第 36 卷 第 6 期 计 算 机 工 程 2010 年 3 月
Vol.36 No.6 Computer Engineering March 2010
·开发研究与设计技术· 文章编号：1000—3428(2010)06—0256—03 文献标识码：A 中图分类号：TP311

基于 3G 网络的绑定式智能卡系统模型

周允强[1]，李代平[1]，刘志武[1]，黄 健[2]，梅小虎[1]，郭鸿志[1]
(1. 广东工业大学计算机学院，广州 510006；2. 南方智能卡公司，广州 510090)

摘 要：在分析 3G 网络下单晶片智能卡芯片操作系统(COS)的结构及关键技术基础上，提出绑定式多晶片智能卡 COS 的覆盖模型，对模型各功能模块结构及构造流程进行研究。通过对模型的裁剪，抽象出符合用户需求的绑定式单晶片智能卡 COS 模型，并对模型进行可行性分析及评估。

关键词：绑定式；单晶片操作系统；多晶片操作系统；覆盖模型

Model of Binding Smart Card System Based on 3G Network

ZHOU Yun-qiang[1], LI Dai-ping[1], LIU Zhi-wu[1], HUANG Jian[2], MEI Xiao-hu[1], GUO Hong-zhi[1]
(1. Faculty of Computer, Guangdong University of Technology, Guangzhou 510006; 2. South Smart Card Company, Guangzhou 510090)

【Abstract】 Based on the analysis of the structure and the key technologies of smart card Mini_COS on 3G network, a cover model of smart card Bind_Max_COS is proposed, the structure and the construction process of this model are studies. A smart card Bind_Mini_COS which suits for the customs taste is made. And a feasibility analysis and evaluation for this model are described.

【Key words】 binding; Mini_COS; Bind_Max_COS; cover model

1 概述

随着 3G 网络的发展以及微电子技术的进步，智能卡硬件资源越来越丰富，使开发一套适应 3G 网络，能在智能卡中实现的芯片操作系统(Chip Operating System, COS)成为可能。同时，为满足 3G 用户存储大容量信息的需求，高端智能卡芯片也在频繁更换，使兼容各大厂家芯片 COS 的研发成为智能卡技术的发展趋势。

2 单晶片操作系统技术分析

智能卡由硬件资源(智能卡芯片)与 COS 组成，COS 是智能卡的核心。而针对某一种特定芯片开发的 COS，简称为单晶片操作系统(Mini_COS)。

2.1 3G 网络 UICC 平台

通用集成电路卡(Universe Integrated Circuit Card, UICC)是用在 3G 网络系统移动终端的智能卡物理载体[1]。同时，智能卡应用功能的实现在3G网络下都需要UICC平台的物理支撑。UICC 内部集成电路一般由多个硬件构成，但是每间公司的芯片设计和市场定位不一，导致各厂家的 UICC 内部结构不一定相同，出现了一套 COS 只能适合一种特定芯片的瓶颈问题。

2.2 Mini_COS 层次调用技术分析

ISO7816 系列规范对智能卡的物理电气特性、文件系统结构、通信协议进行了规定[2]。传统智能卡的 COS 离不开四大功能模块与硬件底层的设计和开发[3]。Mini_COS 的层次调用模型如图 1 所示。

Mini_COS 层次模型从整体上分为功能模块层和微内核层。功能模块层主要实现 COS 的应用逻辑处理功能，并调用微内核层中的底层驱动模块实现对硬件的操作，该层主要包含通信管理模块、安全管理模块、命令处理模块及文件管理模块。微内核层主要对功能层的逻辑处理提供硬件支持，并直接实现对 UICC 硬件的具体操作，如 flash, DES, RNG, TIMER 等硬件的读写程序[4]。

