI; CCM; Ubiquitous computing; Component; Embedded OS; Middleware; Distributed system

近几年来国内外都逐渐将构件化的软件开发作为一种设计方法学应用于操作系统的设计实践中。本文重点研究如何将构件化的软件设计方法学应用于嵌入式应用设计并实现可重配置操作系统;着重介绍我们自己设计的构件化嵌入式操作系统——Liquid。

1 Liquid构件化的嵌入式操作系统及开发环境

1.1 研究方向

针对不同的应用(如信息家电、武器系统和通信设备),通过构件化操作系统的灵活配置特点,提供能在各种应用环境下,方便易用、符合嵌入式系统特点、具有自适应性的应用相关(Application-specific)的嵌入式操作系统。

1.2 Liquid的设计与实现

Liquid是一个以构件技术、中间件技术、实时技术等为技术核心,研究具有高度可配置性、重用性、自适应性、可移植性和健壮性的新型构件化嵌入式操作系统。向用户提供灵活、透明的构件服务,提供普适计算支撑,提供实时多任务机制和低功耗能源管理功能,提供包括系统调试工具和测试工具在内的系统开发环境。其中,构件化嵌入式操作系统内核、由应激服务构件群构成的系统自适应服务环境和支持网络分布式环境的构件化系统服务层这3方面内容是本课题的研究内容的主要组成部分。

Liquid系统架构为分层结构(Layered Architecture),整个系统从宏观上分为2层: Liquid内核层和构件化系统服务层。系统架构如图1所示。

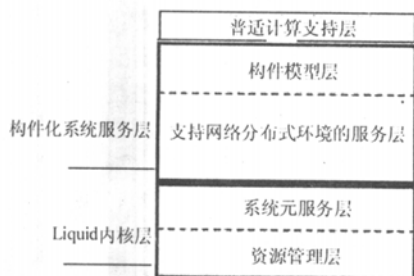


图1 Liquid 体系结构

(1) Liquid内核层

Liquid内核层分为系统元服务层与资源管理层。资源管理层的任务是实现对设备本地资源的管理,并向上层提供基本服务。系统元服务完全封装了本地资源管理的实现。元服务的设计是层次式的,如图2所示。



图2 元服务层次结构

元服务适配层在功能上基本上是对系统服务的封装和整合。元服务包装外观层在实现上采纳了大量经过实践检验的设计模式的思想,实现了许多用于嵌入式、分布式软件的核心模式。元服务框架实现了一个网络编程框架,集成并增强较低层次的元服务包装外观。元服务组件提供包装成自包含组件的标准分布式服务库。它们在元服务中扮演了2种角色: 1)分解出可复用分布式应用的“积木”; 2)演示元服务组件的常见用例。

(2) 构件化系统服务层

构件化系统服务层分为支持网络分布式环境的服务层和构件模型层。

1)支持网络分布式环境的服务层(Broker-Liquid系统软总线)

Liquid系统的一个重要特征是对分布式异构环境的有效支持。为此, Liquid系统需要实现一个软件总线,作为本地操作系统和上层服务、应用之间的中间层。我们提出了Broker-Liquid系统软总线。Broker模式包括一个broker,本质上是对象引用的仓库,对于客户端和服务端是全局可见的。客户端不必具备服务器位置的先验知识,增强了基础架

基金项目: 国家“863”计划基金资助项目(2002AA1Z2302)

作者简介: 江 峰(1978—),男,硕士生,研究方向: 嵌入式操作系统; 陈文智,副教授; 吴朝晖,教授

定稿日期: 2004-03-11 **E-mail:** data@cad.zju.edu.cn

构支持的分布式计算对上层应用的透明性。Liquid系统选择CORBA作为Liquid的软总线技术。

在兼容CORBA标准的前提下，对CORBA规范进行了裁减，遵循Minimum CORBA和Real-time CORBA规范，内置了对服务质量的支持，实现了TORB。TORB具有良好的(甚至实时)性能，高度可裁减，具备最小的、足够系统运行的功能。

TORB的整体结构图如图3所示。

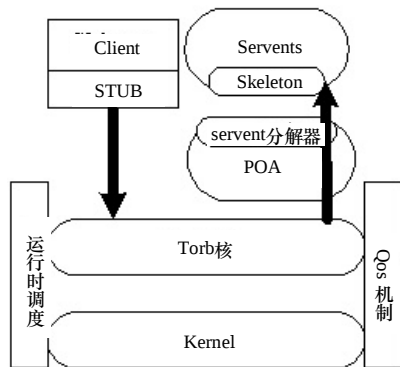


图3 Torb架构图

2) 构件模型层

在Liquid系统设计思想中，构件模型是创建可重用的构件并将其组合到容器中，以得到新的应用系统。构件模型层需要有定义构件的基本体系结构、构件界面的结构、与其他构件及容器相互作用的机制等功能。利用构件模型规范说明，构件开发人员开发那些实现了应用系统逻辑的构件，而应用系统开发人员则把这些预先开发好的构件组合成应用系统，这些应用系统也可以作为新的构件。当前OMG提出的CORBA Component Model(CCM)作为CORBA3.0标准的一部分，很好地实现这种功能。

