



(12) 发明专利

(10) 授权公告号 CN 118132060 B

(45) 授权公告日 2024.08.02

(21) 申请号 202410544115.0

G06F 8/38 (2018.01)

(22) 申请日 2024.05.06

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 110825362 A, 2020.02.21

申请公布号 CN 118132060 A

CN 113886362 A, 2022.01.04

(43) 申请公布日 2024.06.04

审查员 沈宇扬

(73) 专利权人 苏州锐盈智能科技有限公司

地址 215128 江苏省苏州市工业园区金鸡

湖大道88号人工智能产业园G3-1901-

070单元

(72) 发明人 周培勇 钟鸣 周新花

(74) 专利代理机构 北京中企鸿阳知识产权代理

事务所(普通合伙) 11487

专利代理师 刘立国

(51) Int. Cl.

G06F 8/34 (2018.01)

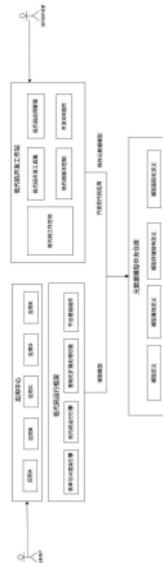
权利要求书2页 说明书10页 附图2页

(54) 发明名称

一种基于元数据驱动的低代码软件开发方法

(57) 摘要

本发明公开了一种基于元数据驱动的低代码软件开发方法,属于软件开发技术领域,包括以下步骤:建立低代码开发工作站,用于为低代码开发者提供交互式图形界面软件需求的收集和设计;基于低代码开发工作站,构建元数据模型中央仓库,用于保存和管理低代码开发后的模型元数据;基于元数据模型中央仓库,设计低代码运行框架,构建用于部署前端应用的平台基础框架;基于低代码运行框架,生成应用中心,用于对基于元数据驱动的低代码软件开发出的各类业务应用进行自动化部署。本发明通过元数据驱动的低代码开发方式和通用的平台基础组件,减少和简化了传统开发中的各个环节等,加快了对业务需求变化的响应速度。



1. 一种基于元数据驱动的低代码软件开发方法,其特征在于,包括以下步骤:

建立低代码开发工作站,用于为低代码开发者提供交互式图形界面软件需求的收集和  
设计,包括,数据模型结构设计、业务流程设计、界面和表单设计;

基于所述低代码开发工作站,构建元数据模型中央仓库,用于保存和管理低代码开发  
后的模型元数据,包括模型和模型属性的定义、模型的存储结构定义和模型的国际化定义;

基于所述元数据模型中央仓库,设计低代码运行框架,通过设置表单和UI渲染引擎、低  
代码运行引擎、客制化扩展点以及 workflow 引擎,构建用于部署前端应用的平台基础框架;

基于所述低代码运行框架,生成应用中心,用于对基于元数据驱动的低代码软件开发  
出的各类业务应用进行自动化部署;

在实现模型和模型属性的定义的过程中,通过UI界面创建新的模型后,将模型和模型  
属性的定义进行保存,支持设置模型的关联关系和访问控制的配置,实现模型和模型属性  
的定义;

在实现模型的存储结构定义的过程中,通过定义模型在关系型数据库中的存储方式,  
包括模型所对应的表、模型属性所对应表的字段,实现模型的存储结构定义;

在实现模型的国际化定义的过程中,通过定义模型和模型属性根据各国语言的翻译后  
的内容,包括模型名的翻译内容和模型属性的翻译内容,实现模型的国际化定义。

2. 根据权利要求1所述一种基于元数据驱动的低代码软件开发方法,其特征在于:

在设计低代码运行框架的过程中,通过图形界面创建和配置表单,定义字段和布局,而  
无需编写代码,并通过UI渲染引擎将设计完成的表单和其他页面转换成实际的用户界面,  
处理用户的输入,并将操作结果反馈给系统,完成表单和UI渲染引擎的设计。

3. 根据权利要求2所述一种基于元数据驱动的低代码软件开发方法,其特征在于:

在设计表单和UI渲染引擎的过程中,所述表单和UI渲染引擎的功能包括:

表单布局控制:支持通过界面操作来设定表单的整体布局和结构,包括字段和部分的  
位置、大小和顺序;

表单设计界面:提供直观的图形用户界面,用户通过拖放组件的方式来设计表单,无需  
编写代码;

表单列定义:使用户能够定义表单中各个字段的列属性,以及如何在表单中显示这些  
字段;

表单个性化视图:支持创建自定义的表单视图,选择哪些字段展示以及它们的布局,以  
满足特定角色或工作流程的需求;

表单关联列表定义:在表单中支持定义与表单相关联业务对象的列表,包括相关记录  
的显示和关联数据的管理;

