

SCA 波形组件的可视化装配与部署

洪锡军, 陈小凤, 张 激

(华东计算技术研究所, 上海 200233)

摘要: 针对软件通信结构(SCA)规范中 XML 配置文件在可视化表示波形组件描述和波形应用装配与部署方面的不足, 提出 SCA 波形组件的可视化装配与部署, 为 SCA 兼容系统的架构设计者和系统开发者提供了一种图形化的装配逻辑和部署关系表示方式, 减少了系统设计和开发过程中可能出现的错误。并对 SCA 规范中的组件间端口连接方式进行了归纳和总结, 给出在不同应用场景下选择端口连接方式的原则, 介绍了一种基于 Eclipse 体系框架的 SCA 波形集成开发环境。

关键词: 软件通信结构; 波形; 可视化装配与部署

Visualized Assembly and Deployment of SCA Waveform Component

HONG Xi-jun, CHEN Xiao-feng, ZHANG Ji

(East China Institute of Computer Technology, Shanghai 200233)

【Abstract】 For making up the deficiency of XML profiles of SCA specification in visually describing waveform component and application assembly or deployment, a concept of visualized assembly and deployment of SCA waveform component are proposed. By graphically represent assembly logic and deployment relationships of waveform components, the architects and developers can communicate design clearly and unambiguously. So some potential errors can be avoided during the process of SCA compatible system design and development. Meanwhile, connections between components are also concluded and summarized. The principles of how to choose proper connection type according to different application scenarios are depicted too. At last an IDE for SCA waveform development based on eclipse architecture is introduced.

【Key words】 Software Communication Architecture(SCA); waveform; visualized assembly and deployment

SCA规范通过定义核心框架接口和应用程序环境描述(AEP), 并利用CORBA中间件标准化组件间的通信, 使得上层波形应用具有很好的可移植性和可复用性, 从而大大减少了无线通信系统的开发成本。但是, 为了实现该目标, 要求SCA兼容系统的架构设计者和开发者之间必须以非常清晰的方式进行设计交流。但SCA规范本身并没有提供设计的可视化表现形式, 而是以一套语义复杂的XML描述文件在设计和开发间进行交互, 难以将整个系统的需求分析和设计表达清楚。尤其是在系统相对比较庞大时, 组件间的连接非常多, 连接关系非常错综复杂, 且连接本身的语义也不易理解, 如果用XML描述文件来进行系统设计和开发间的交流, 将会导致后期开发过程中的大量错误^[1-2]。所以, 对SCA兼容系统的设计进行可视化描述, 按照人类最易理解的图形方式建立系统模型, 构建有效的架构设计者与系统开发者之间的交互机制, 进行SCA波形组件的可视化装配与部署是SCA发展过程中必须解决的一个重要技术。为此, 本文基于SCA规范中的端口概念, 对SCA兼容系统中的组件间连接和体系结构各层间的关系进行可视化描述, 在基础平台架构设计者和波形应用开发者之间建立对整个SCA兼容系统顶层设计的可视化交互机制, 以缩短系统的开发周期, 降低系统的开发成本, 提高系统的质量。

1 基本原理

为了对SCA波形组件的可视化装配与部署进行系统设计和建模, 必须对该模型中的相关概念进行抽象, 以形成最小的可视化装配与部署基本单元。本文在遵守SCA规范中相关规则的前提下, 根据可视化装配与部署的需要, 抽象出了以

下一些基本概念^[3-4]:

(1) 组件

在 SCA 规范中, 组件主要指实现了一系列系统功能/波形功能的软件模块或单元。组件对外暴露的是接口和端口, 而封装的则是具体的功能实现。一般来说, 在 SCA 规范中, 组件主要指实现了 Resource 接口的波形组件, 虽然实现了 Device 接口的逻辑设备也是一种组件, 但在 SCA 规范中还是将其称之为逻辑设备。

