



(12) 发明专利申请

(10) 申请公布号 CN 103530134 A

(43) 申请公布日 2014. 01. 22

(21) 申请号 201310524752. 3

(22) 申请日 2013. 10. 30

(71) 申请人 重庆大学

地址 400044 重庆市沙坪坝区沙正街 174 号

(72) 发明人 雷琦 李隆昌 侯坤 王洪

宋豫川

(74) 专利代理机构 重庆博凯知识产权代理有限公司

公司 50212

代理人 张先芸

(51) Int. Cl.

G06F 9/44 (2006. 01)

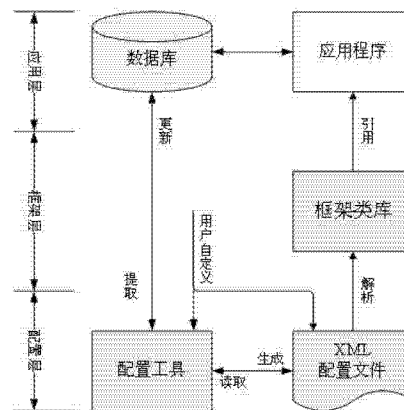
权利要求书1页 说明书6页 附图3页

(54) 发明名称

一种可配置软件平台结构

(57) 摘要

一种可配置软件平台结构,包括可配置平台框架和基础平台框架;可配置平台框架包括数据视图配置系统、计算配置系统、数据扫描配置系统和消息配置系统。本发明针对现有企业计算机信息系统不能随着业务流程的变化而进行动态调整的问题,通过建立可配置的软件平台将解决企业信息系统开发的效率的问题;解决企业信息系统自身的升级和改善造成系统不稳定的问题;解决企业信息系统实施个性化强,实施困难的问题。在该平台上进行企业信息系统的开发和维护,解决企业信息化系统刚性、开发效率低、实施复杂等问题。



1. 一种可配置软件平台结构,其特征在于,包括可配置平台框架和基础平台框架;可配置平台框架用于软件的可配置开发和柔性调整,基础平台框架用于提供平台的基础连接和配置功能;可配置平台框架包括数据视图配置系统、计算配置系统、数据扫描配置系统和消息配置系统;其中,

(1) 数据视图配置系统:

数据视图配置系统通过数据视图的映射配置将数据库对数据的定义和信息系统业务逻辑对数据的处理通过数据视图进行分离,而数据视图通过标准的 XML 方式进行描述和实现;

(2) 计算配置系统:

计算配置系统用于将计算模型的逻辑通过外部的 XML 进行模型的描述和参数的定义,相应配套建立了一个参数文件库,用于保存该计算模型参数配置情况;然后在信息系统中建立一个逻辑计算引擎,对定义在外部的计算模型进行计算;

(3) 数据扫描配置系统:

数据扫描配置系统用于条形码的扫描识别,解决在企业已有编码规则的前提下实现条码扫描的快速定位输入;

首先,对企业的条码进行分类,建立条码分类表,为每类条码分配一个标志;

然后,建立一个输入区域的定义表,将输入区域按照相同的方式定义对应的条码标志;

再后,在企业按照原有规则生成条码打印之前,通过平台将定义的条码分类定义自动作为前缀形成新的条码进行打印;

最后,在应用中扫描该打印条码时,系统首先识别条码首位的类别,通过该类别系统自动识别该条码需要的输入区域,然后系统自动将该输入区域获得焦点完成条码扫描时数据的自动输入;

(4) 消息配置系统:

首先,在平台中对所有的“确定”操作,最后自动由系统产生 2 条消息,一条是成功的消息,一条是失败消息,并且对该消息的产生定义编号和过程序号;

然后,平台中建立了一个消息管理器,用于保存这些发送过来的“确定”操作的消息;

再后,平台建立了一个消息处理表,将该类消息的后处理方式预先进行定义;

