申请同济大学工学硕士学位论文

基于“和欣”嵌入式操作系统的DTV软件开发平台的研究

（国家863“软件重大专项”项目　编号：　2001AA113400）

培养单位：电子与信息工程学院

一级学科：计算机科学与技术

二级学科：计算机应用

研　究　生：杜亮

指导教师：顾伟楠　教授

二〇〇五年二月

A dissertation submitted to
Tongji University in conformity with the requirements for
the degree of Master of Philosophy

School/Department:
Discipline:
Major:
Candidate: Du Liang
Supervisor: Prof. Wei-Nan Gu

February, 2005

基于『和欣』嵌入式操作系统的ＤＴＶ软件开发平台的研究

杜亮　同济大学

学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：

年　　月　　日

经指导教师同意，本学位论文属于保密，在　　年解密后适用本授权书。

指导教师签名：　　学位论文作者签名：

年　　月　　日　　年　　月　　日

同济大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

签名：

年　　月　　日

摘要

数字电视(DTV)是继黑白电视和彩色电视之后的第三代产品，由于采用了数字信号，因此使得计算机技术的发展和介入成为了可能，在这样的背景下,嵌入式操作系统将逐渐成为数字电视软件开发的标准平台。同时由于很多国外厂商普遍采用操作系统与芯片（CPU）捆绑的方式，向我国提供所谓的芯片和操作系统一体化的解决方案。这样就会产生一旦底层平台发生改变，上层的应用程序就必须完全重写的局面，使得基于数字电视的软件开发效率十分底下，严重的阻碍了数字电视的普及。因此，在数字电视系统中，如何解决异构平台和操作系统对上层应用程序支持的问题,是提高数字电视软件开发效率的关键。“和欣”嵌入式操作系统是完全面向构件的操作系统，操作系统提供的功能模块全部基于构件技术，都是可拆卸的构件，因此基于“和欣”很容易开发并部署系统级的构件，以此来扩充操作系统的功能。本文提出了一种基于“和欣”嵌入式操作系统来提供数字电视软件开发平台中间件的解决方案。它使得多媒体应用能够不经任何修改，甚至不必重新编译、链接就可以运行在不同硬件厂商提供的数字电视平台上,实现了 DTV 应用程序的跨平台，从而可以做到“一次编写,多处运行”，大大提高了数字电视软件的开发效率。

关键词：数字电视，构件技术，中间件

第 1 章 引言 1
1.1 数字电视软件开发概述 1
1.1.1 数字电视发展概况 1
1.1.2 数字电视的特点 2
1.2 数字电视与软件中间件 3
1.3 课题来源及研究意义 4
1.4 论文结构 5
第 2 章 “和欣”嵌入式操作系统及其构件技术 6
2.1 “和欣”操作系统概述 6
2.1.1 “和欣”操作系统简介 6
2.1.2 和欣灵活内核简介 6
2.1.3 和欣操作系统提供的功能 7
2.1.4 和欣操作系统的应用软件开发 8
2.1.5 和欣操作系统的优势 9
2.2 和欣构件运行平台 11
2.2.1 和欣构件运行平台简介 11
2.2.2 和欣构件运行平台的功能 11
2.2.3 和欣构件运行平台的技术优势 13
2.2.4 利用和欣构件运行平台编程 13
2.3 ezCOM构件技术 14
2.3.1 ezCOM技术的由来 14
2.3.2 ezCOM构件技术概要 15
2.3.3 ezCOM技术的意义 16
2.3.4 ezCOM技术对软件工程的作用 17
2.3.5 ezCOM技术在“和欣”技术体系中的作用 18
2.3.6 如何用ezCOM技术编程 19
第 3 章 数字电视的技术本质和数字电视操作系统 20
3.1 数字电视的技术架构 20
3.2 数字电视的技术本质 20
3.2.1 未来数字电视的本质特征 20

3.2.2 数字电视对系统软件的要求 21
3.3 数字电视操作系统 21
3.3.1 数字电视操作系统概述 21
3.3.2 “和欣”嵌入式操作系统及其对数字电视的支持 22
3.4 基于中间件技术的数字电视解决方案 23
3.5 数字电视中间件的特征 24
第 4 章 DTV软件开发平台的系统结构和构成特点 26
4.1 基于API函数库的DTV软件开发方法 26
4.2 DTV软件开发方法平台的体系结构设计 27
4.2.1 gm5221 硬件平台 28
4.2.2“和欣”实时操作系统 29
4.2.3 DTV软件开发平台 30
4.3 DTV软件开发平台的构成特点 30
第 5 章 DTV中间件平台的详细设计与实现 32
5.1 DTV中间件平台的系统结构 32
5.2 DTV基础构件库的详细设计与实现 32
5.2.1 OSD(On Screen Display)构件 33
5.2.2 IO（Input/Output）构件 36
5.2.3 音频解码（Audio）构件 38
5.2.4 视频解码（Video）构件 39
5.3 DTV解释层的详细设计与实现 42
5.3.1 GUI（Graphics User Interface）构件 42
5.3.2 频道调节构件（TUNER）构件 44
5.3.3 画质构件（PICTURE）构件 45
5.3.4 音质构件（TONE）构件 47
5.4 DTV中间件平台中的消息机制 48
5.4.1 DTV中间件平台中的消息定义 48
5.4.2 DTV中间件平台中的消息流分析 49
第 6 章 可视化开发工具——DTV Visual Developer 51
6.1 可视化开发方法简介 51

6.2 可视化开发方法在DTV软件开发中的应用 51
6.3 DTV Visual Developer的体系结构设计 52
6.3.1 DTV Visual Developer的系统构成 53
6.4 基于DTV Visual Developer的数字电视软件开发方法..... 54
第 7 章 结论与展望 55

第 1 章 引言

1.1 数字电视软件开发概述

数字电视是继黑白模拟电视，彩色模拟电视之后的第三代电视。区别于传统电视，它是在拍摄、编辑、制作、传输、播出、接收电视信号的全过程都使用数字技术的电视系统。采用数字技术不仅可以使电视设备获得比模拟电视更高的显示效果，而且还具有模拟技术不能提供的诸如视频点播、数字广播、个性化交互电视、远程教育、Internet、三网合一、电视电子商务和日常信息综合服务等增值服务项目。影视数字化从根本上改变了影视的命运。影视节目的制作和播放，由于数字化方式的加入，而变得更加多元化，随机化，全球化和可追求化。

数字电视的诞生是为了满足人们的视觉和控制的需要。数字电视的信号不再是模拟信号，而是采用以数字形式进行传输、处理、存储的数字信号。由于数字电视信号的存储和处理电路便于大规模和超大规模的集成，因而其设备比模拟电路设备元件少，便于调整，其重量轻、体积小、功耗少，寿命长且可靠性高，容易与计算机以及其它数字化设备接口，适合于公用数据通讯网，便于实现生产、运行的自动化和视听信息处理的综合化、网络化。计算机技术的发展和介入，使得正处于方兴未艾的电视工业得到了新的支持，带来了又一次巨大变革的历史机遇。

1.1.1 数字电视发展概况

现有的彩色电视包括以下几种不同的制式——欧洲和我国采取的是 PAL 制式，美国和日本采取 NTSC 制式，前苏联采取 SECAM 制式，但利用人的视觉暂留原理顺序扫描、同步扫描的原理却是一样的，因而面临的问题也是相同的，即由于扫描行数的限制而造成的清晰度不够理想。为了提高电视图像的分辨率，从 70 年代开始，工业发达国家开始了对高清晰度电视系统的研究工作。这个工作最早是从日本的 NHK 开始的，到了 80 年代初获得成效，制作了 1125 线的数字电视机，60 场/秒，2∶1 隔行扫描标准的高清晰度电视，简称 HDTV。到了 90 年代，形成了日本、欧洲、美国三大数字电视制式。日本和欧洲的两种制

式出现比较早，图像压缩比较小。采用模拟信号传送，卫星播出方式适合较宽的信道传输；美国的全数字方案吸收了日本和欧洲的优点，采用数字压缩编码和数字通信技术，传送效率高，有效地压缩了宽带，适合于窄频道传输的地面广播，并且对使用相同频道的其它节目不产生干扰，实现了与先行模拟信号电视兼容过渡的根本目的。

世界各国发展数字电视的情况：美国国家电视网要在 2006 年普及数字电视，全面停止模拟信号。 英国全国由模拟电视向数字电视过渡的时间从 2006 年开始，预计 2010 年结束；日本 2001 年开播 6 套卫星高清晰度数字电视，但地面高清晰度电视要在 2003 年才会在主要的大城市开播。我国计划在 2005 年将进行数字电视的商业播出，2008 年用数字电视转播奥运会，2015 年停止模拟电视的播放，全面推行数字电视。

1.1.2 数字电视的特点

相对于传统电视，数字电视具有以下技术特点：

(1) 收视效果好，图像清晰度高，音频质量高，可以更好的满足人们感官的需求。

(2) 抗干扰能力强。数字电视不易受处界的干扰，避免了串台、串音、噪声等影响。

(3) 传输效率高。利用有线电视网中的模拟频道可以传送 8—10 套标准清晰度数字电视节目。

(4) 兼容现有模拟电视机。通过在普通电视机前加装数字机顶盒即可收视数字电视节目。

(5) 提供全新的业务。借助双向网络，数字电视不但可以实现用户自点播节目、自由选取网上的各种信息，而且可以提供多种数据增值业务。

（6）很容易实现加密/解密和加扰/解扰技术，便于专业应用(如军用)以及广播应用。

（7）具有可扩展性、可分级性和互操作性，便于在各类通信信道特别是异步转移模式(ATM)的网络中传输，也便于与计算机网络联通。

1.2 数字电视与软件中间件

数字电视的推广应用。促进了电视制作、播出、传输、接收诸环节的技术开发和设备制造的全面发展，形成一个比模拟电视更为庞大的产业网络。这个网络主要是由技术服务商、电视运营商、内容制作商、电视厂家和广大数字电视观众组成。由于数字电视已不再是传统意义上的电视机，而相当于一台32位CPU的电脑，由于数字电视接受的是二进位的数字代码，因此电视节目内容制作商有了更大的空间可以动态地控制这些代码，从而达到制作动态节目的目标，这将是电视史上的一次里程碑。类似的动态服务将比比皆是，动态服务将为我们带来除基本音视频业务之外的数字电视增值业务。由于电视节目的多样化，为内容制作商提供技术支持的技术服务商就应运而生，技术服务商的作用就相当于应用软件开发商，他们利用数字电视的操作系统所提供或支持的中间件，开发出大量的数字电视节目开发软件，为内容制作商提供强大的支持。而电视运营商将会从内容制作商那里采购所需的电视节目，编排以后提供给电视观众。电视厂家为电视观众提供数字电视，以便观众们能享用丰富的电视节目。

以下是数字电视产业网的示意图：

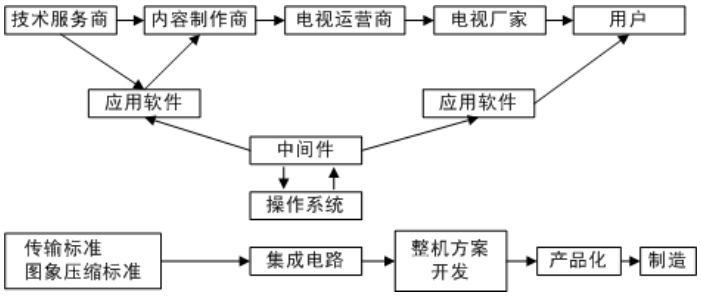


图1.1 数字电视产业网结构图

数字电视服务的最大特点是，它除了支持传统的音频、视频业务以外，还能带来电视增值业务，而这些增值业务大都是建立在基于数字电视的软件产品上的。随着数字电视市场的不断发展，数字电视运营商在寻求一种高效可靠的软件平台，数字电视制造商面对着多种系统实现方案和市场需求，其软件开发占技术总投入的比重日益加大，从而产生了对与平台无关而又能够支持业务扩展的开放系统的需求。数字电视中间件技术就是在这种应用需求下产生的。它定义了一组支持数字电视应用功能的标准程序接口，使应用程序与具体的数字

电视硬件平台无关。通过中间件的平滑嫁接，同一应用软件可以在不同的操作系统上使用。

早期的中间件，如OpenTV, MediaHighway等提供端到端的系统支持，对推动各自数字电视系统和产品的发展起了重要的作用。但是由于这些系统之间互相独立，电视业务互不兼容，阻碍了整个数字电视的推广进程，也制约了自身技术的发展。目前，数字电视中间件正逐步向统一平台和统一业务支持的方向发展，符合国际或工业标准的开放平台逐渐占据了主导地位，因为开放的中间件平台可以保证不同系统应用间的协调操作。由此可见，发展统一、开放的数字电视中间件技术是市场的要求，也是国际化的要求。国外市场现行的三种数字电视标准都有自己的中间件，美国标准采用"DASE" 中间件，日本标准采用"ISDB-BML" 中间件，欧洲标准采用"DVB-MHP" 中间件。我国将要采用的中国数字电视标准，是一套完全具有自我知识产权的标准。

1.3 课题来源及研究意义

本课题来源于国家863重大软件专项项目——“基于中间件技术的因特网嵌入式操作系统及跨操作系统的中间件运行平台”。本课题的主要目标是研究和开发基于嵌入式操作系统的数字电视软件开发平台，改平台作为系统中间件，运行于面向构件的嵌入式操作系统中，为新一代的面向DTV的各类应用系统的开发建立高效可靠的软件平台，基于该平台所开发的DTV应用程序不经任何修改，甚至不必重新编译、链接就可以运行在不同硬件厂商提供的数字电视平台上,实现DTV应用程序的跨平台性，从而可以做到“一次编写,多处运行”的目的，从而加快DTV应用系统的研制与普及，降低开发成本。本文是在完成该课题的基础上进行的。首先研究了嵌入式操作系统和数字电视中间件的关键技术；然后分析了数字电视软件开发的技术架构和技术本质，提出了基于“和欣”操作系统的构件化数字电视软件开发平台的总体设计方案；最后给出了该软件平台的详细设计和实现。