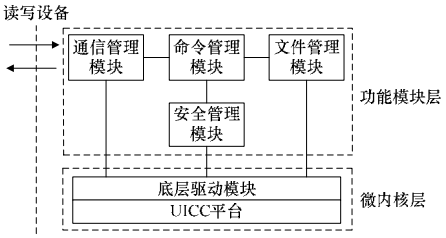

图 1 Mini_COS 层次调用模型

从图 1 模型的层次调用关系来看，在读写设备采用应用协议数据单元(Application Protocol Data Unit, APDU)[2]直接与功能模块层进行通信后，APDU 命令使数据在智能卡层与层之间发生调用关系。与传统 COS 调用方式相比，Mini_COS 具有更高的效率，主要体现在对安全管理模块的设置上。在 Mini_COS 系统中并没有对所有文件系统中存储和读取数据进行安全管理，例如与网络鉴权中，连续访问同一文件时，不需要重复进行安全处理，而是根据命令的类别需要来进行适当的处理，比如部分文件数据的加密等。而传统 COS 中所有与外界通信的数据都需经过安全处理[5]。

2.3 Mini_COS 存在的问题

虽然 Mini_COS 比传统 COS 在数据传输效率上有明显的

基金项目：广州市越秀区自然科学基金资助项目“绑定式近场通信 3G COS 研制”(2008-GX-015)

作者简介：周允强(1983－)，男，硕士，主研方向：3G 智能卡，并行计算；李代平，教授、博士；刘志武，硕士；黄 健，高级工程师、硕士；梅小虎、郭鸿志，硕士

收稿日期：2009-11-20 **E-mail：**begin8329849@126.com

改善，但其针对某一种特定芯片底层来开发，产生如下问题：

(1)在 COS 支持的上层应用不变的情况下，当更换不同的 UICC 硬件时，需要重新了解新硬件 COS 的开发环境及底层技术细节，移植工作量非常巨大，不低于重新编写一次 COS。

(2)使用自然语言开发的 COS，大多采用层次结构，开发效率较低，编写的代码量较大，增加了硬件的效率成本和存储成本。

(3)不同厂家开发各自芯片时，需要研发适合自身芯片的操作系统和数据服务，造成操作系统、同类指令处理逻辑的重复开发及利用[6]。

为了增强 Mini_COS 上层逻辑在不同芯片上的适应性，减小上层逻辑在不同 UICC 上移植的难度，采用模型改进的策略提高 COS 开发效率。

3 绑定式多晶片操作系统模型

针对目前大多数芯片的 COS 和多种 UICC 硬件的不同结构，提出绑定式多晶片 COS 模型，简称多晶片操作系统(Bind_Max_COS)模型。

3.1 Bind_Max_COS 模型技术问题分析

Bind_Max_COS 模型是一个覆盖模型，同时也是一个抽象模型，它并不是一个可以单独编译运行的系统，简单地说只是一种组件的管理概念。因此，直接掩模到具体 UICC 芯片上的是在 Bind_Max_COS 基础上进行建立、裁剪、抽象出来的符合用户需求的 Mini_COS，简称 Bind_Mini_COS。从 Bind_Max_COS 演化为 Bind_Mini_COS 的核心技术问题如下：

