

# 一种适用于服务组合平台的语境信息使用模式

张 程<sup>1),2)</sup> 韩燕波<sup>1)</sup>

<sup>1)</sup>(中国科学院计算技术研究所软件研究室 北京 100080)

<sup>2)</sup>(中国科学院研究生院 北京 100089)

**摘要** 面向服务应用中的语境信息有多种用途,实现过程各异,这增加了支撑平台开发与维护的复杂程度。文章对这些用途进行了分析,并将它们在实现过程中的共性特点进行抽象,以此为基础,提出了一种适用于服务组合平台的语境信息使用模式。按这种模式,服务组合平台能够以统一的方式对语境信息的各种用途提供支撑,有效支持了面向服务应用在运行时的语境敏感需求,同时简化了平台本身的开发与维护。结合在奥运信息服务平台 FLAME2008 上的实现进行测试,并对测试结果做了分析。

**关键词** 服务组合;语境信息;语境敏感;使用模式

**中图法分类号** TP311

## A Context Information Usage Pattern for Service Composition Platforms

ZHANG Cheng<sup>1),2)</sup> HAN Yan-Bo<sup>1)</sup>

<sup>1)</sup>(Software Division, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

<sup>2)</sup>(Graduate School of the Chinese Academy of Sciences, Beijing 100039)

**Abstract** Context information can enrich services-oriented applications in multiple ways. The implementation processes of these usages differ a lot, increasing the complexity to develop and maintain the supporting environment. This paper proposes a context information usage pattern suitable for service composition platforms. Using this pattern, a service composition platform can make use of context information in a uniform way. This not only effectively supports context-aware requirements of applications at run-time but also simplifies the development and the maintenance of composition platforms. The research results are evaluated in terms of experiments supported by FLAME2008 - a service mediation platform for the Olympic Games 2008.

**Keywords** services composition; context information; context-aware; usage pattern

## 1 引言

随着技术的发展,面向服务的应用可以在更多种类的计算设备上运行,而这些设备将融入人们日常生活的各个角落<sup>[1]</sup>。这使得应用运行时环境中的语境信息经常改变,所谓语境信息,我们指用户所处

的周围环境信息、用户自身的个性化信息以及运行应用的硬件环境信息。此时,面向服务的应用期望具备语境敏感的能力,即自动感知环境信息的变化并使用这些信息。

服务组合平台作为上层应用和底层资源之间的软件模块,为构造和运行面向服务的应用提供支撑。在服务组合平台中提供对语境敏感的支持,可以屏

收稿日期:2004-12-05;修改稿收到日期:2005-02-22。本课题得到国家自然科学基金(90412005)、国家“八六三”高技术研究发展计划项目基金(2003AA414330)、中国科学院百人计划(20024040)和中国科学院计算技术研究所青年创新基金(20026180-20)资助。张 程,男,1977 年生,博士研究生,主要研究方向为动态服务组合、语境敏感计算技术。E-mail: zhch@software.ict.ac.cn。韩燕波,男,1962 年生,研究员,博士生导师,主要研究方向为软件集成与服务网格。

蔽底层语境信息的变化,这不仅方便了具备语境敏感特点的面向服务应用的构造,还能够实现应用运行时的敏感需求。此时,对语境信息的处理由组合平台完成,以生命周期的观点来看,该处理过程可分为如下三个阶段:信息的获取、信息的组织以及信息的使用,其中前两个阶段为第三阶段提供了可用的信息,而如何使用这些信息又面临如下问题:首先,要确定语境信息的用途是什么,期望达到什么目的;其次,就是如何利用这些信息实现其用途。然而,在面向服务应用中语境信息有多种用途,而且实现过程各异,这增加了支撑平台的开发与维护的复杂程度。

本文正是针对上面的问题进行研究,分析了语境信息在面向服务应用中的用途,将具有共性特点的执行过程进行抽象,并以状态转换图对此过程做规范化的描述,在此基础上,基于向量运算的方法提出了一种适用于服务组合平台的语境信息使用模式,所谓使用模式是指能够指导实现的规范化的操作步骤,而规范化是指可以用固定的模型进行描述。应用此模式,服务组合平台能够以统一的方式对语境信息的各种用途提供支撑,有效支持了面向服务应用在运行时的语境敏感需求,同时简化了平台本身的开发与维护。结合在奥运信息服务平台 FLAME2008 上的实现进行测试,文中对测试结果做了分析与评价。

本文第 2 节分析语境信息在面向服务的应用中的用途;第 3 节提出语境信息使用模式,介绍其基本思想,并详细阐述工作原理;第 4 节介绍该模式在奥运信息服务平台 FLAME2008 上的实现原理;第 5 节是测试与分析;第 6 节对相关工作进行分析比较;最后在第 7 节进行了总结。