虽然国内企业对数字电视的研发已经起步，但由于我们普遍采用外国厂家提供的与芯片捆绑销售的操作系统，受国外操作系统技术封闭的影响，国内厂家将很难做到自主开发产品，必须高度依赖国外厂家的技术支持。这样就必然产生开发环境、开发手段落后，手工作坊式生产应用软件甚至是电视节目的弊

端。为了更好的保护我国的数字电视产业，一方面要采用具有我国自主知识产权的操作系统，另一方面还必须加快对数字电视软件平台相关技术的研究和开发。因此，本文对构件化数字电视软件开发平台进行的研究，不仅具有重要的理论意义，还具有广阔的应用前景。

1.4 论文结构

本文的内容共分为七章:

第一章简要介绍了数字电视的发展概况、数字电视与软件中间件的关系以及课题来源及研究意义。

第二章重点介绍了“和欣”嵌入式操作系统及其构件技术。

第三章主要介绍了数字电视的技术架构和技术本质，并对数字电视操作系统进行了分析，提出了基于中间件技术的数字电视解决方案。

第四章主要论述了数字电视软件开发平台的体系结构设计及其构成特点。

第五章具体论述了数字电视软件开发平台的详细设计和实现。

第六章论述了基于数字电视中间件平台的可视化开发工具DTV Visual Developer的体系结构，并提出了一种数字电视软件开发的新模式。

第七章对目前的研究工作进行了总结，并提出了未来可能的研究方向和关键性问题。

第 2 章 “和欣”嵌入式操作系统及其构件技术

2.1 “和欣”操作系统概述

2.1.1 “和欣”操作系统简介

“和欣”是32位嵌入式操作系统。该操作系统可以从多个侧面进行描述：

32位嵌入式操作系统。操作系统基于微内核，具有多进程、多线程、抢占式、基于线程的多优先级任务调度等特性。提供FAT兼容的文件系统，可以从软盘、硬盘、FLASH ROM启动，也可以通过网络启动。和欣操作系统体积小，速度快，适合网络时代的绝大部分嵌入式信息设备。

完全面向构件技术的操作系统。操作系统提供的功能模块全部基于ezCOM构件技术，因此是可拆卸的构件，应用系统可以按照需要剪裁组装，或在运行时动态加载必要的构件。

从传统的操作系统体系结构的角度来看，和欣操作系统可以看成是由微内核、构件支持模块、系统服务器组成的。

- 微内核：主要可分为 4 大部分：硬件抽象层（对硬件的抽象描述，为该层之上的软件模块提供统一的接口）；内存管理（规范化的内存管理接口，虚拟内存管理）；任务管理（进程管理的基本支持，支持多进程，多线程）；进程间通信（实现进程间通信的机制，是构件技术的基础设施）。
- 构件支持模块：提供了对 ezCOM 构件的支持，实现了构件运行环境。构件支持模块并不是独立于微内核单独存在的，微内核中的进程间通讯部分为其提供了必要的支持功能。
- 系统服务器：在微内核体系结构的操作系统中，文件系统、设备驱动、网络支持等系统服务是由系统服务器提供的。在和欣操作系统中，系统服务器都是以动态链接库的形式存在。

2.1.2 和欣灵活内核简介

和欣操作系统的实现全面贯穿了ezCOM思想，ezCOM构件可以运行于不同

地址空间或不同的运行环境。可以把操作系统的内核地址区看成是一段特殊的地址空间，用户可以根据运行时的需求，自主选择将操作系统的某些系统服务构件、文件系统、图形系统、设备驱动构件等运行于内核地址空间或用户地址空间。与传统的操作系统的“大内核”、“微内核”体系结构相比，和欣操作系统内核里提供的系统服务，完全可以由用户依据系统自身的需求动态决定。因此称和欣操作系统内核为“灵活内核” （Agile Kernel）。

和欣灵活内核的体系结构，利用构件和中间件技术解决了长期以来困扰操作系统体系结构设计者的大内核和微内核在性能、效率与稳定性、安全性之间不能两全其美的矛盾。

下图来表示和欣灵活内核及其与系统构件和应用构件的关系：

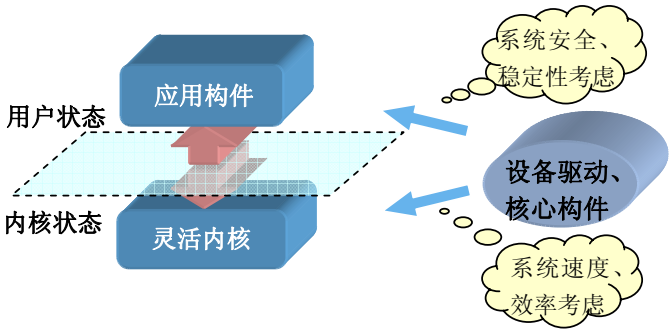


图 2.1 “和欣”灵活内核与系统构件和应用构件的关系

2.1.3 和欣操作系统提供的功能

从应用编程的角度看，和欣操作系统提供了一套完整的、符合ezCOM规范的系统服务构件及系统API，为在各种嵌入式设备的硬件平台上运行ezCOM二进制构件提供了统一的编程环境。

和欣操作系统还提供了一组动态链接构件库，这些构件库通常是开发嵌入式应用系统时不可缺少的：

- 图形系统构件库（方便开发图形用户操作界面）；
- 设备驱动构件库（各种输入输出设备的驱动）；
- 文件系统构件库（FAT 兼容，包括对 FLASH 等的支持）；
- 网络系统构件库（TCP/IP 等网络协议支持）。

系统提供的构件库，以及用户开发的应用程序构件都是通过系统接口与内

核交互，从这个意义上说，他们处于同样的地位。用户可以开发性能更好或者更符合需求的文件系统、网络系统等构件库，替换这些构件库，也可以开发并建立自己的应用程序构件库。这就是基于构件技术操作系统的优势之一。

此外，为了方便用户编程，在和欣SDK中还提供了以下函数库：

- 与微软 Win32 API 兼容的应用程序编程接口(zeew32 API)；
- 标准 C 运行库（libc）；
- 和欣提供的工具类函数（zeeutil）。

对程序员来说，和欣操作系统提供的用户编程接口与上一节中介绍的和欣构件运行平台完全一样。所以，在相互兼容的硬件平台上，不管运行的是和欣操作系统还是Windows操作系统，应用程序可以不加区分地在其上运行。

和欣操作系统实现并支持系统构件及用户构件相互调用的机制，为ezCOM构件提供了运行环境。关于ezCOM构件的运行环境，其描述与“和欣构件运行平台”是一样的，在此从简。因此，可以把和欣操作系统看成是直接运行在硬件平台上的“和欣构件运行平台”。

可以用下图来表示和欣操作系统及其主要构成：

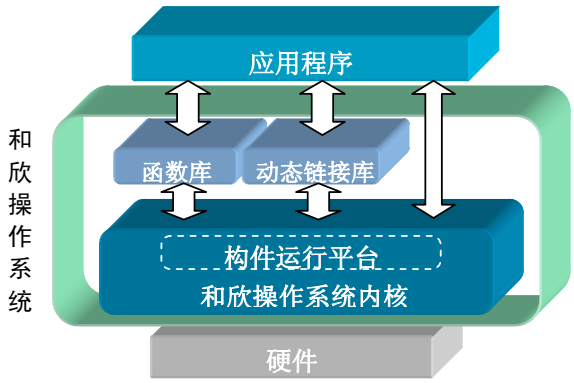


图 2.2 “和欣”操作系统的系统结构图

2.1.4 和欣操作系统的应用软件开发

和欣SDK提供了应用软件的开发环境和工具。开发“和欣”应用软件的开发环境如下图所示：

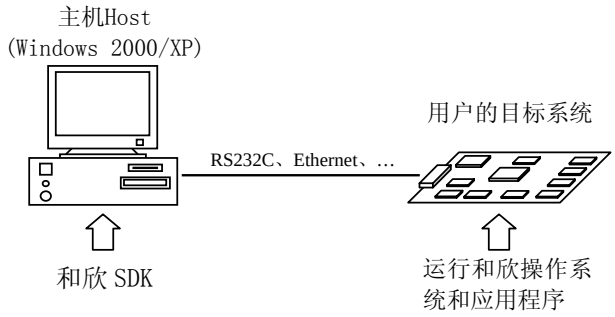


图 2.3 “和欣”应用软件的开发环境

开发“和欣”应用软件的过程，如下图所示：

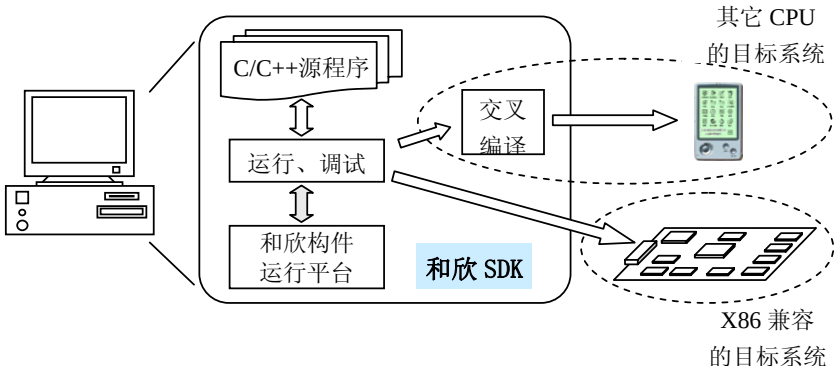


图 2.4 “和欣”应用软件的开发过程

2.1.5 和欣操作系统的优势

和欣操作系统的最大特点就是：

- 全面面向构件技术，在操作系统层提供了对构件运行环境的支持；
- 用构件技术实现了“灵活”的操作系统。

这是和欣操作系统区别于其它商用嵌入式操作系统产品的最大优势。

在新一代因特网应用中，越来越多的嵌入式产品需要支持网络服务，而网络服务的提供一定是基于构件的。在这种应用中，用户通过网络获得服务程序，这个程序一定是带有自描述信息的构件，本地系统能够为这个程序建立运行环境，自动加载运行。这是新一代因特网应用的需要，是必然的发展方向。和欣操作系统就是应这种需要而开发，率先在面向嵌入式系统应用的操作系统中实现了面向构件的技术。

因此，构件化的和欣操作系统可以为嵌入式系统开发带来以下好处：

- 在嵌入式软件开发领域，导入先进的工程化软件开发技术。嵌入式软件一般用汇编语言、C 语言，在少数系统中已经支持了 C++开发，但是由于还没有一个嵌入式操作系统能够提供构件化的运行环境，可以说，嵌入式软件开发还是停留在手工作坊式的开发方式上。和欣操作系统使得嵌入式应用的软件开发能够实现工程化、工厂化生产。
- 可以动态加载构件。动态加载构件是因特网时代嵌入式系统的必要功能。新一代 PDA 和移动电话等移动电子产品，不能再像以前那样由厂家将所有的功能都做好后固定在产品里，而要允许用户从网上获得自己感兴趣的程序。
- 随时和动态地实现软件升级。动态加载构件的功能，同样可以用于产品的软件升级，开发商不必为了添加了部分功能而向用户重新发布整套软件，只需要升级个别构件。
- 灵活的模块化结构，便于移植和剪裁。易于定制成针对不同硬件配置的紧凑高效的嵌入式操作系统。添加或删除某些功能模块也非常简单。
- 嵌入式软件开发商容易建立自己的构件库。在不同开发阶段开发的软件构件，其成果很容易被以后的开发所共享，保护软件开发投资。软件复用使得系列产品的开发更加容易，缩短新产品开发周期。
- 容易共享第三方软件开发商的成果。面向行业的构件库的建设，社会软件的丰富，使得设备厂家不必亲自开发所有的软件，可以充分利用现有的软件资源，充分发挥自己的专长为自己的产品增色。
- 跨操作系统平台兼容，降低软件移植的风险。在和欣开发环境上开发的软件所具有的跨平台特性，使得用户可以将同样的可执行文件不加修改地运行在和欣操作系统（嵌入式设备）与 Windows 2000/XP（PC）上。特别是对于需要将 Windows 上的软件移到嵌入式系统以降低产品成本的用户，这一特点不仅可以大大节约软件移植的费用，还可以避免因移植而带来的其它隐患。
- 功能完备的开发环境和方便的开发工具，帮助嵌入式开发人员学习和掌握先进的构件化软件编程技术，提高软件开发效率。应用软件可以在开发环境下开发调试，与硬件研制工作同时进行，缩短产品研制周期。

2.2 和欣构件运行平台

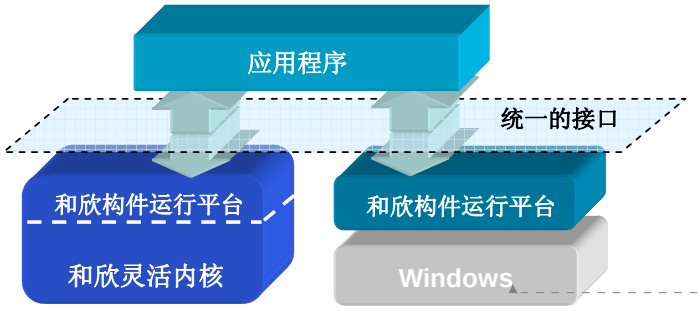
2.2.1 和欣构件运行平台简介

和欣构件运行平台提供了一套符合ezCOM（详见2.3节）规范的系统服务构件及支持构件相关编程的API函数，实现并支持系统构件及用户构件相互调用的机制，为ezCOM构件提供了编程运行环境。和欣运行平台有在不同操作系统上的实现，符合ezCOM编程规范的应用程序通过该平台实现二进制级跨操作系统平台兼容。

在和欣操作系统中，和欣构件运行平台与“和欣灵活内核”共同构成了完整的操作系统。

在Windows 2000、WinCE、Linux 等其它操作系统上，和欣构件运行平台屏蔽了底层传统操作系统的具体特征，实现了一个构件化的虚拟操作系统。在和欣构件运行平台上开发的应用程序，可以不经修改、不损失太多效率、以相同的二进制代码形式，运行于传统操作系统之上。