(1)需要对不同底层硬件 UICC 进行分析，抽取其中相同或兼容硬件驱动部分，记录到驱动库中。

(2)建立 COS 适配器，针对特定芯片不同的硬件需求将覆盖模型裁剪编译成特定的 COS 掩模灌装到智能卡芯片中。

(3)如何为不同的 UICC 底层硬件驱动在覆盖模型中建立正确的地址映射。

针对上述问题，下文给出一个绑定式多晶片 COS 模型。

3.2 Bind_Max_COS 模型整体结构

模型的设计原则遵从 IS07816 相关规范，以便提高不同芯片的兼容性。模型结构如图 2 所示。

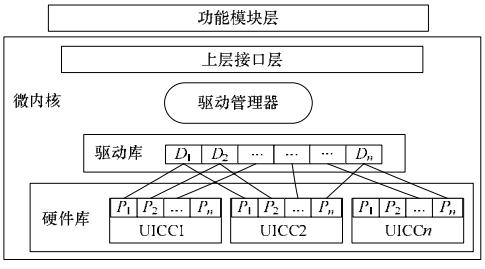

图 2 Bind_Max_COS 模型结构

硬件库表示多个芯片 UICC 硬件驱动程序的集合，不同的 UICC 可以由不同的硬件属性 P_i 构成，属性 P_i 有 FLASH, DES, RNG, I/O, CPU 等硬件电路模块，同时每一种属性只能与驱动库中的一种驱动 D_i 相对应。另外，硬件库采用设备管理表(Driver Manage Table, DMT)对 UICC 属性进行检索管理。

驱动库表示硬件库中所有硬件属性 P_i 对应的驱动 D_i 的集合，一般硬件属性 P_i 的总量上限大于或等于驱动 D_i 的总量上限，并且 D_i 与 P_i 的对应关系为一对一或一对多。驱动库是从硬件库中抽象出来的，不同的 UICC 对 FLASH 的擦除模式、DES 的运算模式等都可以具有通用性，也就是说一个 D_i 属性至少可以同时兼容 2 种以上不同芯片的 P_i 属性。

驱动管理器是微内核设计的核心部分之一，主要实现对下层驱动库程序的管理，并且为上层接口层提供正确的驱动程序映射。管理器通过设置驱动控制表(Driver Control Table, DCT)实现对底层驱动的管理。DCT 中每一项可以代表一种具体硬件属性的驱动，但实际上存储的是硬件属性对应的在驱动库中放置的驱动映射地址，DCT 只是一种管理机制。

上层接口层表示微内核与上层功能模块通信的接口。该层设计的好坏直接影响到上层逻辑应用到不同的 UICC 时，对上层代码修改工作量的大小。因此，进行 COS 设计时必须考虑底层接口对上层应用逻辑的通用性，尽量使用不同 UICC 均支持的函数接口。

本模型将 UICC 硬件底层的设计与上层功能模块的设计进行分离，使得底层硬件驱动的动态部署成为可能。模型中的微内核层由四大部件构成，其中硬件库集成了不同 UICC 属性的驱动，通过抽象的对比与筛选，把不同芯片的等价驱动采用驱动控制表(DCT)记录入驱动库中，大大减少了驱动程序代码量。而且在上层功能模块不变的情况下，相对 Mini_COS 模型，Bind_Max_COS 模型解决了更换硬件时不需重新移植的问题，实现了重新选择相应硬件的驱动组件编译构成新 COS 的功能，做到了“一次开发，到处运行”。另外，采用了模型改进的策略，提高了 COS 的运行效率等。

3.3 DCT 和 DMT 表建立工作流程

DCT 和 DMT 表分别表示对硬件库和驱动库中硬件属性驱动程序在库中放置的映射地址的记录，实质是一种管理机制，在库中驱动代码散列放置，采用上述管理机制来实现对驱动代码的检索和管理。工作流程如图 3 所示。

