



(12) 发明专利

(10) 授权公告号 CN 111831258 B

(45) 授权公告日 2024. 07. 02

(21) 申请号 202010284995.4

(22) 申请日 2020.04.13

(65) 同一申请的已公布的文献号
申请公布号 CN 111831258 A

(43) 申请公布日 2020.10.27

(30) 优先权数据
19000184.2 2019.04.15 EP

(73) 专利权人 门德斯科技有限公司
地址 荷兰鹿特丹

(72) 发明人 安德烈·科勒韦因

(74) 专利代理机构 北京康信知识产权代理有限
责任公司 11240
专利代理师 陈方鸣

(51) Int.Cl.

G06F 8/20 (2018.01)

G06F 8/34 (2018.01)

G06F 8/41 (2018.01)

(56) 对比文件

US 2007234290 A1, 2007.10.04

审查员 张露

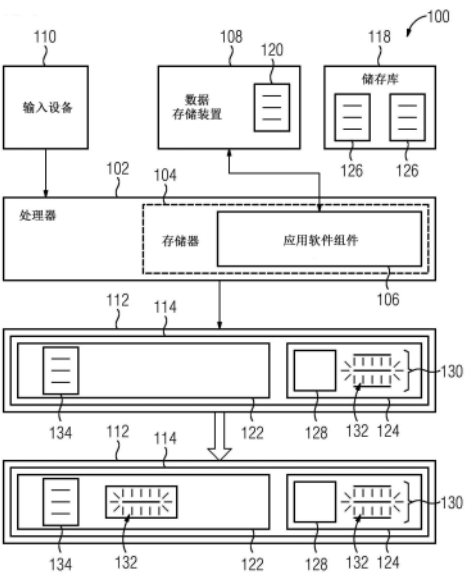
权利要求书2页 说明书12页 附图18页

(54) 发明名称

创建应用程序的方法和系统

(57) 摘要

本发明公开了一种创建应用程序的方法和系统。计算机实施的方法包括：向用户提供应用程序开发用户界面 (UI)，用于开发应用程序；向用户提供搜索UI，用于在存储可重复使用于应用程序开发的工件的储存库中搜索；响应于用户与搜索UI的交互，捕获用户的搜索意图；将对应于所捕获的用户的搜索意图的至少一个工件从储存库导入到应用程序开发UI；以及通过使用所导入的至少一个工件通过应用程序开发UI来开发应用程序。



1. 一种创建应用程序的计算机实施的方法,包括:
向用户提供应用程序开发用户界面,用于开发所述应用程序;
向所述用户提供搜索界面,用于在存储可重复使用于应用程序开发的工件的储存库中搜索;
响应于用户与所述搜索界面的交互,捕获所述用户的搜索意图;
将对应于所捕获的用户的搜索意图的至少一个工件从所述储存库导入到所述应用程序开发界面;以及
通过使用所导入的至少一个工件通过所述应用程序开发界面来开发所述应用程序;
部署所创建的应用程序,其中部署所述创建的应用程序包括将所述应用程序发布给用户,由所述用户安装和激活所发布的应用程序,并且在成功部署之后,所创建且所部署的应用程序的至少一部分作为所述工件被存储在储存库中,被用于重复使用于应用程序开发。
2. 根据权利要求1所述的计算机实施的方法,还包括:
在所述搜索界面中显示存储在所述储存库中且对应于所捕获的用户的搜索意图的一组工件;以及
响应于所述用户与所述搜索界面的交互,捕获所述用户的意图以重复使用所显示的一组工件中的至少一个工件。
3. 根据权利要求2所述的计算机实施的方法,还包括:
检索存储在所述储存库中的所述工件的元数据;
确定存储在所述储存库中的所述工件的排名,所述排名对应于所检索的元数据与所捕获的用户的搜索意图的符合度;以及
根据所确定的所述工件的排名显示所述一组工件。
4. 根据权利要求3所述的计算机实施的方法,所述元数据包括以下中的至少一个:工件所有者、工件标签、工件描述、工件关键词、工件流行度、工件版本、工件相似度、工件目的、工件区域或它们的任何组合。
5. 根据前述权利要求中任一项所述的计算机实施的方法,其中所述工件包括以下中的至少一个:应用程序数据、应用程序组件、应用程序架构、应用程序编程接口、应用程序服务、应用程序用法、应用程序链接、应用程序描述、应用程序相关性、工件相关性、应用程序环境、应用程序标签、业务事件、通知、应用程序界面、触发信息、用户界面设计、知识图、可重复使用IT资产图、到页面的链接或它们的任何组合。
6. 根据权利要求1-4中任一项所述的计算机实施的方法,其中创建所述应用程序包括使用基于视觉模型表示。
7. 根据权利要求1-4中任一项所述的计算机实施的方法,其中所创建的应用程序和/或存储在所述储存库中的所述工件包括与以下中的至少一个有关的标准化模块:数据提供者、数据消费者、数据更新机制、数据格式、可用数据处置操作、应用程序身份和访问管理或它们的任何组合。
8. 根据权利要求1-4中任一项所述的计算机实施的方法,还包括:
在所述储存库中存储所创建的和部署的应用程序的所存储的至少一部分的元数据。
9. 根据权利要求8所述的计算机实施的方法,其中所存储的元数据包括以下中的至少一个:部署信息、应用程序服务、以及关于所创建的和部署的应用程序的所存储的至少一部

分与已经用于开发所创建的和部署的应用程序的所导入的至少一个工件之间的相关性的信息、或它们的任何组合。

10. 根据权利要求8所述的计算机实施的方法,还包括:

在修订所开发的和部署的应用程序中的至少一部分时,更新所述储存库中的所存储的至少一部分,并且/或者

在修订所存储的至少一部分的元数据时,更新所述储存库中的所存储的至少一部分的所存储的元数据。

11. 根据权利要求8所述的计算机实施的方法,其中所存储的至少一部分的所存储的元数据包括关于用于访问所存储的至少一部分的访问权限的信息;并且其中所存储的至少一部分仅对于具有合适访问权限的用户是可访问的。

12. 根据权利要求8所述的计算机实施的方法,其中所存储的至少一部分的所存储的元数据包括关于用于部署和/或使用应用程序的访问权限的信息,其中使用所存储的至少一部分来开发所述应用程序;

其中所存储的至少一部分仅针对具有合适访问权限的用户是可访问的,并且/或者所开发的应用程序仅能够由具有合适访问权限的用户部署和/或使用。

13. 根据权利要求1-4中任一项所述的计算机实施的方法,其中所创建的应用程序适合于部署到包括智能电话、手持式计算机、平板计算机、台式计算机、智能手表、TV的硬件设备,部署到云计算平台或到物联网操作系统。

14. 一种被布置和配置为执行根据前述权利要求中任一项所述的计算机实施的方法的步骤的计算机系统。

15. 一种编码有可执行指令的计算机可读介质,当执行所述指令时,使得根据权利要求14所述的计算机系统执行根据权利要求1至13中任一项所述的方法。

创建应用程序的方法和系统

技术领域

[0001] 本公开主要涉及软件开发系统,特别地涉及用于开发应用程序的系统,该系统(在本文中统称为产品系统)可用于构建、测试、部署和迭代此类应用程序。

背景技术

[0002] 最近,越来越多的计算机软件产品以应用程序(在本专利文档中简称为“app”)的形式既用于个人需要又用于业务需要。此类应用程序可用于移动情境,并且可提供具体的一组功能。本发明主要涉及此类应用程序的开发。

[0003] 当前,存在支持应用程序开发的产品系统和解决方案。此类产品系统可从改进中受益。

发明内容

[0004] 各种公开的实施方式包括可用于便于创建应用程序的方法和计算机系统。

[0005] 根据本发明的第一方面,创建应用程序的计算机实施的方法可包括:向用户提供应用程序开发用户界面(UI)以用于开发应用程序;向用户提供搜索UI,用于在存储可重复使用于应用程序开发的工件的储存库中搜索;响应于用户与搜索UI的交互,捕获用户的搜索意图;将对应于所捕获的用户的搜索意图的至少一个工件从储存库导入到应用程序开发UI;以及通过使用所导入的至少一个工件通过应用程序开发UI来开发应用程序。以示例的方式,可通过至少一个处理器的操作执行所描述的计算机实施的方法。

[0006] 根据本发明的第二方面,一种计算机系统可被布置和配置为执行该创建应用程序的计算机实施的方法的步骤。特别地,所描述的计算机系统可被布置和配置为执行以下步骤:向用户提供应用程序开发用户界面(UI)用于开发应用程序;向用户提供搜索UI,用于在存储可重复使用于应用程序开发的工件的储存库中搜索;响应于用户与搜索UI的交互,捕获用户的搜索意图;将对应于所捕获的用户的搜索意图的至少一个工件从储存库导入到应用程序开发UI;以及通过使用所导入的至少一个工件通过应用程序开发UI来开发应用程序。以示例的方式,计算机系统可包括被配置为执行这些描述的步骤的至少一个处理器。

[0007] 根据本发明的第三方面,一种编码有可执行指令的计算机可读介质,当执行可执行指令时,使得所描述的计算机系统执行所描述的创建应用程序的方法。以示例的方式,所描述的计算机可读介质可为非暂态的,并且还可为存储设备上的软件组件。

[0008] 前述已经相当广泛地概述本公开的技术特征,使得本领域技术人员可更好地理解下文的详细描述。在下文中将描述形成权利要求主题的本公开的附加特征和优点。本领域技术人员将了解,他们可易于使用所公开的概念和具体实施方式作为用于修改或设计用于完成本公开的相同目的其它结构的基础。本领域技术人员也将意识到此类等同构造不脱离本公开最广泛形式的精神和范围。

[0009] 再者,在进行下面的详细描述之前,应当理解,在整个该专利文档中提供了某些词语和短语的各种定义,并且本领域普通技术人员将理解,此类定义在很多(即使不是大多

数) 实例中适用于此类定义的词语和短语的先前以及未来使用。虽然一些术语可包括多种多样的实施方式,但是所附权利要求可明确地将这些术语限制于具体实施方式。

附图说明

[0010] 图1示出便于在产品系统中创建应用程序的示例系统的功能框图。

[0011] 图2至图11示出包括结果的一系列示例性图形用户界面,这些结果可使用图1所示的产品系统显示给用户。

[0012] 图12示出便于在产品系统中创建应用程序的示例方法的流程图。

[0013] 图13示出其中可实施实施方式的数据处理系统的框图。

[0014] 图14至图18示出关于便于在产品系统中创建应用程序的另外的信息。

具体实施方式

[0015] 现在将参考附图描述涉及用于在产品系统中创建应用程序的系统和方法的各种技术,其中贯穿全文,相似的附图标记表示相似的元素。下面讨论的附图以及在本专利文档中用于描述本公开的原理的各种实施方式仅为例示的方式,并且不应当以任何方式被解释为限制本公开的范围。本领域技术人员将理解,可在任何合适布置的仪器中实施本公开的原理。应当理解,被描述为由某些系统元素完成的功能可由多个元素实行。类似地,例如,元素可被配置为实行被描述为由多个元素完成的功能。将参考示例性非限制性实施方式描述本专利文档的许多的创新性教导内容。

[0016] 应用程序通常是指软件程序,软件程序在执行时实行具体期望的任务。一般而言,在包含一个或多个操作系统(“OS”)、虚拟机(例如,支持Java™编程语言的)、设备驱动程序等的运行时环境中执行若干应用程序。

[0017] 可使用传统源代码创建、编辑和表示应用程序。此类传统源代码的示例包括C、C++、Java、Flash、Python、Perl和表示应用程序的其它基于脚本的方法。可通过经合适训练的用户的手动编码实现开发和创建此类基于脚本的应用程序或此类基于脚本的应用程序的部分。

[0018] 开发者经常使用应用程序开发框架(“ADF”)(其本身是应用程序或app)用于实施/开发期望的应用程序。ADF提供可在应用程序的开发中直接/间接使用的一组预定义代码/数据模块。ADF也可提供工具,诸如集成开发环境(“IDE”)、代码生成器、调试器等,其便于开发者以更快/更简单的方式编码/实施期望的应用程序逻辑。

[0019] 一般而言,ADF通过提供可重复使用组件来简化应用程序开发,可重复使用组件可由应用程序开发者通过例如选择组件以实行期望的任务且定义所选择组件的外观、行为和交互,用于定义用户界面(“UI”)和应用程序逻辑。一些ADF基于模型-视图-控制器设计模式,该模式促进松耦合和更容易的应用程序开发和维护。

[0020] 根据另一个方法,也可使用基于视觉模型的表示来创建、编辑和表示应用程序。与传统源代码实施方式不同,可通过在视觉建模环境内绘制、移动、连接和/或断开逻辑元素的视觉描绘来创建、编辑和/或表示此类应用程序。应用程序的基于视觉模型的表示可使用符号、形状、线、颜色、阴影、动画和/或其它视觉元素来表示逻辑、数据或内存结构或用户界面元素。为了对传统的基于脚本的应用程序编程,通常需要程序员根据复杂的一组编程语

