

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 17/00 (2006.01)

G06F 9/45 (2006.01)



[12] 发明专利说明书

专利号 ZL 03146664.8

[45] 授权公告日 2008 年 1 月 9 日

[11] 授权公告号 CN 100361121C

[22] 申请日 2003.7.11 [21] 申请号 03146664.8

[73] 专利权人 北京直真节点技术开发有限公司
地址 100080 北京市海淀区北四环西路 9
号银谷大厦 1506 室

[72] 发明人 金建林 马克 袁隽 胡明

[56] 参考文献

WO03/009071A1 2003.1.30

CN1154776A 1997.7.16

CN1302413A 2001.7.4

审查员 尹春梅

[74] 专利代理机构 北京同达信恒知识产权代理有限公司
代理人 郭润湘

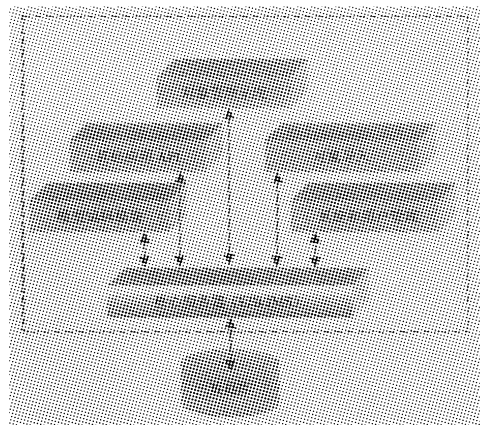
权利要求书 3 页 说明书 15 页 附图 5 页

[54] 发明名称

一种通用对象建模方法及通用对象管理系统

[57] 摘要

本发明有关一种通用对象建模方法及通用对象管理系统，该方法包括采用通用对象模型、特殊对象模型和应用业务逻辑三层对象模型来建立被管对象的描述和应用的扩展，所述通用对象模型层描述和定义一般对象的属性和呈现方式，将所有被管对象采用统一标准的格式来描述，形成有序组织的对象；特殊对象模型层定义对象和对象之间的相互关系，以及对象特定的应用含义，从而形成特定技术领域的对象模型描述；应用业务逻辑层定义基于特定对象模型的特定操作或方法，并与相关的应用逻辑相结合。本发明通过对通用对象模型层的描述，可以建立所有被管理对象的模型，适应对多技术领域、多专业领域管理对象的管理。



1、一种通用对象建模方法，其特征在于包括：采用通用对象模型层、特殊对象模型层和应用业务逻辑层三层对象模型来建立被管对象的描述和应用的扩展；所述通用对象模型层描述和定义一般对象的属性和对象呈现方式，将所有被管对象采用统一标准的格式来描述，形成有序组织的对象；所述特殊对象模型层在通用对象模型层的基础上进一步定义对象和对象之间的相互关系，以及对象特定的应用含义，形成特定技术领域的特定对象模型；所述特殊对象模型层中通过定义不同对象之间不同的相互关系，生成多个对象管理域；所述应用业务逻辑层在所述特殊对象模型层的基础上进一步定义基于特定对象模型的特定操作或方法以及与操作或方法相关的应用逻辑。

2、如权利要求1所述的通用对象建模方法，其特征在于：所述一般对象的属性包括基本属性、传递属性、计算属性和对象属性。

3、如权利要求1所述的通用对象建模方法，其特征在于：所述对象呈现方式包括图标呈现或线呈现；

所述图标呈现方式包括图标文字、图标颜色、图标闪烁、图标隐藏和图标提示五者之一，或其中任意两者、三者、四者或全部的组合；

所述线呈现方式包括线颜色、线宽度、线闪烁、线型和线隐藏五者之一，或其中任意两者、三者、四者或全部的组合。

4、如权利要求1所述的通用对象建模方法，其特征在于：所述特定对象模型的描述方式包括：文件模型描述方式和库表模型描述方式；还包括：所述文件模型描述方式和库表模型描述方式相互转换。

5、如权利要求1所述的通用对象建模方法，其特征在于：所述对象之间的相互关系包括包含、关联、承载或引用关系。

6、如权利要求1所述的通用对象建模方法，其特征在于：所述被管对象在不同的对象管理域中生成相同或不同的界面呈现方式，所述界面呈现方式包括拓扑图、树图、属性图或列表图。

7、一种通用对象管理系统，其特征在于包括：模型编辑模块、核心服务器模块、对象导入模块、其他应用接口模块、对象管理呈现模块以及系统和用户管理模块；

所述模型编辑模块实现建立被管对象模型，并提供对所建模型的编辑、存储以及与所述核心服务器模块交互，实现对所建模型的导入功能；

所述模型编辑模块建立被管对象模型的具体方法包括：采用通用对象模型层、特殊对象模型层和应用业务逻辑层三层对象模型来建立被管对象的描述和应用的扩展；所述通用对象模型层描述和定义一般对象的属性和对象呈现方式，将所有被管对象采用统一标准的格式来描述，形成有序组织的对象；所述特殊对象模型层在通用对象模型层的基础上进一步定义对象和对象之间的相互关系，以及对象特定的应用含义，形成特定技术领域的特定对象模型；所述特殊对象模型层中通过定义不同对象之间不同的相互关系，生成多个对象管理域；所述应用业务逻辑层在所述特殊对象模型层的基础上进一步定义基于特定对象模型的特定操作或方法以及与操作或方法相关的应用逻辑；

所述核心服务器模块用于实现模型访问、对象管理和导入操作，并提供对象数据维护和内存对象维护功能；

所述对象导入模块提供对象导入规则定义以及与所述核心服务器模块交互，实现对象导入的功能；

所述其他应用接口模块提供用户开发特定的应用相关的接口库，其它应用程序通过该接口库来访问存储在该系统中的对象和模型；

所述对象管理呈现模块提供对管理对象按照对象呈现规则进行呈现的功能，提供拓扑图、树图、属性图或列表图来展现对象；

所述系统和用户管理模块实现对通用对象管理系统的监控、图标管理、组织机构管理、权限和用户管理，并实现整个通用对象管理系统和架构在该通用对象管理系统上的特定应用的权限控制。

8、如权利要求7所述的通用对象管理系统，其特征在于：所述被管对象

模型包括对象基本模型、对象关系模型、对象呈现模型和对象操作模型。

9、如权利要求8所述的通用对象管理系统，其特征在于：所述接口库中包括有客户端应用程序接口，通过该接口，实现访问另一通用对象管理系统的对象和模型数据，实现多套通用对象管理系统之间数据信息的透明访问。