我们在兼容CCM标准的前提下，对CCM规范进行了适当裁减，结合当前各种实现的CCM版本，实现了LCCM，其结构如图4。

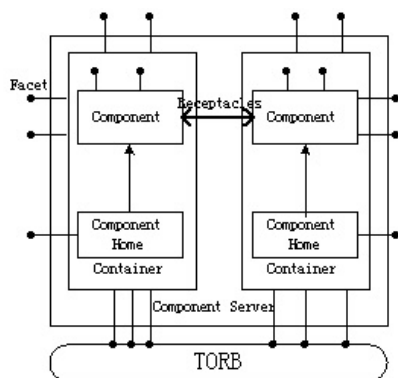


图4 LCCM结构

图中，①ComponentServer：在构件配置过程中负责container生成与销毁。

②Container：运行在ORB上的服务器端的一个框架，包含了对POA和部分CORBA Service的一些调用接口，它给ComponentHome及Component实例提供运行环境。

③Home：负责构件的生命周期管理。提供相关接口负责构件的生成与销毁。

LCCM这样的层次结构拥有以下一些优点：利于软件复用；降低了构件和中间件之间的耦合度；详细说明了构件的相互联络；使用更好的策略来创建构件；可以通过“元数据”对构件配置；并且提供了Component Server来对系统进行维持。

同时LCCM还提供相应的编译、联结、装配等工具，提供完整的构件生成、联结、装配流程。

LCCM具有较为完善的流程和管理，在这种模式下进行开发具有可重用、方便等优势。

通过LCCM构件模型的程序设计可以达到以下这些目的：重用，高层开发，通过工具进行自动化开发、简化开发过程等。

(3) 普适计算环境支持层

Liquid系统支持适应普适计算(Pervasive Computing)中高不确定性特点的环境，达到稳定、健壮的系统特性，适应无线移动等环境。

普适计算对计算环境提出了新的定义和要求。人们对计算环境的理解将由个人计算机转变为一个物理空间。把整个物理空间当作一个计算系统；Liquid OS就是在这个计算系统上运行的操作系统，它屏蔽掉了物理空间中各个设备的细节，向上提供了标准程序接口，使得这个物理空间变成了一个能被感知的空间。我们设计的操作系统就是要对这个物理空间的资源进行管理，使应用程序能够与物理空间进行交互。

在构件化系统服务层的基础上，通过物理空间内的多个Liquid操作系统实例之间的协作，我们能够得到支持普适计算环境的系统服务层，实现物理空间与计算空间的统一。

1.3 一个简单实验场景设计

一个房间，安装了机顶盒(with a TV SET)，及一台构件服务器，各设备均安装了Liquid操作系统，这些设备与系统构成了一个智能空间。如图5所示。

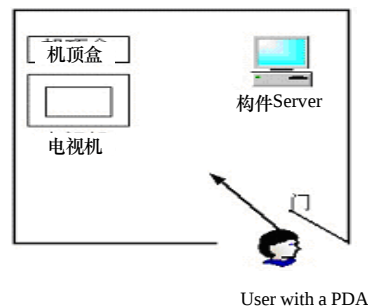


图5 试验场景

场景中存在若干个构件，机顶盒对外提供服务的SetupBox_Comp构件，主要提供若干服务接口；构件Server上有一个构件仓库，内有若干构件。当实验者手持PDA(安装有Liquid)进入这一智能空间，PDA利用Discovery Service自动发现房间内的机顶盒设备及构件服务器。当PDA准备与机顶盒进行交互时，将会出现2种情况：(1)PDA内部已经安装控制机顶盒的构件；(2) PDA没安装该构件。对于情况 (1)，实验者可直接通过PDA控制机顶盒进行视频点播；对于情况(2)，PDA连接构件server，从构件server中下载控制机顶盒构件，结合当前PDA内部资源状况，对PDA内部系统进行重配置，然后进行操作。

(下转第87页)

染色体 $U^t = \{\gamma_j^t\}$ ，其中 t 表示代数。显然，适应值较大的个体参与产生下一代的概率较大，为了加快算法的收敛速度，可以将每代中适应值较大的一些个体强行传到下一代，而不受选择过程的限制；

3) 插入式交叉操作；

4) 从交叉操作后得到的个体中，以一定的概率 P_m 随机选取某个个体，按照前面的变异算子定义，进行变异操作。至此我们得到了经过遗传操作后获得的下一代种群 $U^{t+1} = \{\gamma_j^{t+1}\}$ ；

5) $gen = gen + 1$ ；

end while

(4) 将评价函数值最好的一组结果指定为遗传算法的结果；

(5) 根据各个最优个体，提取特征词条，重新计算词条权重，生成合取群体特征向量。

3 实例验证

测试过程分为特征向量的获取和正确性验证两个阶段。测试数据来自新浪财经新闻稿，共摘录了50篇文章。其中有40篇关于股票评论的文章，还有10篇关于其它方面的内容。我们从40篇股票评论文章中选出20篇作为训练样本，由协同演化遗传算法进行特征提取，作为股评类的特征。全部50篇文章都作为测试文本对所获取的特征进行准确性验证。