法规则打出详细脚本。与此相反,在一些情况下,可通过连接各种逻辑元素(例如,动作块和/或决策块)以创建定义应用程序的操作的视觉流程图,完成对基于视觉模型的应用程序进行编程。类似地,与在脚本中打出详细的命令相反,在基于视觉模型的应用程序中定义数据结构(例如,变量类型、数据库对象或类别)和/或用户界面元素(例如,下拉框、列表、文本输入框)可通过在虚拟工作空间内绘制、放置或连接逻辑元素的视觉描绘完成。因而,与传统基于脚本的应用程序相比,基于视觉模型的应用程序可更直观地编程和/或编辑。

[0021] 为简洁起见,对“模型”、“视觉模型”或“应用程序”或“app”的引用可被理解为指基于视觉模型的应用程序,除非具体指示。在一些情况下,此类基于视觉模型的应用程序可表示用于在计算机系统上执行的完整的、独立的应用程序。基于视觉模型的应用程序也可表示被配置为实行某些任务或功能的离散模块,但是其不表示完整的应用程序,代替地,此类离散模块可插入到更大的应用程序中或与其它离散模块组合以实行更复杂的任务。此类离散模块的示例可包括用于验证ZIP编码、用于从天气预报来源接收关于当前天气的信息和/或用于呈现图形的模块。

[0022] 可以两种形式表示视觉模型:内部表示和一个或多个相关联的视觉表示。内部表示可为根据由建模环境用于捕获和定义应用程序(或应用程序的一部分)的操作的文件格式编码的文件。例如,内部表示可定义应用程序可接收什么输入、应用程序可提供什么输出、应用程序通过其可达到结果的算法和操作、应用程序可显示什么数据、应用程序可存储什么数据等。内部表示也可用于指导执行环境如何在运行时期间执行应用程序的逻辑。内部表示可以非人类可读代码(例如,二进制代码)的形式存储。内部表示也可根据二进制存储的JSON(java脚本对象表示法)格式和/或XML格式存储。在运行时,执行引擎可使用内部表示以编译和/或生成可执行机器代码,当由处理器执行可执行机器代码时,使得处理器实施模型的功能。

[0023] 内部表示可与一个或多个视觉表示相关联。视觉表示可包括视觉元素,视觉元素描绘应用程序的逻辑如何流动,但是其没有被设计为被编译或执行。这些视觉表示可包括例如向用户示出应用程序将如何运行的流程图或决策树。视觉模型也可视觉上描绘将从用户接收的数据、将被存储的数据、以及将显示给用户的数据。这些视觉模型也可交互式的,这允许用户以直观方式操纵模型。例如,视觉表示可被配置为默认显示一定水平的细节(例如,分支的数量、显示的参数的数量、显示的逻辑的粒度)。然而,用户可与视觉表示交互,以便示出期望水平的细节-例如,用户可显示或隐藏逻辑的分支和/或显示或隐藏参数集合。与视觉模型的元素相关的细节可默认隐藏不可见,但是可出现在当用户点击适当元素时出现在屏幕上的滑动窗或弹出窗口中。用户也可放大或缩小模型,并且/或者平移模型的不同部分,以检查模型的不同部分。用户也可将逻辑的分支从模型的一个部分复制或粘贴到另一个部分中,或者将逻辑的分支从第一模型复制/粘贴到第二模型中。在一些情况下,模型的部分可包含到模型的其它部分的链接,使得如果用户点击链接,用户将自动被带到模型的另一个部分。如果模型被显示在建模环境中,则观看用户可以观看用户可与模型交互的相同方式中的至少一些方式与视觉表示交互。换句话说,视觉表示可被配置为模仿当模型被显示在视觉建模环境内时模型将如何表现。单个内部表示可对应于使用不同样式或格式规则以显示应用程序逻辑的多个视觉表示。例如,对应于相同内部表示的多个视觉表示可在其颜色使用、包括或省略的元素、以及用于描绘逻辑流程的符号、形状、线、颜色

和/或阴影的使用上彼此不同。

[0024] 参考图1,示出便于创建应用程序的示例计算机系统或数据处理系统100。处理系统100可包括至少一个处理器102,其被配置为执行来自由处理器102访问的存储器104的至少一个应用软件组件106。应用软件组件106可被配置(即,编程)为促使处理器102完成本文所描述的各种动作和功能。例如,所描述的应用软件组件106可包括和/或对应于应用程序开发应用程序的一个或多个组件,其被配置为生成产品数据,并且将产品数据存储于数据存储装置108(诸如数据库)中。

[0025] 可适于包括本文所描述的应用程序创建特征的产品系统的示例可包括荷兰鹿特丹的Mendix Technology B.V.的低代码软件平台。该平台提供用于构建、测试、部署和迭代应用程序的工具,并且基于视觉模型驱动的软件开发。然而,应当了解,本文所描述的系统和方法可用于其它产品系统(例如,PLM、PDM、ERP、ALM系统)和/或生成产品数据并将产品数据存储于数据库中的任何其它类型的系统。再者,可被用作本文所描述的一个或多个数据存储装置的数据库的示例包括数据库服务器应用程序,诸如Oracle、微软SQL服务器或可运行为存储数据记录的任何其它类型的数据存储装置。

[0026] 以示例的方式,创建应用程序包括使用例如包括在视觉模型驱动的软件开发平台(例如,低代码软件平台)中的基于视觉模型的表示。在一些实施方式中,基于视觉模型的表示可为上面描述的那些,可用于创建、编辑和表示应用程序。另外,视觉模型驱动的软件开发平台可例如提供用于构建、测试、部署和/或迭代应用程序的工具。

[0027] 应当了解,在复杂应用程序开发环境中创建应用程序困难且耗时。例如,可需要用户的高级编码知识,或者需要有意识地作出很多选项的选择,两者都涉及很多手动步骤,这是长期且不是有效的过程。

[0028] 为了使得能够增强应用程序的创建,所描述的产品系统或处理系统100可包括至少一个输入设备110和至少一个显示设备112(诸如显示屏)。所描述的处理器102可被配置为通过显示设备112生成GUI 114。这样的GUI可包括可由用户使用以通过输入设备110提供促使创建应用程序的输入的GUI元素(诸如按钮、链接、搜索框、列表、文本框、图像、滚动条)。特别地,GUI可包括用于用户开发应用程序的应用程序开发UI和用于用户在用于存储可重复使用于应用程序开发的工件的储存库118中搜索的搜索UI。

[0029] 在示例实施方式中,应用软件组件106和/或处理器102可被配置为将应用程序开发UI 122提供到用户用于开发应用程序120,并且将搜索UI 124提供到用户用于在用于存储可重复使用于应用程序开发的工件126的储存库118中搜索。为了增强应用程序120的创建,应用软件组件106和/或处理器100可被配置为响应于用户与搜索UI 124的交互,捕获用户的搜索意图,并且将对应于所捕获的用户的搜索意图的至少一个工件132从储存库118导入到应用程序开发UI 122。应用软件组件106和/或处理器102可被配置为通过使用所导入的至少一个工件132通过应用程序开发UI 122开发应用程序120。

[0030] 以示例的方式,所提供的应用程序开发UI 122可与上面提及的可用于开发应用程序的ADF、IDE或基于视觉模型的表示交互。应用程序开发UI 122以及搜索UI 124可使用显示设备122显示给用户。用户可使用输入设备110经由他的输入将他的搜索意图提供到搜索UI 124。通过与搜索UI 124的交互,用户可搜索合适的工件126,其被存储在储存库118中且可重复使用于应用程序开发。储存库118可为用户经由搜索UI 124可访问的数据存储装置

或存储器。储存库118可包含用户或其它用户的工件126。可在存储于储存库118中且从储存库118导入到应用程序开发UI 122的工件126之中选择对应于所捕获的用户的搜索意图的至少一个工件132。以示例的方式,可通过在应用程序开发UI 122中“拖放”完成导入。使用所导入的至少一个工件132,通过应用程序开发UI 122开发应用程序120,以示例的方式,应用程序开发UI 122可与ADF、IDE或基于视觉模型的表示交互实现应用程序120的开发。

[0031] 在一些示例中,工件的导入可包括导入软件库等。对于工件的导入,例如,可使用冻结引用将引用目标处的当前修订并入到将被开发或创建的应用程序中,其中所述的引用目标的未来修订不应当自动并入到将被开发或创建的应用程序中。在其它示例中,工件的导入可包括创建与运行服务的连接,使得将被开发或创建的应用程序被链接到运行中的服务、其它应用程序等,并且可定期、基于事件和/或根据请求将更新的信息提供到将被开发或创建的完成的或部署的应用程序。与运行服务的这样连接可例如被理解为实时引用,其允许并入在引用目标处可用的未来信息,使得可通过将被开发或创建的完成的或部署的应用程序从引用目标检索更新的和始终最新的信息。

[0032] 换句话说,工件的导入可包括使用静态信息,例如,软件库、冻结引用和/或在引用目标处可用的信息等,或者使用动态信息,例如,通过创建或使用与运行的服务的连接和/或在连接目标处可用的信息、实时引用和/或在引用目标处可用的信息等。在一些示例中,导入可使得在应用程序的开发和/或创建阶段期间或甚至在所创建的应用程序的部署和/或操作期间可使用相应的信息。

[0033] 搜索UI 124可包括用于用户输入该搜索意图的输入框128。应用程序开发UI 122可已经包括当前开发的应用程序120的一些其它应用程序部分134。

[0034] 应当了解,通常,本发明可便于与资产(诸如由其它应用程序/服务提供的工件)的视觉建模连接,其中使由其它服务提供的(例如,在组织内的)所有或几乎所有或几乎所有软件资产是可发现的,并且其中实际连接的技术构造被抽象为使用由其它应用程序提供的资产。

[0035] 在示例实施方式中,应用软件组件106和/或处理器102可被配置为在搜索UI 124中显示存储在储存库118中的且对应于所捕获的用户的搜索意图的一组工件130,并且响应于用户与搜索UI 124的交互,捕获用户的意图以重复使用所显示的一组工件130中的至少一个工件132。

[0036] 所显示的一组工件130可每个至少在一定程度上对应于所捕获的用户的搜索意图,并且可使用显示设备112被显示在搜索UI 124中。以示例的方式,显示一定数量的工件,在储存库118中可用的工件116之中,该一定数量的工件与所捕获的用户的搜索的意图具有最大的符合度。通过与搜索UI 124交互,用户可在所显示的一组工件130之中选择,例如,使用输入设备110提供对应的输入。捕获用户重复使用所显示的一组工件130中的至少一个工件132的意图。以示例的方式,仅由用户选择用于重复使用的那些工件132从储存库118被导入到应用程序开发UI 122。

[0037] 对于这些示例,应用软件组件106和/或处理器102可被配置为检索存储在储存库118中的工件126的元数据,以确定存储在储存库118中的工件126的排名,该排名对应于所检索的元数据与所捕获的用户的搜索意图的符合度,并且根据所确定的工件的排名显示一组工件130。

[0038] 鉴于所捕获的用户的搜索意图可分析存储在储存库118中的工件126的元数据,以便确定元数据与所捕获的用户的搜索意图的符合性的程度。以示例的方式,符合性的程度可与工件相似度相关,工件的类型(例如,应用程序、应用程序的应用区域或应用程序的目的)可被认为是由工件相似度引起的。然后,该符合度可用于确定存储在储存库中的工件的排名。排名可涉及比较例如工件的流行度(例如,给定应用程序的部署的数量)、关键词的匹配、工件的相似度(例如,工件的应用区域或使用区域、工件的目的)、或其任何组合。以示例的方式,显示一定数量的工件130,在储存库118中可用的工件126之中,一定数量的工件130的元数据与所捕获的用户的搜索意图具有最大符合性程度。在另选的示例中,显示元数据超过与所捕获的用户的搜索意图特定最小符合性程度的那些工件130。

[0039] 也应当了解,对于这些示例,元数据可包括以下中的至少一个:工件所有者、工件标签、工件描述、工件关键词、工件流行度、工件版本、工件目的、工件区域或其任何组合。

[0040] 以示例的方式,可通过给定应用程序或给定工件的部署的数量测量或比较流行度。另外或另选地,可例如通过给定工件的连接和用法的数量来测量或比较流行度。对于连接和用法的数量,可考虑用于应用程序开发或其它应用程序创建的相应工件的导入的数量,或通过导入相应的工件已经开发或创建的应用程序的数量。在另外的示例中,工件描述可包括某些关键词。

[0041] 在另外的示例中,工件可包括以下中的至少一个:应用程序数据、应用程序组件、应用程序架构、应用程序编程接口(API)、应用程序服务、应用程序用法、应用程序链接、应用程序描述、应用程序相关性、工件相关性、应用程序环境、应用程序标签、业务事件、通知、应用程序界面、触发信息、用户界面(UI)设计、知识图、可重复使用IT资产图、到页面的链接或其任何组合。

[0042] 提及的应用程序特性也可被实施为对应的工件特性。应用程序相关性可特别地为此类种类,即对源工件或系统的修订将影响目标工件或系统,工件以已知或可预测方式被导入到目标工件或系统。这可通过验证所有改变和相关性是正确的而有助于确保新的应用程序或软件的“安全”安装或部署。

