

支持上下文感知的构件适配模型研究

郑 笛¹, 王 俊², 贲可荣¹

(1. 海军工程大学计算机工程系, 武汉 430033; 2. 空军雷达学院重点实验室, 武汉 430019)

摘 要: 为了对普适环境下的构件化应用提供支持, 扩展传统的构件适配技术, 在上下文感知中间件的基础上, 提出支持上下文感知的构件适配模型 CACAM 和构件适配算法 CACA, 实现对环境、用户上下文的动态感知和重配置。实验结果表明, 该模型能够有效地实现普适计算环境下基于上下文感知的构件适配。

关键词: 普适计算; 上下文感知; 构件适配; 中间件; 反射

Research on Component Adaptation Model Supporting Context-aware

ZHENG Di¹, WANG Jun², BEN Ke-rong¹

(1. Department of Computer Engineering, Naval University of Engineering, Wuhan 430033, China;

2. Key Research Lab, Air Force Radar Academy, Wuhan 430019, China)

【Abstract】 In order to support component based applications of pervasive environments, traditional component adaptation is extended and a Context-aware Component Adaptation Model(CACAM) is put forward based on component middleware and Context-aware Component Adaptation(CACA) algorithm according to support dynamic awareness and reconfiguration for the context based adaptation. Experimental results show that this model can support the context-aware component adaptation efficiently in the pervasive computing environment.

【Key words】 pervasive computing; context-aware; component adaptation; middleware; reflection

DOI: 10.3969/j.issn.1000-3428.2012.02.012

1 概述

近年来, 对基于构件的分布式应用进行动态适配逐渐成为活跃的一个领域。各研究项目提出的自适应、自管理系统具有相似的体系结构, 都由系统状态收集部件、系统分析部件、系统配置部件 3 类部件组成, 从而形成一个闭环反馈控制系统。其中, 系统状态收集部件由若干探测器组成, 负责在不同地点、不同时间收集不同种类的系统状态信息。根据系统的体系结构信息和 QoS 要求, 系统分析部件对收集到的系统状态信息进行综合、转化和分析, 给出对系统各种特性的评测, 包括系统是否足够安全、负载是否均衡和实时性能是否满足要求等, 从而做出是否需要系统调控以及如何通过改变系统配置实现系统调控的决策。系统配置部件则负责根据系统分析部件下达的系统配置更改命令, 实施动态适配。显然, 动态适配技术是自适应、自管理系统实现的基础。

近年来, 普适计算技术逐渐成为未来计算技术发展的重要方向。普适计算环境由大量计算和通信设备构成, 这些设备无缝地将人的思维和活动联系在一起。这类环境中的应用通常是上下文感知的, 适用不同类型的上下文, 如环境和时间以适配演化的环境, 并提供更灵敏的服务^[1-5]。

传统的构件化中间件缺乏对上下文感知适配的支持, 因此, 本文通过上下文感知服务和不一致性管理算法负责上下文信息的收集、解释和管理, 并对传统的构件适配技术进行扩展, 提出支持上下文感知的构件适配模型(Context-aware Component Adaptation Model, CACAM)和支持上下文感知的构件适配算法(Context-aware Component Adaptation, CACA)。