一种通用对象建模方法及通用对象管理系统

技术领域

本发明涉及对象建模及对象管理，尤指一种不局限于某一具体应用领域的通用对象建模方法及其管理系统。

背景技术

面向对象技术和理论的出现对于计算机领域乃至整个制造业都引起了轩然大波，通过对客观世界的真实抽象，采用面向对象语言来描述对象本身的属性、操作和展现以及对象与相关对象之间的关系，使对象描述变得更加简单，模型清晰而易于维护。人们越来越深刻的认识到“对象”是真正能够描述客观世界一切存在实体的最简单的方式，更多的人都投入来这个研究中来，面向对象技术也得到的飞速的发展，各种面向对象的语言也应运而生，如UML，XML、IDL（CORBA、TINA）等，应用这些工具可以很快的建立各种对象模型。

现有技术中，电信领域作为一个新兴的高科技领域，也吸收了面向对象技术的优点，在其标准化的进程中，形成了适合于电信领域的通用对象建模技术，如对管理信息的描述的ASN.1、对电信管理对象的描述的M.3100、GDMO、GNM等，推动着电信业的发展。当然，这些标准为了能够全面的描述所有的管理对象，同时为了能够和以前的标准有所融合，内部结构比较复杂，给使用者造成了一定的难度。M.3100采用GDMO和ASN.1来描述对象的结构和意义，它将网络抽象为CIM通用信息模型，见图1，网络由网元和网元之间的连接组成，网络可以包括子网络，网元是由多个构件组成的，构件可以包含多个子构件，所有的组成部分都可以通过对象的形式进行描述。在该信息模型的基础上，形成了许多网络管理系统，图2是基于M.3100的CIM模型

形成的网络管理系统的基本结构。一个网络管理系统基本上由两个层次组成，底层是网络资源信息模型，上层是网管应用层，下层是网络资源管理系统，通过GDMO和ASN.1来抽象的描述网络的信息模型，通过定义对象的各种属性信息来描述对象自身，并通过定义对象之间的包含、关联、承载、引用等来描述对象之间的各种关系，上层是网管的各种应用系统，如故障管理、性能管理、路由管理、话务管理、业务管理等各种业务的应用。上层的各种应用依赖于下层网络信息模型，而下层网络信息模型是一个庞大的复杂模型，建立和维护的代价非常大。

现有技术一的缺点：

1、它建立了一个特定网络的信息模型，只使用于对电信网络的描述，模型本身比较复杂，同时不容易实现对管理的信息模型的扩展，如不能实现对组织机构的描述。

2、随着INTERNET技术的发展，信息交互问题已经不仅仅局限于电信网络的内部，需要广域网上实现和其他多个系统之间信息的交互，所以对信息描述方式必须采用通用、易懂且灵活扩展的语言来进行描述，而无论是GDMO，还是ASN.1都只是电信领域的标准，而且十分复杂，需要专业人员才能够理解。

3、系统的扩展性差，主要表现在扩展新的功能的时候需要重新构造网络模型，应用和模型紧密相关。

4、不能提供一个完全可视化的对象建模平台，需要通过程序来实现对实际对象的模型创建。

发明内容

本发明提供一种通用对象建模方法及通用对象管理系统，采用多层模型结构来描述通用模型和特定对象模型，适应对多技术领域、多专业领域管理对象的管理。

本发明的通用对象建模方法，包括：采用通用对象模型层、特殊对象模型层和应用业务逻辑层三层对象模型来建立被管对象的描述和应用的扩展；所述通用对象模型层描述和定义一般对象的属性和对象呈现方式，将所有被管对象采用统一标准的格式来描述，形成有序组织的对象；所述特殊对象模型层在通用对象模型层的基础上进一步定义对象和对象之间的相互关系，以及对象特定的应用含义，形成特定技术领域的特定对象模型；所述特殊对象模型层中通过定义不同对象之间不同的相互关系，生成多个对象管理域；所述应用业务逻辑层在所述特殊对象模型层的基础上进一步定义基于特定对象模型的特定操作或方法以及与操作或方法相关的应用逻辑。

所述一般对象的属性包括基本属性、传递属性、计算属性和对象属性。

所述对象呈现方式包括图标呈现或线呈现；所述图标呈现方式包括图标文字、图标颜色、图标闪烁、图标隐藏和图标提示五者之一，或其中任意两者、三者、四者或全部的组合；所述线呈现方式包括线颜色、线宽度、线闪烁、线型和线隐藏五者之一，或其中任意两者、三者、四者或全部的组合。

所述特定对象模型描述方式包括：文件模型描述方式和库表模型描述方式；还包括：所述文件模型描述方式和库表模型描述方式相互转换。

所述对象之间的相互关系包括包含、关联、承载或引用关系。

所述被管对象在不同的对象管理域中生成相同或不同的界面呈现方式，所述界面呈现方式包括拓扑图、树图、属性图或列表图。

本发明还提供一种采用上述通用对象建模方法的通用对象管理系统，包括：模型编辑模块、核心服务器模块、对象导入模块、其他应用接口模块、对象管理呈现模块以及系统和用户管理模块；

所述模型编辑模块实现建立被管对象模型，并提供对所建模型的编辑、存储以及与所述核心服务器模块交互，实现对所建模型的导入功能；所述模型编辑模块建立被管对象模型的具体方法包括：采用通用对象模型层、特殊对象模型层和应用业务逻辑层三层对象模型来建立被管对象的描述和应用的

扩展；所述通用对象模型层描述和定义一般对象的属性和对象呈现方式，将所有被管对象采用统一标准的格式来描述，形成有序组织的对象；所述特殊对象模型层在通用对象模型层的基础上进一步定义对象和对象之间的相互关系，以及对象特定的应用含义，形成特定技术领域的特定对象模型；所述特殊对象模型层中通过定义不同对象之间不同的相互关系，生成多个对象管理域；所述应用业务逻辑层在所述特殊对象模型层的基础上进一步定义基于特定对象模型的特定操作或方法以及与操作或方法相关的应用逻辑；

所述核心服务器模块用于实现模型访问、对象管理和导入操作，并提供对象数据维护和内存对象维护功能；

所述对象导入模块提供对象导入规则定义以及与所述核心服务器模块交互，实现对象导入的功能；

所述其他应用接口模块提供用户开发特定的应用相关的接口库，其它应用程序通过该接口库来访问存储在该系统中的对象和模型；

所述对象管理呈现模块提供对管理对象按照对象呈现规则进行呈现的功能，提供拓扑图、树图、属性图或列表图来展现对象；

所述系统和用户管理模块实现对通用对象管理系统的监控、图标管理、组织机构管理、权限和用户管理，并实现整个通用对象管理系统和架构在该通用对象管理系统上的特定应用的权限控制。

所述被管对象模型包括对象基本模型、对象关系模型、对象呈现模型和对象操作模型。所述接口库中包括有客户端应用程序接口，通过该接口，实现访问另一通用对象管理系统的对象和模型数据，实现多套通用对象管理系统之间数据信息的透明访问。

本发明的优点在于：

1、灵活的管理对象模型建立；通过对通用对象模型层的描述，可以建立所有被管理对象的模型，适应对多技术领域、多专业领域管理对象的管理。如在网络管理中，可以建立传输网、传送网的网络模型，实现跨网网络管理。