对消息管理器进行扫描,如果某些消息没有定义下一步的处理方式,则该消息后续不做任何处理就过滤掉;而一旦在消息处理表中有后续处理过程的定义,则平台将按照预先定义的方式对“确定”后的数据进行处理。

一种可配置软件平台结构

技术领域

[0001] 本发明用于企业信息系统和软件的设计开发和实施运行领域,特别涉及对企业信息系统和软件进行可配置开发和标准化实施的方法。

[0002]

技术背景

[0003] 信息化系统已经全面应用于企业的经营和生产、销售等各个方面,并在实际应用中取得了良好的效果。

[0004] 但是,在进行信息系统开发时,由于各个企业之间情况差距大,企业自身情况复杂多变,开发时首先要解决的就是系统数据复杂多变、信息量大,以及企业的个性化需求等问题,因此在系统开发过程中,因其严密的逻辑性和现实的高度耦合,导致信息系统缺乏一定的柔性。

[0005] 在传统的信息系统开发过程中,由于信息化系统的巨大刚性和企业信息化需求的柔性之间的矛盾,导致了信息系统的可配置差、可重构差和柔性差等问题。在传统的信息系统开发过程中,对于数据库的访问操作、计算逻辑、界面信息数据结构等信息是直接嵌入代码中的,导致信息系统连接的数据库或企业的实际需求发生微小改变后,原有的信息系统就不能正常运行或满足要求,必须由程序员进行代码级的修改或更新,这就给企业的信息化系统实施带来了繁重的工作,也造成了人力资源、时间、金钱的巨大浪费,不利于信息化系统的商品化生产,也不利于信息化系统在企业中的广泛应用。

[0006] 在信息系统的实施过程中,随着企业的不断发展和客户的需求的不断变化,企业需要更加快速高效的信息化运作模式,以更快更方便地适应企业环境的快速变化。企业的业务流程和运作模式,以及系统的具体功能都将会经常性的变化,这就给企业信息化系统带来了很大的挑战,要求企业的信息化系统具有适应企业业务功能和个性化功能及界面的能力。

[0007] 不同行业、不同企业的生产组织模式不尽相同,即便在同一企业中,企业的业务流程也在不断变化之中。传统的信息系统开发时基于特定行业 and 企业的固定业务流程进行开发,信息系统不能随着业务流程的变化而进行动态调整。传统的信息系统由于缺乏柔性,不易修改和配置,当其移植到其他生产业务环境、组织结构中时,信息系统将不能完成其功能。此外,在传统的信息系统开发过程中,由于不重视其可配置性,不同行业或企业间的信息系统之间很难相互借鉴,使得信息化系统开发周期长、难度大、成本高,不利于信息化系统在信息化系统的广泛应用。

[0008]

发明内容

[0009] 针对现有企业计算机信息系统不能随着业务流程的变化而进行动态调整的问题,本发明的目的是提供一种可配置软件平台结构,在该平台上进行企业信息系统的开发和维

护,解决企业信息化系统刚性、开发效率低、实施复杂等问题。

[0010] 解决上述技术问题,本发明采用的方法手段如下:

一种可配置软件平台结构,其特征在于,包括可配置平台框架和基础平台框架;可配置平台框架用于软件的可配置开发和柔性调整,基础平台框架用于提供平台的基础连接和配置功能;可配置平台框架包括数据视图配置系统、计算配置系统、数据扫描配置系统和消息配置系统;其中,

(1) 数据视图配置系统:

数据视图配置系统通过数据视图的映射配置将数据库对数据的定义和信息系统业务逻辑对数据的处理通过数据视图进行分离,而数据视图通过标准的 XML 方式进行描述和实现;

(2) 计算配置系统:

计算配置系统用于将计算模型的逻辑通过外部的 XML 进行模型的描述和参数的定义,相应配套建立了一个参数文件库,用于保存该计算模型的参数配置情况;然后在信息系统中建立一个逻辑计算引擎,对定义在外部的计算模型进行计算;

(3) 数据扫描配置系统:

数据扫描配置系统用于条形码的扫描识别,解决在企业已有编码规则的前提下实现条码扫描的快速定位输入;

首先,对企业的条码进行分类,建立条码分类表,为每类条码分配一个标志;

然后,建立一个输入区域的定义表,将输入区域按照相同的方式定义对应的条码标志;

再后,在企业按照原有规则生成条码打印之前,通过平台将定义的条码分类定义自动作为前缀形成新的条码进行打印;

最后,在应用中扫描该打印条码时,系统首先识别条码首位的类别,通过该类别系统自动识别该条码需要的输入区域,然后系统自动将该输入区域获得焦点完成条码扫描时数据的自动输入;

(4) 消息配置系统:

首先,在平台中对所有的“确定”操作,最后自动由系统产生 2 条消息,一条是成功的消息,一条是失败消息,并且对该消息的产生定义编号和过程序号;

然后,平台中建立了一个消息管理器,用于保存这些发送过来的“确定”操作的消息;

再后,平台建立了一个消息处理表,将该类消息的后处理方式预先进行定义;

对消息管理器进行扫描,如果某些消息没有定义下一步的处理方式,则该消息后续不做任何处理就过滤掉;而一旦在消息处理表中有后续处理过程的定义,则平台将按照预先定义的方式对“确定”后的数据进行处理。

[0011] 相比现有方法,本发明具有如下有益效果:

(1) 本发明通过建立可配置的软件平台将解决企业信息系统开发的效率的问题。由于企业信息系统对数据的操作主要是查询、增加、删除和修改等固定的操作,只是面向的数据和操作过程不同,造成企业信息系统在开发过程中存在大量的重复编程操作,大大降低了开发的效率。

[0012] (2) 本发明通过可配置的软件平台解决企业信息系统自身的升级和改善造成系统

不稳定的问题。由于信息系统通过不同企业的应用会产生许多为了符合企业个性化需求的应用,而这些应用于信息系统本身具有密切关联,它们妨碍了信息系统本身的升级,因为升级可能会造成企业信息系统应用的不稳定性。

[0013] (3)本发明利用配置的方式解决企业信息系统实施个性化强,实施困难的问题。企业信息化系统由于企业的性质不一样,产品不一样,管理不一样,流程不一样,数据要求不一样等差异性,对企业信息系统的实施提出了很高的要求,因此,目前大量的企业信息系统都通过开发人员参与的二次开发方式,通过源代码级别的修改满足特定企业的需求,造成实施过程不可控,过程复杂。

[0014]

附图说明

[0015] 图 1 是本发明中可配置的软件平台的示意图;

图 2 是本发明中基于可配置平台的系统应用模型图;

图 3 是本发明中基于可配置平台的软件开发模型图;

图 4 是本发明中 XML 配置文档的结构图;

图 5 是本发明中可配置平台核心类库组织结构图。

[0016]

具体实施方式

[0017] 下面结合具体实施例和附图对本发明作进一步详细说明。

[0018] 如图 1 和图 2 所示,一种可配置的软件平台结构,主要包括可配置平台框架和基础平台框架;其中,以可配置平台框架为主。可配置平台框架涉及到数据视图配置、界面配置、流程配置、计算配置、数据扫描配置、主从结构配置、报表配置、消息配置、菜单工具配置和集成配置等方面。上述软件平台结构是为了保持平台的完整性而设计的,其中有些方面已经很成熟,本发明的创新是通过硬件和软件的结合,尤其是通过软件方法实现;主要包括以下几个方面:

(1) 数据视图配置系统:

数据视图配置主要解决企业信息系统中由于各个企业对数据的要求和显示不同,而将数据操作从系统开发过程,即从系统源代码中剥离出来,包括数据的定义、显示和 SQL 查询语句。通过数据视图的映射配置将数据库对数据的定义和信息系统业务逻辑对数据的处理通过数据视图进行分离,使企业信息系统对数据的处理不是直接通过数据库处理,而是通过中间的数据视图进行处理。而数据视图通过标准的 XML 方式进行描述和实现。一旦企业信息系统对数据的要求发生变化,将直接体现在对数据库和数据视图的配置完成从而满足数据变化,避免了信息系统业务逻辑的处理,即采用相同的业务逻辑处理完成对变化数据的处理。参见图 3。