## 2 面向服务应用中语境信息的用途

面向服务应用中语境信息的用途与面向服务应用的构造和使用方式密切相关。文献[2,3]从面向应用的角度描述了 Web 服务的基本架构,其中的发布—查找—绑定过程是所有面向服务应用都遵循的使用模式,使用服务组合平台可以方便地构造和运行这样的应用,参考当前服务组合方面的主流工作 SWIG<sup>[4]</sup>, SWORD<sup>[5]</sup>, SELF-SERV<sup>[6]</sup>, FLAME2008<sup>[7]</sup>等,应用的构造与使用过程可以归纳如下:首先要为用户呈现一个符合其个性化需求的服务视图;然后用户可以使用组合工具将所需服务按一定的逻辑组合在一起,编制成流程,并对其中各节点的执行条

件、质量约束、数据关联等因素进行配置;最后执行引擎根据配置信息执行应用。

针对上面所述的面向服务应用的特点及使用模式,目前我们将语境信息在应用运行时的用途细化成如下几类:

(1) 服务选取。应用中使用的是抽象服务,运行时要选择一个与之对应的具体服务,此时将语境信息作为选取条件;

(2) 路径选择。作为流程中分支节点的路径选择条件,根据语境信息的值决定执行哪个路径;

(3) 服务触发。执行特定服务的触发条件,即处于特定环境时触发特定的服务;

(4) 隐含输入。运行具体服务时将特定的语境信息值自动作为服务的输入,而不需用户参与。

下面对这几种用途的核心操作进行分析,为了更清楚地阐述问题,首先给出如下定义。

**定义 1.** 称  $\sigma_{M(CtxInfo, Objprop)}$  ( $ObjectSet$ ) 为抽象操作,其中  $ObjectSet$  是目标对象的集合;  $M$  代表完成运算的函数,它包含两个参数: $CtxInfo$  是使用到的语境信息的集合,  $Objprop$  是相应的目标对象属性的集合,  $M$  对这两个集合中各元素的值进行计算;  $\sigma$  代表操作的功能。

该操作的含义是:对目标对象集合  $ObjectSet$  通过运算  $M$  完成  $\sigma$  功能。

**性质 1.** 服务选取(用途 1)和路径选择(用途 2)的操作过程符合抽象操作的定义。

事实上,服务选取和路径选择的操作过程的实质是对多个目标对象进行最优选取,此时的目标对象在服务选取中就是各候选服务,在路径选择中就是各候选路径,集合  $CtxInfo$  和  $Objprop$  中所包含的元素依赖于流程中该节点上的配置信息,每次使用都可能不同,  $M$  是完成匹配的函数,  $\sigma$  代表选择操作,因此符合抽象操作定义。

**性质 2.** 服务触发(用途 3)的操作过程符合抽象操作的定义。

服务触发的操作过程是将语境信息的值与配置信息中的预定义值进行比较判断,如果条件成立,由系统触发对服务的调用。此时  $ObjectSet$  中只有一个元素,即将要触发的特定服务。集合  $Objprop$  和  $CtxInfo$  中所包含的元素来源于配置信息,同时,  $Objprop$  中元素的值来源于配置信息中的预定义值,  $M$  是判断相等的函数,  $\sigma$  代表触发,因此符合抽象操作定义。

**性质 3.** 隐含输入(用途 4)的操作过程符合抽

象操作的定义。

隐含输入的操作过程只是将语境信息的值作为服务调用时的输入参数,此时 *ObjectSet* 中只有一个元素,即将要调用的服务,集合 *CtxInfo* 中所包含的元素来源于配置信息, *Objprop* 为空, *M* 代表取值函数,  $\sigma$  代表输入操作,因此符合抽象操作的定义。

性质 1~3 说明上面所述的几种用途的操作过程可以归结为一个规范的过程,这为使用模式的提出奠定了基础。

### 3 语境信息的使用模式

#### 3.1 基本思想

根据上一节的分析可以看出,各种用途都是将语境信息当作数据来使用,而敏感最终体现在应用层面,各种用法在实现敏感的过程具有一些共性特点,并可以用规范的方式描述为一个抽象操作,以此为基础,我们从实现的角度来考虑,将语境信息的使用过程按抽象操作的定义转化为一个操作步骤,如图 1 所示。

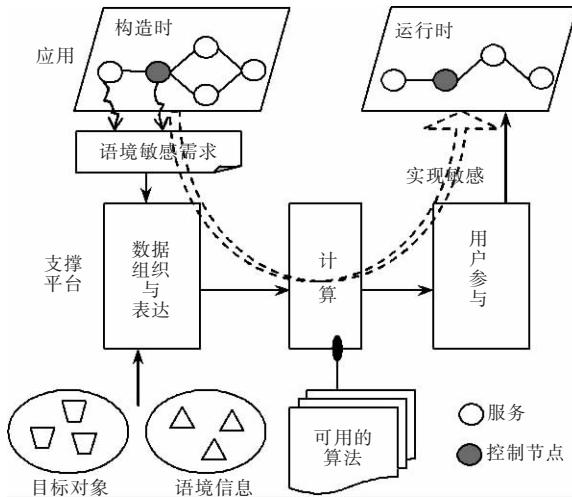


图 1 使用语境信息的操作步骤

如图 1 所示,在构造应用的时候设定语境敏感需求,所谓语境敏感需求就是对语境信息的使用需求,完成这项工作只需利用组合平台,将应用中某些节点上的数据项与相应的语境信息项关联。