[0043] 以示例的方式,业务事件、通知和/或触发信息通常可为从另一个应用程序、服务可获得的数据和/或从另一个应用程序、服务发送的数据,或者通常可为可促使在将被创建的应用程序中或通过将被创建的应用程序触发动作的的数据源。例如,初始时,新用户或客户订阅或注册系统或服务,并且然后,在将被创建的应用程序中或借助于将被创建的应用程序触发动作,例如,创建新的账户或租户,例如,在IIOT系统中发送文档或确认、提供对新用户或客户的访问、显示更新的状态信息、供应另一个应用程序等。在其它示例中,业务事件、通知和/或触发信息可包括项目的购买,硬件和/或软件产品的产生的一个、若干或所有步骤的完成,项目或给定项目的可用性或非可用性。应用程序界面可例如包括关于应用程序的接收和/或发出信息、支持或接受的数据协议、支持或接受的加密协议等的信息。附加地或另选地,其它应用程序可订阅将被创建的应用程序,将对应的触发信息等提供到将被创建的应用程序,并且促使某些动作被完成。以示例的方式,将被创建的应用程序可被用于数据消耗、覆盖应用程序的数据源,其中覆盖应用程序可订阅底层应用程序,订阅为触发信息,并且底层应用程序依照请求、根据相关数据的可用性和/或以给定频率将相关数据提供到覆盖应用程序。

[0044] 如上面提及的,工件可包括应用程序链接、到页面的链接或通常可包括链接。这些链接可源于其它应用程序或服务,在其它应用程序或服务中,这些链接可在应用程序或服务的运行期间使用,并且可用于导入储存库中。工件的导入也可包括上面在软件库、与运行服务的连接、冻结引用、实时引用、静态或动态信息的上下文中诠释的示例。

[0045] 可用于应用程序开发和/或应用程序创建的各种各样的工件可被认为是数据目录,在该数据目录中,使组织或公司中的多种(如果不是所有的)可用的信息、数据、服务和/或应用程序可用于使用。该数据目录可包括来自产品生命周期管理、企业资源计划、产品设计、制造操作等的信息、数据、服务和/或应用程序。以示例的方式,数据目录可包括将被用于其它应用程序、服务、度量、流数据、事件、业务事件中的操作,并且/或者通常可包括数据集(例如,包括上面诠释的业务事件、通知和/或触发信息)。以该方式,可使可用信息(例如,软件应用程序)的异构的场景规范化,使得其可被用于开发、创建和/或甚至运行应用程序。

[0046] 也应当了解,在其它示例中,应用程序开发UI 122可为由用户通过浏览器或客户端可访问的,并且搜索UI 124可为由用户通过应用程序开发UI 122可访问的。

[0047] 在另外的示例中,所创建的应用程序120和/或存储在储存库126中的工件可包括标准化的模块,标准化的模块涉及以下中的至少一个:数据提供者、数据消费者、数据更新机制、数据格式、可用数据处置操作、应用程序身份和访问管理(IAM)或其任何组合。

[0048] 此类标准化模块可有助于创建有利的低代码抽象,其允许用户导入工件或部分应用程序,凭此,不需要所导入的信息的详细知识。以示例的方式,数据更新机制可涉及周期性数据更新或推或拉数据更新。特别地基于当用户选择找到的工件用于重复使用时在储存库中可用的元数据,此类特征可支持所导入的工件、应用程序的部分或来自开发环境的应用程序的自动和正确集成和同步。这允许在存储在储存库118中的工件126与该工件120重复使用于应用程序开发的应用程序之间建立相关性。然后,所建立的相关性变成所部署的应用程序的元数据的一部分,并且对于每个部署,通过相应地更新对应的元数据,将通知储存库118相关性存在于一个或多个部署状态中。

[0049] 在示例实施方式中,应用软件组件106和/或处理器102可被配置为部署所创建的应用程序120,将所创建的和部署的应用程序的至少一部分存储为工件126(其可重复使用于储存库118中的应用程序开发),并且将所创建的和部署的应用程序120的所存储的至少一个部分126的元数据存储于储存库118中。

[0050] 所创建的应用程序120的部署可包括将应用程序120发布给用户,然后,用户可安装和激活所发布的应用程序,以满足其具体个人或业务需要。以示例的方式,在成功的第一部署之后,至少一部分作为工件126被存储在储存库128中,该工件可被用于重复使用于应用程序开发。附加地,所存储的部分的元数据也可被存储在储存库118中。

[0051] 对于这些示例,所存储的元数据可包括以下中的至少一个:部署信息、应用程序服务、以及关于所创建的和部署的应用程序的所存储的至少一部分与已经被用于开发所创建的和部署的应用程序的所导入到至少一个工件之间的相关性的信息、或其任何组合。

[0052] 也应当了解,在其它示例中,应用软件组件106和/或处理器102可被配置为在修订所开发和部署的应用程序120中的至少一部分时更新在储存库118中的所存储的至少一部分,并且/或者在修订所存储的至少一部分的元数据时,更新在储存库118中的所存储的至

少一部分的所存储的元数据。

[0053] 对于此类示例,所存储的至少一部分的所存储的元数据可包括关于用于访问所存储的至少一部分的访问权限的信息,其中所存储的至少一部分可仅对于具有合适的访问权限的用户是可访问的。

[0054] 对于此类示例,所存储的至少一部分的所存储的元数据可包括关于用于部署和/或使用应用程序的访问权限的信息,使用所存储的至少一部分开发该应用程序,其中所存储的至少一部分可仅对于具有合适访问权限的用户是可访问的,并且/或者所开发的应用程序可仅由具有合适访问权限的用户部署和/或使用。

[0055] 对于所有示例,所创建的应用程序可适合于部署到包括智能电话、手持式计算机、平板计算机、台式计算机、智能手表、TV的硬件设备,部署到云计算平台或到物联网(IoT)操作系统(特别是工业物联网(IIoT)系统)。

[0056] 所描述的示例可通过使得能够进行所描述的增强的应用程序创建,为用户提供创建应用程序的有效方式。因此,一般而言,所描述的示例可通过减轻或至少最小化对用户或管理员具有高级编码知识或手动作出很多选项的有意识选择的需要,减少应用程序和此类应用程序的创建或开发的总拥有成本。可在任何行业(例如,航空与国防、汽车与运输、消费品与零售、电子与半导体、能源与公用事业、工业机械与重型装备、航海或医疗设备与药物)中利用此类有效应用程序创建。此类应用程序创建也可适用于面临快速创建自己的应用程序的需要的消费者。

[0057] 特别地,上面的示例分别同样适用于创建应用程序的计算机实施的方法以及本专利文档中诠释的对应的计算机可读介质。

[0058] 现在参考图2-图11,示出包括可使用图1所示的产品系统显示给用户的结果的一系列示例性图形用户界面。

[0059] 图2示出包括应用程序开发UI 122的示例性GUI,其包含当前部署的应用程序的其它应用程序部分。示例性GUI此外包括搜索UI 124,其包括输入框128,其允许用户输入搜索意图。搜索UI 124此外包括对应于所捕获的用户的搜索意图的一组工件130。

[0060] 图3示出在由用户选择一组工件130中的一个工件132时,所选择的工件132被导入到应用程序开发UI中。在该示例中,用户选择工件“Contact(联系人)”,然后,工件“Contact(联系人)”作为新框被导入到应用程序开发UI 122。

[0061] 图4示出用户此外将工件“Company(公司)”导入到应用程序开发UI 122。

[0062] 图5示出除最近导入到工件“Contact(联系人)”和“Company(公司)”的之外用户已经移除所有其它工件。

[0063] 图6-图9示出用户如何调整两个导入的工件“Contact(联系人)”和“Company(公司)”的性质以及用“Contact_Company(联系人_公司)”指示的关系链接。

[0064] 因此,图10与图11示出可如何使用应用程序开发UI 122运行和部署所开发的应用程序。

[0065] 重要的是注意到,虽然本公开包括在全功能系统和/或一系列动作的上下文中的描述,但是本领域技术人员将了解,本公开的机制的和/或所描述的动作的至少部分能够以多种形式中的任一种包含在非暂态机器可用、计算机可用或计算机可读介质内的计算机可执行指令的形式分发,并且不管用于实际完成分发的指令或数据承载介质或存储介质的特

定类型,本公开同样适用。非暂态机器可用/可读或计算机可用/可读介质的示例包括:ROM、EPROM、磁带、软盘、硬盘驱动器、SSD、闪存存储器、CD、DVD和蓝光光盘。计算机可执行指令可包括例程、子例程、程序、应用程序、模块、库、执行线程等。另外,方法的动作的结果可被存储在计算机可读介质中,显示在显示设备上等。

[0066] 现在参考图12,示出便于创建应用程序的方法1200。方法可开始于1202,并且方法可包括通过至少一个处理器的操作完成的若干动作。

[0067] 这些动作可包括将应用程序开发用户界面(UI)提供到用户用于开发应用程序的动作1204;将搜索UI提供到用户,用于在用于存储可重复使用于应用程序开发的工件的储存库中进行搜索的动作1206;响应于用户与搜索UI的交互,捕获用户的搜索意图的动作1208;将对应于所捕获的用户的搜索意图的至少一个工件从储存库导入到应用程序开发UI的动作1210;以及通过使用所导入的至少一个工件通过应用程序开发UI开发应用程序的动作1212。在1214,方法可结束。

[0068] 应当了解,方法900可包括先前关于处理系统100讨论的其它动作和特征。例如,方法可包括在搜索UI中显示存储在储存库中且对应于所捕获的用户的搜索意图的一组工件的动作;以及响应于用户与搜索UI的交互,捕获用户的意图以重复使用所显示的一组工件中的至少一个工件的动作。

[0069] 除此之外,方法可包括检索存储在储存库中的工件的元数据的动作;确定存储在储存库中的工件的排名的动作,该排名对应于所检索的元数据与所捕获的用户的搜索意图的符合性程度;以及根据所确定的工件的排名显示一组工件的动作。

[0070] 另外,方法可包括部署所创建的应用程序的动作;将所创建的和部署的应用程序的至少一部分存储为工件的动作,工件在储存库中可重复使用于应用程序开发;以及在储存库中存储所创建的和部署的应用程序的所存储的至少一部分的元数据的动作。

[0071] 方法的示例实施方式也可包括在修订所开发的和部署的应用程序中的至少一部分时更新储存库中的所存储的至少一部分的动作;和/或在修订所存储的至少一部分的元数据时更新储存库中的所存储的至少一部分的所存储的元数据的动作。

[0072] 如先前讨论的,与这些方法相关联的动作(除了任何描述的手动动作诸如通过输入设备手动作出选择的动作之外)可通过一个或多个处理器完成。一个或多个此类处理器可包括在一个或多个数据处理系统中,例如,一个或多个数据处理系统执行可操作为促使这些动作由一个或多个处理器完成的软件组件。在示例实施方式中,此类软件组件可包括对应于例程、子例程、程序、应用程序、模块、库、执行的线程等的计算机可执行指令。另外,应当了解,软件组件可由软件环境/语言/框架(诸如Java、JavaScript、Python、C、C#、C++)或能够产生被配置为完成本文所描述的动作和特征的组件和图形用户界面的任何其它软件工具编写和/或产生。

[0073] 图13示出数据处理系统1000(也称为计算机系统)的框图,在数据处理系统中,例如,实施方式可被实施为产品系统的一部分,和/或由软件或以其它方式可操作地配置以实行如本文所描述的过程的其它系统。描绘的数据处理系统包括可连接到一个或多个桥接器/控制器/总线1004(例如,北桥、南桥)的至少一个处理器1002(例如,CPU)。例如,总线1004中的一个可包括一个或多个I/O总线,诸如PCI Express总线。同样在所描绘的示例中连接到各种总线的可包括主存储器1006(RAM)和图形控制器1008。图形控制器1008可连接

到一个或多个显示设备1010。也应当注意,在一些实施方式中,一个或多个控制器(例如,图形、南桥)可与CPU集成(在相同芯片或管芯上)。CPU架构的示例包括IA-32、x86-64和ARM处理器架构。

[0074] 连接到一个或多个总线的其它外围设备可包括通信控制器1012(以太网控制器、WiFi控制器、蜂窝控制器),通信控制器可操作为连接到局域网(LAN)、广域网(WAN)、蜂窝网络和/或其它有线或无线网络1014或通信装备。

[0075] 连接到各种总线的另外的组件可包括一个或多个I/O控制器1016,诸如USB控制器、蓝牙控制器和/或专用音频控制器(连接到扬声器和/或麦克风)。也应当了解,各种外围设备可(经由各种端口和连接)连接到一个或多个I/O控制器,包括输入设备1018(例如,键盘、鼠标、指针、触摸屏、触摸板、绘图板、轨迹球、按钮、小键盘、游戏控制器、游戏手柄、相机、麦克风、扫描仪、捕获运动手势的运动感测设备)、输出设备1020(例如,打印机、扬声器)或可操作为将输入提供到数据处理系统或从数据处理系统接收输出的任何其它类型的设备。再者,应当了解,被称为输入设备或输出设备的很多设备可既提供输入,又接收与数据处理系统通信的输出。例如,处理器1002可被集成到包括用作输入和显示设备的触摸屏的外壳(诸如平板电脑)中。另外,应当了解,一些输入设备(诸如膝上型计算机)可包括多个不同类型的输入设备(例如,触摸屏、触摸板、键盘)。再者,应当了解,连接到I/O控制器1016的其它外围硬件1022可包括被配置为与数据处理系统通信的任何类型的设备、机器或组件。

