

一种全面支持后续开发的嵌入软构件模型

夏 苑 张为群

(西南大学计算机与信息科学学院 重庆 400715)

摘 要 基于构件的软件开发(CBD)逐渐成为嵌入式软件开发的主要开发方法之一。它可将构件开发的诸多优点(如降低开发成本,提高开发效率,缩短产品上市时间等)带入到嵌入式软件开发当中去。一个好的嵌入软构件模型能对基于构件的嵌入式软件的后续开发起到至关重要的作用。提出了一种嵌入软构件模型——ESCM。它能对之后的嵌入式软件构件的检索、组装及集成测试等提供有力的支持。

关键词 CBD,嵌入软构件模型,ESCM

Comprehensive Follow-up Support for the Development of Embedded Software Component Model

XIA Yuan ZHANG Wei-qun

(College of Computer and Information Science, Southwest China University, Chongqing 400715, China)

Abstract Component-based Software Development (CBD) has gradually become the main of the embedded software development methods. It will develop component of the many advantages (such as lower development costs, increase development efficiency, reduce time to market, etc.) to bring them to embedded software development. A good embedded software component model plays a crucial role to the development follow-up. This paper presented a model of embedded software component ——ESCM. It has provided strong support for the retrieval of embedded software components, assembly and integration testing.

Keywords CBD, Embedded software component, ESCM

1 引言

嵌入式系统已具有相当长的历史,最初用于军事、航空航天、工业过程控制等领域。从 20 世纪 90 年代中后期开始,互联网、通信、信息家电等的兴起使嵌入式系统渗透到了人们的日常生活当中。嵌入式应用的迅速增长要求有更先进的软件开发方法来满足市场的需求。

基于构件的开发(CBD)是当前软件开发的一个主流和发展趋势。把基于构件的软件工程(CBSE)的诸多方法和工具应用到嵌入式软件的开发中来将极大地提高嵌入式产品的质量及生产效率。

CBSE 的生命周期分为单个构件的生产和基于构件的软件开发两个并发的过程,由两个不同的生产实体来完成。单个构件的生产是为基于构件的软件开发提供可复用的软件单元以及为系统集成测试提供可利用的测试信息,因此是整个 CBSE 的基石。

构件的生产者必须在一定的构件模型指导下进行构件的生产,因此构件模型的优劣直接关系到基于构件的软件开发成败。

在嵌入式软件开发中,由于嵌入软构件的特殊性,如资源有限性、与硬件环境的高依赖性、时间限制性等使得嵌入软构件模型对于单个嵌入软构件的生产及基于构件的嵌入式软件开发及测试有着更大的指导意义,这也是本文研究的重点。

2 对现有的嵌入式软构件模型的分析

基于构件的嵌入式软件开发还是一个较新的研究热点和应用领域,因此对嵌入式软构件模型的研究还不充分。

在通用软件生产领域的几种构件模型,如 3C 模型,COM/DCOM,JavaBeans,CORBA 等,由于嵌入式软件的特殊性,都几乎无法应用到嵌入式软件开发中去。

在嵌入式软件领域,国外的研究者提出了几种嵌入式软件模型^[12],如:比利时 SEESCOA 项目的 CCOM 模型,飞利浦公司用于消费电子嵌入式系统的 Koala 组件模型,ABB 等公司参与的用于现场设备中嵌入式系统的 PECOS 组件模型。但这些模型缺少对嵌入软构件的非功能属性这一重要特性进行描述和度量,也缺乏对嵌入式软件的后续开发,如构件检索和集成测试的支持。

因此,本文针对当前嵌入式软构件模型的研究现状,提出一种嵌入式软构件模型——ESCM,此模型对构件的功能和非功能属性都进行了描述,对嵌入式软件构件的检索、组装及集成测试提供了有力的支持。

3 嵌入式软构件模型——ESCM

本文提出的嵌入式软构件模型是基于体系结构方法下的。为了支持系统的层次化分解,引入了复合构件的概念。系统可被看作是特殊的复合构件,当一个小系统集成到大系

统中时,仅仅是一个普通的复合构件,在这个意义上,我们认为原子构件是在系统开发中的基本单元,原子构件有其对应的实现体,复合构件在规约层次上表达了成员构件之间的复合,本身并不对应于任何实现体。