下图显示了和欣构件运行平台在Windows 2000/XP、和欣操作系统中的位置。



带格式的

图 2.5 “和欣”构件运行平台与操作系统的关系

2.2.2 和欣构件运行平台的功能

从和欣构件运行平台的定义，知道该平台为ezCOM提供了运行环境。从这个意义上，这里说的ezCOM技术也可以理解为在运行环境中对ezCOM规范提供支持的程序集合。

从编程的角度看，和欣构件运行平台提供了一套系统服务构件及系统API

（应用程序编程接口），这些是在该平台上开发应用程序的基础。

和欣操作系统提供的其它构件库也是通过这些系统服务构件及系统API实现的。系统提供的这些构件库为应用编程开发提供了方便：

- 图形系统构件库；
- 设备驱动构件库；
- 文件系统构件库；
- 网络系统构件库。

从和欣构件运行平台来看，这些构件和应用程序的构件是处于同样的地位。用户可以开发性能更好或者更符合需求的文件系统、网络系统等构件库，替换这些构件库，也可以开发并建立自己的应用程序构件库。

右图显示出和欣构件运行平台的功能及其与构件库、应用程序的关系。

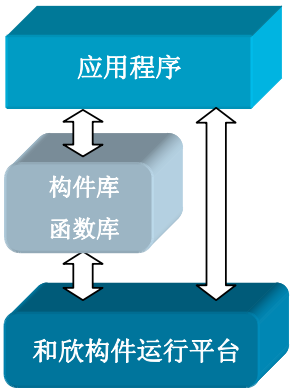


图 2.6 和欣构件运行平台功能图

从支持ezCOM构件的运行环境的角度看，和欣构件运行平台提供了以下功能：

- 根据二进制构件的自描述信息自动生成构件的运行环境，动态加载构件；
- 提供构件之间的自动通信机制，构件间通信可以跨进程甚至跨网络；
- 构件的运行状态监控，错误报告等；
- 提供可干预构件运行状态的机制，如负载均衡、线程同步、访问顺序控制、安全（容错）性控制、软件使用权的控制等；
- 构件的生命周期管理，如进程延续（Persistence）控制、事务元（Transaction）控制等；

总之，构件运行平台为ezCOM构件提供了对程序员完全透明的运行环境，构件可以运行在不同地址空间，不同环境，甚至跨网络。构件运行平台自动为构件运行提供支持，配置必要的网络协议、针对不同的输入输出设备的协议。程序员不必过多地去关心诸如网络协议转换及构件运行控制等与其它构件互操作时的协调问题，只需专注于自己需要解决的程序算法的实现。从而可以从繁杂庞大的应用环境体系中解放出来，大大提高编程的效率。

和欣构件运行平台直接运行二进制构件，而不是像JAVA和.NET那样通过虚

拟机在运行程序时解释执行中间代码。因此，与其它面向构件编程的系统相比，具有资源消耗小，运行效率高的优点。

2.2.3 和欣构件运行平台的技术优势

作为总结，和欣构件运行平台的主要技术优势列举如下：

- 开发跨操作系统平台的应用软件；
- 对程序员透明的 ezCOM 构件运行环境，提高编程的效率；
- 直接运行二进制构件代码，实现软件运行的高效率；
- 构件可替换，用户可建立自己的构件库。

需要说明的是，和欣构件运行平台实现的应用软件跨操作系统平台兼容是以具有同样的硬件体系结构为前提的。目前，和欣构件运行平台还不能支持不同指令系统的CPU间的“跨平台”兼容。

2.2.4 利用和欣构件运行平台编程

对程序员来说，编写运行于和欣构件运行平台上的程序，运用ezCOM技术和跨平台技术的具体方法，体现在对构件库的接口方法、通用API函数的调用上。应用程序运行所需要的动态链接库，则是在程序运行时由和欣构件运行平台自动加载的。

下图简明地表示了编写运行于和欣构件运行平台上的应用程序所需的相关要素之间的关系的示意。

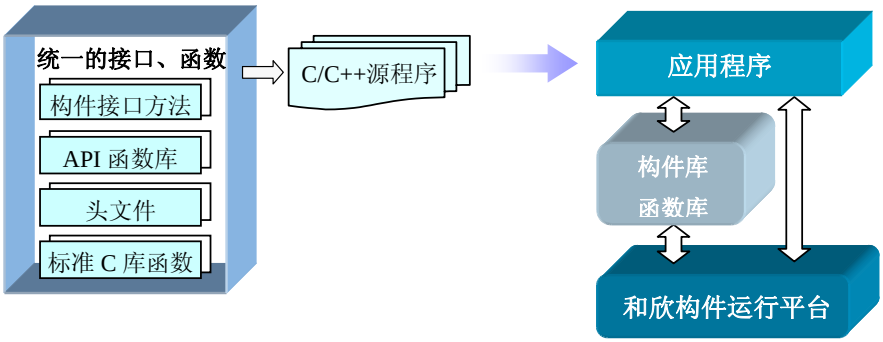


图 2.7 如何编写基于“和欣”构件运行平台的应用程序

2.3 ezCOM 构件技术

2.3.1 ezCOM 技术的由来

80年代以来，目标指向型软件编程技术有了很大的发展，为大规模的软件协同开发以及软件标准化、软件共享、软件运行安全机制等提供了理论基础。其发展可以大致分为以下几个阶段。

（1） 面向对象编程

通过对软件模块的封装，使其相对独立，从而使复杂的问题简单化。面向对象编程强调的是对象的封装，但模块（对象）之间的关系在编译的时候被固定，模块之间的关系是静态的，在程序运行时不可改变模块之间的关系，就是说在运行时不能换用零件。其代表是C++语言所代表的面向对象编程。

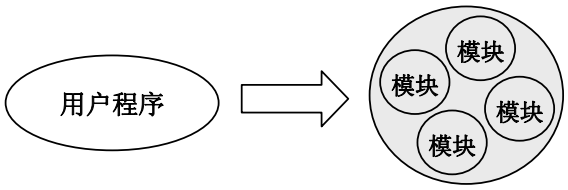


图 2.8 面向对象的编程模型

（2） 面向构件编程

为了解决不同软件开发商提供的构件模块（软件对象）可以相互操作使用，构件之间的连接和调用要通过标准的协议来完成。构件化编程模型强调协议标准，需要提供各厂商都能遵守的协议体系。就像公制螺丝的标准一样，所有符合标准的螺丝和螺母都可以相互装配。构件化编程模型建立在面向对象技术的基础之上，是完全面向对象的，提供了动态构造部件模块（运行中可以构造部件）的机制。构件在运行时动态装入，是可换的。其代表是COM技术。

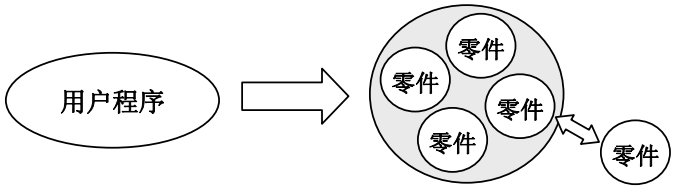


图 2.9 面向构件的编程模型

（3） 面向中间件编程

由于因特网的普及，构件可来自于网络，系统要解决自动下载，安全等问题。因此，系统中需要根据构件的自描述信息自动生成构件的运行环境，生成代理构件即中间件，通过系统自动生成的中间件对构件的运行状态进行干预或控制，或自动提供针对不同网络协议、输入输出设备的服务（即运行环境）。中间件编程更加强调构件的自描述和构件运行环境的透明性，是网络时代编程的重要技术。其代表是ezCOM、JAVA和.NET（C#语言）。

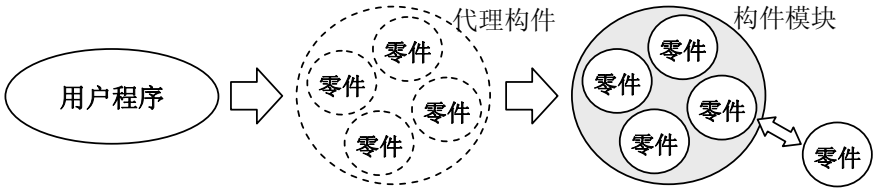


图 2.10 面向中间件的编程模型

在这样的发展过程中，人们逐步深化了对大规模软件开发所需的科学模型、网络环境下软件运行必要机制的理解，使软件技术达到了更高的境界，实现了：

- 构件的相互操作性。不同软件开发商开发的具有独特功能的构件，可以确保与其他人开发的构件实现互操作。
- 软件升级的独立性。实现在对某一个构件进行升级时不会影响到系统中的其它构件。
- 编程语言的独立性。不同的编程语言实现的构件之间可以实现互操作。
- 构件运行环境的透明性。提供一个简单、统一的编程模型，使得构件可以在进程内、跨进程甚至于跨网络运行。同时提供系统运行的安全、保护机制。

ezCOM 技术技术就是在总结面向对象编程、面向构件编程技术的发展历史和经验，为更好地支持面向以 Web Service（WEB 服务）为代表的下一代网络应用软件开发而发明的。ezCOM 很大程度地借鉴了 COM 技术，保持了和 COM 的兼容性，同时对 COM 进行了重要的扩展。

2.3.2 ezCOM 构件技术概要

ezCOM构件技术是面向构件编程的编程模型，它规定了一组构件间相互调用的标准，使得二进制构件能够自描述，能够在运行时动态链接。

ezCOM兼容微软的COM。但是和微软COM相比，ezCOM删除了COM中过时的约定，禁止用户定义COM的非自描述接口；完备了构件及其接口的自描述功能，实现了对COM的扩展；对COM的用户界面进行了简化包装，易学易用。

从上面的定义中，我们可以说ezCOM是微软COM的一个子集。ezCOM很大程度地借鉴了COM技术，保持了和COM的兼容性，同时对COM进行了重要的扩展。在和欣SDK工具的支持下，使得高深难懂的构件编程技术很容易被C/C++程序员理解并掌握。ezCOM中的“ez”源自与英文单词“easy”，恰如其分地反映了这一特点。

ezCOM技术是在总结面向对象编程、面向构件编程技术的发展历史和经验，为更好地支持面向Web Service（WEB服务）的下一代网络应用软件开发而开发的。

ezCOM的编程思想是“和欣”技术的精髓，它贯穿于整个技术体系的实现中。

为了在资源有限的嵌入式系统中实现面向中间件编程技术，同时又能得到C/C++的运行效率，ezCOM没有使用JAVA和.NET的基于中间代码—虚拟机的机制，而是采用了用C++编程，用和欣SDK提供的工具直接生成运行于和欣构件运行平台的二进制代码的机制。用C++编程实现构件技术，使得更多的程序员能够充分运用自己熟悉的编程语言知识和开发经验，很容易掌握面向构件、中间件编程的技术。在不同操作系统上实现的和欣构件运行平台，使得ezCOM构件的二进制代码可以实现跨操作系统平台兼容。

2.3.3 ezCOM 技术的意义

对于面向WEB服务的应用软件开发，以及开发操作系统这样的大型系统软件而言，采用ezCOM构件技术具有以下的意义：

- 不同软件开发商开发的具有独特功能的构件，可以确保与其他人开发的构件实现互操作。
- 实现在对某一个构件进行升级时不会影响到系统中的其它构件。
- 不同的编程语言实现的构件之间可以实现互操作。

- 提供一个简单、统一的编程模型，使得构件可以在进程内、跨进程甚至于跨网络运行。同时提供系统运行的安全、保护机制。
- ezCOM 构件与微软的 COM 构件二进制兼容，但是 ezCOM 的开发工具自动实现构件的封装，简化了构件编程的复杂性，有利于构件化编程技术的推广普及；
- ezCOM 构件技术是一个实现软件工厂化生产的先进技术，可以大大提升企业的软件开发技术水平，提高软件生产效率和软件产品质量；
- 软件工厂化生产需要有零件的标准，ezCOM 构件技术为建立软件标准提供了参考，有利于建立企业、行业的软件标准和构件库。

2.3.4 ezCOM 技术对软件工程的作用

大型软件的生产，已经不是传统的手工作坊式的开发所能够胜任的。所谓的软件工厂化生产，是软件工程研究几十年来追求的理想。实现这样的目标，需要有一个科学的编程模型，为所有参与项目开发的软件工程师开发的软件能够正确地对接运行提供技术上的保障。这些软件可以是由一个公司的不同部门提供，也可以是不同企业，不同地点、不同时间生产的。COM技术在微软经过十多年的应用与提炼，被证明是行之有效的编程模型，基于COM技术，微软造就了Windows NT以及配套应用软件这样的大型软件产品。汲取了COM技术的精华，并对其进行了扩展和简易包装的ezCOM技术，将成为软件工厂化生产的利器。

ezCOM的重要特点就是：构件的简易化包装；构件的相互操作性；软件升级的独立性；编程语言的独立性；进程运行透明度。在实际的编程应用中，ezCOM技术的优势体现在以下方面：

（1）易学易用

基于COM的构件化编程技术是大型软件工程化开发的重要手段。但是微软COM的繁琐的构件描述体系令人望而生畏。ezCOM的开发环境和欣SDK提供了结构简洁的构件描述语言和自动生成辅助工具等，使得C++程序员可以很快地掌握ezCOM编程技术。

（2）可以动态加载构件

在网络时代，软件构件就相当于零件，零件可以随时装配。ezCOM技术实

现了构件动态加载，使用户可以随时从网络得到最新功能的构件。

（3）采用第三方软件丰富系统功能

ezCOM技术的软件互操作性，保证了系统开发人员可以利用第三方开发的，符合ezCOM规范的构件，共享软件资源，缩短产品开发周期。同时用户也可以通过动态加载第三方软件扩展系统的功能。

（4）软件复用

软件复用是软件工程长期追求的目标，ezCOM技术提供了构件的标准，二进制构件可以被不同的应用程序使用，使软件构件真正能够成为“工业零件”。充分利用“久经考验”的软件零件，避免重复性开发，是提高软件生产效率和软件产品质量的关键。

