

文章编号: 1001-2486(2009)03-0082-05

面向普适计算应用的构件定义语言 UC DL<sup>\*</sup>刘惠<sup>1</sup>, 丁博<sup>1</sup>, 史殿习<sup>1</sup>, 刘明<sup>2</sup>

(1. 国防科技大学 计算机学院, 湖南 长沙 410073; 2. 九江职业技术学院, 江西 九江 332007)

**摘要:** 普适计算的适应性特征对于软件重用的要求决定了构件模型是目前普适计算软件体系结构的理想选择。在面向普适计算的自适应软件平台 UbiStar 的研究过程中, 为了准确地对普适应用中构件的外部特征进行抽象描述, 参考了 CORBA 构件模型使用 IDL3 和 CIDL 对构件进行特征抽象的方法, 结合 UbiStar 普适计算平台的特点与需求, 提出了一种适用于普适计算环境的构件定义语言 UC DL, 并且给出了一个应用示例。

**关键词:** 普适计算; 自适应体系结构; 构件模型; 构件定义语言

中图分类号: TP311 文献标识码: A

## UC DL—A Ubiquitous Component Definition Language for Pervasive Computing-oriented Applications

LIU Hui<sup>1</sup>, DING Bo<sup>1</sup>, SHI Dianxi<sup>1</sup>, LIU Ming<sup>2</sup>

(1. College of Computer, National Univ. of Defense Technology, Changsha 410073, China;

2. Jiujiang Vocation and Technique College, Jiujiang 332007, China)

**Abstract:** Because of the requirement of software reuses determined by adaptability, the component model has become an ideal choice of software architecture in research of pervasive computing. To abstract the characteristics of components precisely in the pervasive computing-oriented environments, as a part of research in a self-adaptive pervasive computing-oriented software platform UbiStar, a component definition language UC DL is presented. The design of UC DL refers to the manner of using IDL3 and CIDL to abstract the characteristics of components in CORBA Component Model. In the end, an example of the application of UC DL is also presented.

**Key words:** pervasive computing; self-adaptive architecture; component model; component definition language

随着芯片制造、无线通信、人机交互和网络计算技术等持续快速发展, 以泛在性、便捷性和适应性为特点的普适应用也正从纸面上的构思逐渐成为现实<sup>[1]</sup>, 国内外很多著名的研究机构和大专院校都已经开展了以面向普适计算的软件平台为核心的相关研究工作, 形成了 Oxygen<sup>[2]</sup>、Gaia<sup>[3]</sup>、Aura<sup>[4]</sup>、iRos<sup>[5]</sup>、One World<sup>[6]</sup>、Smart Classroom<sup>[7]</sup> 等一系列重要的研究成果。

在构造普适计算软件平台以支撑具有普适特性的典型应用的过程中, 如何满足适应性要求是绝大多数普适计算研究工作的重点。作为一种有效的软件重用手段<sup>[8]</sup>, 以 CORBA 构件模型 CCM 为代表的分布式构件技术, 尤其是适应嵌入式环境下系统资源受限特点的轻量级 CORBA 构件模型, 在普适计算相关的研究中延续并且增强了其基础软件的作用。另一方面, 普适应用在实时机制、上下文相关机制、与应用程序框架的交互机制等方面有别于传统分布应用的独特要求, 决定了在设计普适计算软件体系结构、实现普适计算软件平台以支撑普适应用时, 不能原样照搬已有的 CCM 或者轻量级 CCM 等构件模型, 必须有针对性地进行相应的裁剪或者扩展以适应普适计算环境的特点。

以 CCM 为基础, 结合普适计算应用的具体需求, 面向普适计算的自适应软件平台 UbiStar 提出了一种基于自主单元的自适应体系结构来对普适计算空间及其中的实体进行构件建模<sup>[9]</sup>。作为 UbiStar 平台开发工具的有机组成, 在参考 CCM 的接口定义语言 IDL3 和构件实现定义语言 CIDL 规范的基础上, 本文提出了一种普适计算环境下的构件定义语言 UC DL, 对普适应用中构件的外部特征进行抽象描述,

\* 收稿日期: 2008-09-04

基金项目: 国家 863 计划资助项目(2006AA01Z198)