在测试中，设定最大遗传迭代次数为500代。为每篇文章建立15条染色体的小种群，这样共有20篇文章300条染色体参与协同演化。另外，设置停机条件为各小种群的平均适应值不再发生较大变化，则停止进化。

经过平均80代的迭代(共测试20次)，个体收敛，从20个小种群中获得最佳个体15条。我们对这15个染色体中的基因(词条)进行基因重组，组成新的类别特征向量。合并相同词条，并按下列公式计算各词条的权重：

$$W_{word_j}^{new} = \frac{\sum_{i=1}^{10} W_{word_j}^{old} \times D_i}{\sum_{i=1}^{10} D_i}$$

至此，我们获得了测试样本的特征表示。下面来验证其准确性。首先对测试集中的每篇文章进行分词处理，并计算词条权重。在本测试中，将直接采用文本的原始向量，避免由于特征向量获取所带来的误差造成测试结果的不准确。同时设定与类别向量相似度大于80%的文章，就认为它与训练文本为一类。最后的验证结果如表1。

(上接第78页)

PDA自身的重配置过程对于实验者是完全透明的。

1.4 Liquid 设计的优势

与Choices、Os-Kit、2K及GAIA相比较，在构件化嵌入式的今天，Liquid有着其天然的优势。首先Liquid是专门为嵌入式应用设计的操作系统；其次，Liquid是一个比较彻底的构件化的操作系统，它继承了诸多构件化系统的优势；最后，CCM的采用使得Liquid拥有了一套较为完整的构件开发及管理策略。

2 结论

构件化的软件设计方法与嵌入式操作系统的开发的结合将是现代操作系统发展的一个趋势，随着各种带有构件化特性的嵌入式系统的出现，正逐步论证这一发展趋势的必然性

从实验结果来看，作为训练样本的20篇文章都得到了正确归类，说明我们所获得的类别特征向量能基本体现该20个训练样本的特征，保证了准确率的前提。另外对同类的另20篇文章进行测试，有3篇文章的相似度为75.8%、72.1%和65.4%，没有通过80%的阈值，漏判率为15%。总的来说，从实验数据和结果分析来看，基于协同演化的遗传算法在保证准确率的基础上，获得了同类多文本的特征。

表1 验证结果

测试文本	分为同类	分为异类	准确率
20篇训练样本	20	0	100%
20篇测试样本(同类)	17	3	85%
10篇测试样本(异类)	1	9	90%

4 结束语

本文主要讨论了文本的特征向量获取问题。文本的特征向量获取是文本信息处理中又一个重要的处理步骤。我们根据协同演化的思想，提出了基于协同演化遗传算法的群体同类文本的特征提取算法，有效地解决了同一主题众多文本的集体特征向量的获取问题。在所获得的每个子种群的最优个体，已经不是体现该文本的最佳个体，但它是反映整个文本的集合某些特征的最佳个体。因此合取向量不再是个体特征向量的简单叠加，而是体现了所有个体的共同特性。集体合取特征向量的获取在用户访问兴趣模板和文本分类模板的获取方面有着较为广泛的应用。

参考文献

- 1 Gudivada V N. Information Retrieval on the World Wide Web. IEEE Internet Computing, 1997, 1(5): 58-68
- 2 张晓辉. WWW上的信息发现与搜索引擎技术. 小型微型计算机系统, 1998, 19(6): 66-71
- 3 王伟强, 高文, 段立娟. Internet上的文本数据挖掘. 计算机科学, 2000, 27(4): 32-36
- 4 刘明吉, 饶一梅. 基于模糊近似度的Web文本过滤模型. 计算机科学, 2001, 28(12): 55-58
- 5 王继成, 潘金贵, 张福炎. Web文本挖掘技术研究. 计算机研究与发展, 2000, 37(5): 513-520
- 6 刘明吉. Web文本信息的特征获取算法. 小型微型计算机系统, 2002, 23(6): 683-686
- 7 Liu Mingji, Wang Xiufeng. A Knowledge Discovery Algorithm Based on Genetic Algorithm. The Third World Congress on Intelligent Control and Automation, IEEE, WCICA, 2000

和可行性。构件化嵌入式操作系统开发技术的进一步成熟及各种嵌入式设备的普及给与普适计算提供了充分的条件。

参考文献

- 1 Crnkovic I. Component-based Software Engineering——New Challenge in Software Development. Department of Information and Electronics Engineering, Niigata Institute of Technology, 2000
- 2 Aoyama M. New Age of Software Development: How Component-based Software Engineering Changes the Way of Software Development. Department of Information and Electronics Engineering, Niigata Institute of Technology, 2001
- 3 Satyanarayanan M. Pervasive Computing: Vision and Challenges. OMG, Inc., 1999
- 4 The Common Object Request Broker: Architecture and Specification. OMG, Inc., 1987