[0076] 连接到各种总线的附加组件可包括一个或多个存储控制器1024(例如,SATA)。存储控制器可连接到存储设备1026诸如一个或多个存储驱动器和/或任何相关联的可移除介质,存储设备可为可用的任何合适的非暂态机器或机器可读存储介质。示例包括非易失性设备、易失性设备、只读设备、可写设备、ROM、EPROM、磁带存储装置、软盘驱动器、硬盘驱动器、固态驱动器(SSD)、闪存存储器、光盘驱动器(CD、DVD、蓝光光盘)、以及其它已知的光、电或磁存储设备驱动器和/或计算机介质。再者,在一些示例中,存储设备诸如SSD可直接连接到I/O总线1004,诸如PCI Express总线。

[0077] 根据本公开的实施方式的数据处理系统可包括操作系统1028、软件/固件1030和数据存储装置1032(数据存储装置可被存储在存储设备1026和/或存储器1006上)。这样的操作系统可采用命令行界面(CLI)壳体和/或图形用户界面(GUI)壳体。GUI壳体准许在图形用户界面中同时呈现多个显示窗口,其中每个显示窗口将界面提供到不同应用程序或相同应用程序的不同实例。可由用户通过点击设备诸如鼠标或触摸屏操纵图形用户界面中的光标或指针。光标/指针的位置可被改变,并且/或者可生成事件,诸如点击鼠标按钮或触摸一触摸屏,以促动期望的响应。可用于数据处理系统中的操作系统的示例可包括微软视窗、Linux、UNIX、iOS和安卓操作系统。再者,数据存储装置的示例包括数据文件、数据表、关系数据库(例如,Oracle、微软SQL服务器)、数据库服务器或能够存储可由处理器检索的数据的任何其它结构和/或设备。

[0078] 通信控制器1012可连接到网络1014(不是数据处理系统1000的一部分),如对本领域技术人员已知的,网络1014可为任何公用或私有数据处理系统网络或网络的组合,包括互联网。数据处理系统1000可经过网络1014与一个或多个其它数据处理系统诸如服务器1034(也不是数据处理系统1000的一部分)通信。然而,另选的数据处理系统可对应于实施为分布式系统的一部分的多个数据处理系统,其中与若干数据处理系统相关联的处理器可

借助一个或多个网络连接而通信,并且可共同实行被描述为由单个数据处理系统实行的任务。因此,应当理解,当提到数据处理系统时,可在经由网络彼此通信的在分布式系统中组织的若干数据处理系统上实施这样的系统。

[0079] 另外,术语“控制器”意指控制至少一个操作的任何设备、系统或其部分,无论这样的设备以硬件、固件、软件还是硬件、固件、软件中的至少两个的某种组合实施。应当注意,与任何特定控制器相关联的功能可为集中式的或分布式的,无论是本地的还是远程的。

[0080] 除此之外,应当了解,数据处理系统可被实施为虚拟机架构或云环境中的虚拟机。例如,处理器1002和相关联的组件可对应于在一个或多个服务器的虚拟机环境中执行的虚拟机。虚拟机架构的示例包括VMware ESXi、Microsoft Hyper-V、Xen和KVM。

[0081] 本领域普通技术人员将了解,对于特定的实施方式,针对数据处理系统描绘的硬件可变化。例如,在该示例中,数据处理系统1000可对应于计算机、工作站、服务器、PC、笔记本电脑、平板电脑、移动电话、和/或可操作为处理数据和完成与本文讨论的数据处理系统、计算机、处理器和/或控制器的操作相关联的本文描述的功能和特征的任何其它类型的仪器/系统。仅为了诠释的目的提供所描绘的示例,并且所描绘的示例不意味着暗示关于本公开的架构限制。

[0082] 再者,应当注意,本文描述的处理器可位于服务器中,该服务器远离本文所描述的显示设备和输入设备。在这样的示例中,所描述的显示设备和输入设备可包括在客户端设备中,客户端设备通过有线或无线网络(其可包括互联网)与服务器(和/或在服务器上执行的虚拟机)通信。在一些实施方式中,例如,这样的客户端设备可执行远程桌面应用程序,或者可对应于门户设备,门户设备与服务器完成远程桌面协议,以便将输入从输入设备发送到服务器,并且从服务器接收视觉信息以通过显示设备进行显示。此类远程桌面协议的示例包括Teradici的PCoIP、微软的RDP和RFB协议。在这样的示例中,本文描述的处理器可对应于在服务器的物理处理器中执行的虚拟机的虚拟处理器。

[0083] 如本文所使用的,术语“组件”和“系统”旨在涵盖硬件、软件或硬件和软件的组合。因此,例如,系统或组件可为过程、在处理器上执行的过程或处理器。附加地,组件或系统可定位在单个设备上或分布在若干设备上。

[0084] 再者,如本文使用的,处理器对应于经由硬件电路、软件和/或固件配置以处理数据的任何电子设备。例如,本文描述的处理器可对应于以下中的一个或多个(或组合):微处理器、CPU、FPGA、ASIC或能够在数据处理系统中处理数据的任何其它集成电路(IC)或其它类型的电路,数据处理系统可具有控制器板、计算机、服务器、移动电话和/或任何其它类型的电子设备的形式。

[0085] 本领域技术人员将认识到,为了简单和清晰起见,本文没有描绘或描述适合于与本公开一起使用的所有数据处理系统的全结构和操作。代替地,仅描绘和描述本公开独有的或对于理解本公开所必需的那么多的数据处理系统。数据处理系统1000的剩余构造和操作可符合本领域已知的各种当前实施方式和实践中的任一种。

[0086] 现在参考图14至图18,示出关于便于在产品系统中创建应用程序的另外的信息。

[0087] 通常,图14至图18中提及的“目录”可引用储存库118。

[0088] 再者,应当理解,应当广泛地解释本文使用的单词或短语,除非在一些示例中明确地限制。例如,术语“包含”和“包括”以及其派生词意指包括而没有限制。单数形式“一个/一

种”和“所描述的”旨在也包括复数形式,除非上下文以其它方式清楚地指示。另外,如本文使用的术语“并且/或者”指代且涵盖相关联的列出的项目中的一个或多个中的任一个和所有可能组合。术语“或”是包括的,意指和/或,除非上下文以其它方式清楚指示。短语“与……相关联”和“与此相关联”以及其派生词可意味着包括、被包括在……内、与……互连、包含、包含在……内、连接到或与……连接、联接到或与……联接、可与……通信、与……合作、交织、并置、接近、结合到或与……结合、具有、具有……的性质等。

[0089] 再者,虽然术语“第一”、“第二”、“第三”等可在本文中用于描述各种元素、功能或动作,但是这些元素、功能或动作不应当由这些术语限制。相反,这些数字形容词用于将不同元素、功能或动作彼此区分开。例如,第一元素、功能或动作可被称作第二元素、功能或动作,并且类似地,第二元素、功能或动作可被称作第一元素、功能或动作,而不脱离本公开的范围。

[0090] 除此之外,短语诸如“处理器被配置为”完成一个或多个功能或过程可意指处理器可操作地配置为或在操作上被配置为经由软件、固件和/或有线电路完成功能或过程。例如,被配置为完成功能/过程的处理器可对应于执行软件/固件的处理器,软件/固件被编程以致使处理器完成功能/过程,并且/或者可对应于在存储器或存储设备中具有软件/固件的处理器,软件/固件可用于由处理器执行以完成功能/过程。也应当注意,被“配置为”完成一个或多个功能或过程的处理器也可对应于特定制备或“布线”以完成功能或过程的处理器电路(例如,ASIC或FPGA设计)。另外,在被配置为完成多于一个功能的元素(例如,处理器)之前的短语“至少一个”可对应于每个完成功能的一个或多个元素(例如,处理器),并且也可对应于分别完成一个或多个不同功能中的不同的功能的元素(例如,处理器)中的两个或更多个。

[0091] 除此之外,术语“与……相邻”可意味着:元素相对靠近另外的元素但不与另外的元素接触;或者意味着元素与另外的部分接触,除非上下文以其它方式清楚指示。

[0092] 虽然已经详细描述本公开的示例性实施方式,但是本领域技术人员将理解,可作出本文公开的各种改变、替代、变型和改进,而不脱离本公开最广泛形式的精神和范围。

[0093] 本专利文档中的描述均不应当被解读为暗示任何特定元素、步骤、动作或功能为必须包括在权利要求范围内的必要的元素:仅由所允许的权利要求书定义申请专利的主题范围。而且,这些权利要求均不旨在援引35USC§112(f),除非确切的词语“用于……的装置”后面是分词。