作者简介: 刘惠(1971-), 女, 副研究员, 硕士。

并且给出了 UC DL 的应用示例。

## 1 自适应普适计算平台 UbiStar 概况

UbiStar 普适计算平台将普适计算空间中的计算设备或者资源抽象封装为自主单元。自主单元是一类具有 Join/Adapt 操作语义的软件节点。所谓 Join 操作语义,是指自主单元在运行时刻可以按需加入或者退出应用系统;所谓 Adapt 操作语义,是指自主单元能够根据当前上下文进行决策,适应性地修改自身的行为,从而表现出某种自主性。如图 1 所示,在普适应用中,UbiStar 普适计算平台采用“基层构件+ 元层容器”的体系结构来实现自主单元及其 Join/Adapt 语义。这样,整个普适计算空间就被抽象为由若干自主单元所组成的一个开放集合。

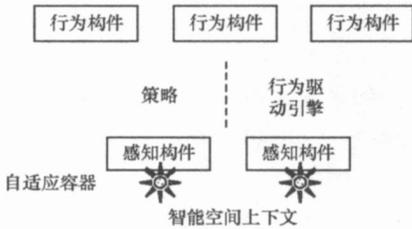


图 1 自主单元结构

Fig. 1 Architecture of autonomic unit



图 2 UbiStar 普适计算平台基本部件

Fig. 2 Components of UbiStar

UbiStar 普适计算平台的基本部件如图 2 所示,提供了完整的普适应用运行环境和相应的开发工具,具体包括自主单元体、发起者、UC DL 编译器和应用系统开发工具等部分。

## 2 普适计算环境下的构件定义语言 UC DL

考虑到普适计算环境泛在性的特点,即有多种异构的计算设备和网络资源可能参与到普适应用之中,上文所述的 UbiStar 普适计算平台势必要解决这样一个问题,即必须有对各类构件的外部特征做一个独立于具体构件实现的抽象描述,并且这种抽象描述在构件模型的层面上不与具体的程序设计语言绑定。这样,使用不同编程语言、运行于不同网络环境中的代码实现才能对某个具体的构件有准确的理解,从而实现构件之间的互操作。

### 2.1 UC DL 的提出

在典型的分布构件应用中,CCM 的 IDL3 语言提供了一个比较理想和相对成熟的方式来解决构件外部特征抽象描述的问题:通过抽象构件模型提供 CCM 构件接口特性的定义,通过构件实现框架引入 CIDL 语言定义构件实现的编程模型。但是,如果在普适计算环境下直接使用 CCM 的这种构件描述方式可能不完全切合很多相对轻量级的普适应用场景,而且 IDL3 没有提供描述上下文接口的方式,从而无法区分感知构件和行为构件。因此,以 Gaia 等为代表的一部分普适计算研究在对构件进行抽象时,还是被迫在一定程度上牺牲了对构件之间互操作特性的支持,转而使用具体的编程语言,例如 C++ 等来描述这些上下文相关的构件接口。

鉴于此,在参考 CCM 模型使用 IDL 语言描述构件特性这个思路的基础上,综合考虑 UbiStar 普适计算平台所使用的基于自主单元的自适应体系结构,以及 UbiStar 普适计算平台中构件实现更加接近于自描述 CORBA 对象的特点,UC DL 在设计上以 IDL2 为蓝本,首先对其中不应用于普适计算环境的元素进行了针对性的裁剪,其次加入了普适应用中构件描述所必需的“context”等关键字,相应地对其描述能力进行了扩充,使得 UbiStar 普适计算平台支撑的应用中不同的构件提供者或者实现者之间可以对某个构件有着相同的、准确的理解。

### 2.2 UC DL 语法规则

在所使用的数据类型及接口的定义规则上,UC DL 采用了与 CORBA 规范接口定义语言 IDL2 相一致

的定义,本节仅补充以BNF范式表示的UCDL语法中对构件部分进行定义的相关内容,具体描述如下(其中未展开的符号请参见CORBA规范中有关IDL2的部分章节):

UCDL 语法规范:

`< component > ::= < component _ header > " { " < component __ body > " } "`

`< component __ header > ::= " component " < identifier > [ < support _ interfaces > ]`

`< support _ interfaces > ::= " supports " < interface __ name > { " , " < interface _ name > } *`

`< interface _ name > ::= < scoped _ name >`

`< scoped _ name > ::= < identifier > | " :: " < identifier > | < scoped _ name > " :: " < identifier >`