表单个性化区块定义:支持个性化定义表单中的部分区域,以逻辑地组织信息并提高  
用户界面的可读性,并支持以HTML和JavaScript的方式动态编写前端组件;

列表定义:支持设定列表视图,包括列表中应显示的字段,字段的排序和过滤器设置;

列表元素定义:支持定义列表中各个元素,包括格式、样式和交互行为。

4. 根据权利要求3所述一种基于元数据驱动的低代码软件开发方法,其特征在于:

在设计低代码运行框架的过程中,通过低代码运行引擎打通前端界面和后端业务逻辑,  
使得业务逻辑的扩展和修改只需要通过脚本编写即可完成。

5. 根据权利要求4所述一种基于元数据驱动的低代码软件开发方法,其特征在于:

在设计低代码运行框架的过程中,通过整合面向切面编程,设计客制化扩展点,用于支持在业务数据的创建、更新、删除或展示数据的前后进行客制化规则扩展点的自定义设置,包括数据的处理与验证,以及基于特定条件控制规则的执行,其中,

在服务端,通过所述客制化扩展点,在数据的生命周期内进行客制化规则的定义,包括数据处理与验证以及执行条件控制,通过根据自定义条件执行相应的业务规则,同时,根据在用户界面中添加的自定义按钮和链接,通过客户端或服务端脚本实现自定义功能的触发;

在客户端,通过所述客制化扩展点,对客户端低代码进行编写,用于增强UI表单的交互性,包括前端数据验证、表单项联动和用户体验的增强,并将客户端低代码配置为在不同的表单事件触发,同时设置触发条件,其中,UI策略用于控制和管理用户界面元素的行为,包括字段的显示/隐藏、编辑权限和必填条件,并支持基于自定义条件的控制执行。

6. 根据权利要求5所述一种基于元数据驱动的低代码软件开发方法,其特征在于:

在构建平台基础框架的过程中,平台基础组件包括:

多租户管理:用于以租户为单位,对平台功能和数据进行隔离;

用户管理:用于生成新的用户账户,并管理现有账户;

角色管理用于定义、分配和管理用户角色,并结合用户组对用户群组的分组管理和权限控制,简化访问权限的分配,其中,

权限管理,包括:

RBAC权限管理,用于对角色的访问控制,支持对不同角色的特定实体设置权限;

ACL权限管理:用于对表、字段和记录级别进行细粒度的访问控制,包括对特定对象执行特定操作的授权,基于自定义条件的授权,对RESTAPI的授权;

数据权限管理:用于对多维度的数据的可见性进行权限控制;

在设计低代码运行框架的过程中,通过定义和建模复杂的工作流程,并设计多种类型的工作流节点以及图形化流程,对所述工作流引擎进行设计,其中,定义和建模复杂的工作流程,包括:任务、决策点、分支和循环,并设计工作流调度器对工作流的任务创建、执行和运行过程进行监控;多种类型的工作流节点包括运行脚本、输出日志、条件分支和审批;基于可视化的用户界面,通过拖放方式设计和定制工作流程,完成图形化流程设计。

## 一种基于元数据驱动的低代码软件开发方法

### 技术领域

[0001] 本发明涉及软件开发技术领域,具体而言,涉及一种基于元数据驱动的低代码软件开发方法。

### 背景技术

[0002] 在当前的软件开发领域,低代码开发方法已成为快速构建应用程序的重要手段,这种方法使业务人员能够通过图形化界面和少量低代码开发而不是传统的需要编写大量代码的方式来构建应用,软件开发效率得到了大幅度提高。然而,现有的低代码开发方法受限于低代码平台的实现方式,在处理大型、复杂的业务场景方面存在显著的限制,低代码平台的灵活性和可扩展性也需要更先进的低代码引擎进行改善和提高。因此,急需设计一种通过模型驱动架构的软件设计方法,旨在通过元数据定义模型的低代码软件开发方法,针对构建复杂业务应用,满足高级业务逻辑处理和数据集成的需求。

### 发明内容

[0003] 为了解决上述问题,本发明的目的是提供一种基于元数据驱动的低代码软件开发方法,利用元数据来驱动应用的动态生成和运行,从而进一步降低了代码编写的需求,加快了软件开发的进度。

[0004] 为了实现上述技术目的,本申请提供了一种基于元数据驱动的低代码软件开发方法,包括以下步骤:

[0005] 建立低代码开发工作站,用于为低代码开发者提供交互式图形界面软件需求的收集和设计,包括,数据模型结构设计、业务流程设计、界面和表单设计;

[0006] 基于低代码开发工作站,构建元数据模型中央仓库,用于保存和管理低代码开发后的模型元数据,包括模型和模型属性的定义、模型的存储结构定义和模型的国际化定义;

[0007] 基于元数据模型中央仓库,设计低代码运行框架,通过设置表单和UI渲染引擎、低代码运行引擎、客制化扩展点以及工作流引擎,构建用于部署前端应用的平台基础框架;