[0019] 通过基于 XML 文档的数据视图的枚举和数据关联的配置,解决个性化数据的显示和关联问题。例如,人员信息在系统数据库中是一张表,性别 Gender 属性下“0”代表“男”,“1”代表“女”,则将“0”和“男”、“1”和“女”的映射关系保存在 XML 配置文档中,在查看系统界面时,系统将自动解析 XML 配置文档,将“0”转化为“男”、“1”转化为“女”进行显示,

当系统的数据视图需求发生改变时,修改配置文档中的映射关系即可方便的个性化设置数据视图;另外例如,信息系统中某人的级别必须由另外一个级别数据来约束,原有技术就必须在信息系统内部通过严格的逻辑处理来解决,而这里则通过数据视图对这两组具有约束关系的数据进行配置定义即可,而信息系统不做任何处理则可以通过配置文件满足数据之间的约束。

[0020] 以上这些原来都是通过源代码级别的开发操作来解决企业对数据的不同要求,而本方法则通过数据视图的配置来交互式完成,与应用系统完全无关,因此不用任何的源代码级别的开发。

[0021] XML 文档的结构图参见图 4。

[0022] (2) 计算配置系统:

在企业信息系统中经常遇到大量计算模型,传统的处理都是将这些计算模型在源代码中通过变量等定义,通过严格的编程进行逻辑实现,一旦计算模型或者计算参数发生变化,则必须通过源代码进行逻辑调整。

[0023] 本发明指的计算配置技术则将计算模型的逻辑通过外部的 XML 进行模型的描述和参数的定义,相应配套建立了一个参数文件库,用于保存该计算模型的参数配置情况。然后在信息系统中建立一个逻辑计算引擎,对定义在外部的计算模型进行计算。这样,如果信息系统的计算模型发生变化,则只需要交互式修改外部 XML 文档中的计算模型和参数即可,而信息系统本身不做任何调整和修改。

[0024] (3) 数据扫描配置系统:

本发明中数据扫描主要用于条形码的扫描识别。由于在企业信息系统中编码存在太多类型,例如人员编码、设备编码、物料编码等,现在大量的采用了条码技术来解决信息系统中编码的使用。但是在条码扫描时,一般要求屏幕界面输入焦点在对应的输入区域才能正确扫描条码,例如在人员输入区域扫描人员条码,而不能扫描设备条码,因此传统的采用交互式的移动输入区域来对应条码的扫描,这样时工作效率较低,而且容易产生错误。

[0025] 本方法中将通过配置解决条码对应的快速输入问题。解决这个问题首先是在不能改变企业本身的编码规则约束之上的,即解决在企业已有编码规则的前提下实现条码扫描的快速定位输入。

[0026] 首先对企业的条码进行分类,建立条码分类表,为每类条码分配一个标志,例如人员条码用 P,设备条码用 D 等。

[0027] 然后建立一个输入区域的定义表,将输入区域按照相同的方式定义对应的条码标志。

[0028] 再后,在企业按照原有规则生成条码打印之前,通过平台将定义的条码分类定义自动作为前缀形成新的条码进行打印

最后,在应用中扫描该打印条码时,系统首先识别条码首位的类别,通过该类别系统自动识别该条码需要的输入区域,然后系统自动将该输入区域获得焦点完成条码扫描时数据的自动输入,从而不用手工切换输入区域焦点,提高了效率,同时保证和输入的正确性。

[0029] (4) 消息配置系统:

企业信息系统通常由不同的系统所构成,例如管理、生产、销售、设计等系统,因此企业的信息系统完成一个业务后,一般而言会对后续其他的应用系统产生影响,包括数据的处

理和流程的处理。而传统的方式很多也都是采用二次开发的方式,在获取两个系统的源代码的基础上通过编程实现。本方法通过消息的配置来解决某一业务完成后对企业系统的连接处理问题。