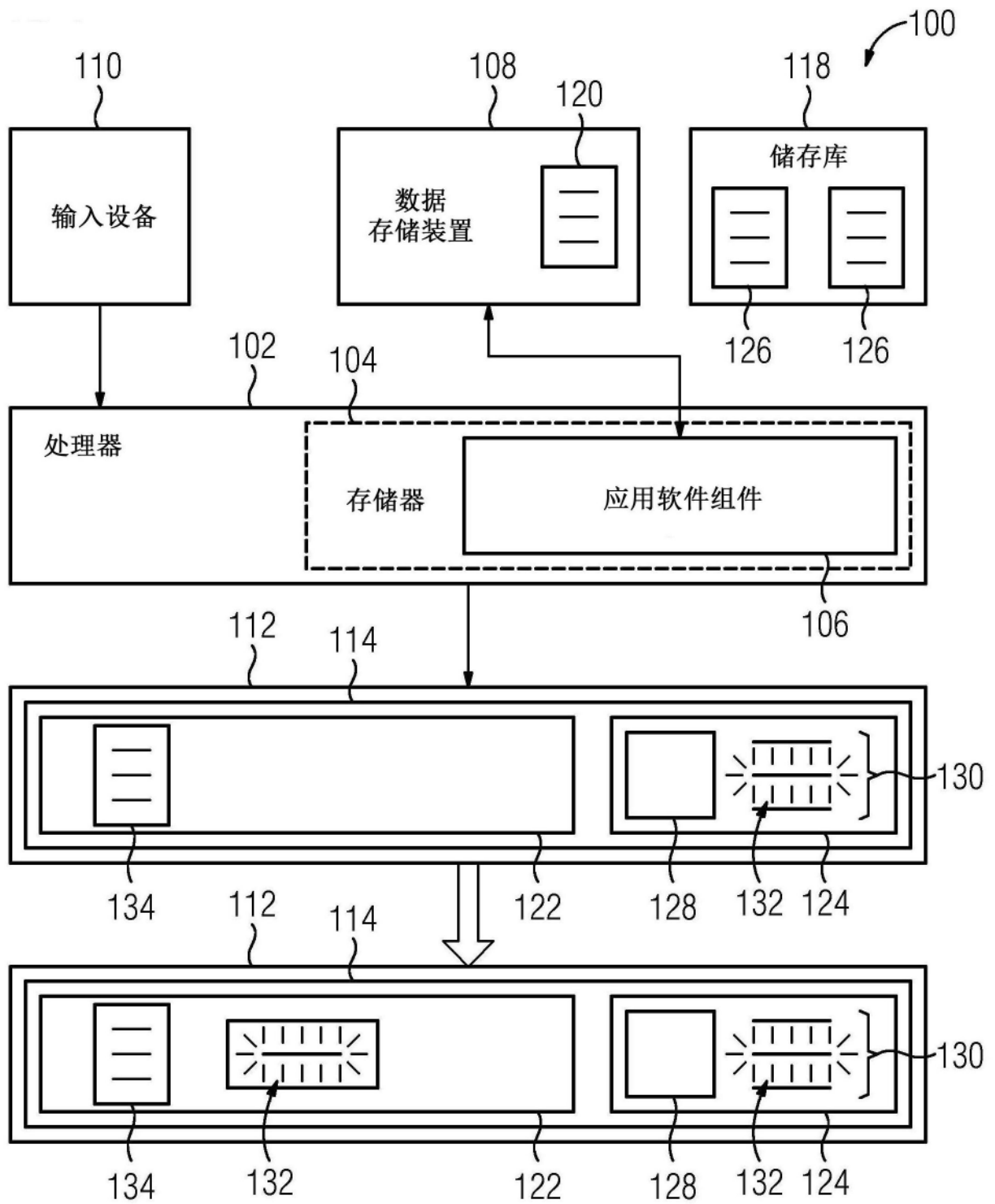


图1

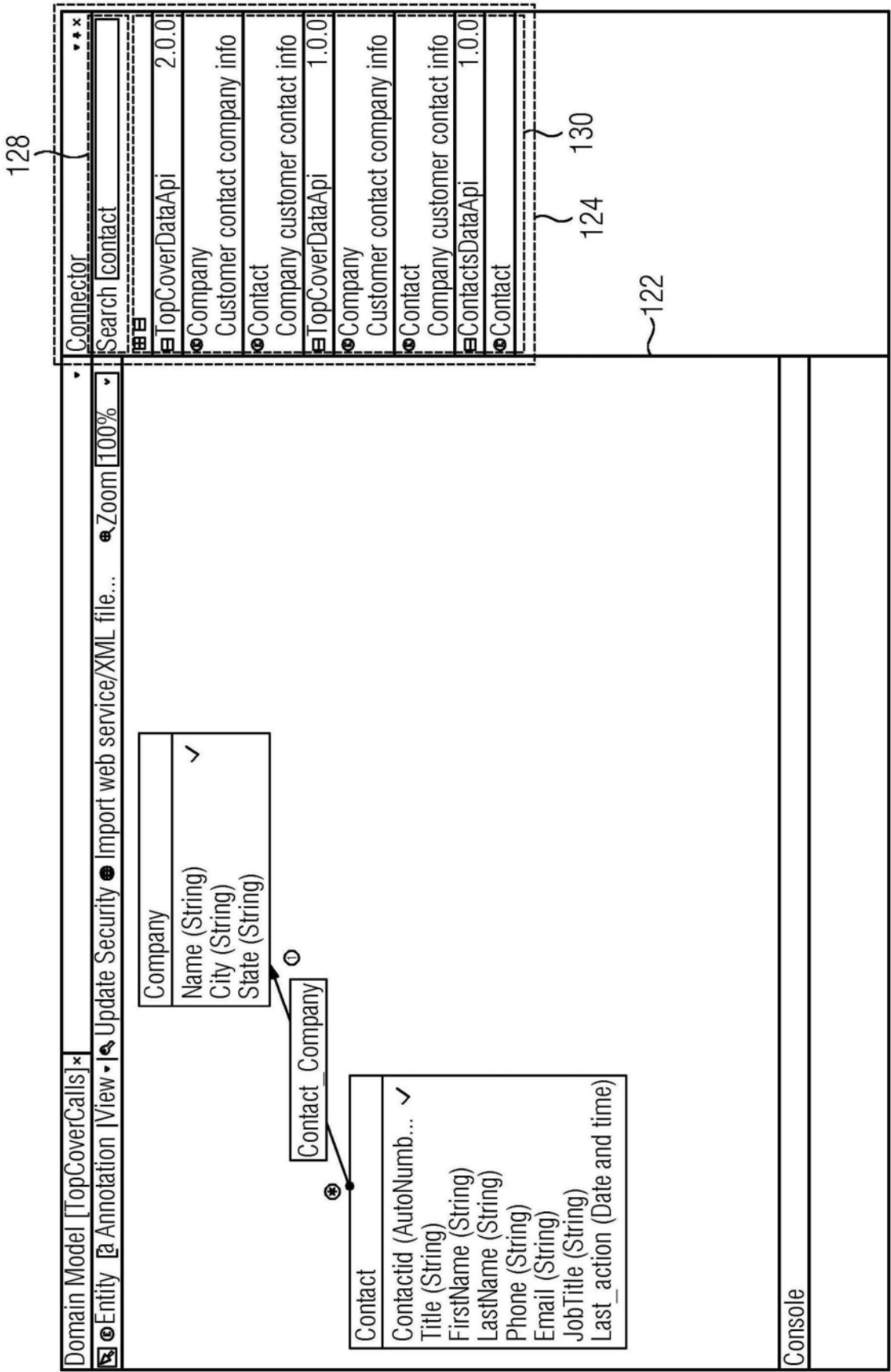


图2

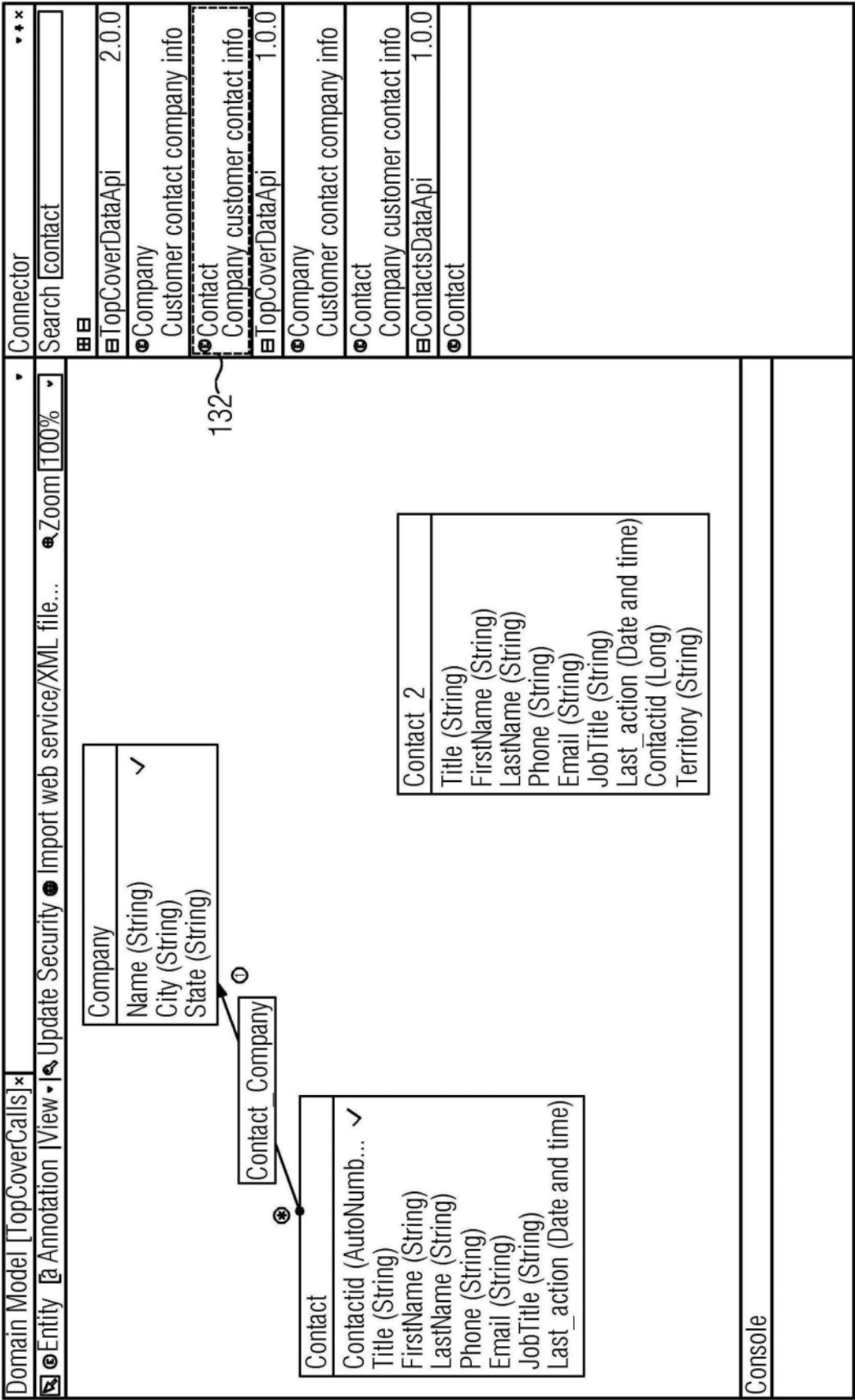


图3

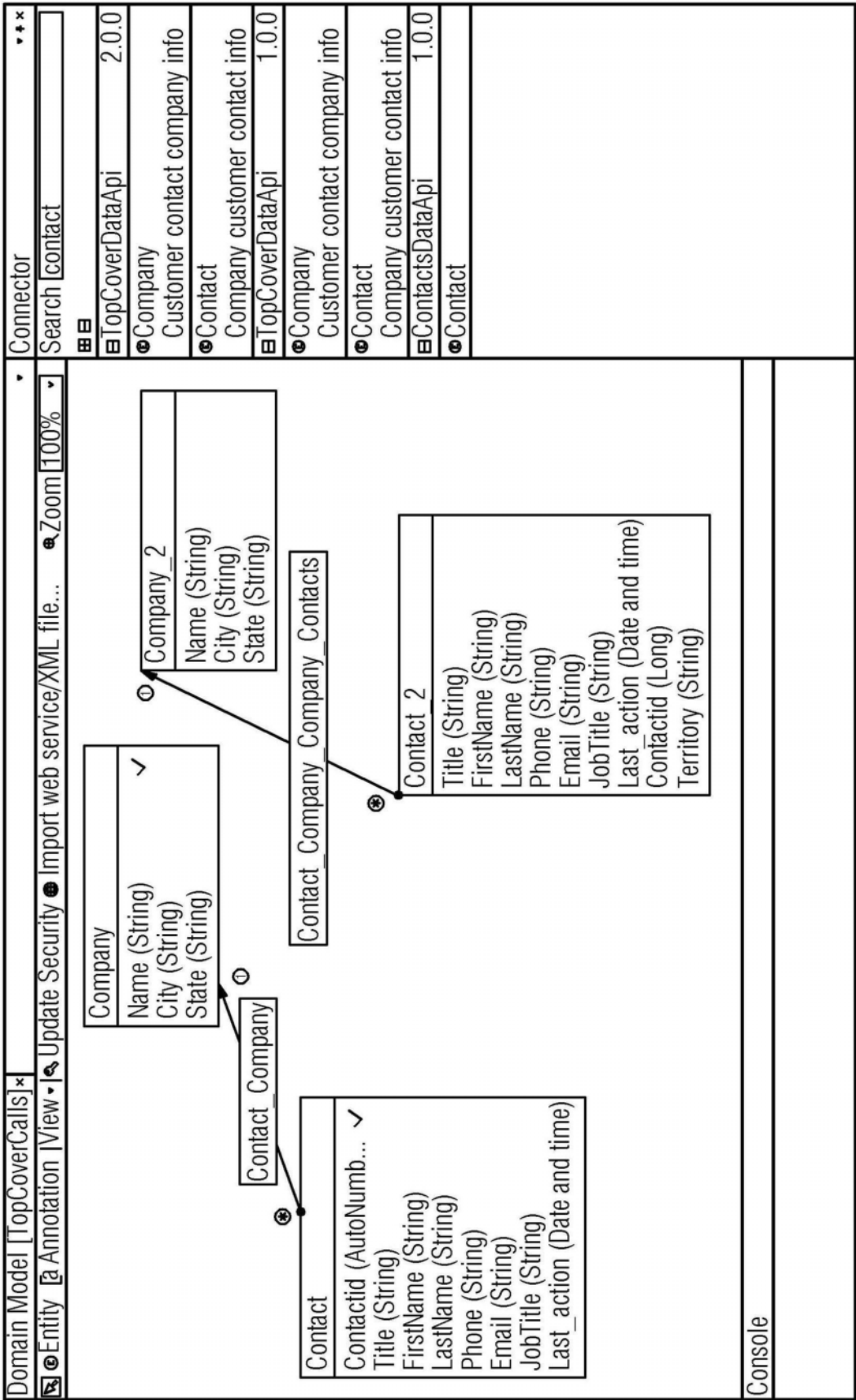


图4

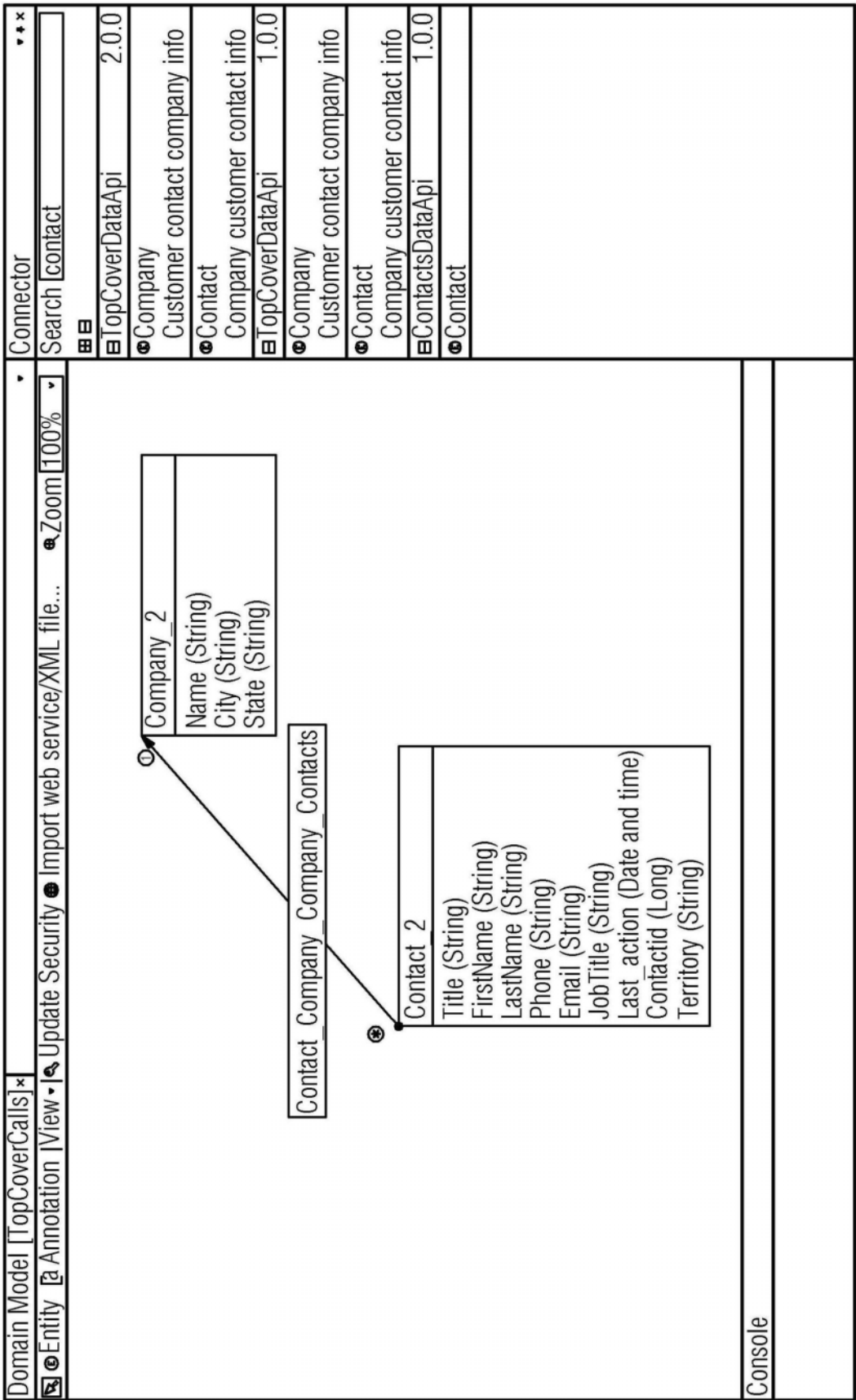


图5

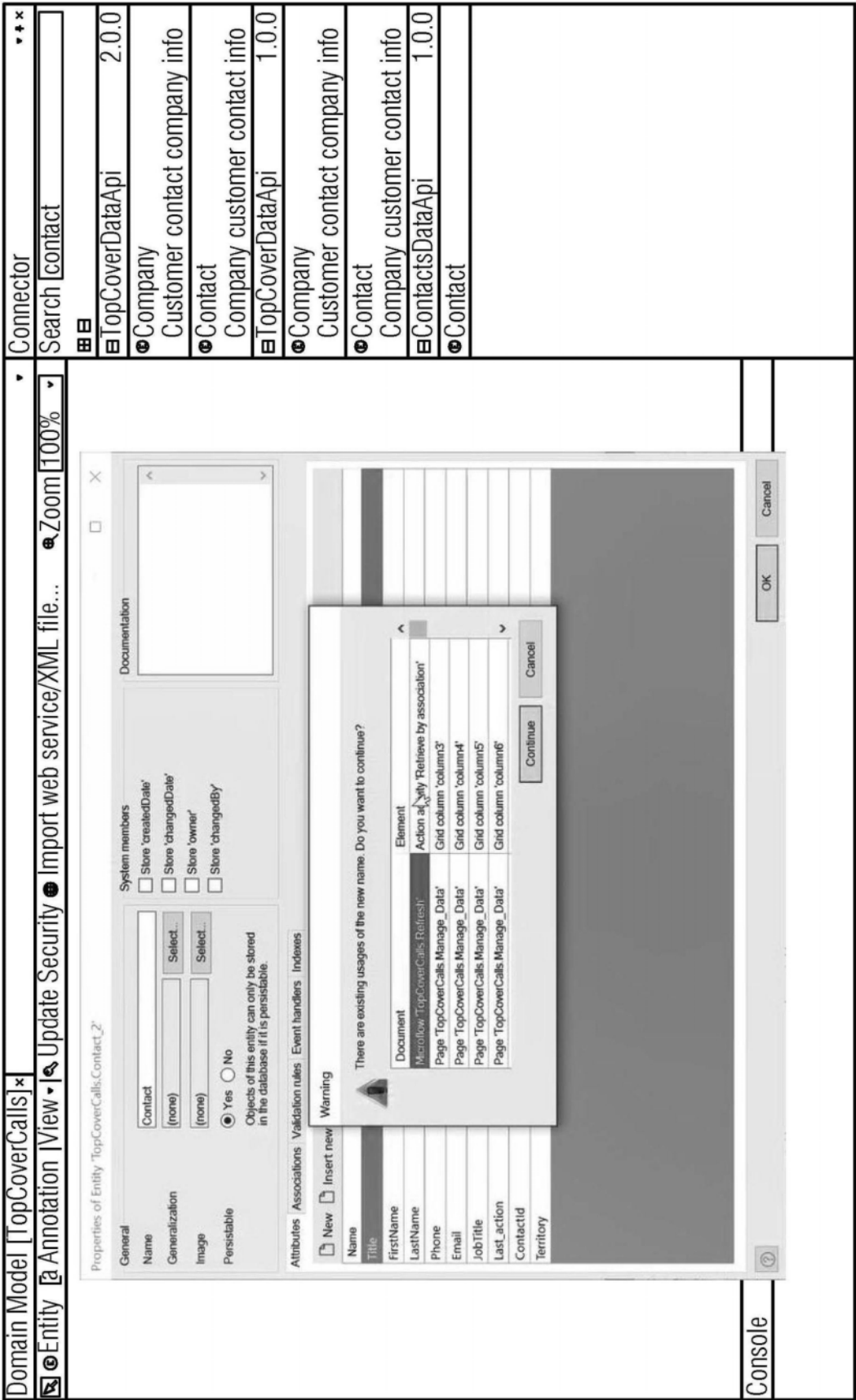


图6

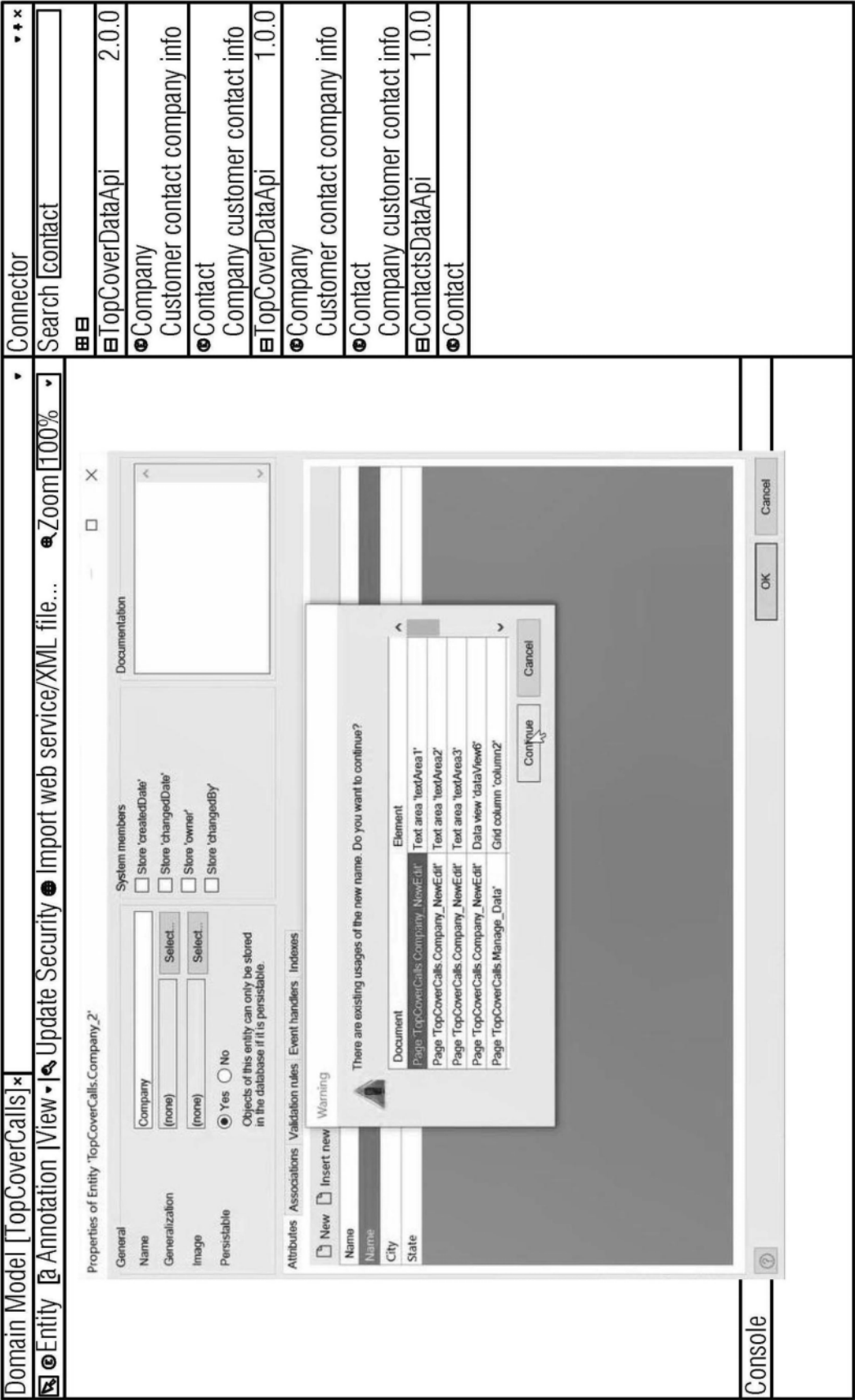


图7

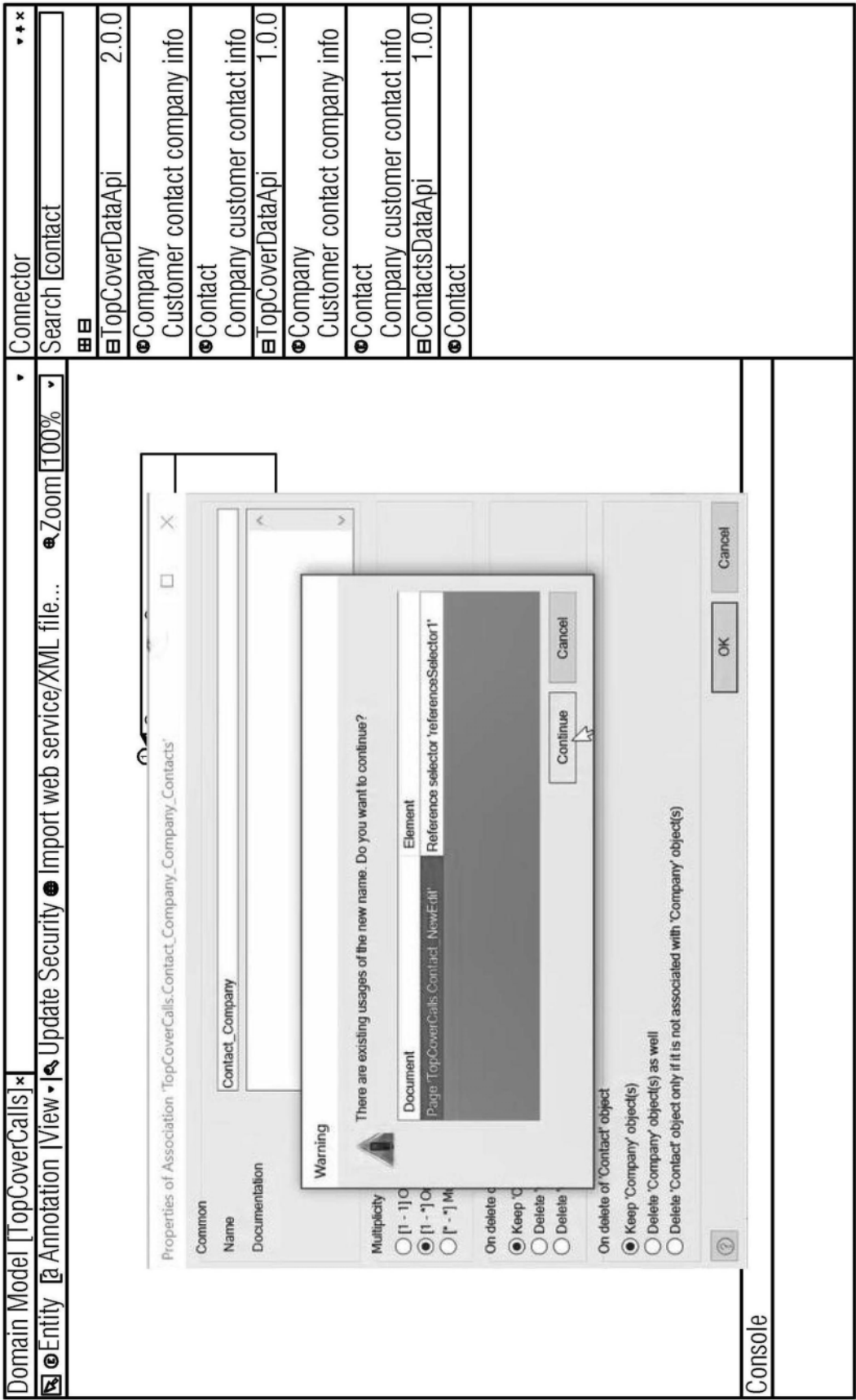


图8

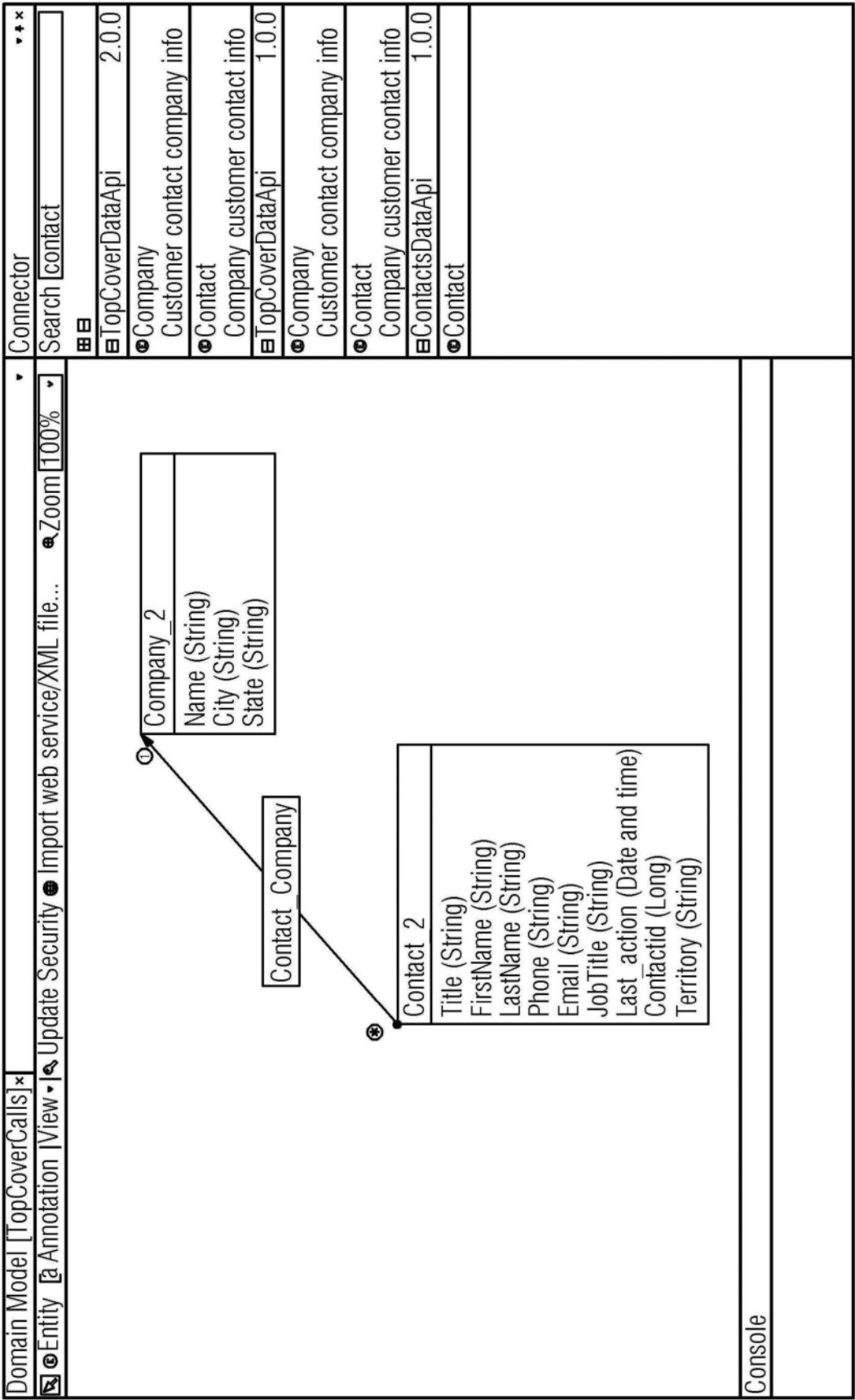


图9

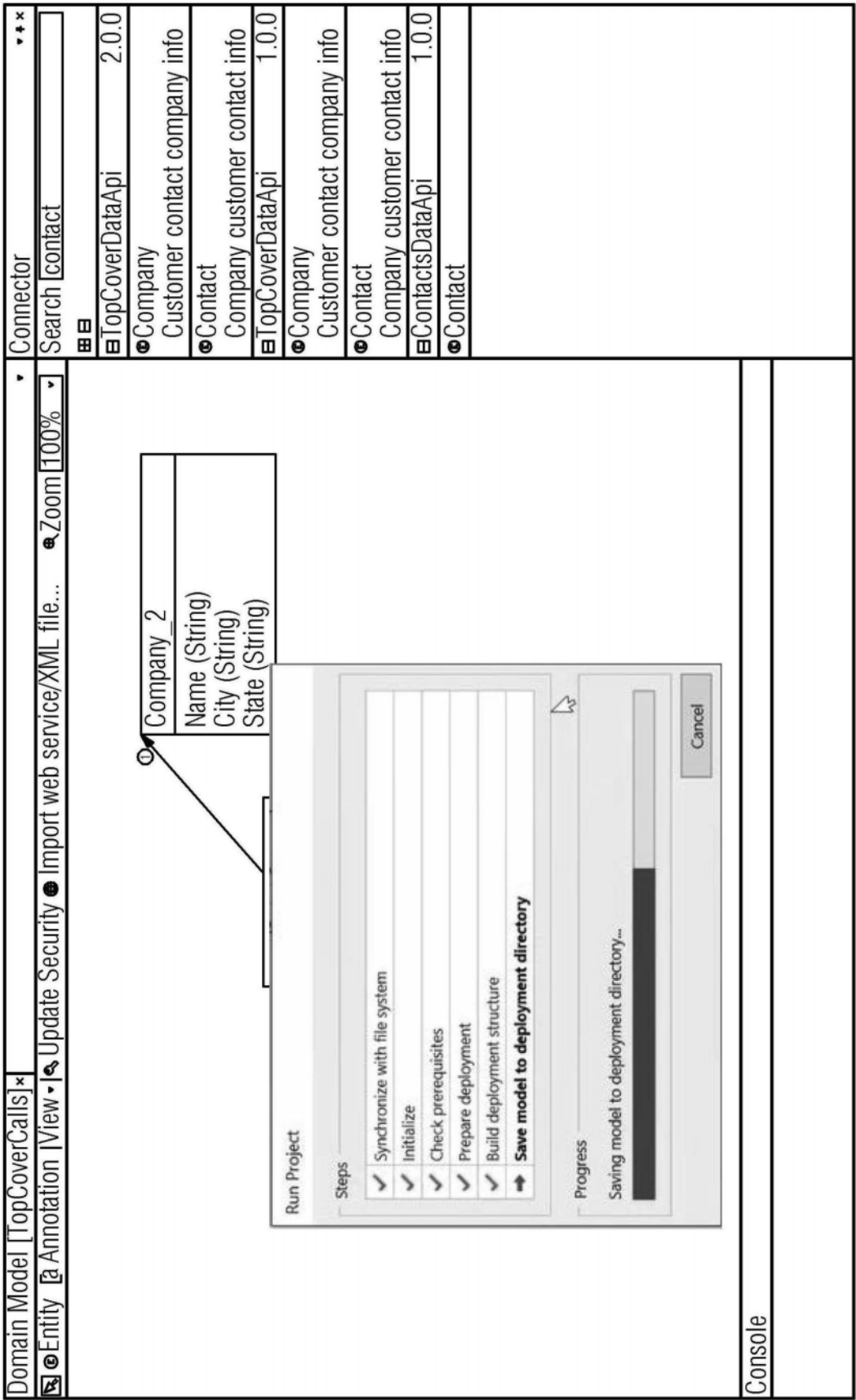