[0008] 基于低代码运行框架,生成应用中心,用于对基于元数据驱动的低代码软件开发出的各类业务应用进行自动化部署。

[0009] 优选地,在实现模型和模型属性的定义的过程中,通过UI界面创建新的模型后,将模型和模型属性的定义进行保存,支持设置模型的关联关系和访问控制的配置,实现模型和模型属性的定义。

[0010] 优选地,在实现模型的存储结构定义的过程中,通过定义模型在关系型数据库中的存储方式,包括模型所对应的表、模型属性所对应表的字段,实现模型的存储结构定义。

[0011] 优选地,在实现模型的国际化定义的过程中,通过定义模型和模型属性根据各国语言的翻译后的内容,包括模型名的翻译内容和模型属性的翻译内容,实现模型的国际化定义。

[0012] 优选地,在设计低代码运行框架的过程中,通过图形界面创建和配置表单,定义字

段和布局,而无需编写代码,并通过UI渲染引擎将设计完成的表单和其他页面转换成实际的用户界面,处理用户的输入,并将操作结果反馈给系统,完成表单和UI渲染引擎的设计。

[0013] 优选地,在设计表单和UI渲染引擎的过程中,表单和UI渲染引擎的功能包括:

[0014] 表单布局控制:支持通过界面操作来设定表单的整体布局和结构,包括字段和部分的位置、大小和顺序;

[0015] 表单设计界面:提供直观的图形用户界面,用户通过拖放组件的方式来设计表单,无需编写代码;

[0016] 表单列定义:使用户能够定义表单中各个字段的列属性,以及如何在表单中显示这些字段;

[0017] 表单个性化视图:支持创建自定义的表单视图,选择哪些字段展示以及它们的布局,以满足特定角色或工作流程的需求;

[0018] 表单关联列表定义:在表单中支持定义与表单相关联业务对象的列表,包括相关记录的显示和关联数据的管理;

[0019] 表单个性化区块定义:支持个性化定义表单中的部分区域,以逻辑地组织信息并提高用户界面的可读性,并支持以HTML和JavaScript的方式动态编写前端组件;

[0020] 列表定义:支持设定列表视图,包括列表中应显示的字段,字段的排序和过滤器设置;

[0021] 列表元素定义:支持定义列表中各个元素,包括格式、样式和交互行为。

[0022] 优选地,在设计低代码运行框架的过程中,通过低代码运行引擎打通前端界面和后端业务逻辑,使得业务逻辑的扩展和修改只需要通过脚本编写即可完成。

[0023] 优选地,在设计低代码运行框架的过程中,通过整合面向切面编程,设计客制化扩展点,用于支持在业务数据的创建、更新、删除或展示数据的前后进行客制化规则扩展点的自定义设置,包括数据的处理与验证,以及基于特定条件控制规则的执行,其中,

[0024] 在服务端,通过客制化扩展点,在数据的生命周期内进行客制化规则的定义,包括数据处理与验证以及执行条件控制,通过根据自定义条件执行相应的业务规则,同时,根据在用户界面中添加的自定义按钮和链接,通过客户端或服务端脚本实现自定义功能的触发;

[0025] 在客户端,通过客制化扩展点,对客户端低代码进行编写,用于增强UI表单的交互性,包括前端数据验证、表单项联动和用户体验的增强,并将客户端低代码配置为在不同的表单事件触发,同时设置触发条件,其中,UI策略用于控制和管理用户界面元素的行为,包括字段的显示/隐藏、编辑权限和必填条件,并支持基于自定义条件的控制执行。

[0026] 优选地,在构建平台基础框架的过程中,平台基础组件包括:

[0027] 多租户管理:用于以租户为单位,对平台功能和数据进行隔离;

[0028] 用户管理:用于生成新的用户账户,并管理现有账户;

[0029] 角色管理用于定义、分配和管理用户角色,并结合用户组对用户群组的分组管理和权限控制,简化访问权限的分配,其中,

[0030] 权限管理,包括:

[0031] RBAC权限管理,用于对角色的访问控制,支持对不同角色的特定实体设置权限;

[0032] ACL权限管理:用于对表、字段和记录级别进行细粒度的访问控制,包括对特定对

象执行特定操作的授权,基于自定义条件的授权,对RESTAPI的授权;

[0033] 数据权限管理:用于对多维度的数据的可见性进行权限控制。

[0034] 优选地,在设计低代码运行框架的过程中,通过定义和建模复杂的工作流程,并设计多种类型的工作流节点以及图形化流程,对工作流引擎进行设计,其中,定义和建模复杂的工作流程,包括:任务、决策点、分支和循环,并设计工作流调度器对工作流的任务创建、执行和运行过程进行监控;多种类型的工作流节点包括运行脚本、输出日志、条件分支和审批;基于可视化的用户界面,通过拖放方式设计和定制工作流程,完成图形化流程设计。