（5）系统升级

传统软件的系统升级是一个令软件系统管理员头痛的工程问题，一个大型软件系统常常是“牵一发而动全身”，单个功能的升级可能会导致整个系统需要重新调试。ezCOM技术的软件升级独立性，可以圆满地解决系统升级问题，个别构件的更新不会影响整个系统。

（6）实现软件工厂化生产

上述几个特点，都是软件零件工厂化生产的必要条件。构件化软件设计思想规范了工程化、工厂化的软件设计方法，提供了明晰可靠的软件接口标准，使软件构件可以像工业零件一样生产制造，零件可用于各种不同的设备上。

（7）提高系统的可靠性、容错性

由于构件运行环境可控制，可以避免因个别构件的崩溃而波及到整个系统，提高系统的可靠性。同时，系统可以自动重新启动运行中意外停止的构件，实现系统的容错。

（8）有效地实现系统安全性

系统可根据构件的自描述信息自动生成代理构件，通过代理构件进行安全控制，可以有效地实现对不同来源的构件实行访问权限控制、监听、备份容错、通信加密、自动更换通信协议等等安全保护措施。

2.3.5 ezCOM 技术在“和欣”技术体系中的作用

ezCOM构件技术是“和欣”技术体系的精髓，它贯穿于整个技术体系的实

现中。可以说“和欣”操作系统是基于ezCOM实现的，同时它也是全面面向ezCOM构件技术的。

- 构件化的操作系统内核，以及“灵活内核”体系结构；
- 构件化的图形系统、设备驱动、文件系统等操作系统的系统服务；
- 支持应用软件跨平台二进制兼容的“和欣构件运行平台”；
- 构件化的应用软件，支持WEB服务，支持动态加载。

在面向嵌入式系统应用的操作系统中全面导入构件技术，对于开发功能越来越复杂的应用系统有着重要的意义。功能丰富的嵌入式设备将更多地要和网络发生关系，要支持 WEB 服务，前一节中所介绍的软件工程的意义，对于嵌入式系统开发来说同样是重要的。

2.3.6 如何用 ezCOM 技术编程

ezCOM是一个面向构件的编程模型，也可以说是一种编程思想，它表现为一组编程规范，包括构件、类、对象、接口等定义与访问构件对象的规定。

ezCOM的实现，是由一个配套的开发环境和运行环境完成的。

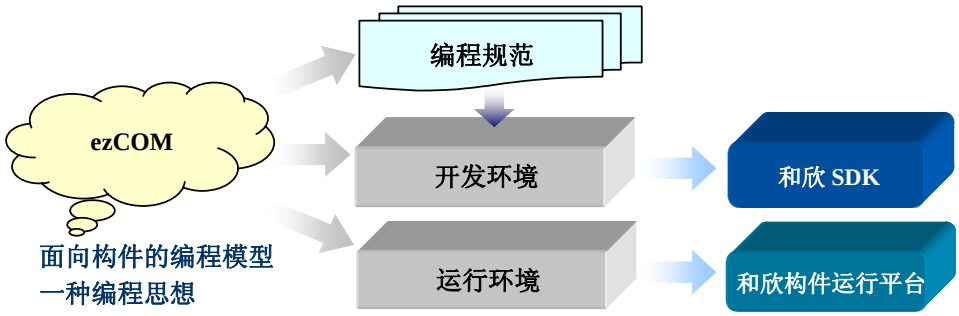


图 2.11 面向中间件的编程模型

第 3 章 数字电视的技术本质和数字电视操作系统

3.1 数字电视的技术架构

数字电视的硬件部分除了包含传统的显示器和喇叭等装置以外,还包含信道解调器、MPEG2 解码器和 CPU 等。这些硬件加上控制及协调他们共同工作的嵌入式操作系统，就构成了数字电视的基础技术架构。然后通过操作系统所支持的中间件，就可以在数字电视的硬件平台上，增添许多软件功能。例如浏览器，它可以帮助人们通过数字电视上网和观看电视节目。还有一些可以提供增值服务的应用软件，这些应用软件可以为我们提供视频点播、数据广播、个性化交互电视、远程教育等增值服务。数字电视的硬件和操作系统以及中间件和众多的应用软件将共同组成我们未来的数字电视。

以下是数字电视的技术架构图：

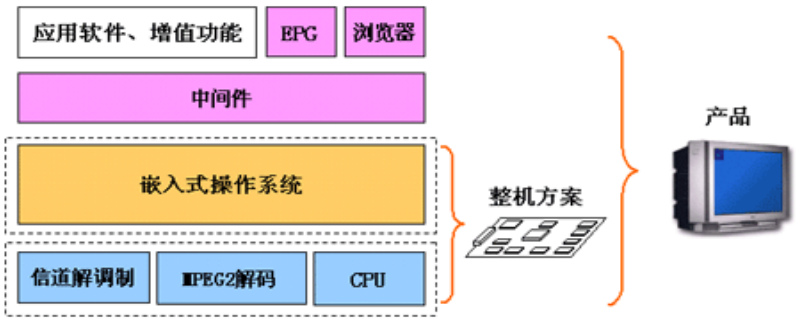


图 3.1 数字电视的技术架构图

3.2 数字电视的技术本质

3.2.1 未来数字电视的本质特征

随着数字电视技术的不断发展，未来的数字电视将具有以下特点。

（1）数字电视相当于一台电脑，它的功能不仅仅是“看电视”，而是一台具有 32 位 CPU 的 PC，靠软件来实现“功能”，由操作系统来提供软件运行环

境。

(2) 从使用者的角度来看，它决不能象电脑那样复杂， 使用者只需要关心自己想做的事，像使用“傻瓜相机”一样，不需要预备知识，这主要通过用高科技手段屏蔽复杂的动作。

(3) “功能”可扩展，自动加载运行应用程序，可支持各种增值业务系统可实现动态升级，“外设”即插即用，具有人性化的操作界面。

3.2.2 数字电视对系统软件的要求

由于数字电视具有支持动态电视节目和网络服务等功能，它对自身所采用的操作系统有较高的要求。这个操作系统应该具备以下的功能：

(1) 系统自动生成软件构件的运行环境，即自动生成中间件。

(2) 通过中间件可以使应用软件能够的动态加载到电视节目里， 做到即插即用。

(3) 支持数字电视的硬件设备的即插即用。

(4) 能够生成统一的构件标准和统一的软件工厂标准， 这样一来技术服务商只要遵循这两个标准，就可以自由开发应用软件，技术服务商的工作效率将大为提高。

(5) 支持高效率的构件运行平台。

(4) 安全性，防病毒。

另外, 操作系统还应该为用户提供方便、自然、人性化的操作方式和界面。为整机厂家提供自主可控的关键技术，使厂家对市场需求能做成快速响应。为技术服务商和内容制作商提供标准的、开放的开发平台，便于他们开发应用软件，和电视节目。

3.3 数字电视操作系统

3.3.1 数字电视操作系统概述

目前国外数字电视采用的操作系统主要有：Nucleus、PSOS、 OS20(ST)、Linux、VxWorks 和 Windows CE 等。所有这些操作系统中国都不占有任何产

权。同时某些国外厂商采用操作系统与芯片（CPU）捆绑的方式，向我国提供所谓的芯片和操作系统一体化的解决方案。因为我国在短时间范围内还无法产出数字电视所需的 CPU，还得依赖外国产品，因此外国厂商就借 CPU 捆绑外国操作系统来打压中国自己的操作系统。同时外国芯片厂家也能提供的自己的操作系统，但这种操作系统比较"原始"，特点如下：

（1）基本上是实时内核 + 设备驱动。

（2）设备驱动、API 等没有标准。

（3）缺少网络支持（这些功能都"交"给了"中间件"）。

（4）图形系统功能弱，或没有（这些功能都"交"给了"中间件"）。

（5）有些操作系统没有文件系统 （这些功能都"交"给了"中间件"）。

如果我们采用外国厂家提供的操作系统，由于受国外操作系统技术封闭的影响，国内厂家将很难以自主开发产品，必须高度依赖国外厂家的技术支持。必然产生开发环境、开发手段落后，手工作坊式生产应用软件甚至是电视节目的弊端，下图具体展示了采用国外操作系统后，给我国数字电视产业界带来的不利影响。由此可见，在我国的数字电视领域采用具有我国自主知识产权的操作系统是大势所趋。

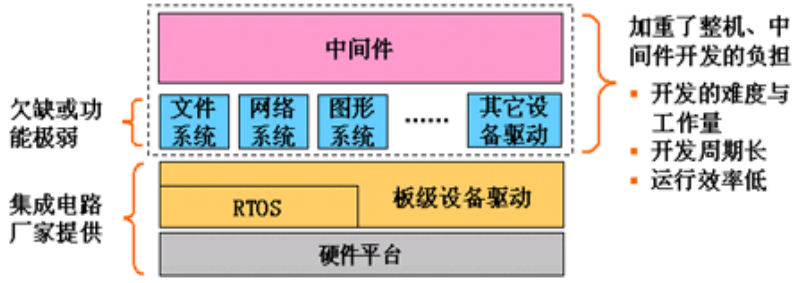


图 3.2 当前的数字电视开发模式

3.3.2 “和欣”嵌入式操作系统及其对数字电视的支持

“和欣”操作系统是国内唯一能够提供构件化数字电视软件平台解决方案的操作系统。作为中国唯一的具有完全自主知识产权的网络操作系统，“和欣”操作系统是世界网络化潮流的必然产物。“和欣”操作系统完全能够满足我国的

乃至全球的数字电视的要求。以下是“和欣”操作系统的基本架构，已及与中间件和硬件平台的关系。

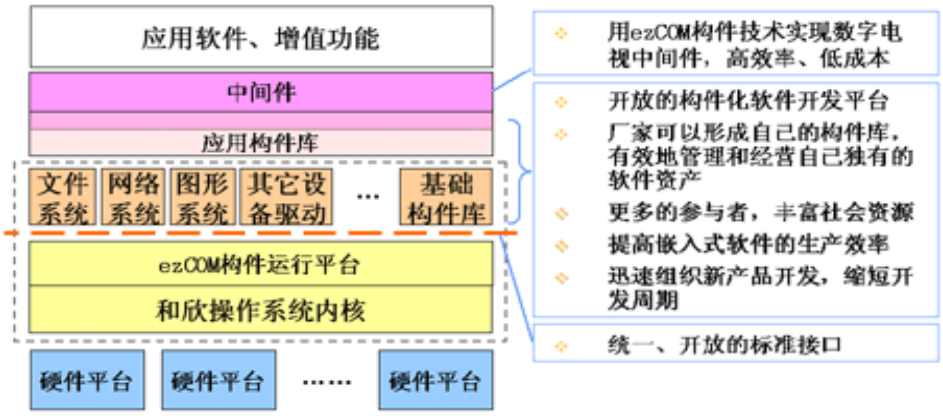


图 3.3 “和欣”操作系统的系统架构

“和欣”操作系统是构件化操作系统，因此变更系统配置非常容易，适应不同档次的硬件平台。 以下是“和欣”操作系统的技术优势所在：

- 在操作系统层面上提供了高效率的中间件运行平台，对数字电视的中间件提供最好的支持；
- 动态加载软件，不需用户干预；
- 系统动态升级，便于厂家售后维护和管理；
- 支持设备的"即插即用"，便于今后的高端产品与其它信息家电、数码相机等随时接入交换数据信息；
- 开放的构件化软件开发平台；
- 构件化的图形系统，方便地实现各种友好、生动的用户界面，便于画面重叠、各种窗口配置等程序编制；
- 安全性高，能防病毒，从而保证了数字电视节目播出的连贯性。

3.4 基于中间件技术的数字电视解决方案

“和欣”在操作系统层面上提供了高效率的中间件运行环境，基于这个环境可以为用户提供一个完善的数字电视软件开发平台，它提供一套标准的 API

（应用程序接口）或 IDE（集成开发环境），使得基于该平台所开发的应用程序不再依赖各个不同的硬件平台而正常的工作。它能够高效地运行在“和欣” 操作系统上，为应用程序提供一个高速通用的软件平台；同时也能支持其它实时操作系统，为用户提供一个完整的中间件解决方案。

基于中间件技术的数字电视解决方案能对网络服务商和硬件制造商提供最大的便利，帮他们降低成本，缩短研发周期。对机顶盒制造商来说，采用该解决方案可以缩短机顶盒软件开发周期，使产品更快的进入市场，而且能够保留各自产品的特色；对于数字电视服务商来说，采用该解决方案可以让他们自由选择适合自己的、最具性价比的机顶盒硬件，而不必关心是否兼容，同时可以方便地开发自己的应用程序。

由于我国无法在短时间内出产数字电视的芯片，所以在开始阶段我们仍将会基于外国的芯片进行开发。由于“和欣”操作系统的内核是灵活内核，因此将它移植到其它芯片上相对其他操作系统而言，是容易的。

下图表示了当前环境下基于中间件技术的数字电视中间件平台的系统示意图以及未来的发展方向。

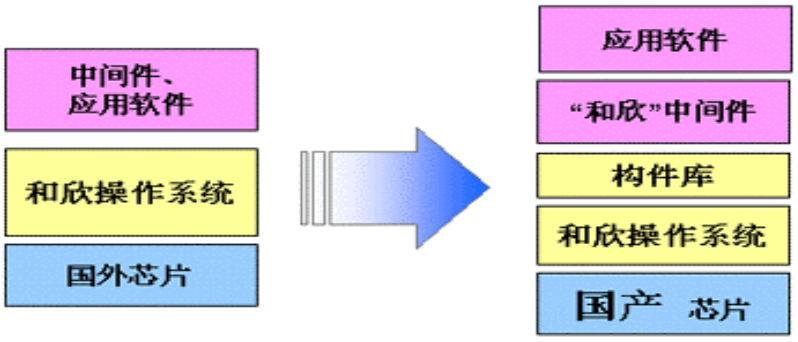


图 3.4 数字电视中间件平台的系统示意图

3.5 数字电视中间件的特征

数字电视中间件是将应用程序与操作系统平台和具体的硬件细节隔离开来的软件环境，允许应用的设计、运行不用考虑底层软硬件体系结构，它以应用程序接口API的形式存在，整个API集合通常被存储在数字电视的flash中。它从本质上是对分布式应用的抽象，因而抛开了与应用相关的业务逻辑的细节，保