下面给出嵌入式软构件模型 ESCM 的形式化定义:

嵌入式软构件 ::= (<嵌入式软构件复合信息>, <嵌入式软构件描述规约>, <嵌入式软构件组装规约>, <嵌入式软构件测试规约>)

其中,嵌入式软构件复合信息是为了标志构件是原子构件还是复合构件;嵌入式软构件描述规约是为了在构件库中能准确、快速地找到所需要的构件;嵌入式软构件组装规约是为构件的组装及组装的自动化提供支持;嵌入式软构件测试规约是为构件集成测试提供依据。

下面将分别对其具体内容及其对后续开发的支持进行详细的阐述。

3.1 嵌入式软构件复合信息

嵌入式软构件复合信息是为了标识构件是原子构件还是复合构件,为检索、组装、测试提供这一必需的信息。我们给出嵌入式软构件复合信息的形式化定义如下:

嵌入式软构件复合信息 ::= (<原子构件结构> | <复合构件结构>)

原子构件结构 ::= (<嵌入式软构件实现的引用>)

复合构件结构 ::= (<构件集合>, <构件间的组合关系>)

其中,构件集合是合成复合构件的构件的集合,可能是原子构件,也可能是复合构件,当然,复合构件还可以再分。构件间的组合关系是指合成复合构件的这些构件是按怎样的关系组合起来的,可以表示为一个组装表达式^[11]:

$$c_1 \theta^1 c_2 \theta^2 c_3 \theta^3 \dots \theta^{n-1} c_n$$

其中, c_i 表示待组装的嵌入式软构件, $i \in (1, n)$; θ^j 表示组装关系, $\theta^j \in \{>>, [,], ||, [,], [>, \cdot, \odot\}$, $j \in (1, n-1)$ 。

3.2 嵌入式软构件描述规约

嵌入式软构件描述规约是对构件信息的描述,是为了在构件库中能准确、快速地找到所需要的构件。这些构件信息包括:

名称:嵌入式软构件的名字。

ID:嵌入式软构件在构件库中的唯一标识号,方便构件的检索。

作者:嵌入式软构件生产者的详细信息。

接口描述:嵌入式软构件的 IDL 定义,为构件的功能匹配查找提供依据。

非功能属性 nfr:对于嵌入软构件来说,除了一些嵌入特性如环境依赖性、特殊硬件设备需求等外,非功能属性对开发成功与否至关重要,这也是它与通用构件之间的最大的区别。在进行构件匹配与检索的时候,非功能属性的匹配检查是非常关键的。

执行情况要求:这是嵌入式软构件实施运行参数,如可执行文件的名称、适用的操作系统、处理器型号、编程语言等等。

复用数:它表明该构件被复用的情况,在某种意义上代表了构件的通用性。

我们用 xml 语言来描述嵌入式软构件描述规约,下面以数据采集构件为例,给出嵌入式软构件的描述规约如下:

```
<? xml version="1.0"?>
```

```
<! DOCTYPE component SYSTEM "component.dtd">
```

```
<component name=" DataCollector " version="1.0">
```

```
<componenttype> Single Component </componenttype>
```

```
<author>
```

```
<name> Xia Yuan </name>
```

```
<company> SWU </company>
```

```
<email add="summerxl@swu.edu.cn"/>
```

```
</author>
```

```
<description/>
```

```
<idl id="IDL:DataCollector;1.0">
```

```
<interface property="request"/>
```

```
<interface property="provide">
```

```
<itf name=" Datacollect ()"/>
```

```
</interface>
```

```
</idl>
```

```
<nfr>
```

```
<time runtime="20ms"/>
```

```
<static_memory s_memory="50k"/>
```

```
<temporary_memory t_memory="200k"/>
```

```
</nfr>
```

```
<implementation id=" DataCollectorImpl">
```

```
<os />
```

```
<processor name="80951"/>
```

```
<compile/>
```

```
<programminglanguage name="c++"/>
```

```
<code filename=" DataCollector.exe"/>
```

```
<runtime/>
```

```
</implementation>
```

```
<reuse>
```

```
<number="5"/>
```

```
</reuse>
```

```
</component>
```