[0035] 本发明公开了以下技术效果:

[0036] 本发明简化了软件开发流程,有效缩短了从设计到部署的整个开发周期。传统的开发模式需要经过原型设计、产品需求设计、概要设计、详细设计、数据库设计、编码、测试、部署、上线等完整的阶段。本发明通过元数据驱动的低代码开发方式和通用的平台基础组件,减少和简化了传统开发中的各个环节等,加快了对业务需求变化的响应速度。

[0037] 本发明提高了代码的可复用性。通过引入基于元数据的页面动态生成技术和模块化功能,使产品功能更易于管理和重用,减少了因理解和修改旧代码所带来的额外成本。

[0038] 在维护成本方面,本发明通过大幅度传统的前后端代码量和提高代码质量,显著降低了长期的维护成本。减少了因业务变更导致的大量代码重写需求,使得系统架构更加稳定,减少了潜在的系统缺陷和不稳定性。

[0039] 本发明通过提高开发效率、代码复用性和降低维护成本,提升了软件的整体成熟度,增强了系统的稳定性和可靠性,减少了由于缺陷修复和功能调整导致的重复测试周期,为用户带来了更加稳定和可靠的软件产品。

## 附图说明

[0040] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0041] 图1是本发明所述的低代码软件开发方法的技术架构图;

[0042] 图2是本发明所述的客制化扩展点的设计示意图。

## 具体实施方式

[0043] 为使本申请实施例的目的、技术方案和优点更加清楚,下面将结合本申请实施例中附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。通常在此处附图中描述和示出的本申请实施例的组件可以以各种不同的配置来布置和设计。因此,以下对在附图中提供的本申请的实施例的详细描述并非旨在限制要求保护的本申请的范围,而是仅仅表示本申请的选定实施例。基于本申请的实施例,本领域技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0044] 本发明提供了一种基于元数据驱动的低代码软件开发方法,包括以下步骤:

[0045] 建立低代码开发工作站,用于为低代码开发者提供交互式图形界面软件需求的收

集和设计,包括,数据模型结构设计、业务流程设计、界面和表单设计;

[0046] 基于低代码开发工作站,构建元数据模型中央仓库,用于保存和管理低代码开发后的模型元数据,包括模型和模型属性的定义、模型的存储结构定义和模型的国际化定义;

[0047] 基于元数据模型中央仓库,设计低代码运行框架,通过设置表单和UI渲染引擎、低代码运行引擎、客制化扩展点以及 workflow 引擎,构建用于部署前端应用的平台基础框架;

[0048] 基于低代码运行框架,生成应用中心,用于对基于元数据驱动的低代码软件开发出的各类业务应用进行自动化部署。

[0049] 进一步优选地,本发明提到的低代码软件开发方法,在实现模型和模型属性的定义的过程中,通过UI界面创建新的模型后,将模型和模型属性的定义进行保存,支持设置模型的关联关系和访问控制的配置,实现模型和模型属性的定义。

[0050] 进一步优选地,本发明提到的低代码软件开发方法,在实现模型的存储结构定义的过程中,通过定义模型在关系型数据库中的存储方式,包括模型所对应的表、模型属性所对应表的字段,实现模型的存储结构定义。

[0051] 进一步优选地,本发明提到的低代码软件开发方法,在实现模型的国际化定义的过程中,通过定义模型和模型属性根据各国语言的翻译后的内容,包括模型名的翻译内容和模型属性的翻译内容,实现模型的国际化定义。

[0052] 进一步优选地,本发明提到的低代码软件开发方法,在设计低代码运行框架的过程中,通过图形界面创建和配置表单,定义字段和布局,而无需编写代码,并通过UI渲染引擎将设计完成的表单和其他页面转换成实际的用户界面,处理用户的输入,并将操作结果反馈给系统,完成表单和UI渲染引擎的设计。

[0053] 进一步优选地,本发明提到的低代码软件开发方法,在设计表单和UI渲染引擎的过程中,表单和UI渲染引擎的功能包括:

[0054] 表单布局控制:支持通过界面操作来设定表单的整体布局和结构,包括字段和部分的位置、大小和顺序;

[0055] 表单设计界面:提供直观的图形用户界面,用户通过拖放组件的方式来设计表单,无需编写代码;

[0056] 表单列定义:使用户能够定义表单中各个字段的列属性,以及如何在表单中显示这些字段;

[0057] 表单个性化视图:支持创建自定义的表单视图,选择哪些字段展示以及它们的布局,以满足特定角色或 workflow 的需求;

[0058] 表单关联列表定义:在表单中支持定义与表单相关联业务对象的列表,包括相关记录的显示和关联数据的管理;