2 上下文感知的构件适配模型及算法

2.1 上下文感知中间件

上下文感知中间件选择前期工作中建立的基于上下文感

知中间件体系结构^[6], 如图 1 所示。

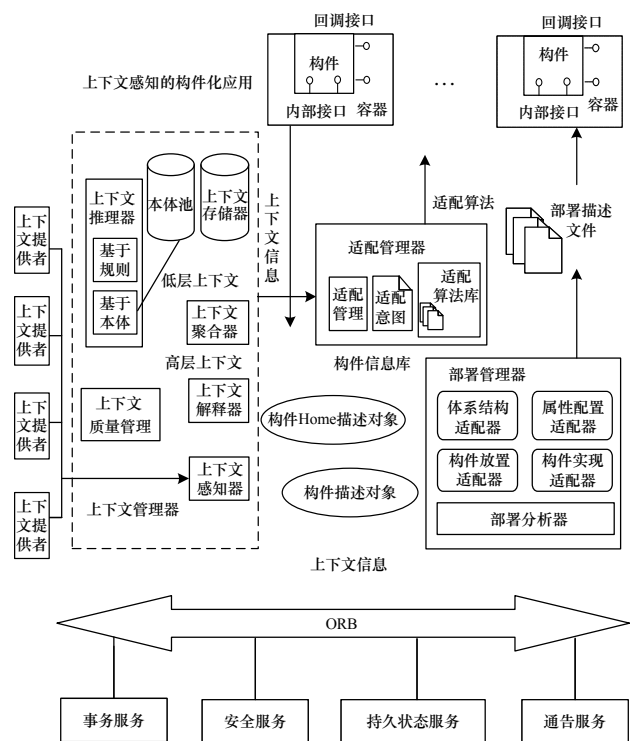


图 1 用于构件化应用的上下文感知中间件结构

基金项目: 海军工程大学自然科学基金资助项目“面向普适服务的无线传感器网络中间件研究”(20100406020005)

作者简介: 郑 笛(1977—), 女, 讲师、博士, 主研方向: 普适计算, 上下文感知中间件; 王 俊, 讲师、博士; 贲可荣, 教授、博士

收稿日期: 2011-07-06 E-mail: dizheng1216@163.com

核心构件管理模块为在前续工作所开发的遵循 CORBA 构件模型 CCM 的构件化中间件，为基于构件的应用、构件实例提供平台无关的服务，为执行平台资源提供了统一的平台无关访问。

构件在容器中执行，为分布式应用的事务、安全、一致性和资源管理提供隐式的支持。容器能够提供包括生命周期管理和构件发现在内的非功能性特性。

2.2 上下文感知的构件适配结构模型

根据适配执行的时序特点，将构件适配分为以下 3 个阶段：适配执行前，适配执行中，适配执行后。其中，适配执行后又可分为有过渡期适配和无过渡期适配 2 种。

(1) 适配执行前

通过中间件的外省和内省机制，感知上下文状态的变化和中间件内部构件接口信息、资源使用情况的变化；通过自主决策推理，得到最优的“自适应动作执行序列”。

(2) 适配执行中

根据阶段 1 得到的适配执行动作序列，采用优化的策略串行或并行执行相关动作。

(3) 适配执行后

1) 有过渡期

在执行了自适应动作之后，处于过渡状态，如果效果不好，仍可以逆向执行刚才的适配动作，使之回滚到适配之前的状态。

2) 无过渡期

在执行了适配动作之后，处于新的状态，不可回到适配之前的状态。

如图 2 所示，上下文管理器负责上下文信息的收集、解释与推理，并负责通知构件适配管理器所需的上下文信息。利用这些上下文信息，适配管理器可获得应用系统的全局视图，并基于对系统的分析以及参照要达到的特定管理目标，产生相应的动态适配方案。适配管理器利用上下文管理器提供的上下文信息，同时通过系统信息收集部件获取构件实时运行状态信息，从而在适当时机调控构件行为，实现动态适配意图。

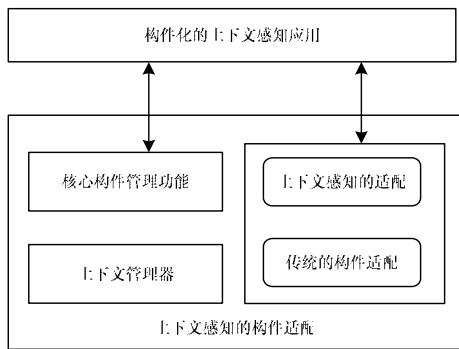


图 2 上下文感知的构件适配结构

适配管理器可以被划分为更为详细的管理器和服务的集合。这些管理器可以借助事件服务向上下文管理器对自身感兴趣的上下文进行注册，当对应的上下文发生变化，对应的管理器将获得通知。

上下文感知的适配系统事实上就是一个反射系统，其中：上下文感知的适配系统的基层是应用系统；上下文感知的适配系统的元层是适配管理器；系统的元控层对应于配置者、管理者。