留了典型的分布交互模式的关键特征。经过抽象，将纷繁复杂的分布式系统经过提和必要的隔离后，以统一的层面形式呈现给应用。应用在数字电视中间件提供的环境中可以更好地集中于业务逻辑上，并以构件化的形式存在，最终自然而然地在异构环境中实现良好的协同工作。

经过以上对“和欣” 嵌入式操作系统的特性和对数字电视中间件功能的分析，我们可以看到数字电视中间件应具有如下特征:

(1) 面对特定应用领域，具有中间件的各项特点。

数字电视中间件针对数字广播系统设计，其架构应满足数字广播领域中间件的需求；其采用技术符合数字广播的诸多标准和协议，比如MPEG-2标准。满足大量应用的需要；运行于多种硬件和操作系统平台，比如x86体系结构或是Mips体系结构；支持分布计算，提供跨网络和平台的透明性的应用或服务的交互，比如通过广播网络传输的各种多媒体应用；支持标准的接口。

(2) 具有实时嵌入式特性

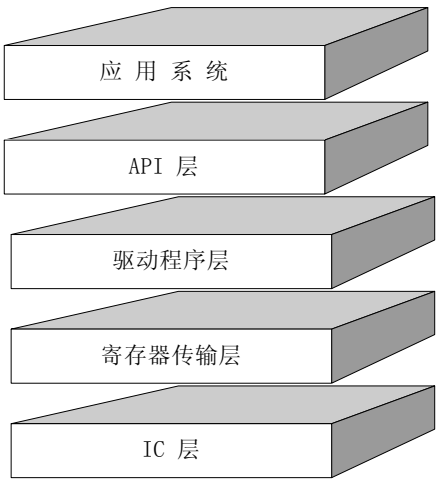
数字电视作为传统电视的延伸及扩一展，其硬件资源受限，成本敏感，一般的数字电视系统都采用flash作为存储和运行程序的空间。数字电视中间件受限于数字电视所提供的系统资源，在设计及实现时必须考虑系统资源对软件运行的限制。数字电视中间件必须对用户及通过网络传输的事件做出及时响应，在响应时间上有所要求。但它又不同于诸如航天领域等所采用的实时系统，它的时限是一个柔性灵活的，比如在用户发出事件后，若处理器负载较重，可以允许响应有相应的延迟。在偶然的超时错误发生时，失败造成的后果并不严重。仅仅是轻微的降低了系统的吞吐量。

第 4 章 DTV 软件开发平台的系统结构和构成特点

4.1 基于 API 函数库的 DTV 软件开发方法

目前，绝大多数的 DTV 软件开发都是在芯片厂商提供的 API 函数库的基础上进行功能扩展或设计的。这些 API 函数库往往被设计成为层次结构的平面型 API 接口，用户实现了自己的应用系统后通过编译、链接把应用系统和底层函数库连接起来，使得它们协同工作。由于这些 API 函数库提供了调用芯片基本功能的大部分方法，所以它可以用于各种各样的基于该芯片的应用软件开发，这充分体现了 API 函数库的可重用性的优势，当然这种优势是建立在函数库具有良好的接口内涵的基础上的。

下图显示了这种基于 API 函数库的 DTV 软件开发方法。其中应用程序通过一个平面结构的 API 层与芯片进行交互。



基于API函数库的DTV软件开发方法

基于 API 函数库的 DTV 软件开发方法可以很好地实现软件开发的可重用性，但也存在以下一些问题：

（1）当 API 非常多时，使用会非常不方便，需要对函数进行组织。往往很多厂商提供的 API 函数库都附带了大量的函数使用手册，开发人员首先要花费很长的时间来熟悉这些函数的功能和用法，而这些函数由于数量庞大而又缺乏组织，所以难于被开发人员尽快掌握。

（2）当更换芯片时尽管有可能不用修改源程序，但由于 API 函数库发生了改变，那么应用系统仍然需要重新进行编译、链接，否则就无法正常运行。

（3）开发效率低下。由于采用了较为原始的开发方法，应用系统从无到有的开发周期会比较漫长，缺少流行的诸如可视化开发方法的支持。

（4）应用系统可移植性差。由于上层应用高度依赖基于特定厂商芯片 的 API 函数库，使得应用系统的可移植性很差，即便是同一应用要想移植到不同

厂商提供的芯片上也需要进行重复开发。

4.2 DTV 软件开发方法平台的体系结构设计

“嵌入式操作系统 + DTV软件开发平台中间件”构成了基于嵌入式操作系统的数字电视系统的开发运行环境。嵌入式实时操作系统及设备的上层应用接口通过驱动程序控制相应设备，并对中间件提供系统服务。在数字电视软件系统中，DTV软件平台中间件基于操作系统为应用软件提供运行支持，同时对操作系统提供的设备控制接口进行适当的封装，由此形成的新接口解除了不同的应用提供商与不同的数字电视中特定的硬件和软件细节间的高度藕合关系，从而实现了内容只需创作一次就可在“任何”地方运行。

DTV软件开发平台采用立体层次结构，作为系统构件运行在操作系统中间件运行平台上，由于构件与外部应用程序以及操作系统的交互完全通过接口进行，保证了构件的实现细节不会影响到使用构件的程序,使得它可以不断的根据不同的功能需求和所支持应用的类型增加新的功能和要求。如图4.2 所示。

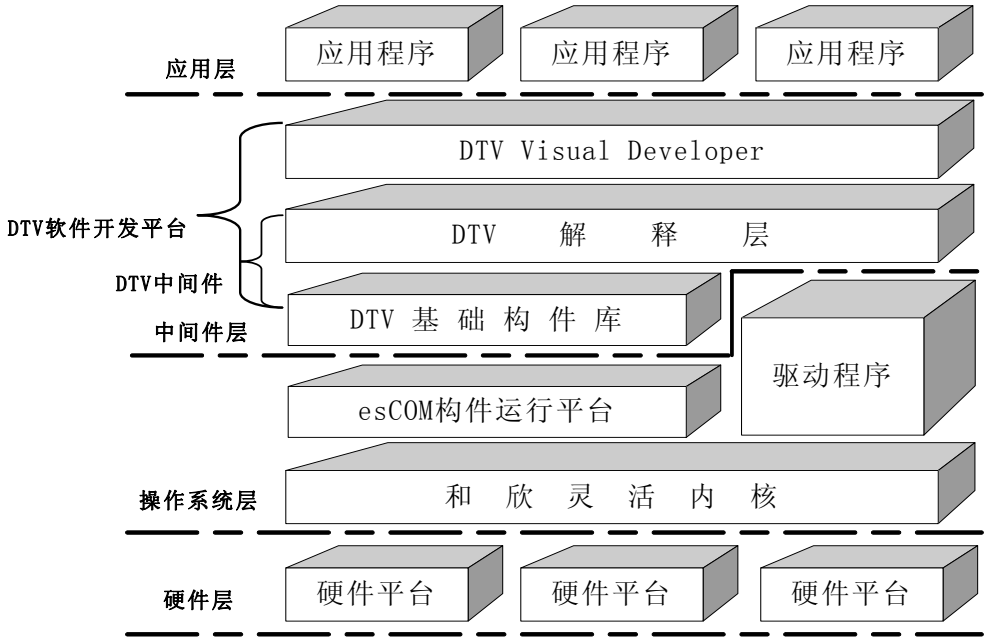


图4.2 DTV软件开发平台的系统结构

基于DTV软件开发平台的软件体系结构可以分为四层:

(1) 硬件层: 它是整个软件体系的基础，我们目前的硬件平台采用Genesis Microchip公司的gm5221系列处理芯片，它是高度综合化的显示器控制器，既支持入门级LCD电视也支持多媒体LCD监视器应用。

(2) 操作系统层: 这一层包括和欣灵活内核、esCOM构件运行平台和硬件驱动程序。这一层的作用主要是为中间件层提供运行环境和提供操纵数字电视硬件设备的软件编程接口。

(3) 中间件层: 这一层包括“DTV解释层”、“DTV基础构件库”以及可选的可视化开发工具——DTV Visual Developer，为数字电视的应用提供开发和运行环境，并提供统一的构件化接口.

(4) 应用层: 主要是一些面向数字电视的应用，如视频点播等。

下面将重点介绍前三层的构成细节。

4.2.1 gm5221 硬件平台

gm5221 系列芯片是Genesis Microchip公司推出的应用于平面电视及LCD监视器领域的控制器。该系列产品产品具有如下特点：

(1) gm5221 系列是高度综合化的显示器控制器，既支持入门级LCD电视也支持多媒体LCD监视器应用。

(2) 它配备了一个NTSC/PAL视频输入端口及完整的AD转换器 （带有PLL和LVDS传送装置)。

(3) 除了采用了 Genesis Microchip 公司拥有专利的图象处理先进技术（图象格式转化）外，gm5221 系列产品同时具备了色彩及对比度可选择（ACC）和色彩灵活管理（ACM-II）性能，可以改善对比度及增强图象色彩度。

(4) gm5221 系列支持PC机频率输入 （VGA、UXGA）和 标准画质及高画质图象输入格式。图象最大显示分辨率可达 1280x1024。

(5) gm5221 和它的同系列产品 gm2221、gm5321、gm2321 均具有兼容性，支持输入输出多机构配置。

下面是gm5221 的硬件系统设计图。

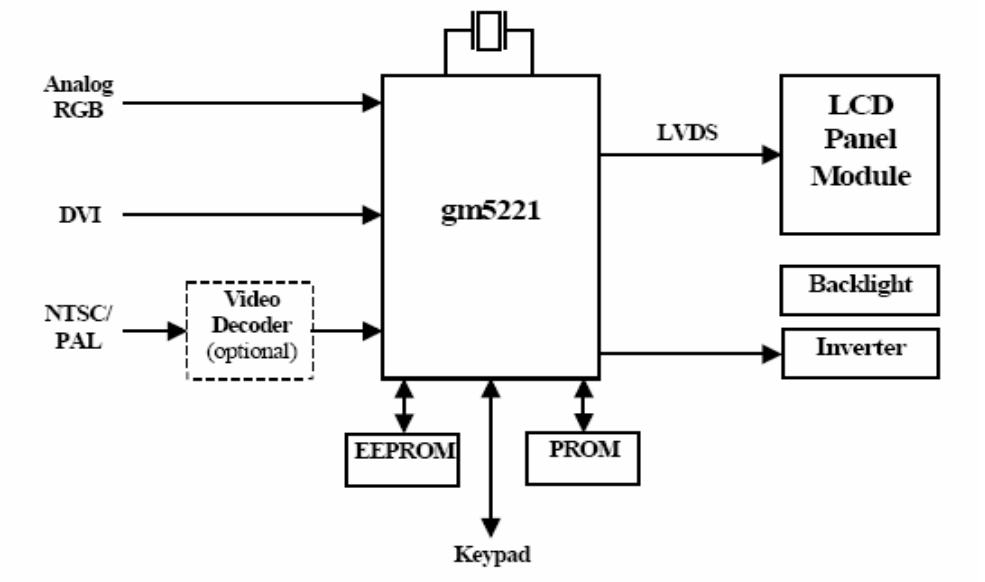


图 4.3 gm5221 的硬件结构图

4.2.2 “和欣”实时操作系统

由于数字电视系统中的资源稀缺，在操作系统层采用了“微内核” 形态的“和欣灵活内核” ，它主要可由4大部分组成：硬件抽象层（对硬件的抽象描述，为该层之上的软件模块提供统一的接口）；内存管理（规范化的内存管理接口，虚拟内存管理）；任务管理（进程管理的基本支持，支持多进程，多线程）；进程间通信（实现进程间通信的机制，是构件技术的基础设施）。

构件运行平台为上层的DTV软件开发平台提供了完全透明的运行环境，使其可以运行在不同地址空间，不同环境，甚至跨网络。构件运行平台自动为构件运行提供支持，配置必要的网络协议、针对不同的输入输出设备的协议。程序员不必过多地去关心诸如网络协议转换及构件运行控制等与其它构件互操作时的协调问题，只需专注于自己需要解决的程序算法的实现。

和欣构件运行平台直接运行二进制构件，而不是像JAVA和.NET那样通过虚拟机在运行程序时解释执行中间代码。因此，与其它面向构件编程的系统相比，具有资源消耗小，运行效率高的优点，而这一点则正是当前数字电视软件开发的重要要求。

4.2.3 DTV 软件开发平台

“DTV解释层”和“DTV基础构件库”构成了DTV中间件平台，而与上层的可视化开发工具——DTV Visual Developer一起组成了DTV软件开发平台。其中“DTV基础构件库”运行于esCOM构件运行平台之上，它提供了对DTV芯片底层功能的封装，并为“DTV解释层”提供服务；DTV解释层根据下层基础构件库提供的“原材料”形成对应DTV基本功能的构件集合，此外，为了满足某些实时性要求较高的功能需求，一部分构件可以拥有跨越构件运行平台而直接访问设备驱动程序的权限；DTV Visual Developer则为开发者提供了一个适于DTV软件开发的可视化的集成开发环境，它根据DTV软件开发的特点，利用DTV解释层所提供的接口将一些原本分散的功能要素封装成为具有一定属性并可以响应相应事件的控件，这样就可以借助流行的可视化开发方法来提高DTV软件开发的效率。

以下分别介绍了各部分的基本构成：

DTV基础构件库：包括OSD(On Screen Display)构件、IO（Input/Output）构件、色彩处理构件、通讯协议构件、时钟构件、内存管理构件、视频解码构件、音频解码构件等。