[0059] 表单个性化区块定义:支持个性化定义表单中的部分区域,以逻辑地组织信息并提高用户界面的可读性,并支持以HTML和JavaScript的方式动态编写前端组件;

[0060] 列表定义:支持设定列表视图,包括列表中应显示的字段,字段的排序和过滤器设置;

[0061] 列表元素定义:支持定义列表中各个元素,包括格式、样式和交互行为。

[0062] 进一步优选地,本发明提到的低代码软件开发方法,在设计低代码运行框架的过程中,通过低代码运行引擎打通前端界面和后端业务逻辑,使得业务逻辑的扩展和修改只

需要通过脚本编写即可完成。

[0063] 进一步优选地,本发明提到的低代码软件开发方法,在设计低代码运行框架的过程中,通过整合面向切面编程,设计客制化扩展点,用于支持在业务数据的创建、更新、删除或展示数据的前后进行客制化规则扩展点的自定义设置,包括数据的处理与验证,以及基于特定条件控制规则的执行,其中,

[0064] 在服务端,通过客制化扩展点,在数据的生命周期内进行客制化规则的定义,包括数据处理与验证以及执行条件控制,通过根据自定义条件执行相应的业务规则,同时,根据在用户界面中添加的自定义按钮和链接,通过客户端或服务端脚本实现自定义功能的触发;

[0065] 在客户端,通过客制化扩展点,对客户端低代码进行编写,用于增强UI表单的交互性,包括前端数据验证、表单项联动和用户体验的增强,并将客户端低代码配置为在不同的表单事件触发,同时设置触发条件,其中,UI策略用于控制和管理用户界面元素的行为,包括字段的显示/隐藏、编辑权限和必填条件,并支持基于自定义条件的控制执行。

[0066] 进一步优选地,本发明提到的低代码软件开发方法,在构建平台基础框架的过程中,平台基础组件包括:

[0067] 多租户管理:用于以租户为单位,对平台功能和数据进行隔离;

[0068] 用户管理:用于生成新的用户账户,并管理现有账户;

[0069] 角色管理用于定义、分配和管理用户角色,并结合用户组对用户群组的分组管理和权限控制,简化访问权限的分配,其中,

[0070] 权限管理,包括:

[0071] RBAC权限管理,用于对角色的访问控制,支持对不同角色的特定实体设置权限;

[0072] ACL权限管理:用于对表、字段和记录级别进行细粒度的访问控制,包括对特定对象执行特定操作的授权,基于自定义条件的授权,对RESTAPI的授权;

[0073] 数据权限管理:用于对多维度的数据的可见性进行权限控制。

[0074] 进一步优选地,本发明提到的低代码软件开发方法,在设计低代码运行框架的过程中,通过定义和建模复杂的工作流程,并设计多种类型的工作流节点以及图形化流程,对工作流引擎进行设计,其中,定义和建模复杂的工作流程,包括:任务、决策点、分支和循环,并设计工作流调度器对工作流的任务创建、执行和运行过程进行监控;多种类型的工作流节点包括运行脚本、输出日志、条件分支和审批;基于可视化的用户界面,通过拖放方式设计和定制工作流程,完成图形化流程设计。

[0075] 本发明还公开了一种基于元数据驱动的低代码软件开发系统,用于实现上述提到的低代码软件开发方法,包括:

[0076] 工作站构建模块,用于建立低代码开发工作站,用于为低代码开发者提供交互式图形界面软件需求的收集和 design,包括,数据模型结构设计、业务流程设计、界面和表单设计;

[0077] 元数据模型中央仓库构建模块,用于基于低代码开发工作站,构建元数据模型中央仓库,用于保存和管理低代码开发后的模型元数据,包括模型和模型属性的定义、模型的存储结构定义和模型的国际化定义;

[0078] 运行框架设计模块,用于基于元数据模型中央仓库,设计低代码运行框架,通过设

置表单和UI渲染引擎、低代码运行引擎、客制化扩展点以及 workflow 引擎,构建用于部署前端应用的平台基础框架;

[0079] 应用部署模块,用于基于低代码运行框架,生成应用中心,用于对基于元数据驱动的低代码软件开发出的各类业务应用进行自动化部署。

[0080] 实施例1:如图1所示,本发明提供了一种基于元数据驱动的低代码软件开发方法,该方法利用元数据来驱动应用的动态生成和运行,从而进一步降低了代码编写的需求,加快了软件开发的进度,具体方案如下:

[0081] 1、低代码开发工作站的开发:

[0082] 低代码开发者在工作站中通过交互式图形界面软件需求的收集和的设计,包括:数据模型结构设计、业务流程设计、界面和表单设计。

[0083] 2、元数据模型中央仓库的开发:

[0084] 保存和管理,低代码开发后的模型元数据,内容包括:

[0085] 模型和模型属性的定义:

[0086] 通过UI界面创建新的模型后,将模型和模型属性的定义进行保存,支持设置模型的关联关系、访问控制等配置。可为模型表加不同类型的属性,如文本、数字、日期等多种类型,支持设置字段的属性和行为,如:属性之间的联动规则、属性的可选项等。

[0087] 模型的存储结构定义:

[0088] 定义模型在关系型数据库中的存储方式,包括模型所对应的表、模型属性所对应表的字段。

[0089] 模型的国际化定义:

[0090] 定义模型和模型属性根据各国语言的翻译后的内容,包括模型名的翻译内容和模型属性的翻译内容。

[0091] 3、低代码运行框架的开发:

[0092] 3.1、表单和UI渲染引擎:

[0093] 通过图形界面创建和配置表单,定义字段和布局,而无需编写代码,并通过UI渲染引擎将设计完成的表单和其他页面转换成实际的用户界面,处理用户的输入,并将操作结果反馈给系统。

[0094] 3.2、低代码运行引擎:

[0095] 框架使用JavaScript作为低代码编程语言,通过低代码运行引擎打通前端界面和后端业务逻辑,使低代码开发人员对业务逻辑的扩展和修改只需要通过简单的脚本编写即可完成,而不需要像传统开发经过代码编写、编译、打包、上线的复杂流程。

[0096] 3.3、客制化扩展点:

[0097] 支持服务端和客户端的客制化扩展处理。在服务端,它允许在数据的生命周期内(如创建、更新、删除或显示前后)进行客制化规则的定义,包括数据处理与验证以及执行条件控制。用户可以根据自定义条件执行相应的业务规则。同时,支持在用户界面中添加自定义按钮和链接,并通过客户端或服务端脚本实现更复杂的操作,如自定义功能的触发。

[0098] 在客户端,用户可以编写客户端低代码来增强UI表单的交互性,例如前端数据验证、表单项联动和用户体验的增强。这些低代码可以配置为在不同的表单事件(如字段更改、表单加载、表单提交)触发,并可设置触发条件。UI策略则用于控制和管理用户界面元素

的行为,如字段的显示/隐藏、编辑权限和必填条件,同样支持基于自定义条件的控制执行。

[0099] 3.4、平台基础组件:

[0100] 平台基础组件包括:

[0101] 多租户管理:支持以租户为单位,对平台功能和数据进行隔离。

[0102] 用户管理:创建新的用户账户,并管理现有账户,属性包括:姓名、联系方式、部门归属、角色等。

[0103] 角色管理:支持定义、分配和管理用户角色、结合用户组 (User Group) 实现对用户群组的分组管理和权限控制,简化访问权限的分配。

[0104] 权限管理:

[0105] 1) RBAC权限,基于角色的访问控制支持对不同角色的特定实体设置权限,如菜单、服务目录、按钮操作等;

[0106] 2) ACL权限:对表、字段和记录级别进行细粒度的访问控制,包括对特定对象(模型、模型属性、视图等)执行特定操作(如查看、创建、编辑、删除)的授权,基于自定义条件的授权,对RESTAPI的授权等;

[0107] 3) 数据权限:支持以公司、部门、项目、应用、个人为维度的数据可见性的权限控制。

[0108] 工作流引擎:

[0109] 1) 支持定义和建模复杂的工作流程,包括:任务、决策点、分支和循环等。由工作流调度器实现 workflows 的任务创建、执行和运行过程的监控。

[0110] 2) 支持多种类型的工作流节点,如:运行脚本、输出日志、条件分支、审批等。

[0111] 3) 图形化流程设计:支持通过可视化的用户界面,通过拖放方式设计和定制工作流程。

[0112] 5、应用中心的开发:

[0113] 基于元数据驱动的低代码软件开发方法开发出的各类业务应用,支持在应用商店中上线应用,由平台自动对需要投产使用的应用进行自动化部署,终端用户可在登录平台后使用应用中心的各业务应用的功能。

[0114] 对于本发明提到的一种基于元数据驱动的低代码软件开发方法,还具备如下特征:

[0115] 1、元数据驱动:

[0116] 基于元数据模型的定义自动渲染各模型对应的列表页面和表单页面。与传统的编写实现功能不同,平台基于低代码运行框架不需要为每个模型或业务功能编写不同的代码,也不需要与市面上其他的低代码平台一样,需要根据代码模型生成代码并运行系统。基于元数据驱动的低代码引擎,所有的业务功能只需要一套低代码运行框架和一套代码,即可完成所有业务功能对应页面的动态生成和业务功能的实现,对应不同业务功能的个性化处理部分,框架支持通过定义前端或后端的扩展点的方式实现。

[0117] 核心元数据对象的数据结构如表1-4所示:

[0118] 表1模型定义

	元数据名称	元数据定义说明
	模型名	模型的显示名称
	模型编码	模型的唯一英文编码
[0119]	模型所属应用	模型所属的应用
	模型访问控制	哪些角色可对模型进行哪些访问操作，包括：增删改查。
	模型继承关系	模型是否支持继承关系，指定从哪个父模型继承

[0120] 表2模型属性定义

	元数据名称	元数据定义说明
	属性名	模型属性的显示名称
	属性编码	模型属性的唯一英文编码
[0121]	属性类型	支持类型包括：字符串、数字、货币、日期、时间、多行文本等多种不同的类型。
	属性限制条件	指定模型属性的限制，是否只读、是否为必填字段、是否在页面隐藏、最大输入长度、默认值。
	属性联动字段	指定属性的输入与哪个指定的字段联动

[0122] 表3模型存储结构定义

[0123]	元数据名称	元数据定义说明
	存储表名	在关系型数据库中存储表的名称。
	存储列名	在关系型数据库中存储表的列名。
	模型名	在平台中对应模型的名称。
	模型属性名	在平台中对应模型属性的名称。

[0124] 表4模型国际化定义

元数据名称	元数据定义说明
模型名称	模型的唯一英文编码
模型属性名称	模型属性的唯一英文编码
国际化语言类型	国际化语言类型的编码，如：中文、英语、日文等
国际化翻译内容	模型属性对应国际化语言类型的翻译内容

[0125] 2、低代码引擎：

[0126] 2.1、表单和UI渲染引擎

[0127] 表单和UI渲染引擎的功能包括：

[0128] 表单布局控制：支持通过界面操作来设定表单的整体布局和结构，包括字段和部分的位置、大小和顺序。

[0129] 表单设计界面：提供直观的图形用户界面，用户可以通过拖放组件的方式来设计表单，无需编写代码。

[0130] 表单列定义：使用户能够定义表单中各个字段的列属性，如列宽和列顺序，以及在表单中显示这些字段。

[0131] 表单个性化视图：支持创建自定义的表单视图，选择哪些字段展示以及它们的布局，以满足特定角色或工作流程的需求。

[0132] 表单关联列表定义：在表单中支持定义与表单相关联业务对象的列表，包括相关记录的显示和关联数据的管理。

[0133] 表单个性化区块定义：支持个性化定义表单中的部分区域，如信息块或者分组，以逻辑地组织信息并提高用户界面的可读性，并支持以HTML和JavaScript的方式动态编写前端组件。

[0134] 列表定义：支持设定列表视图，包括列表中应显示的字段，字段的排序和过滤器设置。

[0135] 列表元素定义：支持定义列表中各个元素，如行和列的界面元素，包括格式、样式和交互行为。

[0136] 2.2、客制化扩展点：

[0137] 本发明支持客制化扩展点来实现个性化功能，实现原理为整合面向切面编程(AOP)来实现功能增强，如图2所示。

[0138] 支持在业务数据的创建、更新、删除或展示数据的前后进行客制化规则扩展点的自定义设置。在实际的低代码开发过程中，客制化扩展点包括数据的处理与验证，以及基于特定条件控制规则的执行，例如，可定义在添加新记录前检查数据的完整性，或在删除记录后清理关联数据。

[0139] 相对于传统的开发模式需要经过原型设计、产品需求设计、概要设计、详细设计、

数据库设计、编码、测试、部署、上线等完整的阶段。本发明通过元数据驱动的低代码开发方式和通用的平台基础组件,减少和简化了传统开发中的各个环节等,加快了对业务需求变化的响应速度

[0141] 本发明通过提高开发效率、代码复用性和降低维护成本,提升了软件的整体成熟度,增强了系统的稳定性和可靠性,减少了由于缺陷修复和功能调整导致的重复测试周期,为用户带来了更加稳定和可靠的软件产品。

[0142] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0143] 在本发明的描述中,需要理解的是,术语“第一”、“第二”仅用于描述目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个该特征。在本发明的描述中,“多个”的含义是两个或两个以上,除非另有明确具体的限定。

[0144] 显然,本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样,倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内,则本发明也意图包含这些改动和变型在内。