`< component _ body > ::= < label > *`

`< label > ::= < use _ context _ dcl > | < provide _ context _ dcl > | < use _ interface _ dcl >`

`< use _ context _ dcl > = " uses context " < context _ type _ spec > < identifier >`

`< context _ type _ spec > = < floating _ pt _ type > | < integer _ type > |  
< char _ type > | < boolean _ type > | < string _ type >`

`< provide _ context _ dcl > = " context " < context _ type _ spec > < identifier >`

`< use _ interface _ dcl > = " uses interface " < interface _ name > < identifier >`

### 2.3 UCDL 描述构件外部特征示例

在基于UbiStar普适计算平台开发普适应用的过程中,使用上文定义的UCDL语言对各类构件的外部特征进行抽象描述需要完成的工作如图3所示。

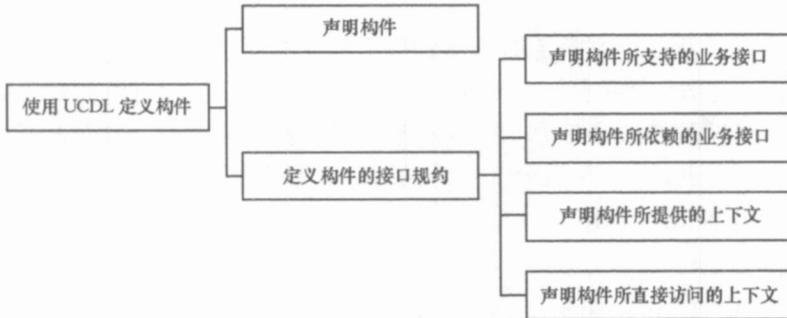


图3 使用UCDL定义构件

Fig.3 Using UCDL define components

#### (1) 声明构件

在UCDL中可以使用“component”关键字声明一个构件,如下所示:

使用UCDL声明构件

```

component TempSensor //定义一个名为TempSensor的温度获取构件
{
};
  
```

上述UCDL代码定义了一个既不对外提供接口也不依赖其他接口的构件。接下来将逐一说明如何使用UCDL为该构件添加各种不同的接口规约。

#### (2) 定义构件的接口规约

如图3所述,构件的接口规约由构件所支持的业务接口、构件所依赖的业务接口、构件所提供的上下文和构件所直接访问的上下文四大部分组成。

首先,构件支持的业务接口是构件对外提供功能的主要方式。在 UC DL 中,接口的定义方式与 CORBA IDL 完全一致,声明构件对该接口的支持则使用“supports”关键字。基于 UbiStar 普适计算平台进行普适应用开发时,使用 UC DL 定义的每个构件可以支持多个业务接口。定义构件对外所提供的业务接口可以示例如下:

---

使用 UC DL 声明构件支持的业务接口

```
interface SensorDevice //定义一个名为 SensorDevice 的接口
{
    void TurnOn(); //定义 SensorDevice 接口对外提供的方法
    void TurnOff();
};

component TempSensor supports SensorDevice
{
};
```

---

其次,某个构件所提供的上下文可以作为某条策略的条件而作为前提判断的依据,也可以被其他构件所直接访问。声明构件所提供的上下文使用“context”关键字,上下文的类型可以是所有 CORBA 的基本类型及 string、OctetSeq 等复合类型。下面以温度获取构件 TempSensor 提供一个长整型的上下文为例,说明 UC DL 如何描述某个构件可以提供的上下文信息。

---

使用 UC DL 声明构件提供的上下文

```
component TempSensor supports SensorDevice //形成完整的 TempSensor 定义
{
    context long Temperature;
};
```

---

接下来 UC DL 分别使用“uses interface”和“uses context”关键字声明某个构件所依赖的业务接口和所直接访问的上下文。假设一个温度计构件 Thermometer 依赖一个 SensorDevice 接口和一个长整型的 InputTemp 上下文,从而控制前文所定义的温度获取构件 TempSensor 的动作并且显示当前温度,那么 Thermometer 构件的 UC DL 描述应如下所示:

---

使用 UC DL 声明构件所依赖的接口和直接访问的上下文

```
component Thermometer //定义一个温度计构件
{
    uses interface SensorDevice device;
    uses context long InputTemp;
};
```