使用的过程中,首先获得敏感需求和目标对象集,在此基础上收集相关的目标对象属性信息和语境信息,并对这些数据进行组织,形成抽象操作中的 *Objprop* 和 *CtxInfo*,针对不同的敏感过程其元素是不同的,这要求数据的表达具有可扩展性。

然后,应用外挂的算法 *M*,作用于 *Objprop* 和 *CtxInfo*,完成计算。实现中可能存在多种可选算法,比如用于选取时可以采用 SWA, MCDM<sup>[8]</sup> 等算法。在使用模式中,我们基于向量运算定义了一个计算框架,某个算法只要按规定的形式表达数据,就可以被应用于计算过程中。

另外,针对服务选取和路径选择两类用途中的选取操作,现有的算法,比如 MCDM 等,都只是依据目标对象之间的差异进行比较,常常会出现选出多个最优元素的情况,在本文中称之为存在冲突,为此在使用模式中为用户提供一种参与的手段,对结果进行优化处理,提高用户的满意度和系统的可靠性。

综合以上因素,我们以面向应用的观点,采用数据向量作为基本的数据表达方式,综合考虑目标对象的属性、用户的期望度及系统的使用策略,提出一种适用于服务组合平台的语境信息使用模式。下面详细阐述其工作原理。

#### 3.2 工作原理

为了清楚地描述问题,先给出基于向量运算的语境信息使用模式的形式化定义。

**定义 2.** 基于向量运算的语境信息使用模式 *P* 是一个五元组:

$$P = (S, s_0, s_f, E, \delta),$$

其中:

(1) *S* 是一个有限集,它的每个元素是处理过程中的一个状态,共有 5 个元素。*S* = {初始状态,数据可用状态,中间结果状态,优化结果状态,终止状态};

(2)  $s_0 \in S$ , 是唯一的初始状态;

(3)  $s_f \in S$ , 是唯一的终止状态;

(4) *E* 是一个内部处理(操作)的集合,它的每个元素都是状态变迁的条件,共有 5 个元素。*E* = {数据组织操作,计算操作,优化操作,选择,系统触发};

(5)  $\delta$  是一个从  $S \times E$  到  $S$  的转换法则; $\delta(s_i, e) = s_k$  意味着:在状态  $s_i$  处,如果发生事件  $e$ ,将转换到下一个状态  $s_k$ 。

根据该定义,可以将其工作过程用状态转换图描述,如图 2 中所示,其中包含 5 个状态,由 5 个内部处理实现状态的变迁。针对前面所述的语境信息的不同用途,使用模式会从不同的状态转入终止状态。

下面通过对图中状态及其变迁过程的阐述,说明该模式的工作原理。

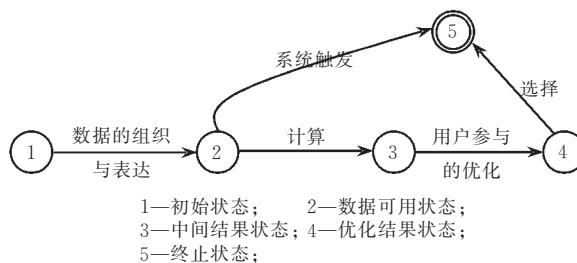


图 2 使用模式的工作过程

### 3.2.1 初始状态

进入初始状态,已经完成了对数据的最初准备,主要包含两项内容:一是得到了针对此次使用的敏感需求;二是得到了初始目标对象集.敏感需求在构造应用时产生,如图1,目标对象集的生成依据语境信息的用途,对于服务选取和路径选择两种用途,可以利用非语境因素对对象全集进行一次过滤而得到;对于服务触发和隐含输入两种用途,当前服务就是目标对象.初始数据准备好以后要对它们进行组织与表达,进入数据可用状态,这些将在下一小节阐述.

### 3.2.2 数据可用状态

数据可用状态形成了一个有效数据集合,该集合是一个有限集,由3个元素构成:

$$DataSet = \{CtxVector, ObjpropMatrix, UserDemand\}.$$

下面介绍集合中各个元素的含义及产生过程.

**定义3.** 称  $CtxVector = (c_1, c_2, \dots, c_n)$  为语境信息的数据向量,其中  $n \leq |S_{userCtx}|$ ,  $S_{userCtx}$  是用户所有语境信息项所构成的集合,  $c_1 \sim c_n$  代表本次所使用的各语境信息项的当前值.

根据初始状态中的敏感需求,得到相关语境信息项,将它们看作一系列的离散量,并调用底层接口获取当前值,就可以生成语境信息的数据向量.如果将该向量看成一个1行  $n$ 列的矩阵,那么矩阵的每列都代表一个语义,它决定了  $c_i$  的类型.