DTV解释层: 包括GUI（Graphics User Interface）构件、模式处理构件、频道调节构件、画质调节构件、音质调节构件等。

DTV Visual Developer: 由开发工具和相关的配置文件组成。

4.3 DTV 软件开发平台的构成特点

DTV软件开发平台是针对当前数字电视软件开发的具体特点进行设计的，它具有很多传统开发方法所无法比拟的技术优势：

（1）是面对数字电视领域的中间件平台。其架构满足数字广播领域中间件的需求，它提供了跨网络和平台的透明性的应用或服务的交互，比如通过广播网络传输的各种多媒体应用。

（2） 通过该平台可以使应用软件能够的动态加载到电视节目里， 做到即插即用。

（3）支持数字电视的硬件设备的即插即用，即实现了软件的跨平台，可以做到“一次编写,多处运行” ，降低软件移植的风险。

（4）平台遵守esCOM标准，这样一来技术服务商只要遵循这标准， 就可以自由开发应用软件，技术服务商的工作效率将大为提高。

（5）具有可拆卸的可视化开发环境。基于DTV中间件平台的可视化开发环境——DTV Visual Developer为应用软件开发者提供了统一的开发方法，它基于“控件”模型，具有可视化开发方法的诸多优点，同时它是可拆卸并支持动态配置的，用户可以根据自己的需要来对其进行自定义。基于该工具可以极大的提高数字电视软件开发的效率。

（6）在数字电视软件开发领域，导入先进的工程化软件开发技术。数字电视软件一般用汇编语言、C语言，在少数系统中已经支持了C++开发，但是由于还没有一个开发环境能够提供构件化的开发运行环境，可以说，数字电视软件开发还是停留在手工作坊式的开发方式上。DTV软件开发平台使得数字电视的应用软件开发能够实现工程化、工厂化生产。

第 5 章 DTV 中间件平台的详细设计与实现

5.1 DTV 中间件平台的系统结构

DTV中间件平台是整个DTV软件开发平台的基础，它由DTV基础构件库和DTV解释层构成。其中DTV基础构件库建立在嵌入式操作系统之上，它屏蔽了对下层的操作系统和芯片的具体操作，使得上层构件只需专注于与数字电视相关的接口信息而不必理会它的实现细节；DTV解释层是对数字电视各种基本功能的具体解释，它体现了设计者对数字电视基本功能的理解，因此是相对灵活的一层。

下图表示了DTV中间件平台的系统结构以及各构件之间的相互关系。

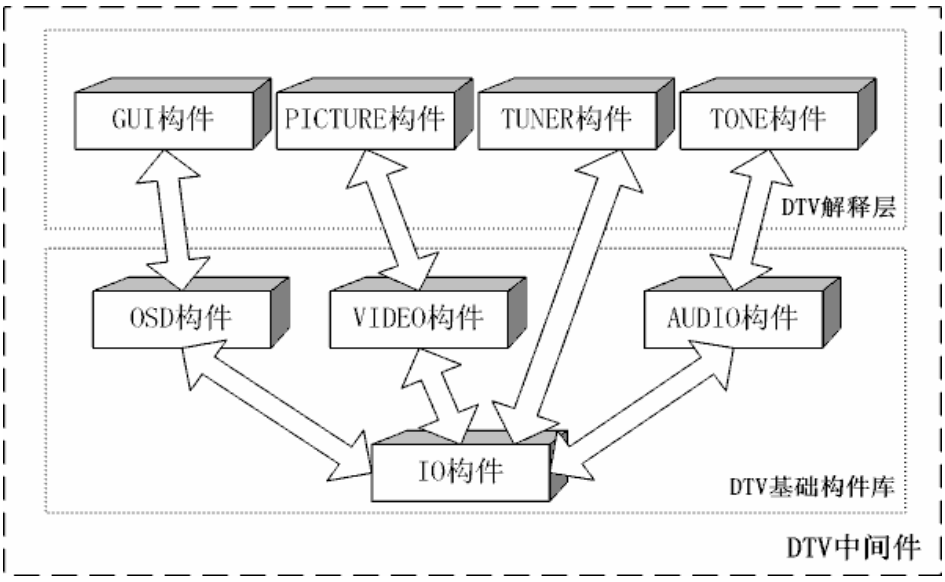


图5.1 DTV中间件平台的系统结构图

5.2 DTV 基础构件库的详细设计与实现

DTV基础构件库是整个DTV中间件平台的基础，它封装了针对数字电视软件开发所必需的函数接口，并以构件化的形式提供给上层用户。这些构件都是可拆卸的，系统可以按照需要进行剪裁组装，或在运行时动态加载必要的构件，

可以很容易定制成为针对不同硬件配置的紧凑高效的DTV中间件平台。

下面将详细的介绍OSD(On Screen Display)构件、IO（Input/Output）构件、视频解码构件和音频解码构件的设计。

5.2.1 OSD(On Screen Display)构件

OSD构件是数字电视屏幕菜单显示的基础，其主要作用是实现显示菜单所需的各功能要素。

```
//OSD 构件设计.
[
    version(1.0), uuid(0000132a-0300-0200-C000-000000320056),
    urn(http://www.koretide.com/repository)
]
component DtvOsd
{
    //提供获取和设置构件属性方法的接口。
    [uuid(0000130b-0000-0000-C000-000000320056)]
    interface IProperty {
        HRESULT SetProperty(
                [in]EzStr prType, [in]EzVar prValue
        );
        HRESULT GetProperty(
                [in]EzStr prType, [out]EzVar * pPrValue
        );
    }

    //提供 OSD 操作的接口。
    [uuid(0000133b-0040-0000-C000-000000000056)]
    interface IOsd {
        HRESULT OsdFontsArrayInit();
        HRESULT OsdGetCurrentFont([out] HANDLE font);
```

```
            HRESULT OsdSetCurrentFont([in] HANDLE font, [out] BOOL success);
            HRESULT OsdDrawString([in] WORD osdNumber, [in] WORD x, [in] WORD
            y, [in] CHAR* pText, [in] UINT background,[in] UINT foreground);
            HRESULT OsdGetFontHeight ([out] WORD height);
            HRESULT OsdGetTextExtent ([in] CHAR* pText, [in] WORD width,
                [in] WORD height);
            HRESULT OsdSetDestination ([in] UINT OSD_Number, [in] UINT
                OSD_X_dstn, [in] UINT OSD_Y_dstn);
            HRESULT OsdDrawRect ([in] UINT OSD_Number, [in] WORD x, [in] WORD
                y, [in] WORD sizex, [in] WORD sizey, [in] UINT fillColor);
            HRESULT OsdDrawBitmap ([in] UINT OSD_Number, [in] WORD startx,
[in]
                 WORD starty, Bitmap* bmp,[in] UINT replacementColor);
        }
        //OSD 类
        [uuid(00001312-0000-0000-C000-000000000066)]
        class COsd {
            interface IProperty;
            interface IOsd;
        }
    }
```

其中 IProperty 接口是 Elastos 提供的一个很重要的接口，很多构件都实现了此接口方法。这个接口的功能类似于哈希表，用于保存构件的属性名，属性值对。该接口实现的目的是对 com 做了改进， MSCOM 中采用 property 关键字来设置或获取构件对象的属性，那么每一个属性在对象的虚函数表中都对应有获取值和设置值两个方法，占用两个指针位大小的空间，当对象的属性比较多的时候，对于内存的消耗比较大；而采用统一的 IProperty 接口，在方法中增加一个参数，那么不论此对象有多少属性，都只有一对方法，在虚表中只占两个指针位大小，从而减小了对象虚函数表的大小。

IProperty 接口只定义了两个方法：GetProperty 和 SetProperty，用来得到和设置对象的属性，方法原型如下：

```
HRESULT GetProperty( [in] EzStr prType, [out] EzVar * pPrValue );
HRESULT SetProperty( [in] EzStr prType, [in] EzVar prValue );
```

SetProperty 方法的第一个参数是属性的名称，第二个参数是用来设置对应的属性的值。第一个参数类型为字符串类型。第二个参数的类型为 EzVar 类型，即一个通用的数据类型，相当于 VB,VC 中 Variant 类型。它是一个变体数据类型，可以兼容构件中的所有数据类型，因此，不论用户所要设置的属性的值是字符串或整数或其他，都可以通过此方法来完成。在 Elastos 内部支持 EzVar 到其他数据类型的显式的转换。

对应的 GetProperty 方法用来得到对象的属性，它的第一个参数也是属性的名称，类型为字符串类型；第二个参数是用来获得属性的值的地址，因此，参数的类型是 EzVar * 类型 — 指向 EzVar 类型数据的指针（地址）。用户通过第二个参数获得相应的属性值后，可以将它转化为确切的数据类型。

在 EzCom 下使用 IProperty 接口方法的示例如下：

```
HRESULT foo() {
        //构件 CFoo 中实现了 IProperty 接口并且有属性名“Name”
        HRESULT hr = E_FAIL;
        EzVar varFooGet;
        EzStr   strGet;
        EzVar varFooSet = EZCSTR(“FooProperty”);
        IPropertyRef   rFoo = NEW_COMPONENT(CTX_SAME_DOMAIN) CFoo;
        If (rFoo.isValid()) {
            hr = rFoo.SetProperty(“Name”，varFooSet);//设置构件的”Name”属性
            If FAILED(hr)       return hr;
            hr = rFoo.GetProperty(“Name”, &varFooGet);
            If FAILED(hr)       return hr;
            strGet = (EZSTR)varFooGet;   //strGet 中得到字符串” FooProperty”
            EzStr::FreeString(strGet);//释放内存空间
            return NOERROR;
        }
        return hr;
    }
```

5.2.2 IO（Input/Output）构件

IO 构件主要包括三个接口：数模转换接口（IDtAC),模数转换接口（IAtDC）和通用输入输出接口（IGPIO)。

```
//IO 构件设计.
[
    version(1.0), uuid(0000147a-0300-0200-C000-000000320048),
    urn(http://www.koretide.com/repository)
]
component DtvIO
{
    //提供获取和设置构件属性方法的接口。
    [uuid(0000133b-0000-0000-C000-000000320056)]
    interface IProperty {
        HRESULT SetProperty(
                [in]EzStr prType, [in]EzVar prValue
        );
        HRESULT GetProperty(
                [in]EzStr prType, [out]EzVar * pPrValue
        );
    }

    //数模转换接口。
    [uuid(0000137b-0040-0000-C000-000000320056)]
    interface IDtac {
        HRESULT DacSetOutput ([in] UINT channel, [in] UINT data);
        HRESULT DacSetContinuous ([in] UINT channel, [in] UINT speed);
        HRESULT DacEnableOutput ([in] UINT channel, [in] UINT on);
        HRESULT DacSetSource ([in] UINT source);
        HRESULT DacLowPowerMode ([in] BOOL enable);
    }
```

```
//模数转换接口。
[uuid(0000137b-0040-0000-C000-000000320056)]
interface IAtdc {
    HRESULT AdcGetValue ([out] UINT data);
    HRESULT AdcStartContinuous ([in] UINT chan, [in] UINT speed);
    HRESULT AdcStopContinuous ();
    HRESULT AdcDoOneShot ([in] UINT chan);
}

//通用输入输出接口。
[uuid(0000137b-0040-0000-C000-000000320056)]
interface IGpio {
    HRESULT GpioSelectPinFunction ([in] UINT port_num, [in] UINT
        bit_num, [in] UINT function);
    HRESULT GpioConfigurePort ([in] UINT port_num, [in] UINT
        direction);
    HRESULT GpioConfigurePin ([in] UINT port_num, [in] UINT
        bit_num, [in] UINT direction);
    HRESULT GpioWritePort ([in] UINT port_num, [in] UINT value);
    HRESULT GpioWritePin ([in] UINT port_num, [in] UINT bit_num,
        [in] UINT value);
    HRESULT GpioReadPort ([in] UINT port_num, [out] UINT value);
    HRESULT GpioReadPin ([in] UINT port_num, [in] UINT bit_num,
        [out] UINT value);
    HRESULT GpioUseExternalSram ([in] BOOL enb, [out] UINT value);
}

//数模转换类
[uuid(00001312-0000-0000-C000-000000000079)]
class CDtac {
    interface IProperty;
```

```
            interface IDtac;
        }

        //模数转换类
        [uuid(00001312-0000-0000-C000-0000000000234)]
        class CAtdc {
            interface IProperty;
            interface IAtdc;
        }

        //通用输入输出类
        [uuid(00001312-0000-0000-C000-000000001066)]
        class CGpio {
            interface IProperty;
            interface IGpio;
        }
    }
```

5.2.3 音频解码（Audio）构件

Audio构件的主要作用是实现音频解码的各功能要素。

```
    //Audio 构件设计.
    [
        version(1.0), uuid(0000132a-0300-0200-C000-000089300666),
        urn(http://www.koretide.com/repository)
    ]
    component DtvAudio
    {
        //提供获取和设置构件属性方法的接口。
        [uuid(0000130b-0000-0000-C000-000000320056)]
        interface IProperty {
```

```
            HRESULT SetProperty(
                    [in]EzStr prType, [in]EzVar prValue
            );
            HRESULT GetProperty(
                    [in]EzStr prType, [out]EzVar * pPrValue
            );
        }
        //提供 Audio 操作的接口。
        [uuid(0000133b-0040-0000-C000-000000320056)]
        interface IAudio {
            HRESULT AudioIsBufferFull ([out] UINT bufferNo);
            HRESULT AudioGetFullBuffer ([in] int bankNo, [out] WORD address);
            HRESULT AudioPlaySdramStart ([in] byte channel, [in] int
               BankBytes, [in] int Numbanks, [in] int startAddr);
            HRESULT AudioGetBufferSize ([out] UINT buffersize);
            HRESULT AudioPlayFile ([in] char* fname, [in] byte channel);
            HRESULT AudioPlayFile_PcmRawData ([in] char* fname);
            HRESULT AudioRecordFile ([in] char* fname, [out] BOOL success);
        }
        //Audio 类
        [uuid(00001312-0000-0000-C000-000000000366)]
        class CAudio {
            interface IProperty;
            interface IAudio;
        }
    }
```