2、快速的部署和无“割接”的系统扩展；提供继承、包含等多种定义方法通过建模可以快速的建立被管理对象的信息模型，同时动态模型的加载技术的使用可以保证系统在无“割接”情况下实现功能的快速扩展。

3、多系统对话成为可能；系统采用流行的XML（扩展标记语言）语言作为对象描述的语言，XML具有标准、通俗易懂等特点，而且可以实现和广域网的信息的交互；另一方面，对象管理系统自身的分布特性和标准接口也保证了对系统信息交互的可行性。

4、“软插件”技术的采用使得通过对象模型扩展可以方便的扩展对象管理的应用逻辑。

附图说明

图 1 为电信网络中的 CIM 信息模型。

图 2 为现有电信网管系统结构示意图。

图 3 为本发明通用对象建模方法的模型层结构示意图。

图 4 为本发明的通用对象管理系统模块结构示意图。

图 5 为本发明的通用对象管理系统内部结构控制示意图。

图 6 为架构在本发明系统上的一个网络管理方案。

图 7 为采用类的继承树进行模型编辑的界面图。

图 8 为对象管理域开发界面图。

图 9 为对象方法及对象呈现开发界面图。

图 10 为本发明系统多套联合应用时的控制方框图。

具体实施方式

本发明的通用对象建模方法（简称为 AMO）是采用通用对象模型、特殊对象模型和特定的应用逻辑三层对象模型来建立被管对象的描述和应用的扩

展,如图3所示,最底层是通用对象模型层(Generic Object Model)层,通用对象模型层描述和定义了一般对象的属性和呈现方式,将所有管理对象采用统一标准的格式来描述,形成有序组织的对象;第二层是特殊对象模型层(Special Object Model),特殊对象模型层定义对象和对象之间的各种关系(如类之间的包含关系、引用关系等),以及对象特定的应用含义,从而形成特定技术领域的对象模型描述;第三层是应用业务逻辑层(Application Logic Procedure),定义基于特定对象模型的特定操作或方法,并与相关的应用逻辑结合起来,如针对一个网络设备可以查看设备的性能指标,针对一个人员可以查看人员相关的联系信息、绩效等。

基于上述通用对象建模方法,本发明提供一种通用对象管理系统(OMP),在该系统之上运用上述建模方法以及特定应用逻辑的开发后,可快速实现新应用的开发,并可非常容易地支持架构在该OMP系统之上的各个应用的集成以及相互之间的信息交换;该系统是一个应用解决方案的支撑平台,它提供统一应用集成的用户界面、中间业务处理逻辑以及后台数据库,完成对被管理对象的通用管理和呈现管理功能,提供了实际应用系统中的数据组织、数据访问接口以及大部分应用的呈现界面。

该OMP系统的模块结构图如图4所示,主要包括以下六个模块组成,即:核心服务器模块、模型编辑模块、系统和用户管理模块、对象导入模块、其它应用接口模块(API)和对象管理呈现模块。

核心服务器模块主要提供用于模型访问、对象管理、导入等操作需要的服务器端软件,并提供对象数据维护和内存对象维护功能,以提高系统的访问速度。

模型编辑模块主要提供对被管理对象的建模功能,并提供对建立好的模型的编辑、存储和导入功能,在导入过程中提供模型的语法检查和与原有模型的比较功能。

系统和用户管理模块主要提供OMP系统的监控、图标管理、组织机构、

权限和用户管理功能，实现整个系统和架构在其上的特定应用的权限的控制。

对象导入模块主要提供对象导入规则定义和对象导入功能。

其他应用API提供用户开发特定的应用相关的接口库，其他应用程序可以通过该接口库来访问存储在OMP中的对象和模型。

对象呈现模块提供对管理对象按照其呈现规则进行呈现的功能，提供拓扑图、树图、属性图、列表图等多种形式来展现对象，同时也是应用集成的界面，用户可以将功能集成到界面的菜单和对象的浮动菜单上，实现功能扩展。

通过上述模块形成了OMP的核心功能，其基本工作过程如下：

1、建立被管对象模型：包括对象基本模型（Object Model）、对象关系模型（Special Object Model）、对象呈现模型（Special Object Model）和对象操作模型（Application Model），这些模型的建立由模型编辑模块实现。

2、将模型加载到系统中：由模型编辑模块和核心服务器模块共同实现按照上述模型动态生成数据库表结构。

3、对象导入：将已有的数据导入到系统中，由对象导入模块和核心服务器模块共同实现。

4、建立系统权限体系：建立组织机构、用户权限，由系统用户管理模块和核心服务器模块共同实现。

5、使用系统：主要由对象管理呈现模块和核心服务器模块实现。

6、扩展系统功能：增加新的应用逻辑开发，采用其它应用程序接口（API）进行开发。

7、修改系统模型：由模型编辑模块实现模型修改，并将新的应用插入系统中，而系统正常运行，然后回到第2步，进入增强功能的管理系统。

以上阐述了OMP系统的模块结构和其工作过程，下面详细描述系统的体系结构和如何根据已有的API来开发新的应用并集成到系统中来。

参见图5，为本发明的OMP系统内部结构控制示意图。图中模型编辑（应用）、对象管理与呈现（应用）、授权管理（应用）与对象导入（应用）为

上层应用可直接使用或者利用的用户界面，模型访问（API）、权限验证（API）和对象管理API为上层应用可以直接访问调用的API，其中模型转换（API）和权限访问（API）对上层应用完全透明，四周框线为消息总线。其中的逻辑层在应用服务器中运行。

在图中显示了几个集成点，在对象管理 API 处：上层应用可以添加在对象创建、删除或者改变前/后需要的附加逻辑判断，这样的逻辑判断可以用来实现对象之间的完整性检查；另外，在应用服务器中还可以配置对收到的各种消息的上层应用的处理动作（也可以没有处理动作）。在对象管理与呈现处：可以通过配置文件决定要进行集成的各个应用系统以及集成时各个系统初始化所要执行的动作；对于每个被集成的架构在 OMP 之上的应用，可以通过配置文件决定该应用要在主界面上出现的菜单，通过模型确定各个应用在各个对象上可出现的右键菜单，以及用户在点取菜单后系统将如何响应；同时，在该处有一个独立的消息处理模块，各个上层应用需要接收的消息以及收到消息后对消息的处理也可以通过配置文件进行挂接。

本发明按照面向对象的思想，将要管理的数据进行抽象，形成类（实例化后的对象即为我们实际的要管理的数据）。传统的对象描述包括了对象的状态（属性）、对象可执行的动作（方法），而在OMP中，除了上述内容之外，还对对象在不同管理域（方案）如何呈现（显示风格）进行描述。OMP中对类的描述方式如下：

一、属性

属性用于记录OMP对象的内部状态，并可规定在进行对象属性呈现时，对属性如何分组显示（在OMP中对应为属性组的概念），以及各个属性是否需要显示（不需要显示表示用户对对象的相应状态的取值不关心）。OMP中属性类型如下表所示：