---

从上面的例子中可以看出,在使用 UC DL 声明所依赖的业务接口和所直接访问的上下文时,不仅要给出类型信息 SensorDevice 和 long,还要给出接口和上下文对应的名称 device 和 InputTemp。这些名称在构件定义时被用来标识该构件所依赖的端口,而与构件运行时具体绑定的变量名称没有关系。运行时的变量绑定是基于相应的类型信息,例如上述温度计构件 Thermometer 所依赖的 InputTemp 上下文在运行时可以绑定到温度获取构件 TempSensor 提供的上下文 Temperature。

## 2.4 基于 UbiStar 的普适应用构件开发流程

基于 UbiStar 普适计算平台进行构件开发的基本流程如图 4 所示。首先,应用开发者或者构件提供者使用 UC DL 语言对各类行为构件或者感知构件的外部特征进行抽象描述。其次,使用 UbiStar 所提供的 UC DL 编译器将构件的 UC DL 定义映射为构件框架和构件的 XML 自描述。然后,构件的编写者基于

UCDL 编译器生成的构件框架完成具体的构件实现, 经过编译后得到构件的可执行代码。最后, 构件的编写者可以指定一些自定义的构件描述信息(可选), 并与 UC DL 编译器生成的构件描述信息一起合成为最终的 XML 构件自描述。基于这样一种与传统分布式应用, 尤其是基于 CORBA 的分布式应用开发过程相类似的流程, 将有助于减少分布式应用开发者学习并且适应一种新的开发模式所需的时间和过程。

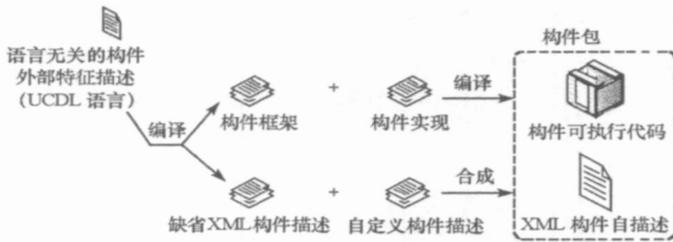


图4 普适应用构件开发流程

Fig. 4 Development of components in pervasive computing-oriented applications

### 3 结束语

基于面向普适计算的自适应软件平台 UbiStar 的实践, 在参考了 CORBA 构件模型 CCM 的接口定义语言 IDL3 和构件实现定义语言 CIDL 规范的基础上, 综合考虑了普适应用的特点与需求, 提出了适用于普适计算环境的构件定义语言 UC DL, 以一种分布式应用开发者所熟悉的方式对普适应用中的构件外部特征进行抽象描述, 使得不同的构件提供者 and 使用者之间可以对每个构件有着准确的理解。

### 参考文献:

- [1] 丁博, 王怀民, 史殿习. 普适计算中间件技术[J]. 计算机科学与探索, 2007, 1(3): 241-254.
- [2] Jean K. Talk to the Machine[J]. IEEE Spectrum, 2002, 39(9): 60-64.
- [3] Ronin M, Hess C K, Cerqueira R, et al. Gaia: A Middleware Infrastructure to Enable Active Spaces[J]. IEEE Pervasive Computing, 2002, 1: 74-83.
- [4] Garlan D, Siewiorek D P, Smailagic A, et al. Project Aura: Toward Distraction-free Pervasive Computing[J]. Pervasive Computing, 2002, 1(2): 22-31.
- [5] Ponnkanti S R, Johanson B, Kiciman E, et al. Portability, Extensibility and Robustness in iROS[C]//1st IEEE International Conference on Pervasive Computing and Communication, 2003.
- [6] Gimm R, Davis J, Lemar E, et al. System Support for Pervasive Applications[J]. ACM Transactions on Computer System, 2004, 22(4): 421-486.
- [7] Gu H L, Shi Y C, Xu G Y. Probe of the Problem, Challenge and Solution of Wireless Network Project Supporting Pervasive Computing in Smart-classroom[J]. Mini+micro Systems, 2005, 26(3): 367-370.
- [8] Frakes W B, Kang K. Software Reuse Research: Status and Future[J]. IEEE Transactions on Software Engineering, 2005, 31(7): 529-536.
- [9] 史殿习, 丁博, 李骁, 等. 面向普适计算的自适应软件平台: 概念与架构[C]//第四届和谐人机环境联合学术会议, 2008.