**定义4.**  $A$  和  $B$  是两个向量,若  $A$  中的元素  $a_i$  与  $B$  中的元素  $b_k$  具有相同的语义,称  $a_i$  与  $b_k$  同源,记为  $a_i \approx b_k$ .若  $A$  和  $B$  具有相同维数,且对应分量同源,即  $a_i \approx b_i, 1 \leq i \leq \text{维}(A)$ ,称向量  $A$  与  $B$  同质,记为  $A \cong B$ .

**定义5.** 有用属性是指在语境信息用法的一次实现过程中将会用到的目标对象属性.

比如针对服务选取,抽象操作中集合  $Objprop$  中的元素所对应的属性就是目标对象的有用属性.

**定义6.** 称  $ObjPV = (p_1, p_2, \dots, p_n)$  为目标对象属性向量,当  $p_i$  是目标对象的有用属性的值,且

$$CtxVector \cong ObjPV.$$

**定义7.** 目标对象属性矩阵定义为

$$ObjpropMatrix = (ObjPV_1, ObjPV_2, \dots, ObjPV_m)^T,$$

$ObjPV_i$  是目标对象的属性向量,  $m$  是目标对象的个数,  $T$  表示矩阵的转置.

与数据向量类似,目标对象属性矩阵的每列代表一个语义,在每次具体的应用中,其列数与数据向量的维度相等,并且对应列具有相同的语义.矩阵中值的产生过程与前面所述的用途相关:

在服务选取用途中,其值来源于服务的非功能属性,首先根据敏感需求中的配置信息确定语境信息与非功能属性项的对应关系,然后由平台从服务的注册信息中获得相关属性项的值;

在路径选择用途中,使用的属性由所有分支中和语境信息相关的条件组成,对于每一行,如果某列所代表的条件在该分支中出现,则这个元素的值为1;否则为0;

在服务触发用途中,属性值来源于配置信息中的预定义值;在隐含输入用途中,目标对象属性向量为空.

**定义8.** 称  $UserDemand = (ud_1, ud_2, \dots, ud_n)^T$  为用户需求向量,其中  $ud_i \in N \wedge 1 \leq ud_i \leq 10$ ,  $N$  代表自然数,  $n=$  维数 ( $CtxVector$  本次使用).

$ud_i$  的值是用户在本次使用中对语境信息数据向量中第  $i$  项的重视程度,代表了各相关语境信息项的使用优先度.该向量是一个列向量,其行数和数据向量的列数相等,其值由用户根据自己的需要进行设定.

对于服务触发和隐含输入两种用途,从该状态直接通过系统触发的操作完成敏感,转入终止状态.对于服务选取和路径选择两种用途,要应用外挂的算法对  $ObjpropMatrix$  和  $CtxVector$  进行运算实现最优选取.由定义2中的计算操作触发,进入中间结果状态,见下一小节.

### 3.2.3 中间结果状态

**定义9.** 中间矩阵按如下公式定义:

$$Temp = F(ObjpropMatrix, CtxVector),$$

其中,  $F$  是可用的选取算法.

为了实现服务选取和路径选择两种用途,需要从数据可用状态通过计算操作进入中间结果状态,此时产生了中间矩阵  $Temp$ .计算操作可以使用不同的算法,只要按向量的形式组织数据即可.比如服务选取时可以采用 SAW<sup>[8]</sup> 算法的原理,来计算每个元素的值:

$$T_{i,j} = \begin{cases} \frac{Q_{i,j} - C_j}{Q_j^{\max} - Q_j^{\min}}, & Q_j^{\max} \neq Q_j^{\min} \\ 1, & Q_j^{\max} = Q_j^{\min} \end{cases} \quad (1)$$

其中  $T_{i,j}$  是中间矩阵第  $i$  行  $j$  列的元素,  $C_j$  是第  $j$  个语境信息项的值,  $Q_{i,j}$  是目标对象属性矩阵中  $i$  行  $j$  列的元素; 而路径选择时可以使用下面公式来计算:

$$L_{i,j} = Q_{i,j} \times (Q_{i,j} - F(C_j)),$$

其中  $F(C_j) = \begin{cases} 0, & \text{当第 } j \text{ 项的值满足条件} \\ 1, & \text{当第 } j \text{ 项的值不满足条件} \end{cases}$

$$T_{i,j} = L_{i,j} \times \prod_{k=1}^n \xi_i(L_{i,k}) \quad (2)$$

其中  $\xi_i$  是第  $i$  个分支条件所构成的逻辑运算函数.

### 3.2.4 优化结果状态

根据中间结果状态中的 **Temp** 矩阵的值完成选取, 经常出现结果集中元素不唯一的情况. 为此我们为用户提供了参与的手段, 对中间结果做进一步优化.

**引理 1.** **Temp** 与 **UserDemand** 可以进行矩阵的乘法运算.

证明. 由定义 3, 7~9 此命题成立.

由此, 我们给出优化操作中所需的数据定义.