大类型	小类型	备注
基本属性	整数	可以是数组，

	实数	
	字符串	
	枚举	
	日期	
	布尔	
传递属性	实数	与不同的管理域有关，支持四种传递规则： 最大值/最小值/总和自定义传递方式
计算属性	整数	其中的变量只能来自于本对象的基本属性， 利用该类型的属性可以对对象自身进行简单 的分析
	实数	
	字符串	
	布尔	
对象属性	OMP类	可以是数组，利用它可以实现任意复杂的数据表示

二、方案

方案用于描述在不同的对象管理域内对象之间可以存在的关系。在现实事件中，从不同的视角观察同一个对象，会发现其与相关的对象的集合是不一样的，彼此之间的相互关联关系也是不同的。如一台具体的路由器，从生产角度来说，会与一个特定的供应商发生联系；从使用角度来说，会与运营商发生联系。这种现实世界中错综复杂的联系，仅用对象自身的属性是无法完全描述清楚的，故在 OMP 中提出了对象管理域——方案的概念，利用方案来描述这种关系。图 6 表明了在一个具体的架构在 OMP 之上的业务开通系统应用中的“网络对象拓扑管理域”方案。

方案中表明：网络可能由其他网络(子网或者网络适配器，从网络继承而来)组成，在网络中存在用户可以接入的端口(用户终端)集合，子网/网络适配器之间通过链路连接。链路作为线显示对象，也就是用于与其它对象的连接。

在模型中建立方案的时候，如果类 A 对象可以包含类 B 对象，可以指定 A 对象与 B 对象的属性之间满足何种条件时，两个对象之间的包含关系即可

存在。如果是线显示对象，还可以指定起点对象所满足的条件以及终点对象所满足的条件。

三、方法

方法代表对象可以执行的动作。对象在执行动作时，需要外部进行触发。如房间中的电灯的开/关动作，是需要人去触发的。同一个对象可以执行的动作可以很多，但是不同的管理域中，用户可以触发的动作可能只是对象所有动作中的一个子集。基于这样一个事实，OMP中在建立模型时，在为对象定义方法时，要求指明可以触发该方法执行的对象管理域——方案。

在实际的使用中，OMP是无法预知对象的方法集合的，同样也不可能决定一个特定方法应该如何实现，如何引起对象的状态（属性）发生变化。建模中所定义的方法，只是定义了对象的方法接口，方法的具体实现需要上层应用做二次开发，在二次开发时，要遵循OMP的接口要求与规范。

本发明的OMP系统提供面向对象进行模型抽象和开发，如图7所示，为采用类的继承树进行模型编辑的界面图，子节点均从父节点继承而来，右边表示在左边继承树上被选中类的各个属性（有的属性是从父类继承而来的，有的属性是对父类的扩展，是本子类所特有的）。图8为对象管理域开发界面图，图中显示了可以在同一个模型中建立多个显示方案，在每个方案中可以对类之间的包含关系与引用关系进行描述，通过该种开发界面只需要简单的鼠标拖动即可进行方案的开发。图9为对象方法及对象呈现开发界面图，由图可知，本发明系统可对对象所采用的方法及对象在不同方案中如何呈现进行统一定制。

由以上的描述可知，本发明的方法及系统，通过进行模型抽象，使得在进行应用开发时，对象的抽象——类的描述不再需要进行程序开发，而只需要利用本发明OMP系统中的图形化的建模工具，即可快速地对类进行描述。这样，在类需要修改的时候，不需要编码，更不需要系统的重新编译、连接，只需要对模型进行修改后导入到我们的系统当中，即可立即生效，快速地实

现业务模型的修改。

本发明系统中对模型的描述，采用了两种方式：文件模型描述方式与库表模型描述方式。文件模型描述方式主要是提供给模型开发人员进行开发，库表模型描述方式主要是为了对对象管理的日常运行提供支持，通过系统提供的模型转换工具，两种描述方式可以相互转换。

在传统的面向对象管理中，对象的呈现通常都是需要开发人员进行编码来表现的，这占据了应用开发相当大的工作量。在OMP中，提供了统一的用户界面，大部分的对象呈现工作已经由系统实现。同时OMP系统所提供的用户界面充分考虑了应用需要进行特殊界面整合的需要，提供了用户界面集成的框架，实现用户界面的可查拔功能。

本发明的OMP系统中，同一对象在不同的管理域中可以有不同的呈现形式，这样的呈现形式不是通过上层应用编写程序来实现的，而只需要在OMP的建模工具中通过定制即可实现（参见图9所示）。目前可定制呈现风格如下表所示：

呈现方式	呈现风格	备注
图标呈现	图标文字	将显示在图标旁边的文字，将与属性关联
	图标颜色	图标的背景颜色，将根据指定属性的取值不同背景颜色可以不同
	图标闪烁	指定图标的闪烁频率，将根据指定属性的取值不同闪烁频率可以不同
	图标隐藏	指定图标显示对象将不显示的条件
	图标Tip	将显示在图标旁的ToolTip中的文字，将与一个给定的属性关联
线呈现	线颜色	根据指定属性的取值不同线色可以不同
	线宽度	根据指定属性的取值不同线宽可以不同
	线闪烁	根据指定属性的取值不同线闪烁的频率可以不同
	线型	根据指定属性的取值不同线型可以不同

	线隐藏	指定线显示对象将不显示的条件
--	-----	----------------

本发明系统实现了主菜单与右键菜单的插拔机制：右键菜单实际上代表了最终用户通过对象可执行的动作。一个对象可以执行的动作可以有多个，但是在不同的管理域，用户可执行的动作可能不一样，OMP系统通过模型定制来实现右键菜单，并且支持这种不同。由于右键菜单是通过模型定制的，所以右键菜单可以很方便地进行添加、修改或者删除，实现查拔。系统的主菜单可以通过配置文件进行配置，在配置文件中，可以指明菜单显示的名称，以及要执行的动作，OMP系统会自动地根据配置文件的内容，决定菜单将如何执行。只要配置文件改变，菜单执行的动作也就改变了，实现了动态查拔机制。

本发明的OMP系统，对于每个需要执行动作的菜单项，点取菜单可能执行的动作分为两类：一类将导致另一个应用（进程）的启动；另一类需要提供一个入口的类名以及方法名，它将与OMP的呈现界面处于同一个进程中，可以访问到呈现界面应用的内部状态。