在进行 SCA 兼容系统开发时, 波形应用是由一系列实现了相关通信功能的组件组成的, 这些组件之间主要通过接口和端口进行通信。

本文抽象 SCA 规范中的组件作为波形可视化装配与部署的基本单元, 将之作为一个独立的模型实体进行操作, 并以矩形块表示。

(2) 设备

从本质上讲, SCA 规范中的逻辑设备也是一种组件, 但它是以软件的形式描述实际的硬件设备, 是对 SCA 兼容系统中硬件平台的抽象。本文为了在可视化装配与部署模型中表示与波形组件的区别, 更好地体现部署的概念, 将 SCA 规范中的逻辑设备也抽象成了一个独立的模型实体。

在 SCA 波形装配与部署时, 波形组件和逻辑设备间的关系主要有两种: 一种是“使用”关系, 即波形组件使用逻辑设备的功能/能力; 另一种是“加载”关系, 即逻辑设备加载

作者简介: 洪锡军(1976 -), 男, 高级工程师, 主研方向: 系统软件, 嵌入式应用和软件无线电; 陈小凤, 工程师; 张 激, 研究员

收稿日期: 2007-01-08 **E-mail:** allenh@online.sh.cn

波形组件。前者体现装配的概念，后者体现的是部署关系。

(3) 端口

在 SCA 规范中，参与组件间连接的对象类型很多，主要有：实现了组件特定接口的 port、组件对外的接口以及设备上的 port 等，为了在可视化装配与部署模型中进行统一处理，本文将其统一抽象为端口对象，并与组件进行分离，用正方形实心方块表示。但从本质上讲，接口和端口是两个不同的概念，其中，接口主要指组件对外公布的一系列操作 (operations) 和属性 (attributes)，用户或其他组件可以通过接口对组件的功能进行调用。接口是实现 SCA 波形组件可重用性和可移植性的重要技术之一，它将波形功能的具体实现进行了隐藏，并使得波形组件不能感知接口调用方是谁，从而提供了对波形组件功能调用的封装。端口主要指组件对外提供的接口或组件需要使用的接口。与接口相比，端口是有方向性的，它包含了两个方面的含义，一个是指该组件能为其他组件提供的接口；另一个是指该组件需要使用的其他组件接口。利用端口的概念，使得组件可以不直接和组件进行通信，而是通过端口间的“使用者”和“提供者”关系来完成，从而将组件间的访问进行了解耦合，提高了组件的可重用性。

(4) 连接

连接是对组件间通过端口进行通信的一种抽象。任何端口需要和其他端口进行通信时，参与通信的两个端口间必须预先建立连接，且该连接具有方向性，必须明确地指出端口间的“使用者”和“提供者”关系。为此，本文在可视化装配与部署模型中将连接也抽象成了一个独立的实体单元，用箭头连线表示，其中，箭头起始端表示“使用者”端口，箭头末端表示“提供者”端口。在本文所示的 SCA 波形组件可视化装配与部署模型中，连接是属于一个波形应用装配的，而端口则是属于组件的。在进行波形组件装配时，组件间连接的两端必须是匹配的合法端口。

2 装配连接处理

根据 SCA 规范，无论是波形应用的装配或者硬件平台结点的配置，都可以指定多种类型的端口连接方式，虽然在一定程度上增加了系统开发者的灵活性，但同时也对系统开发者提出了较高的要求，因为系统开发者只有在非常清楚所有端口连接方式的语义时，才能根据实际应用的需要选择合适的端口连接方式。显然，当系统比较复杂、组件间端口连接比较多时，如果仍然以 XML 配置文件的形式进行系统设计和开发间的交流，势必会增加系统调试和测试的困难，容易导致后期开发过程中的错误^[4]。因此，必须对波形应用的装配或者硬件平台结点的配置进行可视化表示，以图形化的方式建立端口连接，进行组件装配和部署。

为了更好地建立 SCA 波形组件可视化装配与部署模型，本文对 SCA 规范中的连接方式进行了梳理，并在结合应用需求的前提下进行了归纳和总结。虽然 SCA 规范中，参与连接的两个端口间具有明确的“使用者”和“提供者”关系，分别用“usesport”和“providesport”表示，但在本文所建立的可视化装配与部署模型中，“usesport”和“providesport”从端口的概念上来说是同等的、一致的。所以，本文在进行端口连接方式归纳和总结时，只对“providesport”进行了举例说明，但这些原则同样适用于“usesport”。