其中,在数据采集构件的 XML 属性描述文档中,用 author 子元素描述了构件作者的详细信息,idl 子元素描述构件的 idl 定义,nfr 子元素描述了构件的非功能属性需求,implementation 子元素描述了构件的执行情况要求,reuse 子元素描述了构件的复用情况。

3.3 嵌入式软构件组装规约

嵌入式软构件的组装规约,提供了构件组装的上下文语义,构件端口所表示出的接口行为,为构件的自动生成及构件的自动化组装和演化提供了理论基础。下面举例说明:

下面的构件 DataUser 是向一个数据采集构件 DataCollector 发送请求,要求在 2 个单位时间内得到返回数据。

```
component DataUser=
  port DataRead: = (request ! x; Δ2result? y{0}; stop ||| DataRead) [] §
  other ports...
```

```
component DataCollector=
  port DataCollect: = (invoke? x; result? y; stop ||| DataCollect) [] §
  other ports...
```

为了描述一个连接件,需要提供其中每一个角色和粘剂的进程描述,连接件也可以看为一个构件。

```
connector U-C-connector=
  role Datauser: = (request ! x; Δ2result? y{0}; stop) ||| Da-
```

```

taRead [] §
  role DataCollector := (invoke? x ; result? y ; stop ||| DataCollector) [] §
  glue = (Datauser.request ! x ; DataCollector.invoke? x ; DataCollector.result? y ; Datauser.Δ2result? y{0}; stop)
  ||| glue [] §

```

3.4 嵌入式软件测试规约

基于构件的嵌入式软件的测试是基于单个嵌入软构件的集成测试,因此在嵌入式软构件模型中加入测试信息,对之后的集成测试至关重要。但是在现有的对嵌入式软构件模型的研究中,并没有对单个构件测试情况的描述,给后续的集成测试造成了很大的难度。在本模型中给出了嵌入式软构件测试规约,它为构件集成测试提供了强有力的支持。

嵌入式软构件测试规约包括测试计划(Test Plan)和测试过程(Test Procedure)。

我们把嵌入式软构件测试规约定义如下:

嵌入式软构件测试规约 ::= (⟨测试人员⟩, ⟨测试平台⟩, ⟨测试技术⟩, ⟨测试策略⟩, ⟨测试上下文环境⟩, ⟨测试过程⟩, ⟨性能测试⟩)

其中:

测试人员可由代号给出,不需写明具体的名字;

测试平台为测试所需的测试工具及环境;

测试技术为测试所采取的基本技术和方法,如域测试、容量测试、线索测试、用户测试、回归测试、基于风险的测试、声明测试、探索性测试等。

测试策略主要是测试采取的方法:黑盒,白盒等。

测试上下文环境是指对嵌入式构件进行测试所依赖的硬件和软件。因为嵌入式软件与硬件的高依赖性,而且我们在构件组装之后进行的又是集成测试,所以测试上下文环境对我们在嵌入式软构件组装之后的系统集成测试至关重要。

测试过程主要是指测试过程中的测试用例,这对之后的系统集成测试也有很大的指导作用。

性能测试是对嵌入式软构件的非功能属性 nfr 的测试。之所以把它单独列出来,是因为嵌入式软构件对非功能属性的高要求性。嵌入式软构件组装之后的系统非功能性的测试也是系统集成测试中最重要的一项。

4 应用实例

楼宇自控系统(BAS),采用先进的计算机控制技术,给人们带来舒适的工作及生活环境。楼宇自控系统不仅囊括了楼宇所有的基础设施,如动力系统、照明系统、给排水系统、中央空调系统、房间调温、调光系统、电梯控制等,还包括功能设施,如火灾报警系统、保安监控系统、背景音乐、停车场管理系统等。