主菜单的配置文件的格式如下所示：

```
<?xml version="1.0" encoding="GB2312"?>
  -<DynamicMenuBar>
    +<Menu name="配置管理" short_desc="" long_desc="" mnemonic_dey="">
      -<Menu name="告警管理" short_desc="" long_desc="" mnemonic_dey="">
        +<MenuItem name="历史告警查询">
          -<Menu name="查看实时告警" short_desc="" long_desc=""
            mnemonic_key="">
            -<MenuItem name="列表图">
              -<action type ="1">
                <classname name="map.omp.ShowAlarmTable"/>
                -< invokemethod name ="showAlarmTable">
                  <jarg javaclass ="java.util.list"
```



```
        object="CONTEXT_SELECTED_OBJECTS"/>
    </invokemethod>
</action>
</MenuItem>
+<MenuItem name="鸟瞰图">
</Menu>
+</Menu name="告警系统管理">
+</Menu name="告警呈现设置">
</Menu>
+</Menu name="性能管理">
+</Menu name="流量管理">
    </Menu name="路由管理"/>
+</Menu name="报表管理">
</DynamicMenuBar>
```

本发明实现了树状呈现的任意定制：在 OMP 系统中，不同对象管理域中对象之间的关系是通过层次结构来实现的。这种实现方式，为树的构造提供了极为便利的条件。只要对象之间层次关系在各个对象管理域中已经构造好，创建一棵树只需要指定一个对象管理域以及树的根节点对象，OMP 呈现平台就可以通过对象之间的层次结构构造出一棵树来。通过这种方式，可以构件任意多棵树，不同的使用者关注不同的树，便于实现访问权限的控制。

OMP 系统提供了对象管理呈现模块，用户在进行二次开发时，可以以此模块作为呈现的起点（也可以不要这个起点而完全自己重新开发），加入自己所特有的用户界面与呈现逻辑。一对象呈现界面图，左边为树图区域，用于展现登陆用户有权限看到的树；右上角为拓扑图与列表图显示区域，用于显示左边树图上所选中对象在相应的对象管理域（方案）中其所包含的子对象相互之间的连接关系（在列表图中为简单的对象罗列）；右下角为对象的属性图，用于显示在拓扑图或者列表图中被选中对象的各种属性，属性按照模型建立时的分组办法进行呈现。在界面上，系统利用图标、颜色等各种办法，按照建模时候定义的规则进行最直观的对象呈现。

本发明的 OMP 系统提供了对对象数据访问的一套完整的 API，包括对象 API 与模型 API，系统中的界面集成框架也是架构在这一套 API 的基础之上的。当系统应用于分布式环境中时（即多套 OMP 联合使用时），不同 OMP 上的数据访问是透明的。通过这套 API，可以访问全部的 OMP 对象数据，并可实现增删改查等功能。利用该系统的不同应用在进行集成时，由于采用的是同一基础平台，故可以实现无缝集成。这里的无缝集成包括后台数据访问以及前台的界面呈现，因为界面也是统一的集成框架。

在 OMP 系统中，利用 JMS（Java Message Service，Java 消息服务），架构了自己的消息总线。在使用的时候，只需要提供一个配置文件，告诉系统在收到何种消息的时候，要执行何种用户自定义的动作。这样，在二次开发时，只需要提供一个配置文件说明对消息的处理动作，系统即可自动地处理各种消息，而不再需要实现对消息的接收处理编码。

为了支持上层应用的集成，尤其是安全的集成，OMP 提供了系统和用户管理模块实现可配置的权限管理。通过这种方式，提供了统一的授权界面（被集成的各个系统不再需要自己的授权界面），同时提供了权限验证的统一 API，这样各个系统在进行资源访问前，只需要调用 OMP 所提供的权限 API 进行验证，而无需关心权限数据内部如何组织，以及如何通过操作界面进行授权等重复性的工作。

本发明的系统可多套联合实用于大型分布式场合，多套主要是指系统体系三层结构中逻辑层以及数据层有多套（参见图 5），一个集成应用的客户端只需要进行一次登录，即可自动地访问到它所需要的位于各个不同服务器上的正确信息。这个实现，主要是依靠调用 OMP 提供的一套完整的对象访问客户端应用程序接口（Client API）来实现透明访问，最终用户只需要正确地配置各套系统并指明对象数据之间的指向关系，Client API 即可正确地实现对分布式信息访问透明。也就是说，集成应用在进行二次开发的时候，不需要考虑系统的分布式特性，这部分内容已经完全被 OMP Client API 所封装；同样，

集成应用开发者不需要掌握 J2EE 的相关知识才能集成，因为这部分内容已经为 OMP Client API 所完成，他们只需要进行普通的 java 调用，降低了对集成应用开发者的要求。图 10 为系统实现分布式时更为详细的系统结构示意图(省去了单套应用时模型建立工具以及授权工具的相关表示，因为它们不需要集成应用开发者再写代码进行扩展，只需要对配置文件做相应的改动)。

图中虚线表示层次划分；垂直方向带箭头的黑线表示调用（请求）关系；两个带箭头的交叉线表示 OMP Client API 的透明分布式调用；水平方向带箭头的虚线表示应用服务器之间的远程同步调用；Server1、Servern 表示各个独立的应用服务器；图中权限验证（API）、模型访问（API）、权限访问（API）对象管理 API 表示 OMP 提供的可供集成应用调用的 API；图中各椭圆表示 API 内部实现的部分内容。

OMP Client API 对分布式数据的透明访问封装是这样实现的：在 OMP 内部存在一种称为代理（Proxy）的对象（只能挂接在一个方案之下），该对象实际用于指向另一个服务器（Server）上的 OMP 对象（OMP Object）在给定方案中的信息，Client API 在访问到 Proxy 对象时，将根据获取到的 Proxy 对象的信息，再次向另一个 Server 发出请求，得到相关信息。每个 OMP Object 均带有它所属的 Server 信息，后续访问均会直接访问另一个 OMP Server，从而实现分布式的访问，对上层调用者提供透明性，这类似于 http 中的 redirect 操作。

以上所述，仅为本发明较佳的具体实施方式，但本发明的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到的变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应该以权利要求书的保护范围为准。