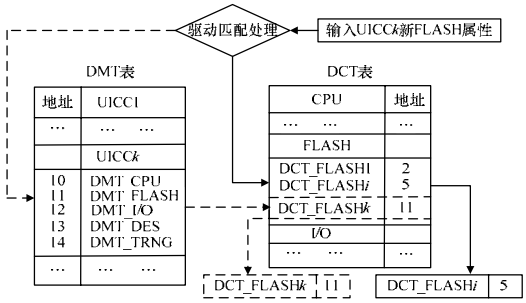

图 3 DCT 和 DMT 表工作流程

DMT 表记录了某个 UICC*k* 硬件中的 CPU, FLASH, I/O, DES, TRNG 等模块信息，以地址值标识该模块驱动程序在内存中的存储位置。DCT 表针对 DMT 表的所有 UICC 硬件中某个同类模块的信息进行记录，例如，DCT 中的 FLASH 表示目前 DMT 表中所有 UICC 具有 FLASH 存储模块这一共性，其中包括 DCT_FLASH1, DCT_FLASH*i*, DCT_FLASH*k* 等不同特征的子属性，也就是说，不同 UICC 的 FLASH 控制方式不一样，比如 FLASH 容量大小不一、擦写模式不统一等，这些都可能导致不同 UICC 具有不同的驱动程序等。从理论上来说，FLASH 所有属性数量不应大于 DMT 表中 UICC 的总数量，而且每一个属性与 DMT 表中同类属性存在一对一或一对多的关系，也就是说 DCT_FLASH*i* 可以适合 UICC1 中的 DMT_FLASH，或者同时适合 UICC1 和 UICC*k* 中的 DMT_FLASH。DCT 表中的其余硬件模块类似。

假设要往两表添加 UICC*k* 的新 FLASH*k* 属性，首先通过驱动管理器的驱动匹配处理，从 DCT 表 FLASH 模块中寻找是否存在与新 FLASH*k* 硬件模块相匹配的子属性，若有匹配

的子属性 DCT_FLASH*i*，就采用当前 DCT_FLASH*i* 作为新 FLASH 的驱动。若不存在，则往 DMT 表相应位置添加新 FLASH*k* 属性，然后把新 FLASH*k* 属性存储信息添加到 DCT 表的 FLASH 模块中，并且采用属性 DCT_FLASH*k* 作为新 FLASH 的驱动。

4 模型可行性分析及评估

4.1 可行性分析

Bind_Max_COS 模型的实现是以 Mini_COS 模型为基础，通过对各厂家的芯片进行分析，把芯片的不同硬件属性及驱动程序记录入库，根据用户的需求，对 Bind_Max_COS 进行裁剪，抽象到符合用户需求的 Bind_Mini_COS。其生成模型如图 4 所示。

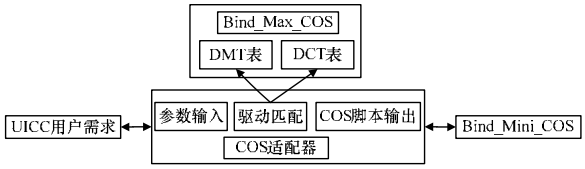

图 4 Bind_Mini_COS 生成模型

Bind_Max_COS 模型只是一个覆盖模型，不能单独编译生成 COS 脚本并掩模到智能卡芯片中，而掩模到芯片中的只能是 Mini_COS 或 Bind_Mini_COS。因此，Bind_Max_COS 中的硬件适配部分可以采用软件形式在 PC 机上实现，用户通过选择界面选择适合的硬件设备及型号，并通过 COS 适配器生成具有绑定用户需求功能的 COS 脚本，利用读卡器把 COS 脚本直接灌装到智能卡芯片中，在芯片中运行的系统就是 Bind_Mini_COS。另外，Bind_Max_COS 可以不被智能卡的代码存储空间和运行时间效率因素所限制，硬件各模块不同类型的驱动程序可以存储在 PC 机上，根据用户的需求来调用。

基于本绑定式智能卡系统模型的实现，通过对芯片进行硬件底层技术分析，根据不同用户需求，对模型中的驱动库和硬件库进行动态部署，实现了将一套智能卡 COS 应用到不同芯片上，证明了方案的可行性。

4.2 模型评估

Mini_COS 模型主要针对的是上层应用逻辑设计和某个具体芯片的底层设计，主要解决上层逻辑应用问题，相比传统的 COS，该模型在安全鉴别及文件数据处理上有较高的效率。Bind_Max_COS 模型针对的是对多个芯片底层的设计，主要解决在多个芯片上的驱动共享和移植问题，相比 Mini_COS，该模型能够支持更多的芯片，提高了 COS 的适应性。Bind_Mini_COS 模型是在 Bind_Max_COS 模型基础上，根据用户的需求，进行裁剪、抽象而成，除了继承 Mini_COS 和 Bind_Max_COS 两大模型的优点外，具有很好的延展性，方便日后在单张智能卡上实现多应用功能[6-7]。