[0030] 首先,在平台中对所有的“确定”操作(一般而言是信息系统界面中的 OK 按钮),的最后自动由系统产生 2 条消息,一条是成功的消息,一条是失败消息,并且对该消息的产生定义编号和过程序号。

[0031] 然后,平台中建立了一个消息管理器,用于保存这些发送过来的“确定”操作的消息。

[0032] 再后,平台建立了一个消息处理表,将该类消息的后处理方式预先进行定义,例如遇到该消息下一步怎么处理进行定义。对消息管理器进行扫描,如果某些消息没有定义下一步的处理方式,则该消息后续不做任何处理就过滤掉;而一旦在消息处理表中有后续处理过程的定义,则平台将按照预先定义的方式对“确定”后的数据进行处理。

[0033] 因此,利用该发明通过对消息处理表的配置,可以定义所有的数据后续处理过程,而不用在原来的应用系统中通过二次开发实现。

[0034] 实施例 1:

首先,本发明涉及的一种可配置的软件平台和结构,将大量的配置操作封装在了一个“配置工具”中,用于帮助完成配置工作。

[0035] 1、企业信息系统开发方面的实施方式:

(1) 一旦开始开发一个新的信息系统功能,利用配置工具初始化,将按照图 5 的结构产生一个统一的开发入口。

[0036] (2) 系统将自动产生一个用于该系统功能的一系列 XML 配置文档,包括如图 2 所示的数据视图配置文档,计算配置文档、条码配置文档等。

[0037] (3) 按照传统的方式设计和形成数据库的数据规划,这个步骤是在数据库中进行。

[0038] (4) 通过配置工具,选取相应的数据库数据表和配置文档关联,平台将自动将所有数据仅识别和关联。

[0039] (5) 根据系统功能的要求,在配置文档中对数据的使用和显示进行配置。

[0040] (6) 在系统功能的界面中通过第一步的开发入口自动生成一个界面配置,用于功能的处理。

[0041] (7) 直接运行系统,系统则按照配置文档的定义,将结果和数据操作进行处理。

[0042] 因此,在系统开发过程中,基本不用多的程序开发,更多的是在进行配置文档的处理,而配置文档本身适合系统分离的。

[0043] 2、企业信息系统的升级和改进:

(1) 根据功能升级和变化的需要,在数据库中对数据进行调整,包括增加、删除、修改等。

[0044] (2) 寻找相应的配置文件,对数据视图、计算模型等进行交互式修改后保存。

[0045] (3) 重新运行系统,系统将按照新的配置文档的定义完成功能和数据的处理。

[0046] 因此,在信息系统升级和改进中,大量的更改依然在配置文档中处理,源代码基本不用修改。当然,如果整个功能的升级已经完全不同,几乎是一个全新的功能,则可以按照上步的企业信息系统开发方面的实施方式进行处理。

[0047] 3、企业信息系统的实施方式：

完全不同于传统企业信息系统的实施过程，新的实施过程如下：

- (1) 分析企业的数据库结构和情况，构建数据库的数据定义；
- (2) 使用配置工具对涉及到个性化的配置文档进行修改；
- (3) 直接运行系统，将按照新的配置进行运行。

[0048] 因此，传统的企业信息系统实施时，由于企业个性化的需求导致大量的功能和数据处理通过二次开发实现，造成了成本高，而且实施周期和效果无法准确预计，对企业实施信息系统带来了极大的难度。而采用本发明的平台，实施中所需要的基本是配置人员，根据企业的个性化需求完成数据和过程的不同配置，开发人员基本不参与实施过程，这样将降低成本，并且实施过程可控，实施周期可控。

[0049] 同时，由于整个平台的核心并没有根据企业的个性化要求进行修改，因此运行核心非常稳定，企业对信息系统的个性化要求都体现在配置文档方面。因此，对系统的测试和维护以及升级都变得非常有效和简单。