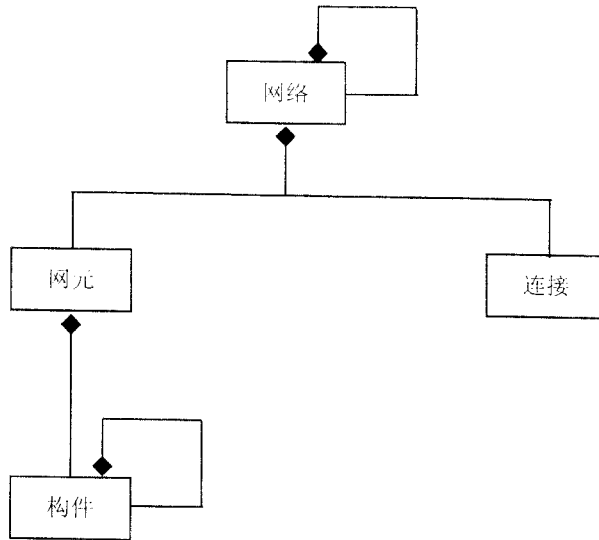


图 1

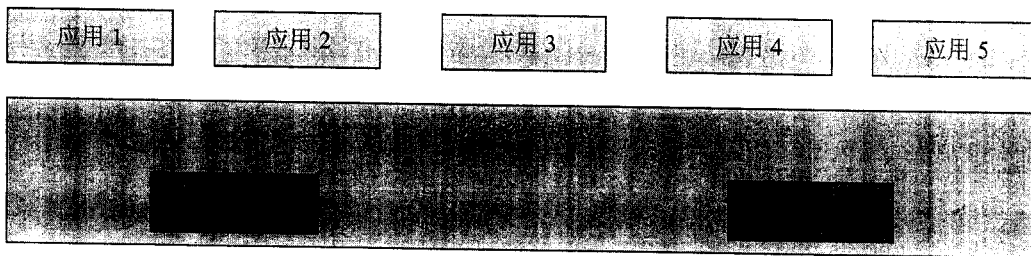


图 2

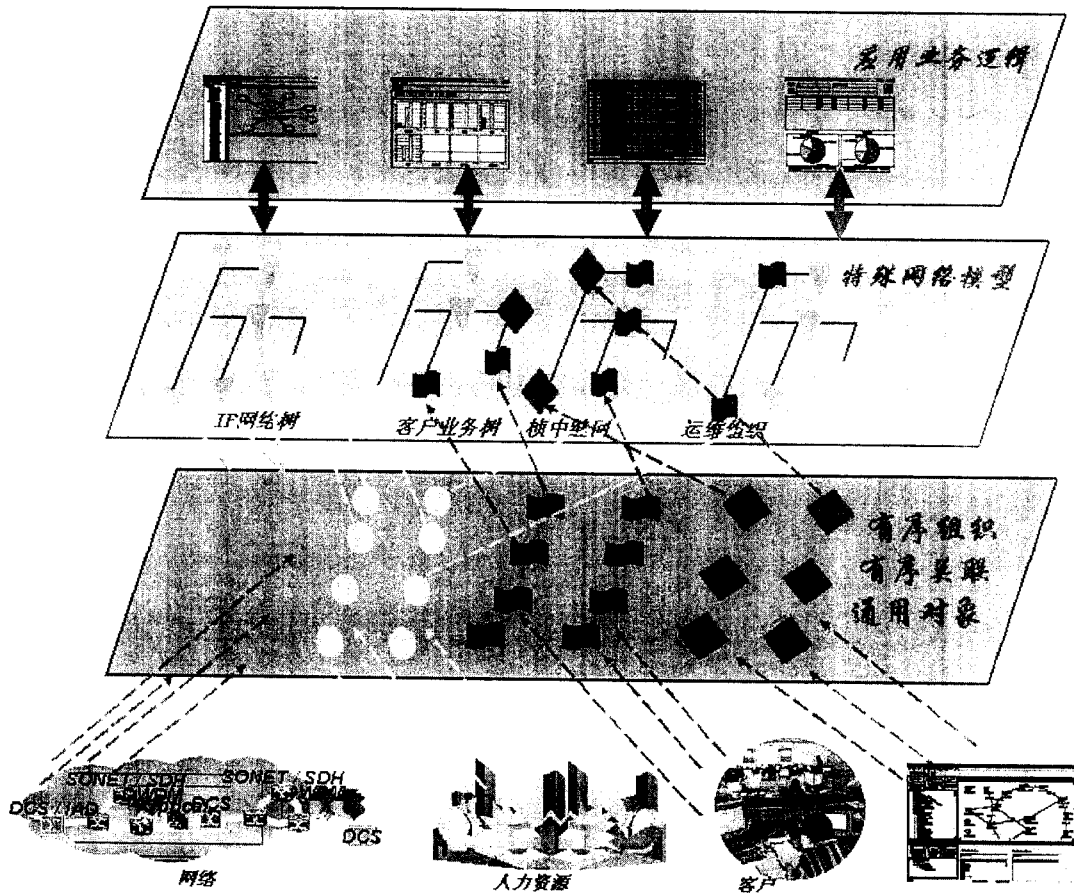


图 3