(1) 波形应用内部组件间的端口连接

由于核心框架在创建波形应用实例时，已经非常清楚各

个组件实例，因此此时应选择 componentinstantiationref 的形式，不要选择 namingservice 的形式。因为后者最终也是通过组件实例的 DCE UUID 来定位组件的，显然，没必要通过名字服务绕一圈，直接从核心框架中取即可。

```
// 应该采用的方式
<providesport>
<providesidentifier>PortName</providesidentifier>
<componentinstantiationref
refid="DCE:482c18a0-62a7-4885-9e8b-37c3b49ae204" />
</providesport>
// 不应该采用的方式
<providesport>
<providesidentifier>PortName</providesidentifier>
</findby>
<namingservice
name="IO_DCE:05ee0091-05bd-4afe-b9d9-13de78e39b92" />
</findby>
</providesport>
```

(2) 波形应用内部组件使用外部通用 CORBA 组件时的端口连接

虽然 SCA 规范详细规定了波形应用组件向名字服务注册的方式，但并没有规定波形应用之外的组件如何向名字服务注册。如果在实际应用中，波形应用内部的组件需要用到波形应用之外的通用 CORBA 组件，则该外部的通用 CORBA 组件必须将其自身绑定到名字服务中。在这种情况下，最直接的端口连接方式是通过 findby。即：

```
<providesport>
<providesidentifier>PortName</providesidentifier>
</findby>
<namingservice name="HCI" />
</findby>
</providesport>
```

不过值得注意的是，此时 namingservice 的 name 应该是相对于名字服务根的路径，如果有多级路径，则各级路径间应以“/”分隔。

(3) 波形应用组件与逻辑设备间的端口连接

根据 SCA 规范，波形应用组件与逻辑设备间有两种访问关系：一种是逻辑设备加载波形应用组件；另一种是波形应用组件使用逻辑设备（此时一般指使用逻辑设备的某种 allocation 属性）。对于第 1 种关系，合理的连接方式为 devicethatloadthiscomponentref；对于第 2 种关系，则应采用 deviceusedbythiscomponentref 的形式。即：

```
// 第 1 种关系时应采用的方式
<providesport>
<providesidentifier>PortName</providesidentifier>
<devicethatloadedthiscomponentref
refid="DCE:5c347fd8-8ad6-4ba6-bb60-1e4ce5fbbe9a" />
</providesport>
<providesport>
<providesidentifier>PortName</providesidentifier>
<deviceusedbythiscomponentref
refid="DCE:1e37517d-7444-4ce8-b785-66f1fe4661d9"
usesrefid="ModemBandwidth" />
</providesport>
```

(4) 与标准服务间的端口连接

SCA 规范中规定的标准服务有 log、eventchannel、

filemanager 和 namingservice。如果波形应用组件需要使用这些标准的服务，应以 domainfinder 的形式进行连接。即：

```
<findby>
<domainfinder type="eventchannel" name="ChannelName" />
</findby>
```

对于上面示例的 findby，当 domainfinder 中没有 name 属性时，任何可获取的事件通道都可以作为参与端口连接的一方；如果指定了 name 属性，则应该获取名字为 "ChannelName" 的事件通道。该原则同样适用于其他的 service。

以上对 SCA 规范中的端口连接方式进行了归纳和总结，并给出了一般的连接原则，该原则是本文中建立可视化装配与部署模型的一个基础。

3 可视化装配与部署

在对 SCA 规范进行了深入理解的基础上，笔者结合实际的 SCA 兼容系统开发需求，基于 Eclipse 框架开发了一个 SCA 波形集成开发环境，并以插件的形式实现了 SCA 波形组件的可视化装配与部署功能。图 1 是一个简单的波形可视化装配与部署示例。

在图 1 中，矩形块表示的是组件；正方形实心方块表示的是组件所发布的端口；箭头连线表示的是连接。根据该图，显然 WF4 波形由 4 个波形组件装配而成，各个组件所发布的端口也一目了然，且组件间的连接关系十分清晰。由此可见，利用波形的可视化装配与部署工具可以在设计者和开发者之间进行最直观的信息交互，能大大减少开发调试过程中由于 XML 域配置文件编写不正确所导致的错误。