5.2.4 视频解码（Video）构件

Video构件的主要作用是实现视频解码的各功能要素。

```
    //Video 构件设计.
```

```
[
    version(1.0), uuid(0000132a-0300-0200-C000-000000410786),
    urn(http://www.koretide.com/repository)
]
component DtvVideo
{
    //提供获取和设置构件属性方法的接口。
    [uuid(0000130b-0000-0000-C000-000000320056)]
    interface IProperty {
        HRESULT SetProperty(
                [in]EzStr prType, [in]EzVar prValue
        );
        HRESULT GetProperty(
                [in]EzStr prType, [out]EzVar * pPrValue
        );
    }

    //提供 Video 操作的接口。
    [uuid(0000133b-0040-0000-C000-000000320056)]
    interface IVideo {
        HRESULT DisplayStart ();
        HRESULT DisplayFullImage ([in] UINT imageSize);
        HRESULT DisplaySetIref ([in] UINT mode);
        HRESULT DisplayEnableCcirLimit ([in] UINT enb);
        HRESULT DisplaySetImageBusMode ([in] UINT imageBus);
        HRESULT DisplaySetVideoFormat ([in] UINT mode);
        HRESULT DisplaySetPixelClockSpeed ([in] UINT speed);
        HRESULT DisplaySetOutputFieldOrder ([in] UINT field);
        HRESULT DisplaySetFieldLock ([in] UINT lock);
        HRESULT DisplaySetFieldPhase ([in] UINT polarity);
        HRESULT DisplaySetCsync ([in] BOOL enb, [in] UINT CSYNCH, [in]
```

```
UINT
                CSYNCV);
            HRESULT DisplaySetHblkTiming ([in] UINT offset);
            HRESULT DisplayEnableHblkVblk ([in] BOOL enb);
            HRESULT DisplaySetHblkVblkWidth ([in] UINT HBLKW, [in] UINT
                VBLKW);
            HRESULT DisplaySetVideoSubCarrier ([in] UINT reset);
            HRESULT DisplayEnableVideoYLimit ([in] BOOL enb);
            HRESULT DisplayEnablePixelClockOutput ([in] BOOL enb);
            HRESULT DisplayEnableLcdDac ([in] BOOL enb);
            HRESULT DisplayEnableVideoDac ([in] BOOL enb);
            HRESULT DisplayGo ([in] UINT Interrupt_mask);
            HRESULT DisplaySetOutputSize ([in] UINT size);
            HRESULT DisplaySetPip ([in] BOOL pipEnb, [in] UINT pipSize, [in]
                BOOL pipZoom);
            HRESULT DisplaySetBackgroundMode ([in] UINT mode, [in] UINT bgcr,
                [in] UINT bgcb);
            HRESULT DisplaySetZoom ([in] UINT direction, [in] UINT factor);
            HRESULT  DisplayBackgroundDisplay  ([in]  UINT  DisplayStartCol,
[in]
                WORD DisplayStartRow, [in] WORD DstStartCol, [in] WORD
                DstStartRow, [in] WORD width, [in] WORD height);
            HRESULT DisplaySetYuvMode ([in] UINT mode);
            HRESULT DisplayGetYuvMode ([out] UINT mode);
            HRESULT DisplaySetToXGA ([in] byte VCLK);
        }

        //Video 类
        [uuid(00001312-0000-0000-C000-000000000366)]
        class CVideo {
            interface IProperty;
```

```
            interface IVideo;
        }
    }
```

5.3 DTV 解释层的详细设计与实现

DTV解释层位于DTV基础构件库之上，它是对DTV各种功能的具体解释。这一层相对于下层的基础构件库是较灵活的一层，因为基础构件库主要与硬件层相关并隔离了硬件层的实现细节，而DTV解释层则往往体现了设计者对数字电视相应功能的理解，随着技术的进步甚至当地消费者的使用习惯不同都有可能导致这一层的相应变化。出于对这一点的考虑，下面将主要介绍比较通用的解释层构件：GUI（Graphics User Interface）构件、频道调节构件、画质调节构件、音质调节构件。

5.3.1 GUI（Graphics User Interface）构件

GUI构件为用户提供了操纵图形界面的常用方法，由于在数字电视应用程序中与用户交互的最主要方式就是通过菜单，所以这个构件中的方法也主要与菜单相关。

```
    //GUI 构件设计.
    [
        version(1.0), uuid(0000132a-0300-0200-C000-0000003007086),
        urn(http://www.koretide.com/repository)
    ]
    component DtvGui
    {
        //提供获取和设置构件属性方法的接口。
        [uuid(0000130b-0000-0000-C000-000000320056)]
        interface IProperty {
            HRESULT SetProperty(
                    [in]EzStr prType, [in]EzVar prValue
```

```
    );
    HRESULT GetProperty(
            [in]EzStr prType, [out]EzVar * pPrValue
    );
}

//提供 GUI 操作的接口。
[uuid(0000133b-0040-0000-C000-000000320056)]
interface IGui {
    HRESULT MenuGetParameterStrings ([in] byte setting, [in] byte
       value, [in] char* settingString, [in] char* valueString);
    HRESULT MenuClearValueList ([in] WORD osdNumber, [out] BOOL
       clearParamName);
    HRESULT MenuBuildTabMenu ([in] ParameterType* param, [in] WORD
       osdNumber);
    HRESULT MenuScrollDownValueList ([in] ParameterType* param,[in]
       WORD osdNumber, [in] WORD valueNum);
    HRESULT MenuScrollUpValueList ([in] ParameterType* param,[in]
       WORD osdNumber, [in] WORD valueNum);
    HRESULT MenuDisplayValues ([in] ParameterType* param,[in]byte
       startValue, [in]byte endValue, [in]byte startScreenPosition
       [in] WORD osdNumber, [in] UINT fColor, [in] UINT bColor);
    HRESULT MenuSelectValue ([in] ParameterType* param, [in] WORD
       OsdNumber, [in] WORD valueNum);
    HRESULT MenuSetValue ([in] WORD parameter, [in] WORD value);
    HRESULT MenuMode ([in] MenuIDType menuID);
    HRESULT MenuTabCtrlCreate ([in] Ctrl* t, [in] WORD osdNumber,
       [in] WORD x, [in] WORD y, [in] WORD width, [in] WORD height,
       [in]UINT bgColor);
    HRESULT MenuCtrlSelect ([in] Ctrl* t);
    HRESULT MenuCtrlClear ([in] Ctrl* t);
```

```
            HRESULT MenuClear ([in] WORD osdNumber);
        }

        //GUI 类
        [uuid(00001312-0000-0000-C000-000000000366)]
        class CGui {
            interface IProperty;
            interface IGui;
        }
    }
```

5.3.2 频道调节构件（TUNER）构件

TUNER构件为用户提供了管理数字电视频道的常用方法，主要包括频道切换和搜台等。

```
    //TUNER 构件设计.
    [
        version(1.0), uuid(0000132a-0300-0200-C000-000000378786),
        urn(http://www.koretide.com/repository)
    ]
    component DtvTuner
    {
        //提供获取和设置构件属性方法的接口。
        [uuid(0000130b-0000-0000-C000-000000320056)]
        interface IProperty {
            HRESULT SetProperty(
                    [in]EzStr prType, [in]EzVar prValue
            );
            HRESULT GetProperty(
                    [in]EzStr prType, [out]EzVar * pPrValue
            );
```

```
    }

    //提供 TUNER 操作的接口。
    [uuid(0000133b-0040-0000-C000-000000320056)]
    interface ITuner {
        HRESULT TunerGetCurrChannel ([out] WORD channelNumber);
        HRESULT TunerSetCurrChannel ([in] WORD channelNumber);
        HRESULT TunerSkipForward ([out] WORD channelNumber);
        HRESULT TunerSkipBackward ([out] WORD channelNumber);
        HRESULT TunerManSearch ([in] UINT extent, [in] WORD start, [in]
           WORD end);
        HRESULT TunerAutoSearch ([in] WORD start, [in] WORD end);
        HRESULT TunerMicroAjust ([in] UINT extent);
    }

    //TUNER 类
    [uuid(00001312-0000-0000-C000-000000000366)]
    class CTuner {
        interface IProperty;
        interface ITuner;
    }
}
```

5.3.3 画质构件（PICTURE）构件

PICTURE构件为用户提供了画质调节的常用方法，主要包括了调节亮度、对比度、色调等的函数接口。

```
//PICTURE 构件设计.
[
    version(1.0), uuid(0000132a-0300-0200-C000-000000322356),
    urn(http://www.koretide.com/repository)
```

```
]
component DtvPicture
{
    //提供获取和设置构件属性方法的接口。
    [uuid(0000130b-0000-0000-C000-000000320056)]
    interface IProperty {
        HRESULT SetProperty(
                [in]EzStr prType, [in]EzVar prValue
        );
        HRESULT GetProperty(
                [in]EzStr prType, [out]EzVar * pPrValue
        );
    }

    //提供 PICTURE 操作的接口。
    [uuid(0000133b-0040-0000-C000-000000320056)]
    interface IPicture {
        HRESULT PictureGetMode ([out] WORD mode);
        HRESULT PictureSetMode ([in] WORD mode);
        HRESULT PictureBrightnessAdjust ([in] UINT value);
        HRESULT PictureContrastAdjust ([in] UINT value);
        HRESULT PictureSaturationAdjust ([in] UINT value);
        HRESULT PictureTintAdjust ([in] UINT value);
        HRESULT PictureSharpnessAdjust ([in] UINT value);
    }

    //PICTURE 类
    [uuid(00001312-0000-0000-C000-000000000366)]
    class CPicture {
        interface IProperty;
        interface IPicture;
```

```
        }
    }
```

5.3.4 音质构件（TONE）构件

TONE构件为用户提供了音质调节的常用方法，主要包括了调节高低音、音量、环绕等的函数接口。

```
    //TONE 构件设计.
    [
        version(1.0), uuid(0000132a-0300-0200-C000-000000301096),
        urn(http://www.koretide.com/repository)
    ]
    component DtvTone
    {
        //提供获取和设置构件属性方法的接口。
        [uuid(0000130b-0000-0000-C000-000000320056)]
        interface IProperty {
            HRESULT SetProperty(
                    [in]EzStr prType, [in]EzVar prValue
            );
            HRESULT GetProperty(
                    [in]EzStr prType, [out]EzVar * pPrValue
            );
        }
        //提供 TONE 操作的接口。
        [uuid(0000133b-0040-0000-C000-000000320056)]
        interface ITone {
            HRESULT ToneGetMode ([out] WORD mode);
            HRESULT ToneSetMode ([in] WORD mode);
            HRESULT ToneVolumeAdjust ([in] UINT value);
            HRESULT ToneTrebleAdjust ([in] UINT value);
```

```
            HRESULT ToneWoofAdjust ([in] UINT value);
            HRESULT ToneBassOpen ([in] BOOL open);
            HRESULT ToneBalanceAdjust ([in] UINT value);
        }
        //TONE 类
        [uuid(00001312-0000-0000-C000-000000000366)]
        class CTone {
            interface IProperty;
            interface ITone;
        }
    }
```

5.4 DTV 中间件平台中的消息机制

5.4.1 DTV 中间件平台中的消息定义

在 DTV 中间件平台中主要定义了四大类的消息，它们基本涵盖了数字电视软件开发过程中所能遇到的消息集合，下面将分别介绍这四类消息。

（1）系统消息:这一类消息来源于DTV解释层中的“模式处理构件”，当前模式类型的状态（包括模式的有效性、模式的幅度等）、模式间的切换、屏幕的刷新等都会产生这一类消息，并由“模式处理构件”将它们发送给允许接收这些消息的其它构件或上层的应用程序。 系统消息的定义及其对应的事件如表 5.1。

表5.1 系统消息及对应事件的定义

消息定义	事件定义
	TVE_NONE
TVM_NO_SYNC	TVE_NoSync
TVM_NO_CABLE	TVE_NoCable
TVM_VALID_MODE	TVE_ValidMode

TVM_MODE_CHANGE	TVE_ModeChange
TVM_OUT_OF_RANGE	TVE_OutOfRange
TVM_SPLASH_SCREEN	TVE_SplashScreen
TVM_AUTO_ADJ	TVE_AutoAdjSuccess
TVM_SYNC_STANDBY	TVE_NoSync
TVM_SYNC_SUSPEND	TVE_NoSync

（2）键盘消息：键盘中某一按键或组合键的按下就会触发键盘事件，对于每个按键的具体解释由用户自定义的按键翻译表来决定。

（3）电源管理消息：电源管理消息主要来源于对电源状态的监测，主要包括电源的打开、关闭和待机等。电源管理消息的定义及其对应的事件如表5.2。

表5.2 电源管理消息及对应事件的定义

消息定义	事件定义
TVM_POWER_ON	TVE_PowerOn
TVM_POWER_STANDBY	TVE_NoSync
TVM_POWER_SUSPEND	TVE_NoSync
TVM_POWER_DOWN_NOTICE	TVE_RemoveOSD
TVM_POWER_DOWN	TVE_Sleep

（4）时钟消息：系统时钟在每个时钟周期都会发出一个时钟消息，这主要用于满足系统同步的需要或用在某些需要计时的场合。

5.4.2 DTV 中间件平台中的消息流分析

从硬件加电到关机，系统中的各种消息按照某种特定的轨迹流动于中间件平台的各部分构件之间，消息流是系统构件之间沟通的最主要方式，图 5.2 表示了 DTV 中间件平台中的消息流动状况。