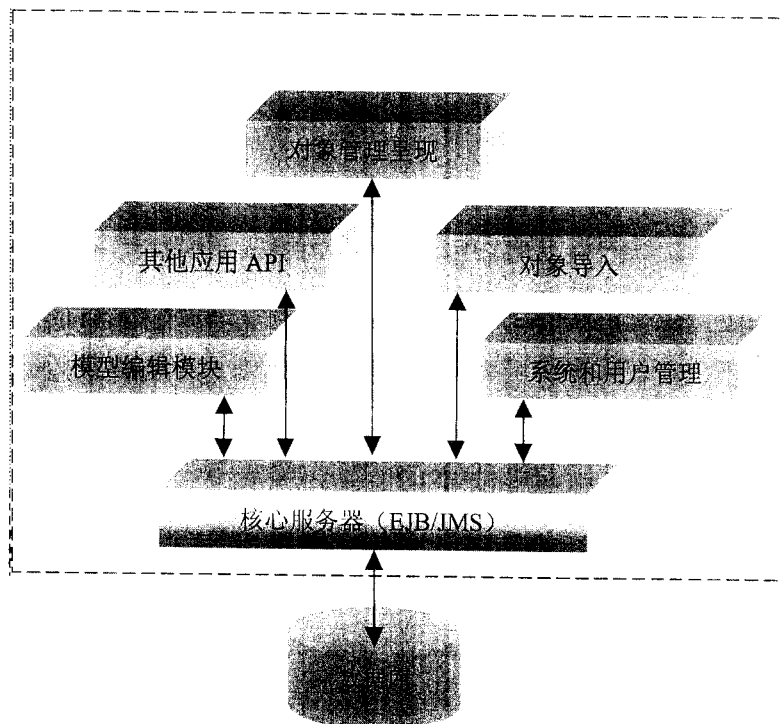


图 4

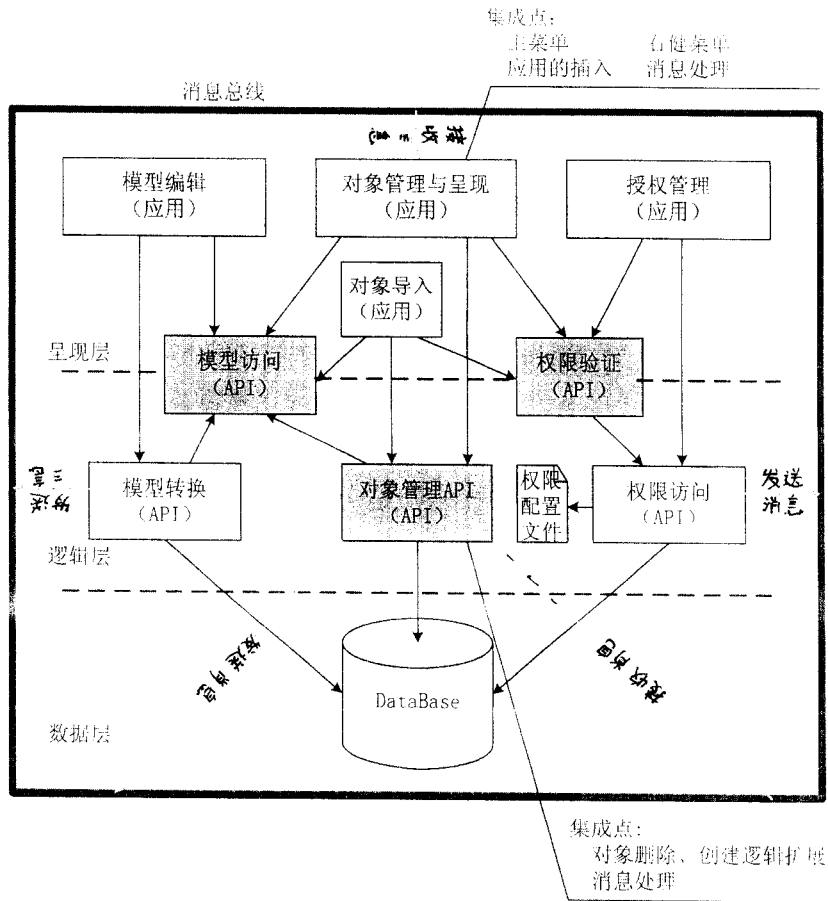


图 5

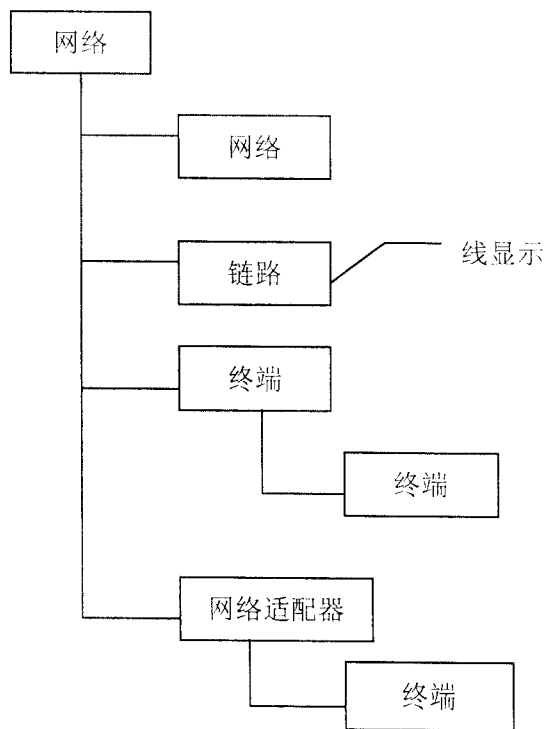


图 6

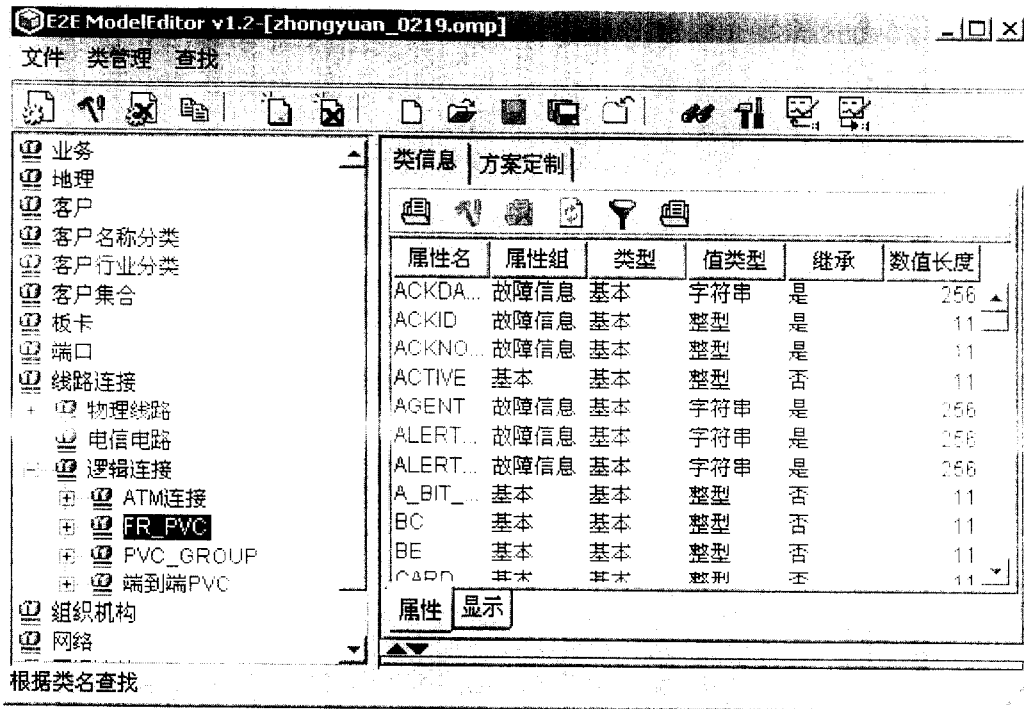