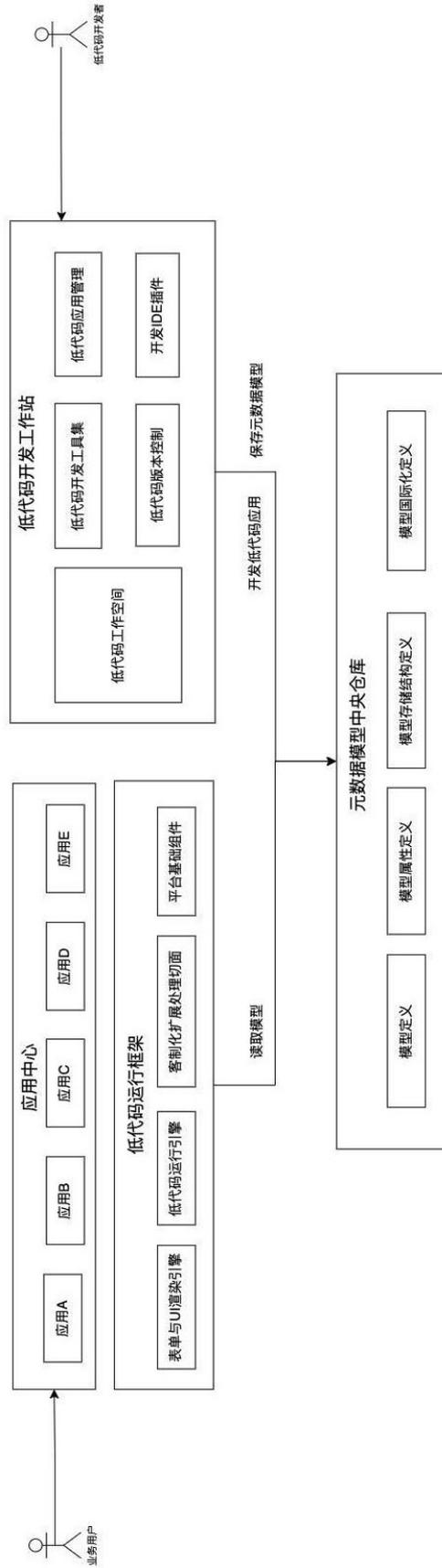


图 1

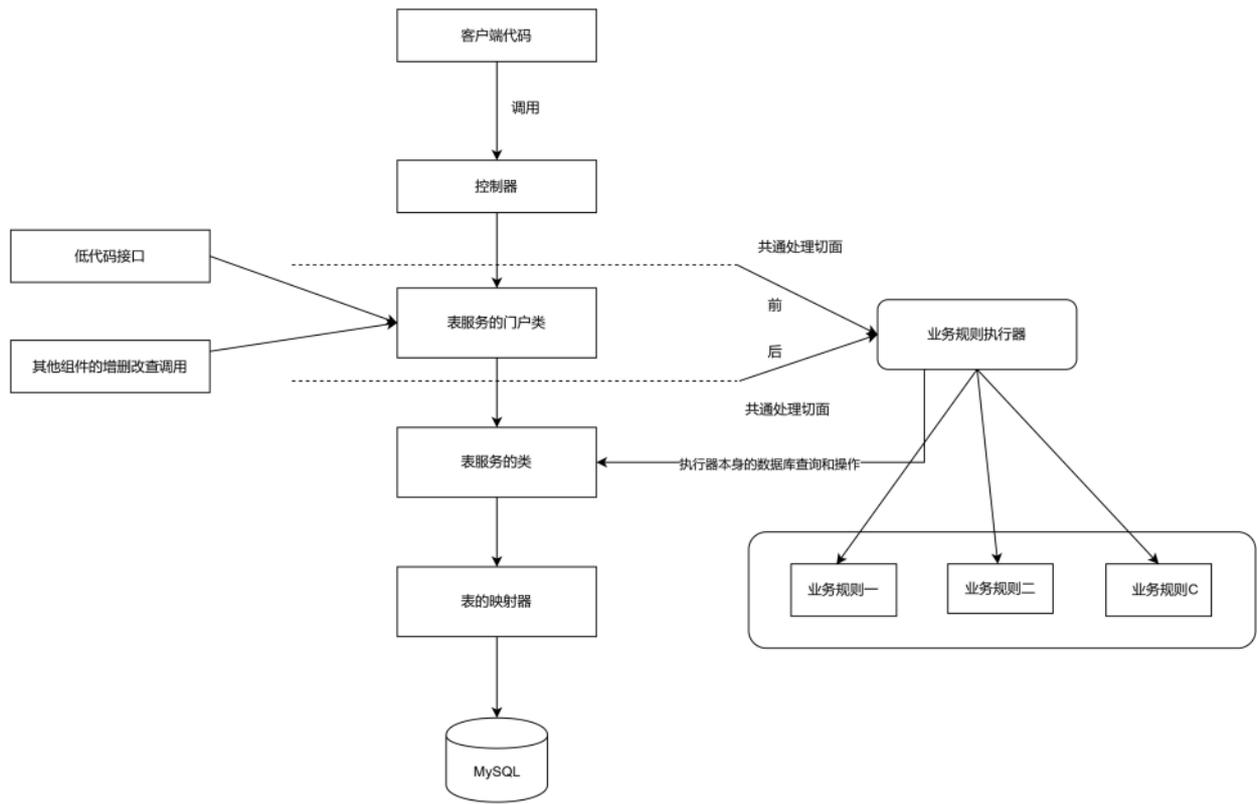


图 2