[0050] 最后说明的是，以上实施例仅用以说明本发明的技术方案而非限制，尽管参照较佳实施例对本发明进行了详细的说明，可以对本发明的技术方案进行修改或者等同替换，而不脱离本发明技术方案的宗旨和范围，其均应涵盖在本发明的权利要求范围当中。



图 1

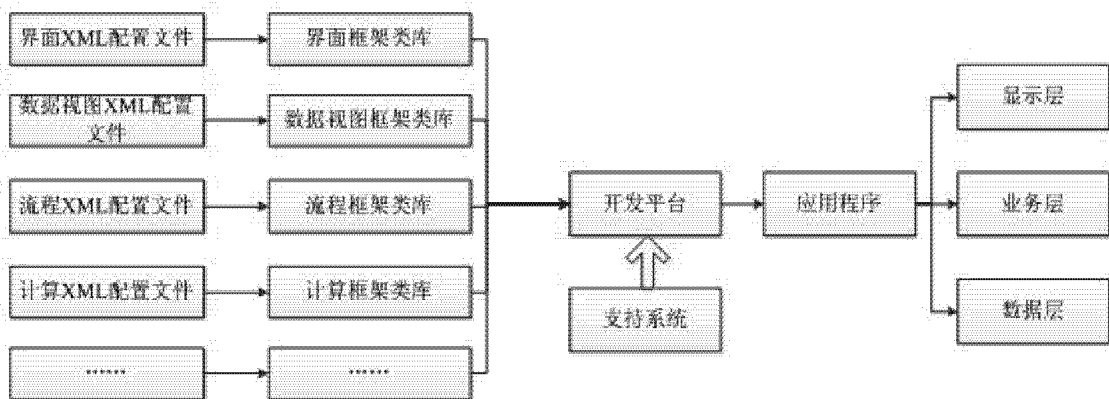


图 2

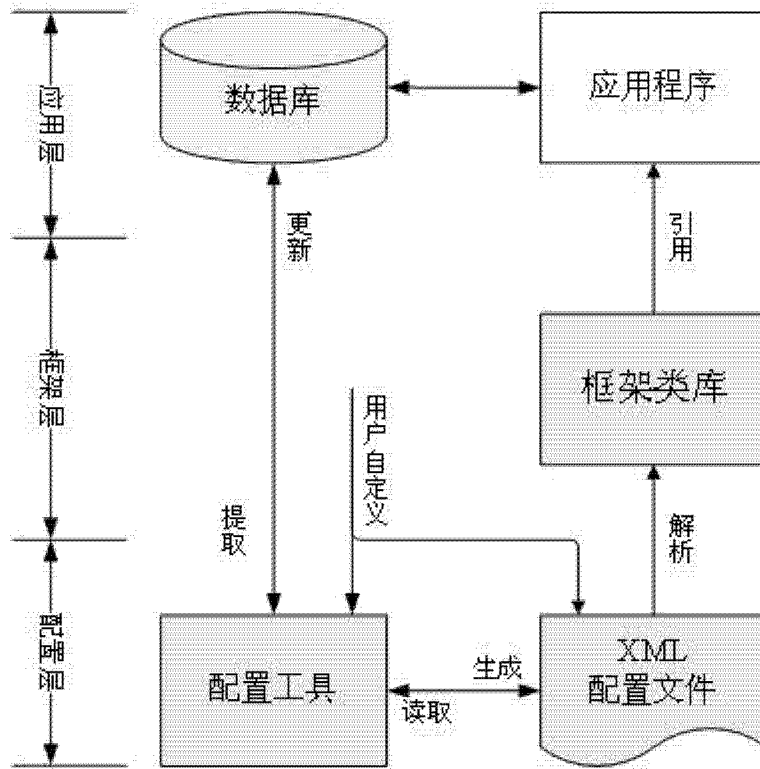


图 3

```

<Entity Alias="emp" Caption="人员信息" Name="t_employee" .....> .....<description of current entity>
.....
<Properties> .....<property set>
<Property Alias="id" Caption="人员编号" DataType="3" .....> .....<property description>
  <Code> .....<code rules>