图 7

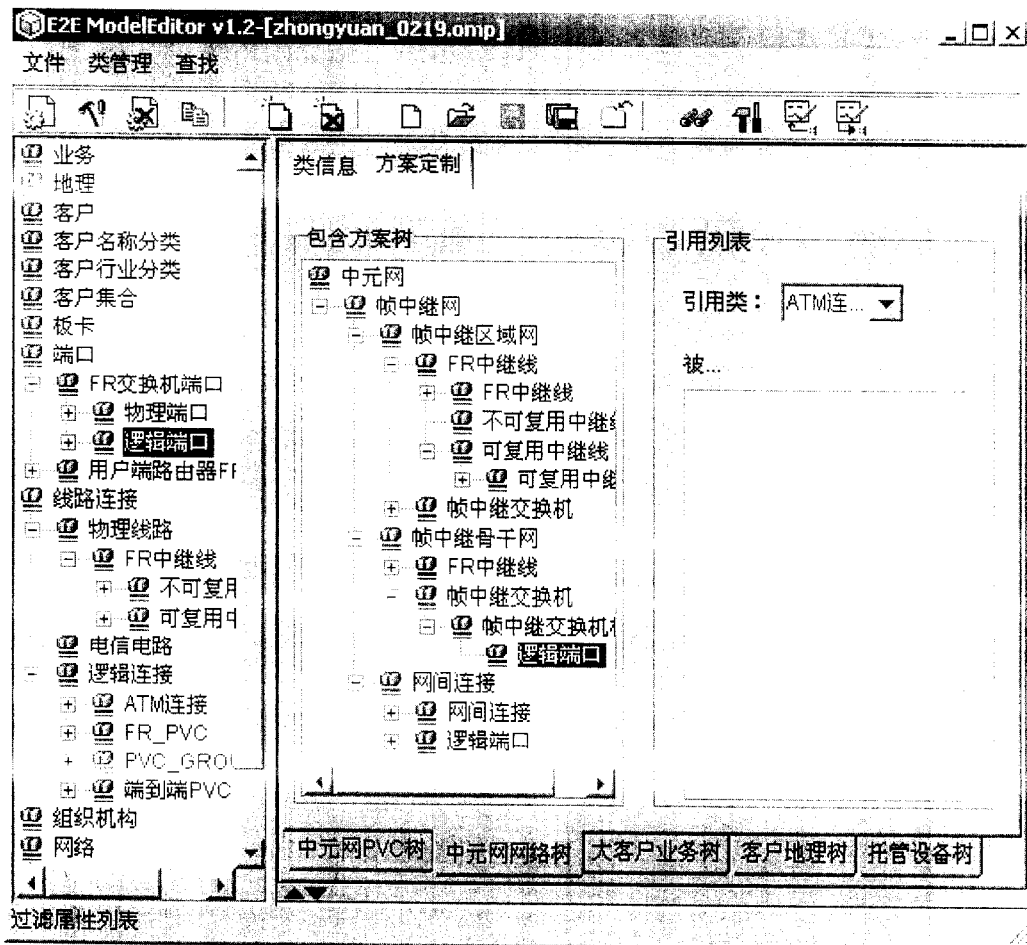


图 8

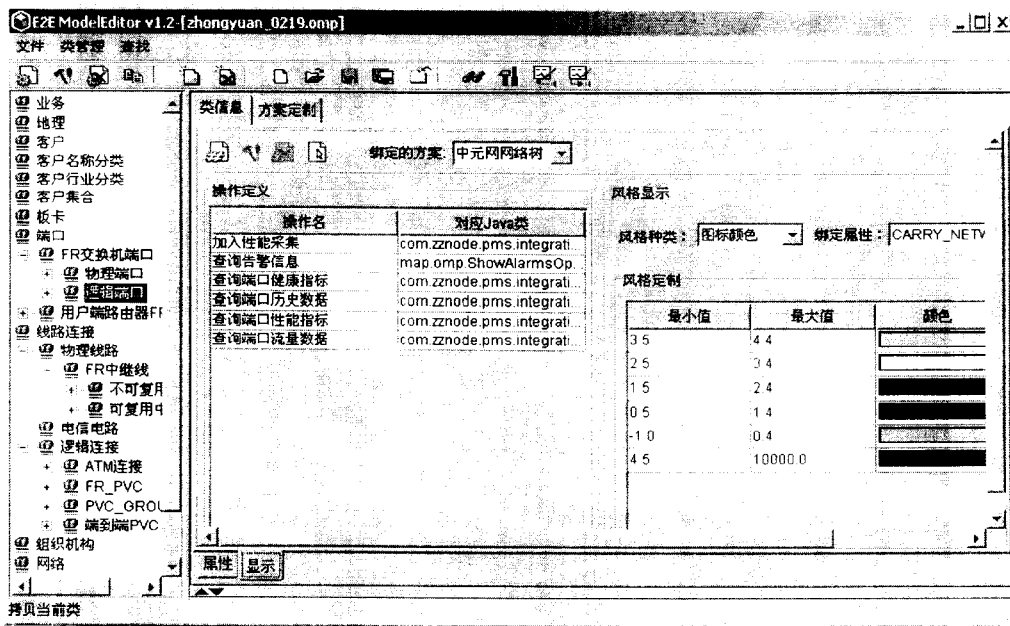


图 9

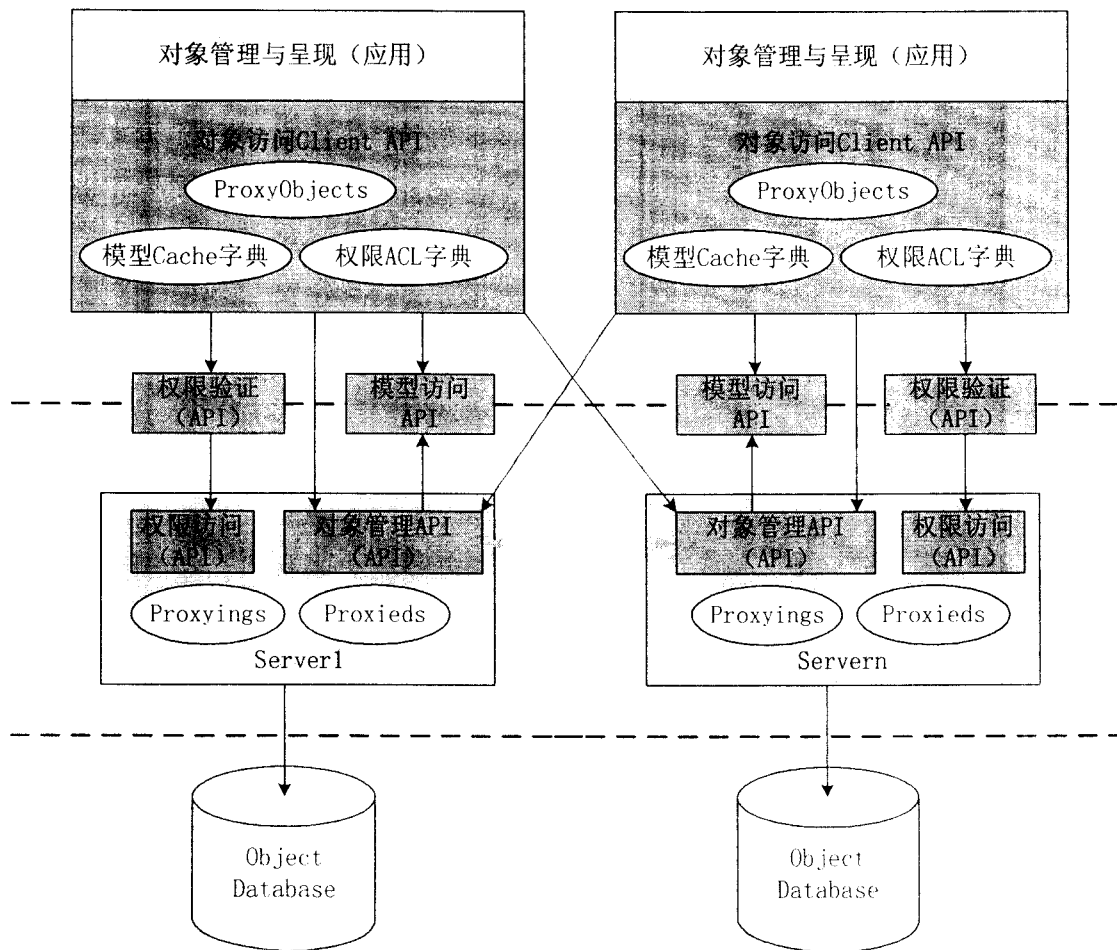


图 10