在图 1 中，通过双击组件对象(即矩形块)，可以显示相应的组件描述信息和组件部署信息；通过双击连接对象(即正方形实心方块)，可以显示组件间的端口连接信息。



图 1 波形可视化装配与部署示例

4 结束语

SCA 波形组件的可视化装配与部署是针对 SCA 波形开发提出来的，其实，该概念同样可以推广到 SCA 兼容系统的开发过程中。在该过程中，所有用 XML 域配置文件进行设计与开发交流的过程都可以利用本文所介绍的“SCA 波形集成开发环境”进行可视化表示。

参考文献

- [1] JTRS JPO. Software Communication Architecture Specification[Z]. 2004-08.
- [2] Joint Tactical Radio System[Z]. (2006-07-06). <http://jtrs.spawar.navy.mil/sca>.
- [3] SDR-Forum Software Defined Radio Forum[Z]. (2006-06-09). <http://www.sdrforum.org>.
- [4] SCARI Software Suite SCA_CF[Z]. (2006-09-06). <http://www.crc.ca/rars>.

(上接第 282 页)

于保存安全隔离卡的安装信息和设置运行时的参数；扩展 ROM 模块主要用于存储隔离卡的控制程序，该控制程序根据用户的需要切换网络终端的工作状态；控制逻辑在控制程序的控制下向切换接口发布通信链路切换命令，由切换接口完成内网与外网、安全区与公共区物理链路的切换，其工作原理参见图 4。

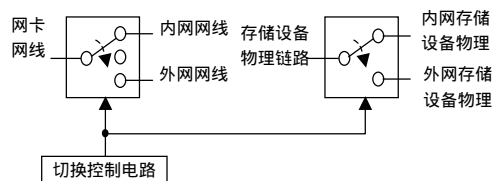


图 4 切换控制示意图

网络接口包括内网网线接口、外网网线接口、网卡网线接口，通过网线分别连接内网交换机、外网交换机和网络终端计算机的网卡。存储设备切换接口包括 3 个 IDE 总线接口(根据需要也可配置不同总线的存储设备接口)，通过 IDE 总线分别连接内网存储设备、外网存储设备和网络终端计算机的主板。切换控制电路在控制程序的控制下完成网络与存储介质的同步切换与锁定。

安装有安全隔离卡的计算机在开机或重启时，启动隔离卡上的控制程序完成 3 项任务：在操作系统启动前获得计算机的控制权；提供用户与安全隔离卡的交互界面；根据用户

的需要驱动切换控制电路完成网络终端工作状态的切换与锁定，之后将控制权交给该状态的操作系统，开始正常的系统使用。由于隔离卡工作在系统的底层，隔离卡控制程序工作于扩展 BIOS 方式，且状态切换后立即被锁定，因此在网络终端正常使用过程中无法再对终端状态进行切换，可有效防止黑客和木马等对切换操作的重置，保证了隔离的安全性。只有当重新启动或开机时，锁定状态才被解除。

3 结束语

物理隔离是目前解决双机工作模式的有效方案，不仅可以降低成本，提高效率，还可以大大提升终端计算机的安全性；另外物理隔离也是一种行之有效的安全防御手段，完全能实现一机双网的彻底分离，可有效阻止来自于外部网络的攻击。经国内多家重要部门安装使用，效果较好。随着物理隔离技术的日趋完善，它将逐渐成为网络信息安全研究的重要组成部分，必然会在我国信息安全领域发挥重要作用。

参考文献

- [1] Shanley T, Anderson D. PCI System Architecture[M]. 4th ed. [S. l.]: Mindshare, 2003.
- [2] PLX Technology Inc. PCI 9052 Data Book Version 2.0[EB/OL]. (2001-09-09). <http://www.plxtech.com>.
- [3] Altera Corporation. MAX 7000 Programmable Logic Device Family Data Sheet Version 6.6[EB/OL]. (2003-06-06). <http://www.Altera.com>.