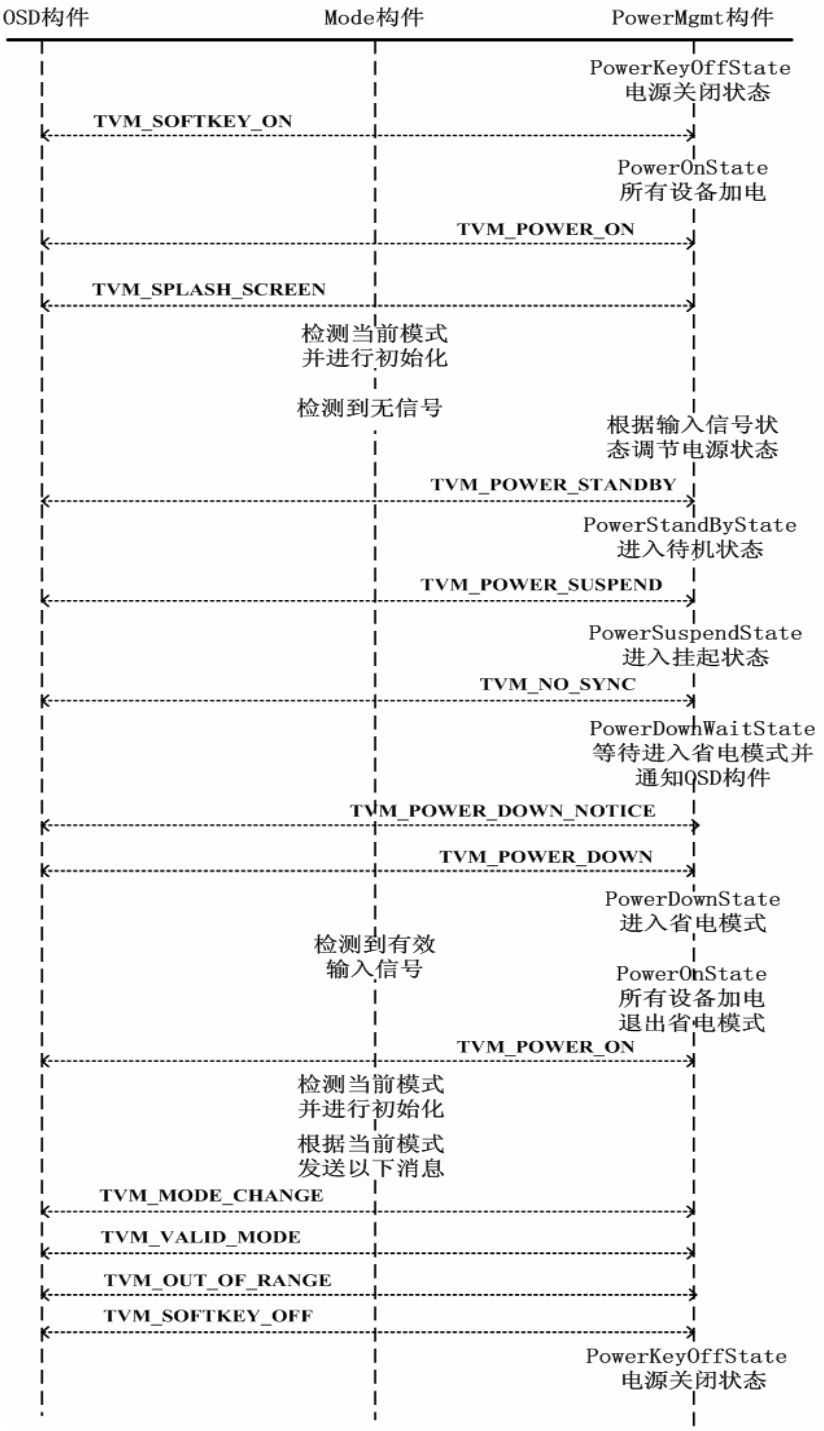


图 5.2 DTV 中间件平台的消息流分析

第 6 章 可视化开发工具——DTV Visual Developer

6.1 可视化开发方法简介

所谓可视化开发，就是指：在软件开发过程中，用直观的具有一定含义的图标按钮、图形化的对象取代原来手工的抽象的编辑、运行、浏览操作，软件开发过程表现为鼠标点击按钮和拖放图形化的对象以及指定对象的属性、行为的过程。这种可视化的编程方法易学易用，而且可以大大提高工作效率。它的特点是把原来抽象的数字、表格、功能逻辑等用直观的图形、图象的形式表现出来。可视化编程技术已经成为当今软件开发最重要的工具和手段，尤其是Power Builder、Visual C++等开发工具的出现，大大推动了可视化编程技术的发展。

传统的编程方法使用的是面向过程、按顺序进行的机制，其缺点是程序员始终要关心什么时候发生什么事情，应用程序的界面都需要程序员编写语句来实现，对于图形界面的应用程序，只有在程序运行时才能看到效果，一旦不符合需求，还需要修改程序，因而使得开发工作非常烦琐。用可视化开发工具进行应用程序开发主要有两部分工作：即设计界面和编写代码。在开发过程中所看到的界面，与程序运行时的界面基本相同，同时可视化开发工具一般还向程序员提供了若干界面设计所需要的对象（称为控件），在设计界面时，只需将所需要的控件放到窗口的指定位置即可，整个界面设计过程不需要编写代码。

可视化编程的另一个重要特点是它为用户提供了所谓的IDE（Integrated Development Environment），即集成开发环境。较早期程序设计的各个阶段都要用不同的软件来进行处理,如先用字处理软件编辑源程序,然后用链接程序进行函数、模块连接,再用编译程序进行编译,开发者必须在几种软件间来回切换，不仅操作繁琐，而且极易出错。而集成开发环境则将编辑、编译、调试等功能集成在一个桌面环境中,开发者所有的工作都可以在这个环境下一次完成,这样就大大方便了用户。

6.2 可视化开发方法在 DTV 软件开发中的应用

数字电视软件开发与传统的 PC 机上的软件开发既有相似之处也有其自身的特点。它们的共同点主要有：

（1）与用户交互的方式都是基于图形界面，用户通过对图形界面中的相关控件的操作对应用程序发出指令。

（2）图形界面中的控件都应该能够响应事件，事件可以由用户操作触发，也可以由来自系统的消息触发。

（3）应用程序没有预定的执行路径，而是在响应不同的事件时执行不同的代码片段。

数字电视软件开发由于受到硬件资源的限制，相对于基于PC机的软件开发又有自身的特点：

（1）用户界面相对简单，主要的表现形式是菜单，图形界面中的控件也只包括按键和滚动条等最基本的元素。

（2）用户主要通过遥控器来发出指令并由此触发系统中相应的事件，这一过程与PC机上通过鼠标和键盘来进行输入操作具有很大的不同，输入手段的单一性和简单性决定了用户界面必须具有易于操作和导航功能强大的特点。

（3）考虑到系统资源对软件运行的限制，用户发出事件后，若处理器负载较重，可以允许响应有相应的延迟。

综上所述，在数字电视软件开发中引入可视化开发方法是可行的，尤其是随着数字电视芯片性能的不断提升以及对数字电视应用软件需求的不断攀升，可视化编程将是大势所趋，这也是提高当前数字电视软件开发效率的重要手段之一。

6.3 DTV Visual Developer 的体系结构设计

DTV Visual Developer 是建立在 DTV 中间件平台上的可视化开发工具，它根据数字电视软件开发的特点，把界面设计中使用频繁的对象封装为控件，以供用户方便的调用。在开发过程中所看到的用户界面框架，与程序运行时的界面基本相同，这就使得软件的设计阶段可以在脱离真机的情况下完成，而不必把大量的时间花在重复下载程序到芯片以及频繁开机调试的过程中，从而大大的提高软件开发的效率。下图是 DTV Visual Developer 的主应用程序界面：

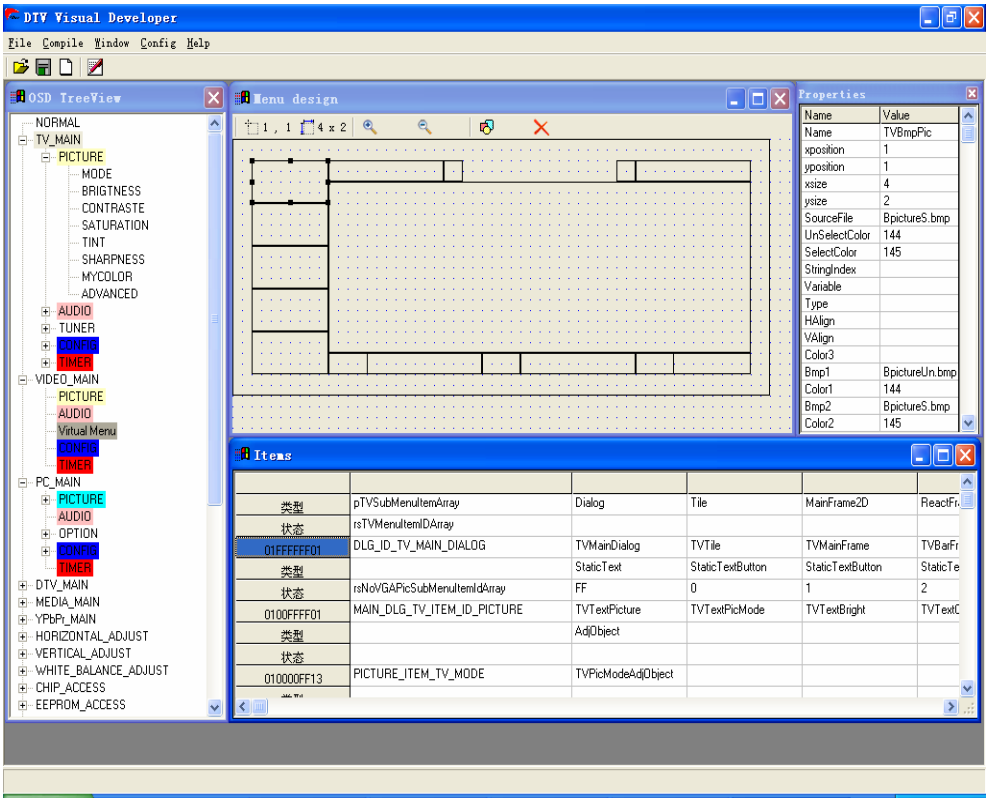


图 6.1 DTV Visual Developer 的主应用程序界面

6.3.1 DTV Visual Developer 的系统构成

DTV Visual Developer 主要由四个工作区组成：

（1）OSD TreeView：是编辑菜单状态的主要场所，菜单状态采用树状结构表示，父菜单与子菜单的关系通过父节点与子节点的关系来模拟。

（2）Menu Designer：是用户界面设计的工作区， 在这里设计的用户界面框架与程序运行时的图形界面基本相同，它提供了添加、删除、以及调整控件形状和位置的功能。

（3）Items Window： 其中的每一项都对应于一个菜单的状态，它包含了这个菜单状态下的所有控件，无论该控件是可见的或是不可见的。

（4）Properties Window：在这里可以编辑选中控件的各种属性和行为。

此外，DTV Visual Developer 还提供了诸如变量表、字符串表和颜色表等

方便用户使用的功能。同时，这个集成开发环境也是可配置的，例如控件的种类，每种控件的属性以及程序中使用的字库等都是可以根据用户的需要来定制的，这就保证了对开发工具的灵活性和适应性的要求。

6.4 基于 DTV Visual Developer 的数字电视软件开发方法

在使用可视化开发工具 DTV Visual Developer 的数字电视软件开发过程中，所有的 DTV 应用程序都将抽象成为“控件集合”＋“资源集合”的模式，图 6.2 解释了这种开发方法。

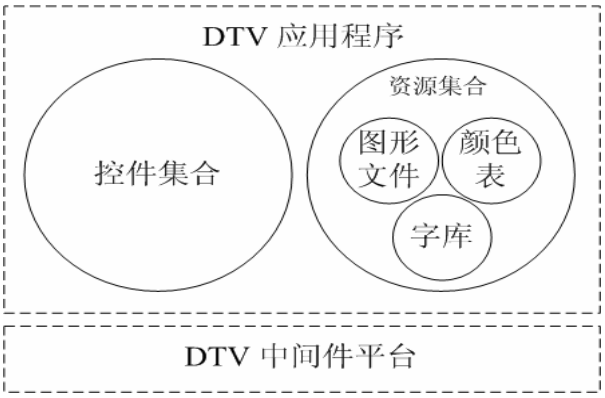


图 6.2 使用 DTV Visual Developer 开发的 DTV 应用程序结构图

这种开发模式的应用使得软件功能的实现将不再以各个功能模块的实现为标志，而是按照“创建用户界面——设置控件对象的属性——编写事件响应代码”这样的流程来完成，这样做的结果不仅极大的提高了软件开发的效率，同时也使得所开发的应用程序逻辑性更强也更易于维护。

第 7 章 结论与展望

本文结合国家863课题“基于构件、中间件技术的因特网操作系统及跨操作系统的构件、中间件运行平台”，首先对当前数字电视软件开发的现状进行了分析，然后提出了一种基于“和欣”嵌入式操作系统的数字电视解决方案，即基于数字电视软件开发平台的全新开发模式。文中具体阐述了数字电视软件开发平台的系统结构和构成特点，并给出了DTV中间件层的详细设计方案，最后简要分析了基于中间件平台的可视化开发工具——DTV Visual Developer。

本文提出的数字电视软件开发平台可以很好的解决多媒体应用的跨平台问题，应用程序能够不经任何修改，甚至不必重新编译、链接就可以运行在不同硬件厂商提供的数字电视平台上,从而可以做到“一次编写,多处运行”，这对提高数字电视软件开发的效率，增强软件产品的开放性以及提高软件产品的可维护性都具有一定的理论和实践意义。

数字电视技术已被国家列入“十五计划”中的十大优先发展的信息技术产业，2005年我国将进行数字电视的商业播出，2008年用数字电视转播奥运会，2015年停止模拟电视的播放，全面推行数字电视，这一时间表，为像我国这样拥有3亿多电视用户的国家来说，是巨大的发展机会。同时，随着中国数字电视产业标准的即将推出，使得数字电视真正成为不同厂家都可参与竞争的一个产业，国外公司的垄断会被彻底打破，这无疑是运营商所要求的，同时也是数字电视生产商所期待的。因此，我国有关部门应该努力加大对数字电视软件开发技术的研究力度，使得我们的数字电视产业能够后来居上，成为信息产业中的新亮点。