**定义 10.** 称  $Score = Temp \times UserDemand$  为得分向量,  $\times$  是矩阵的乘法操作.

**Temp** 和 **UserDemand** 的含义见定义 8 和 9, 优化的过程就是生成得分向量的过程, 然后通过系统支持的操作选择 **Score** 中值最大的元素. 如果这样的元素个数仍不唯一, 可以根据系统策略对结果集中的数据进行处理. 对服务选取来说, 可以随机推荐一个给用户, 也可以为用户呈现结果集中所有服务的列表; 而在路径选择中却造成了逻辑冲突, 此时可以将分支改为并行或者随机选择一个执行次序, 串行执行被选中的分支.

### 3.2.3 终止状态

经过前面的处理, 语境信息已经被有效利用, 此时需要有组合平台的支撑将敏感体现到应用层面, 使用模式的操作过程结束.

### 3.3 使用模式的计算定理

以 3.2 节中的定义 3~10 为基础, 从计算的观点来看, 可以得到如下定理.

**定理 1.** 使用模式的计算过程可表示为

$$Result = Call(Temp \times UserDemand),$$

$$Temp = F(ObjpropMatrix, CtxVector),$$

其中  $Call$  表示系统支撑的操作,  $Result$  是敏感操作产生的结果.

证明. 3.2 节中给出使用模式的状态变迁的过程就是本定理的构造性证明过程.

本定理的意义在于: 将抽象的操作过程转化为具体的计算过程, 直接指导系统的实现, 详见第 4 节实现原理的介绍.

### 3.4 举例

以服务选取的过程为例, 假设希望运行一个查询餐馆的服务, 在所有服务集合中有 3 个具体服务  $S_{R1}, S_{R2}, S_{R3}$  与之对应. 用户对服务的三个非功能属性项设置了语境敏感约束, 它们分别是: 服务的价格、服务提供者位置和服务要求的网络带宽, 所谓语境敏感约束就是根据当前语境信息值来选取服务, 同时用户设置了对这几项的希望度分别是 2, 7, 3, 这几个值构成了用户需求向量, 即  $UserDemand = (2, 7, 3)^T$ . 这些信息构成了该节点的敏感需求.

运行时首先获取初始目标对象集, 在这里采取按功能选取的办法, 从服务全集中将具有查询餐馆功能的服务选中:

$$GoalObjectSet = (S_{R1}, S_{R2}, S_{R3}),$$

进入初始状态. 然后, 根据敏感需求, 调用底层的数据接口获取相关语境信息的当前值, 生成数据向量  $CtxVector = (25, (116.20, 39.30), 128)$ , 其中 3 个数据分别是用户消费水平、用户当前位置和用户设备的网络带宽的当前值. 并将服务的属性组成服务属性矩阵:

$$ObjpropMatrix = \begin{bmatrix} 20, (116.10, 39.40), 56 \\ 30, (116.30, 39.25), 256 \\ 10, (116.20, 39.10), 128 \end{bmatrix},$$

矩阵的第 1, 2, 3 行分别代表  $S_{R1}, S_{R2}, S_{R3}$  三个服务, 第 1, 2, 3 列分别是各个服务的服务价格、服务提供者位置和服务带宽需求的值, 它们与语境信息中的用户消费水平、用户当前位置和用户设备的网络带宽相对应. 此时到达数据可用状态.

再次, 应用选取算法, 在这里我们采用 SAW<sup>[8]</sup> 算法的原理依据公式(1)进行计算, 得到中间矩阵

$$Temp = \begin{bmatrix} 0.25 & 1.6 & 0.28 \\ 0.25 & 1.27 & 1 \\ 0.75 & 2.27 & 0 \end{bmatrix},$$

其中的位置项按距离值进行计算得到, 此时进入中间结果状态.

最后, 按定义 10 经过计算得到得分向量的值  $Score = (12.54, 13.39, 17.39)$ , 因此服务  $S_{R3}$  被选中, 完成优化, 此时  $ResultSet$  中只有一个元素, 不存在冲突, 利用组合平台提供的支撑, 将选中的服务返

回给应用,实现了应用对语境信息的敏感使用,进入终止状态.

## 4 实现原理

FLAME2008<sup>[7,9,10]</sup>是中国科学院计算技术研究所服务网格小组所开发的服务组合平台. 我们以该平台作为支撑,对本文所提出的语境信息使用模式进行了实现,在此过程中采用了一种类似反演的机制来提高系统的灵活性,适应应用对语境信息需求的变化. 此时,将组合平台抽象成满足用户特定需求的提供者,将语境信息使用模式的内部表达(即元模型)的某些方面显示化,使应用可以访问,如图 3 所示.