图10

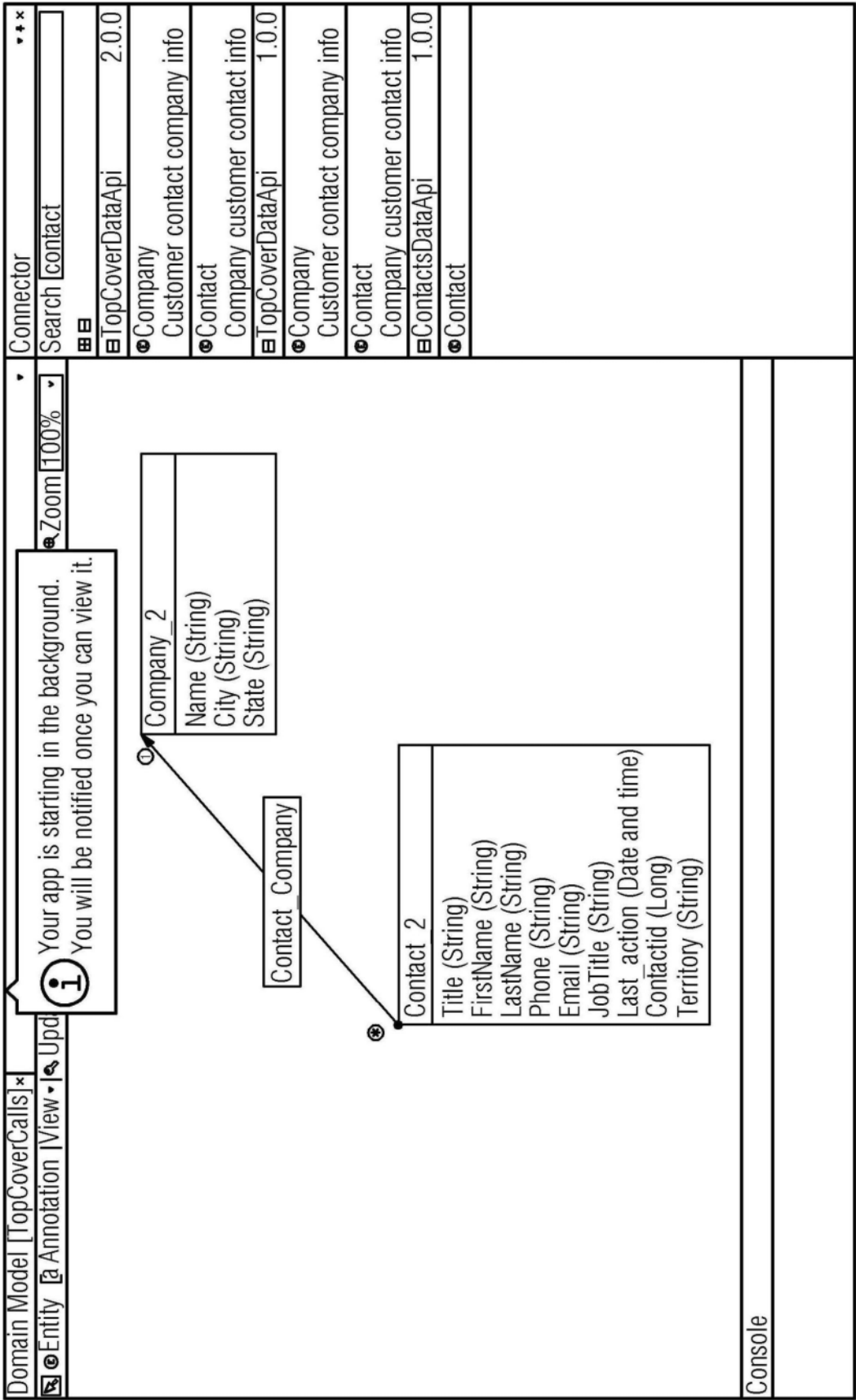


图11

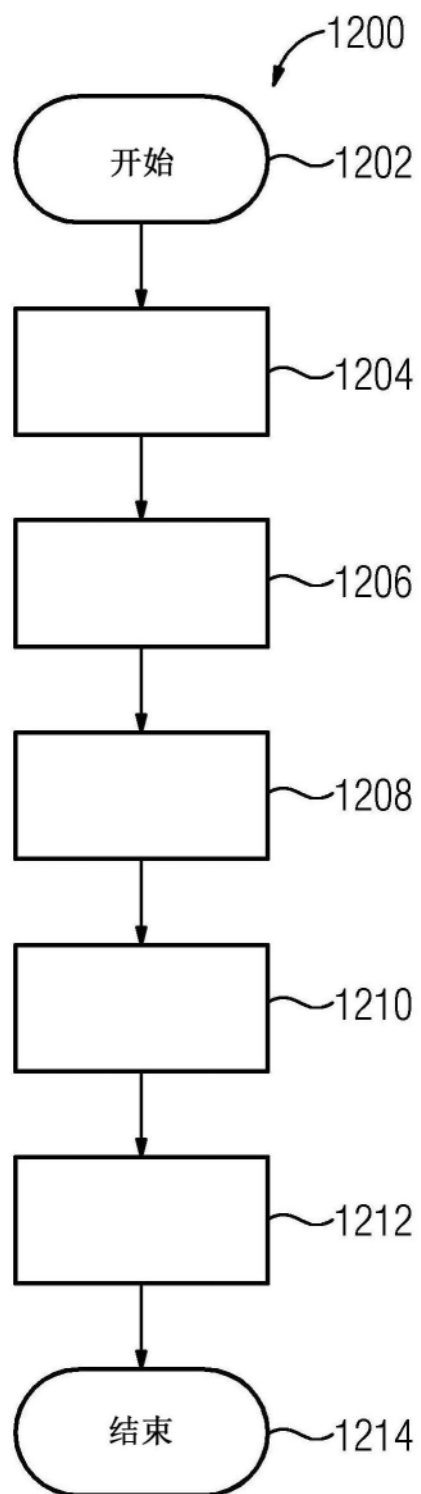


图12

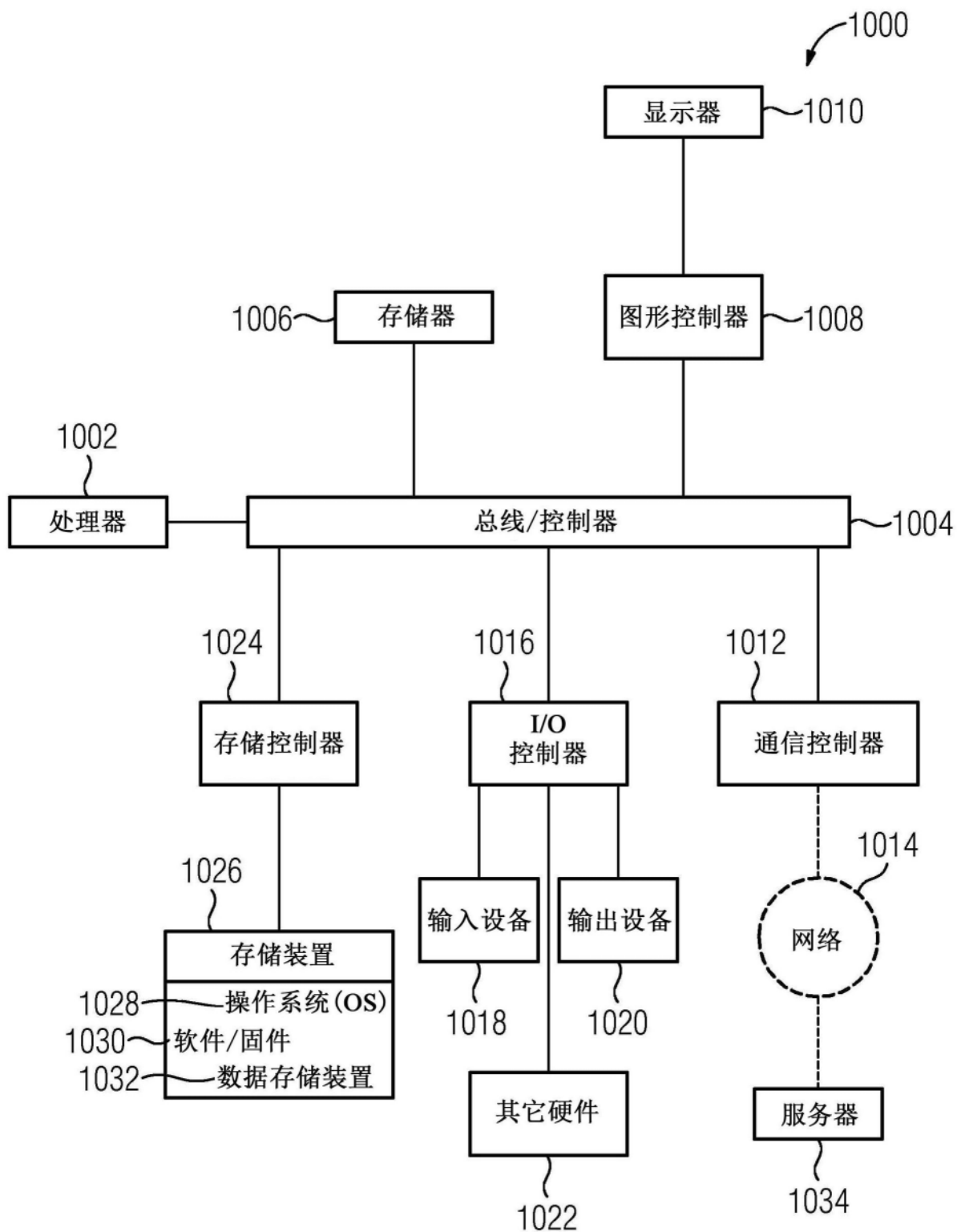


图13

Mendix 企业中枢-总结

• 用于简化软件服务的重复使用的集成系统:

1. 包含所有相关软件场景组件的全知识图的‘目录’ 数据系统: 服务、应用程序、版本、共享的API、使用、相关性
2. 充实有用于智能基于上下文的搜索结果排名以协助应用程序开发者找到他们需要连接到的服务的元数据（所有者、标签、描述）的数据系统
3. 用于搜索的机制
4. 从开发者环境连接到任何企业软件服务以提供最简单和更快的用户体验
5. 对远程服务（数据、api等）的低代码抽象，当开发者选择找到的服务用于重复使用时，开发环境基于目录数据系统中可用的元数据，通过对远程服务（数据、api等）的低代码抽象可自动从开发环境提供正确的集成和同步机制