上下文感知的适配系统的反射体系如图 3 所示。

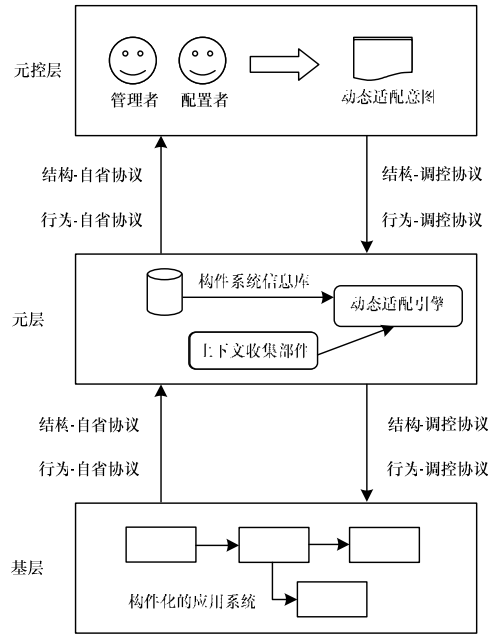


图 3 动态适配系统反射体系示意图

2.3 上下文感知的构件适配算法

本节介绍基于上下文感知的构件适配算法。首先，将上下文感知的构件适配单元定义为：

定义 $CACAU \xrightarrow{define} (A_c, A_{ig}, A_cFLOW, CONTEXT)$

其中：

(1) A_c 为一组上下文感知构件的集合，反映了 CACAU 的组织结构特性，包括具有不同语义信息的上下文感知构件的种类和数目。

(2) A_{ig} 为 A_c 中构件的相互依赖关系，包括上下文感知构件之间的直接、间接依赖；双向、单向依赖；依赖程度反映了 CACAU 内部的聚合程度。

(3) A_cFLOW 为实际运行过程中的构件运行流程，构件之间的接口调用顺序，事件介绍和发送的顺序。

(4) $CONTEXT$ 为 CACAU 运行时的上下文环境，当它发生变化时， A_cFLOW 可能发生相应的变化。

在某一段时间里，不同的构件容器中存在着多个不同的 CACAU，将不同构件容器中的 CACAU 的全体称为上下文感知构件适配群体(Context-aware Component Adaptation Group, CACCG)。

下面介绍 CACA 方法的 4 个阶段：

(1) 选择阶段

根据运行上下文环境的变化，使用其自身的外省和内省机制，选择出所有可以执行的上下文感知构件动作行为的集合(包含完成同一目标的多个动作方案)，以便执行后进行自身结构和功能的变化，选择出执行时间消耗最小的动作并且保证执行后新生成的 CACAG 是安全的，最终生成一个上下文感知构件组合方案。

(2) 准备阶段

1) 对于加入操作 $A_cA(a_c, cacau)$ ，上下文感知构件容器 A_n 。向该 CACAU 中的上下文感知构件广播加入 a_c 的消息以及 a_c 的基本信息，其他构件收到该消息后将会返回 3 种回应：

- ① 返回“ok”，表示同意该构件的加入；
- ② 返回“no”，表示不同意该构件的加入；

③返回“dep”,表示希望依赖 a_c 。

2)对于离开操作 $A_c R_v(a_c, cacau)$,上下文感知构件容器 A_n 向该CACAU中与 a_c 有依赖关系的上下文感知构件单播 a_c 离开的消息。其与 a_c 有依赖关系的构件收到该消息后将会返回2种回应:

- ①返回“ok”,表示同意该构件的离开;
- ②返回“no”,表示不同意该构件的离开。

(3)实施阶段

当上下文感知构件组合方案中的动作都经过了准备阶段,便可以进行相应的动作执行。

(4)更新阶段

在完成上下文感知构件组合方案中的动作执行后,需求同步更新该动作所引起的构件依赖的变化。

3 性能测试

为了测试CACA算法的性能和可靠性,选择具有不同计算资源的6个构件容器,并且在整个测试的过程中不增减其数目和人为改变构件容器中计算资源的数目。

选择预分配的构件数目从50~500变化,初始值为50,每次增加的步长为50。需要说明的是:运行时上下文情况的变化分为CPU计算资源、内存容量和通信带宽3类,并且采用了随机进行这些上下文的变化。

一共进行了30次实验,具体情况如下:

(1)测试情况1,即在相同的运行时上下文情况下,在使用CACA算法和不使用CACA算法执行都成功的情况下时间消耗的对比,如图4所示。

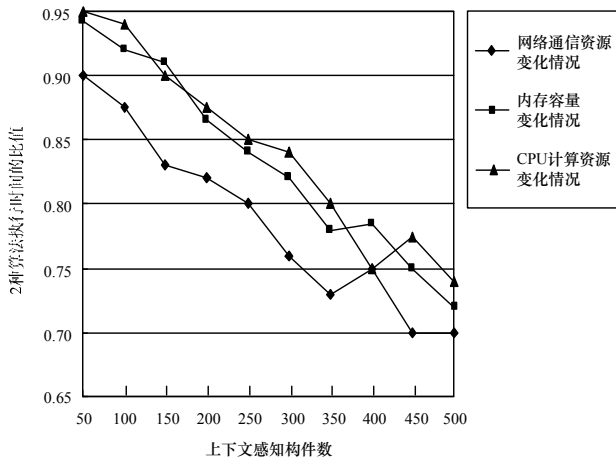


图4 时间消耗比较

在使用了CACA算法之后,对系统的性能影响是客观存在的。其中,在无线网络的动态变化情况下使用CACA算法的时间消耗和性能损失较CPU和内存变化情况下要大。随着构件数目的增加,相互依赖关系的复杂程度变大,使得整体性能损失也随着变大。

然而,总体而言使用CACA算法对于系统的运行连续性是可以令用户满意的。某些点上的性能相差较大是因为在某一情况下执行的上下文感知构件动作较多、较频繁所致。

(2)测试情况2,即在相同的运行时上下文情况下,在使用CACA算法和不使用CACA算法执行成功率的对比,如图5所示。

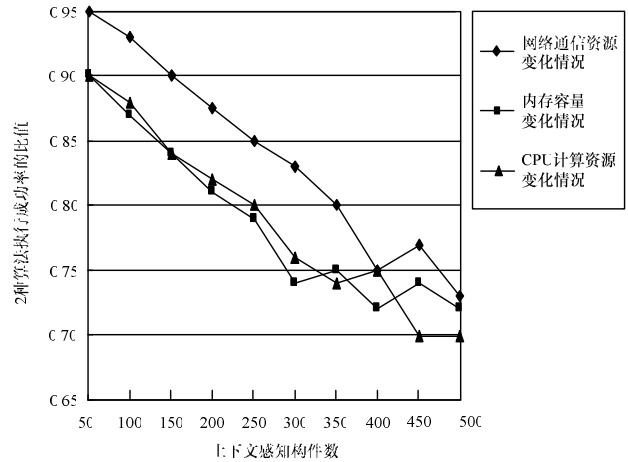


图5 成功率比较

使用了CACA算法之后,在相同的运行时上下文情况下系统执行的成功率比没有使用CACA算法大了很多。其中,在无线网络的动态变化情况下使用CACA算法的系统成功率较CPU和内存变化情况下要大;随着构件数目的增加,整体成功率也随着变大。因此,使用CACA算法使得系统运行的可靠性显著增加。

4 结束语

传统的构件化中间件缺乏对上下文感知适配的支持,本文对传统的构件适配技术进行了扩展,提出了支持上下文感知的构件适配模型CACAM和支持上下文感知的构件适配算法CACA。实验结果表明,该算法能够有效地支持普适计算环境下基于上下文感知的构件适配过程。

参考文献

- [1] 徐光祐, 史元春, 谢伟凯. 普适计算[J]. 计算机学报, 2003, 26(9): 1042-1050.
- [2] 郑增威, 吴朝晖. 普适计算综述[J]. 计算机科学, 2003, 30(4): 18-29.
- [3] 徐文控, 辛运祎, 卢桂章. 普适计算中间件的研究与进展[J]. 计算机科学, 2007, 34(11): 1-5.
- [4] Dey A K. Understanding and Using Context[J]. Personal and Ubiquitous Computing, 2001, 5(1): 4-7.
- [5] 徐磊. 普适计算中间件技术的研究[J]. 计算机工程, 2004, 30(20): 113-116.
- [6] Zheng Di, Wang Jun, Jia Yan, et al. Deployment of Context-aware Component-based Applications Based on Middleware[C]//Proc. of the 4th International Conference on Ubiquitous Intelligence and Computing. Hong Kong, China: [s. n.], 2007: 908-918.

编辑 顾逸斐