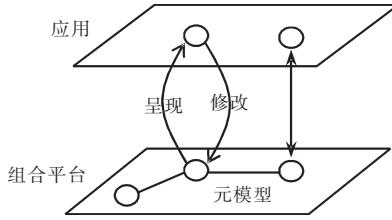


图 3 实现原理

针对每一次具体的使用,定理 1 中的目标对象属性矩阵 *ObjpropMatrix* 和数据向量 *CtxVector* 及用户需求向量 *UserDemand* 中所包含的内容和具体值是不同的,因此将它们作为元模型中对用户呈现的部分,用户根据自己的特定需求,通过一些设置改变这些元级信息的内容,从而使平台根据不同的用户需求产生不同的结果.

在 FLAME2008 中提供了图形化的界面,用户可以使用该工具创建并配置自己的应用,比如构造一个包含两个任务的流程,并设定如下配置信息:第 1 个任务根据当前位置选择服务,第 2 个任务根据服务的开销进行选择(细节参见文献[9]). 运行时,系统读取用户设定的配置信息决定 *ObjpropMatrix* 矩阵和 *CtxVector* 向量中所包含的元素内容,而其数据则由组合平台自动获得. 用户对每个语境信息项的希望度通过单独的工具进行设置(包含在组合工具中),应用运行时,系统根据上面所述的用户配置信息的语义决定使用哪些语境信息项,并取出对应项的希望度的值形成用户需求向量.

## 5 评 价

测试工作是基于目前在 FLAME2008 上的实

现进行的,代码采用 JAVA 语言开发(JDK1.4),以 XML 文档的形式存储各种中间文件及传递消息. 服务组合平台运行在一台联想 PC 机上,该机器采用 Intel Pentium III 处理器,主频 866MHz,320MB 内存,使用 WindowsXP SP2 操作系统. 测试的目的在于对使用模式本身进行分析和评价,而不是针对整个组合平台,因此为了消除网络传输过程中的影响,将 UDDI 服务器也运行在同一台机器上,而且所有的服务信息都注册在这个 UDDI 服务器上. 对该模式进行分析和评价的目标是找出使用的语境因素的个数、目标对象个数与时间及冲突之间的关系(冲突的定义见 3.1 节),而不是对某个算法进行评测,因此我们将模式用于一个具体应用——服务选取,选用一个固定算法(3.3.2 节的示例)进行测试,测试内容由以下两部分组成:

### (1) 影响效率的因素分析

应用使用模式的过程中,考虑的语境因素越多,计算量就越大,因此执行时间会相应地延长,但得到的结果集存在冲突的可能性会降低;相反考虑的语境因素少会减少执行时间,增加结果冲突的可能性. 以服务选取应用进行测试,图 4 和图 5 分别展现了服务选取过程中语境因素的多少、服务的个数与时间和冲突的关系.

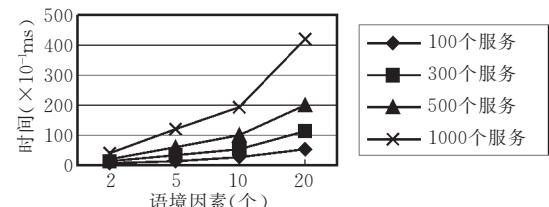


图 4 时间关系

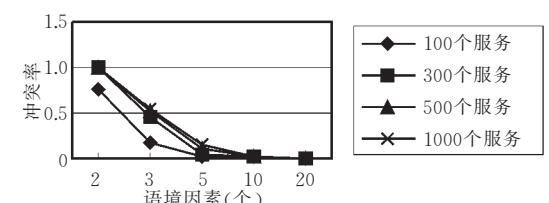


图 5 冲突率

如图 4,图 5,X 轴代表使用的语境因素的个数,Y 轴分别代表执行需要的时间和冲突率,考虑到网格环境下服务数量是巨大的,使用的数据集中分别包含 100 个、300 个、500 个和 1000 个可选具体服务的情形,图中每个节点所代表的含义是:可选服务集中包含  $x$  个具体服务,使用  $n$  个语境因素时进行服务选取所用的时间(或冲突比例),其值通过对每个

固定的选取条件(比如可选服务集中包含 100 个具体服务时,使用两个语境因素)重复执行 10000 次,然后取总平均值所得到。

由图 4 可以看出,当使用的语境因素的个数增加时,所用的时间基本上是线性增加的,由图 5 知,当使用的语境因素的个数小于 5 时,其数目的增加对降低冲突率的作用比较明显,使用 10 个语境因素时,冲突率在 2% 左右。当可选服务集中包含的具体服务数小于 300 个的时候,使用 5 个左右的语境因素比较理想,对应的具体服务大于 300 时,使用 5 到 10 个语境因素比较理想。

### (2) 使用模式的分解执行结果分析

本文所提出的语境信息使用模式的操作过程是个逐步求精的步骤,图 6 和图 7 分别展现了在中间结果状态结束与在优化结果状态结束时,效率与精度的关系。仍以服务选取为例进行测试(其它应用同样),测试数据为一个业务服务对应 300 个具体服务,X 轴代表使用的语境因素的个数,Y 轴分别代表执行需要的时间和冲突率,两条曲线分别是应用需求向量的情况和不应用时的情况,曲线上每个节点是对固定条件执行 10000 次,然后计算平均值得到的。由图 6 可知,在两个状态下终止对执行时间的影响不大,而由图 7 可知在优化结果状态终止,对降低冲突率具有显著的作用。因此在我们的使用模式中不包含中间结果状态到终止状态的转换。