6. 目录提供用于通过验证所有改变和相关性是正确的确保新软件的‘安全’ 安装的机制
7. 每个开发的应用程序的机制部分将自动地将提供的服务和消耗的服务信息和元数据推送到目录系统，以确保目录始终是最新的。

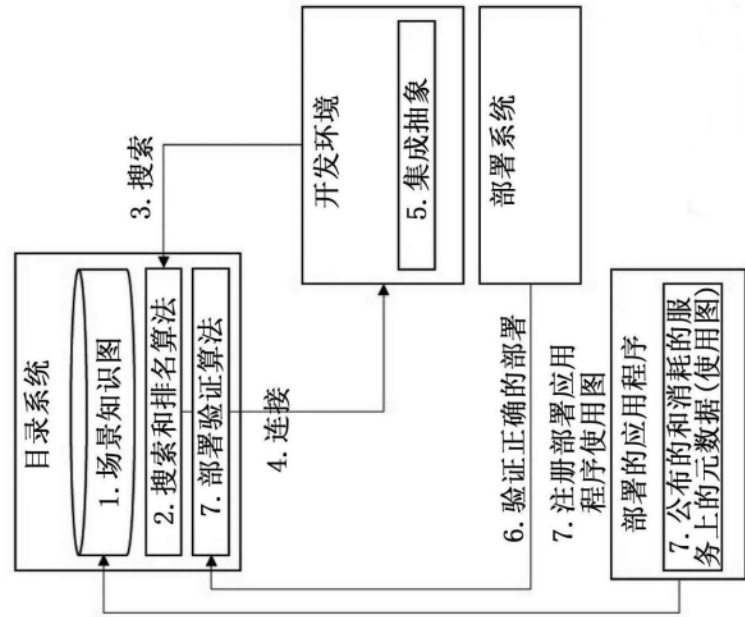


图14

可发现的企业

- 通过直接从Mendix工作室对企业资产进行自助服务重复使用改进团队的生产力
- 使（公民）开发者能够从Web建模器找到重复使用服务、API或数据，并且在没有技术集成知识的情况下开始使用这些构建应用程序
- 工作室中对应用程序进行工作时，通过帮助理解场景相关性改进团队的生产力

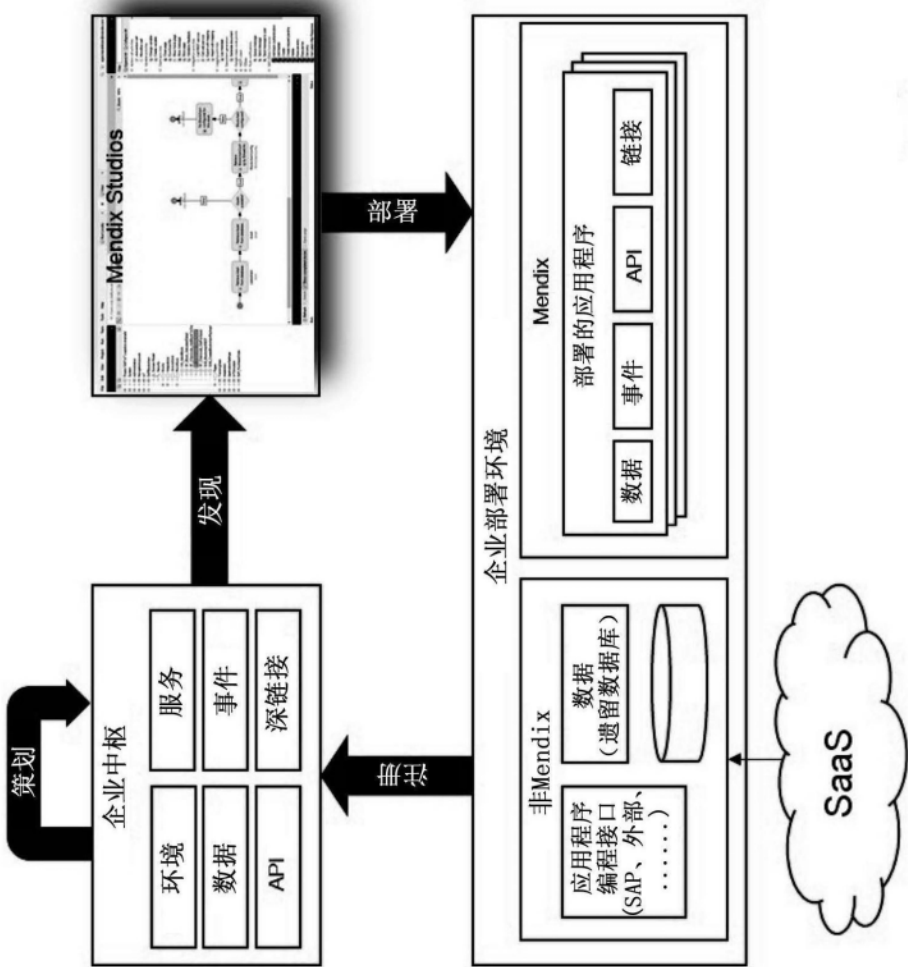


图15

Mendix企业中枢

Mendix企业中枢:

- 简化你的企业应用程序场景上的重复使用
- 发现-发现企业中的企业资产(例如,数据、事件和API),并且在它们的基础上进行构建
- 消耗-通过Mendix低代码平台中的拖放消耗企业资产