  <Segment Digit="2" Fabric="0" IncreaseSeed="0" Increment="1" Matter="Ex" PositionIndex="1"/>
  <Segment Digit="2" Fabric="2" IncreaseSeed="0" Increment="1" Matter="" PositionIndex="2"/>
  <Segment Digit="3" Fabric="1" IncreaseSeed="0" Increment="1" Matter="" PositionIndex="3"/>
  </Code>
.....
</Property>
.....
<Operates> .....<operation set>
  <Operate ID="getpow" SQLString="select [id],[roleid],[powerid] from t_role_power where roleid={role}" />
.....
.....<operation description>
</Operates>
</Entity>

```

图 4

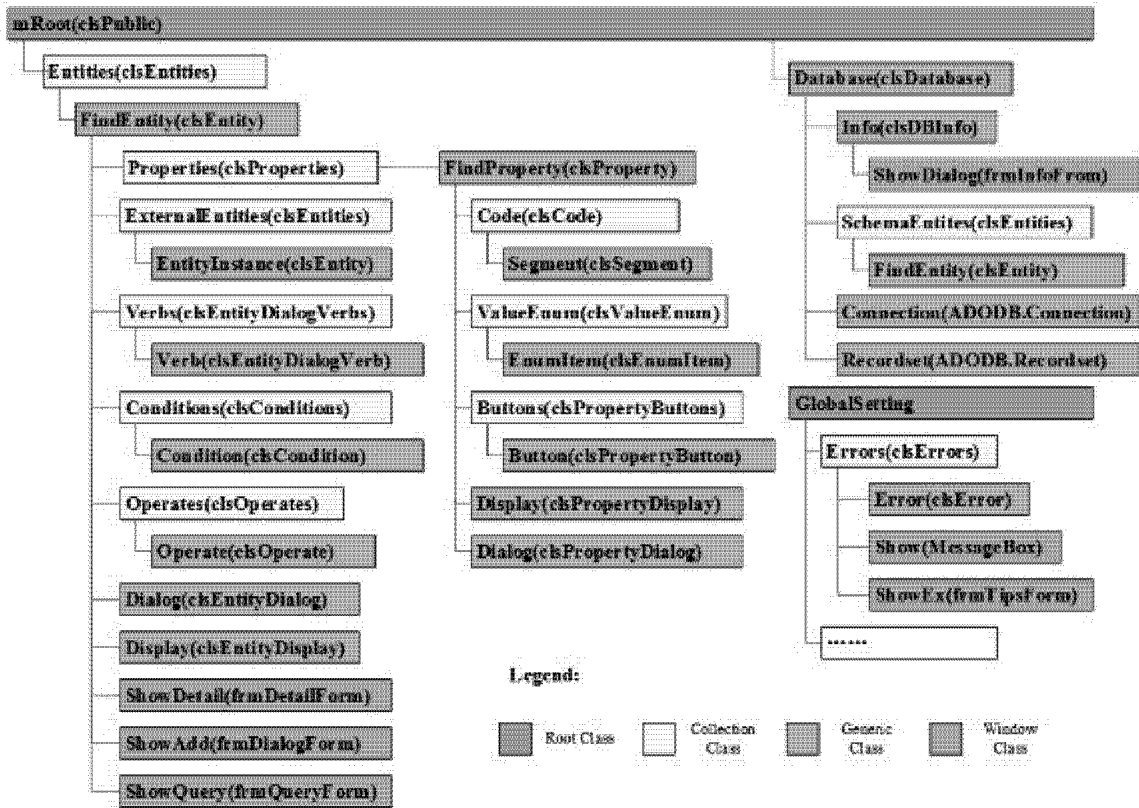


图 5