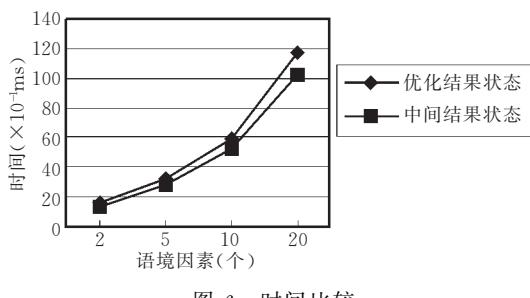


图 6 时间比较

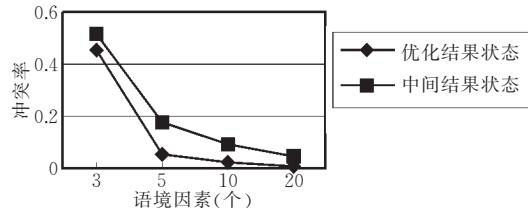


图 7 冲突率比较

中使用语境信息,实现应用层面的敏感,关于单纯的语境敏感领域的相关工作可以参考文献[11~14]等,关于 Web 服务组合和中间件的工作可以参考文献[5,6,15,16]等,本文主要针对已有的具有敏感特性的组合平台中的一些工作进行比较。

在移动计算中间件 CARISMA<sup>[17]</sup> 中采用反演机制来提高移动应用的适应性和语境敏感能力。这里将中间件抽象为用户定制的服务提供者,同时维护和更改语境信息,将特定的语境信息下调用特定的服务的关联关系定义为反演元模型,用户通过修改这些关联关系来改变应用需求,而中间件则自动检测应用需求的变化,将合适的服务呈现给使用者。由于用户可以自己定义服务和语境信息之间的关联关系,因而关系之间可能存在冲突,解决这个问题时采用了微观经济学中的“竞标”原理。这里对语境信息的用法比较单一,将其与具体的服务直接绑定,作为执行该服务的条件,与服务组合中应用语境信息触发服务执行的用法类似,但在服务组合中间件中应用的环境和资源的使用方式与 CARISMA 都不同,而且语境信息还有其它一些重要应用。

服务组合中间件 AgFlow<sup>[18]</sup> 能够自动感知 Web 服务的 QoS 变化,并根据服务的 QoS 和用户的需求实现服务的自动组合。其核心在于给出了一个如何根据用户的全局 QoS 约束完成各个任务节点的服务选取,其实质是一个规划问题。对服务 QoS 的感知和对语境信息的感知是十分类似的,但在 AgFlow 中只是将敏感信息作为规划问题的输入,没有全面考虑敏感对服务组合所带来的其它好处及实现。

与上面两项工作相比,本文的工作在于对几种语境信息用途做分析,将其执行过程中的共性特点进行抽象,提出了一种语境信息的使用模式,同时考虑了更多的相关因素,并提供了在极端情况下的用户最终决定权,而且在计算的过程中可以应用不同的算法,不仅可以有效地使用语境信息,还简化了组合平台本身的维护。

Context ToolKit<sup>[19]</sup> 是一个开放源码的构造语境敏感应用的框架,它以 widgets 为元素,在其内部组件之间以发布订阅机制完成消息的分派,这种结构可以屏蔽传感器的变化对应用带来的影响,具有良好的复用性,软件人员可以扩展框架中所提供的软件组件来创建语境敏感应用。这项工作的重点是语境信息的获取,而不考虑如何使用这些信息,其研究定位与我们是不同的,但其方法与思路可以

## 6 相关工作

本文工作的重点在于研究如何在服务组合平台

用在服务组合平台中对语境信息生命周期第一、二阶段的实现上。

RCSM<sup>[20]</sup>系统是为了适应普适计算环境所开发的一个语境敏感中间件。主要工作在于同时实现语境敏感和自主通信,完成运行时刻语境数据的获取、监视和探测,并触发对象实现的不同行为。它的实现方式与传统中间件类似,将语境敏感的应用建模成语境敏感对象,根据用户的定制产生一个适应性对象容器,完成和底层的交互。它对用户的编程水平具有较高要求,普通用户很难使用。

## 7 总 结

本文的主要贡献在于提出了一个基于向量运算的语境信息使用模式,应用此模式,服务组合平台能够以统一的方式对语境信息的各种用途提供支撑,有效支持了面向服务应用在运行时的语境敏感需求,并简化了平台本身的开发与维护。文章首先分析了语境信息在面向服务的应用中的用途,然后提出并阐述了该模式的基本思想及工作原理,并以实例详细说明其工作过程,最后,结合在 FLAME2008 平台上的实现进行测试,并对测试结果进行了分析与评价。