基于商业理由，大多数厂商对自己的芯片技术都是保密的，在一定程度上阻碍了绑定式智能卡模型的推广。因此，在技术实现共享，对绝大部分厂家芯片技术进行覆盖后，才能真正证明模型的有效性和实用性。

5 结束语

本文针对传统开发的智能卡 COS 不能有效移植到不同芯片上的问题，提出了绑定式多晶片智能卡系统模型，在一定程度上解决了智能卡系统中功能模块与 UICC 硬件底层不相兼容的难题，为日后 COS 的“一次开发，到处运行”奠定了基础。另外，对于如何有效判断已有的驱动能否适合新硬件的问题还没作深入研究，该方向将成为下一步研究的重点。

参考文献

[1] 3rd Generation Partnership Project. 3GPP TS 31.101-2002 UICC-terminal Interface, Physical and Logical Characteristics Version 6.1.0[S]. 2002.

[2] International Standard Organization. Information Technology——Identication Card-integard Circuit(s) Cards with Contacts(Part 4) Inter-industry Commands for Interchange[S]. 2005.

[3] 李 翔. 智能卡研发技术与工程实践[M]. 北京: 人民邮电出版社, 2003.

[4] 郭向荣. THC20F17A-D 接触式智能卡芯片用户手册[Z]. 北京同方微电子有限公司, 2007.

[5] 王卓人. IC 卡的技术与应用[M]. 北京: 电子工业出版社, 1999.

[6] 董 威, 杨义先. 一种跨行业多应用智能卡系统模型及实现[J]. 计算机工程, 2007, 33(8): 230-232.

[7] 徐中华, 刘玉珍. 一种新的“一卡多用”智能卡模型[J]. 计算机工程, 2003, 29(5): 43-45.

编辑 顾逸斐

(上接第 255 页)

5 结束语

本文对合作网络局域世界的演化特性进行了研究，提出了一种合作网络局域世界演化模型(CoLW)。该模型根据合作网络的实际特性，在演化的过程中，既考虑择优选择的局域性，又考虑项目度对网络的影响。CoLW 模型可以生成具有幂律度分布的网络，将实验模拟与实证数据进行了对比分析，结果表明，具有相同网络规模的计算机产生的数据与实证数据具有相同的度分布，可见 CoLW 模型可以较好地刻画真实合作网络的拓扑结构与统计特性。

本文研究了 CoLW 模型的平均路径长度和聚集系数，结果表明，合作网络规模对平均路径长度影响较小，对平均聚集系数基本没有影响，而项目的规模对两者有较大的影响，可见 CoLW 模型具有保持平均聚集系数稳定的能力。

参考文献

[1] ErdÖs P, Rényi A. On the Evolution of Random Grphs[J]. Mathematical Institute of the Hungarian Academy of Science Publications, 1960, 5(1): 17-60.

[2] Watts D J, Strogatz S H. Collective Dynamics of “Small World” Networks[J]. Nature, 1998, 393(6684): 440-442.

[3] Barabási A L, Albert R. Emergence of Scaling in Random Networks[J]. Science, 1999, 286(5439): 509-512.

[4] 张培培, 何 阅, 周 涛, 等. 一个描述合作网络顶点度分布的模型[J]. 物理学报, 2006, 55(1): 60-67.

[5] Ramasco J J, Dorogavtsev S N, Pastor-Satorras R. Self-organization of Collaboration Networks[J]. Phys, Rev. E, 2004, 70(3).

[6] Li Xiang, Chen Guanrong. A Local-world Evolving Network Model[J]. Physics A, 2003, 328(1/2): 274-286.

[7] 袁韶谦, 赵 海, 李 超等. 一种具有指数截断和局部集聚特性的网络模型[J]. 物理学报, 2008, 57(8): 4805-4811.

编辑 索书志