BAS中的火灾报警子系统主要有4个关键构件:数据采集构件、火灾监视构件、用户构件和报警构件。下面以数据采集构件来说明嵌入式软构件模型——ESCM。

DataCollector ::= (⟨DataCollector 复合信息⟩, ⟨DataCollector 描述规约⟩, ⟨DataCollector 组装规约⟩, ⟨DataCollector 测试规约⟩)

DataCollector 复合信息 ::= (⟨原子构件:DataCollector.exe⟩)

DataCollector 描述规约如 3.2 节所示;

DataCollector 组装规约如 3.3 节所示;

DataCollector 测试规约 ::= (⟨测试人员⟩, ⟨测试平台⟩, ⟨测试技术⟩, ⟨测试策略⟩, ⟨测试上下文环境⟩, ⟨测试过程⟩, ⟨性能测试⟩)

测试人员 = “Xia Yuan”;

测试平台 = “Keil uVision” & “80951 调试环境配置”

测试策略 = “白盒/黑盒”

测试上下文环境 = 用测试工具为数据采集构件构造可采集的数据集以及接收数据采集构件数据的上位构件。

测试过程 = (测试用例 1, 测试用例 2, …)

性能测试:

time runtime = “[17 ms, 20ms]”

static_memory s_memory ≤ “50k”

temporary_memory t_memory ≤ “200k”

…

结束语 本文以构造一个对后续开发有良好指导作用的构件模型为旨,构造了一个嵌入软构件模型—ESCM,它可对嵌入式软构件的检索、构件的组装及集成测试提供一定的支持。

本文也有很多不足。此模型的形式化还有待提高;在逐步推广的实践中,模型还需进一步改进和完善。

参考文献

- [1] Kuz I, Liu Y. CAmkES: A component model for secure micro-kernel-based embedded systems. *The Journal of Systems and Software*, 2007, 80: 687-699
- [2] Beydeda S. Self-Metamorphic-Testing Components // *Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC' 06)* 0-7695-2655-1/06. 2006
- [3] Carlson J. SaveCCM: An Analysable Component Model for Real-Time Systems *Electronic Notes in Theoretical Computer Science*, 2006, 160: 127-140
- [4] Polato I. A Component-based Approach to Embedded Software Design *Electronic Notes in Theoretical Computer Science*, 2006, 160: 255-273
- [5] Tseng T-L, Liang W-Y. Applying genetic algorithm for the development of the components-based embedded system. *Computer Standards & Interfaces*, 2005, 27: 621-635
- [6] Arató P, Mann Z Á. Extending component-based design with hardware components. *Science of Computer Programming*, 2005, 56: 23-39
- [7] Bracciali A, Brogi A. A formal approach to component adaptation. *The Journal of Systems and Software*, 2005, 74: 45-54
- [8] Hussein M, Zulkernine M. Intrusion detection aware component-based systems: A specification-based Framework. *The Journal of Systems and Software*, 2007, 80: 700-710
- [9] Salcic Z, Hui D. HiDRA—A reactive multiprocessor architecture for heterogeneous embedded systems. *Microprocessors and Microsystems*, 2006, 30: 72-85
- [10] Carlson J, Akerholm M. An Event Algebra Extension of the Triggering Mechanism in a Component Model for Embedded Systems. *Electronic Notes in Theoretical Computer Science*, 2005, 141: 107-121, 2005
- [11] Zhang Weiqun, Xia Yuan. Research for the ET-LOTOS-Based Embedded Software Component Composition // *ICSCA'2006*
- [12] Douglass B P. 实时 UML—开发嵌入式系统高效对象. 尹浩琼, 等译. 中国电力出版社, 2003
- [13] 王志刚, 王民北, 骆雷飞. 一个嵌入式软件构件的 NFA 量化度量模型. *计算机工程*, 2006(7)
- [14] 夏苑, 张为群. 一种基于 CBSE 的嵌入式实时软件建模方法研究. *计算机科学*, 2005(8)
- [15] 曹严元, 张为群. 一种基于 CBD 的软件测试方法. *计算机科学*, 2005, 32(2)