目前在面向服务的应用中对语境信息的使用主要集中在与服务和过程相关的用途上,具有一定的局限性,今后工作中将考虑在面向服务应用的其它方面应用语境信息,比如在角色与权限的管理中如何根据用户所处的当前情形为其授予不同的权限等。其次,在服务选取等用法中,生成目标对象属性矩阵时,需要组合平台与 UDDI 服务器之间进行交互,其交互协议、缓存机制、数据更新机制及实现与部署方案需要进一步研究。另外,当服务组合平台面向跨领域应用的时候,需要对多个领域本体进行协调,以保证语义的一致性,这是一个难题。

## 参 考 文 献

- 1 Weiser M.. The computer for the 21st century. Scientific American, 1991, 265(3): 66~75
- 2 Feisi Science and Technique Center of Research and Development. Application and Development of Java Web Services. Beijing: Publishing House of Electronics Industry, 2002, 190~196
- 3 Shmueli O.. Architectures for internal Web services deployment. In: Proceedings of the 27th International Conference on

Very Large Data Bases, Roma, 2001, 641~644

- 4 Kiciman E., Melloul L., Fox A.. Towards zero-code service composition. In: Proceedings of the 8th Workshop on Hot Topics in Operating Systems, Schloss Emlau, 2001, 172
- 5 Ponnekanti S. R., Fox A.. SWORD: A developer toolkit for Web service composition. In: Proceedings of the 11th International World Wide Web Conference, Honolulu, 2002
- 6 Benatallah Boualem, Sheng Quan Z., Dumas Marlon. The self-serv environment for Web services composition. IEEE Internet Computing, 2003, 7(1): 40~48
- 7 Holtkamp B., Gartmann R., Han Y.. FLAME2008-Personalized Web services for the olympic games 2008 in Beijing. In: Proceedings of the eChallenges 2003, Bologna, 2003, 93~97
- 8 -L H. C., Yoon K.. Multiple criteria decision making. Lecture Notes in Economics and Mathematical Systems, New York: Springer-Verlag, 1981
- 9 Han Y., Geng H. et al.. VINCA——A visual and personalized business-level composition language for chaining Web-based services. In: Proceedings of the 1st International Conference on Service-Oriented Computing, Berlin, 2003, 165~177
- 10 Han Y., Zhao Z. et al.. CAFISE: An approach enabling on-demand configuration of service grid applications. Journal of Computer Science and Technology, 2003, 18(4): 484~494
- 11 Chen G., Kotz D.. A survey of context-aware mobile computing research. Department of Computer Science, Dartmouth College, Technical Report: TR 2000-381, 2000
- 12 Eli Rohn. Predicting Context Aware Computing Performance. New York: ACM Press, 2003, 3(50): 1~17
- 13 Dey A. K.. Providing architectural support for building context-aware applications[Ph. D. dissertation]. Georgia Institute of Technology, Atlanta, 2000
- 14 Hong J., Landay J. A.. An infrastructure approach to context-aware computing. Human- Computer Interaction, 2001, 16(2~4): 287~303
- 15 Casati F., Ilnicki S., Jin L.-J., Krishnamoorthy V., Shan M-C.. Adaptive and dynamic service composition in eFlow. In: Proceedings of the International Conference on Advanced Information Systems Engineering, Stockholm, 2000, 13~31
- 16 Mascolo C., Capra L., Emmerich W.. Mobile computing middleware. In: Proceedings of Advanced Lectures on Networking, Pisa, 2002, 20~58
- 17 Capra L., Emmerich W., Mascolo C.. CARISMA: Context-aware reflective middleware system for mobile applications. IEEE Transactions on Software Engineering, 2003, 29(10): 929~945
- 18 Zeng Liang-Zhao, Benatallah B. et al.. QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering, 2004, 30(5): 311~327
- 19 Dey Anind K. et al.. A conceptual framework and a toolkit for supporting the rapid prototyping of Context-aware applications. Human-Computer Interaction, 2001, 16(2~4): 97~166

- 20 Yau S. S. , Karim F. , Wang Y. , Wang B. , Gupta S.. Reconfigurable context-sensitive middleware for pervasive computing.

IEEE Pervasive Computing, 2002, 1(3): 33~40



**ZHANG Cheng**, born in 1977, Ph.D. candidate. His current research interests include dynamic services composition and context-aware computing.

**HAN Yan-Bo**, born in 1962, Ph. D. , professor, Ph. D. supervisor. His current research interests include software integration and service grid.

## Background

The major research interests of this group are software integration, dynamic services composition, service grid, and context-aware computing. This paper is primarily supported by the National Nature Science Foundation of China(NSFC) under grant No. 90412005, which involves in the research works about dynamic services composition including context-aware technology. This paper analyses the usage of context information in services-oriented applications, and proposes a

context information usage pattern based on vector operation, which is suitable for service composition platforms. Using this pattern, a service composition platform can make use of context information in a uniform way. This not only effectively supports context-aware requirements of applications at run-time but also simplifies the development and the maintenance of composition platforms.