- 策划-自动控制对企业资产的访问,并且确保治理和遵守最佳实践

2019年蓝图: 解决数据共享的复杂性:

- 数据中枢-消除团队之间的协调开销,以理解使用哪些共享数据。任何开发者可发现和重复使用数据
- 远程实体-使用来自带有原装数据共享功能的其它应用程序的数据
- 不需要提供自定义微流逻辑以在系统之间同步数据
- 使用oData/REST作为初始协议以确保低摩擦进入: 没有附加基础设施
- 使用oData/REST提供与SAP的兼容性

图16

什么是Mendix企业中枢?

- 什么是企业中枢?
 - 发现可重复使用企业资产
 - 数据、API、服务、链接、业务事件、用户界面设计
 - 消耗
 - 与工作室拖放集成
 - 策划
 - 简化治理和监管
- 与其它中枢集成:
 - 在你的组织中找到任何东西
 - 在Mendix工作室中拖放非Mendix资产

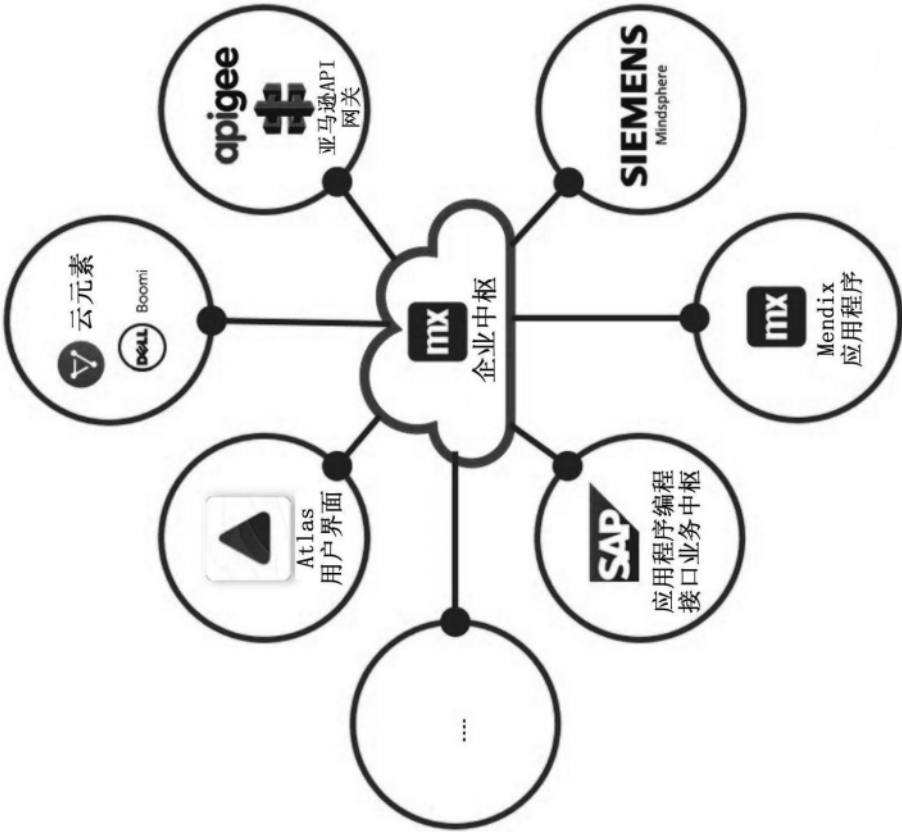


图17

Mendix企业中枢-全局

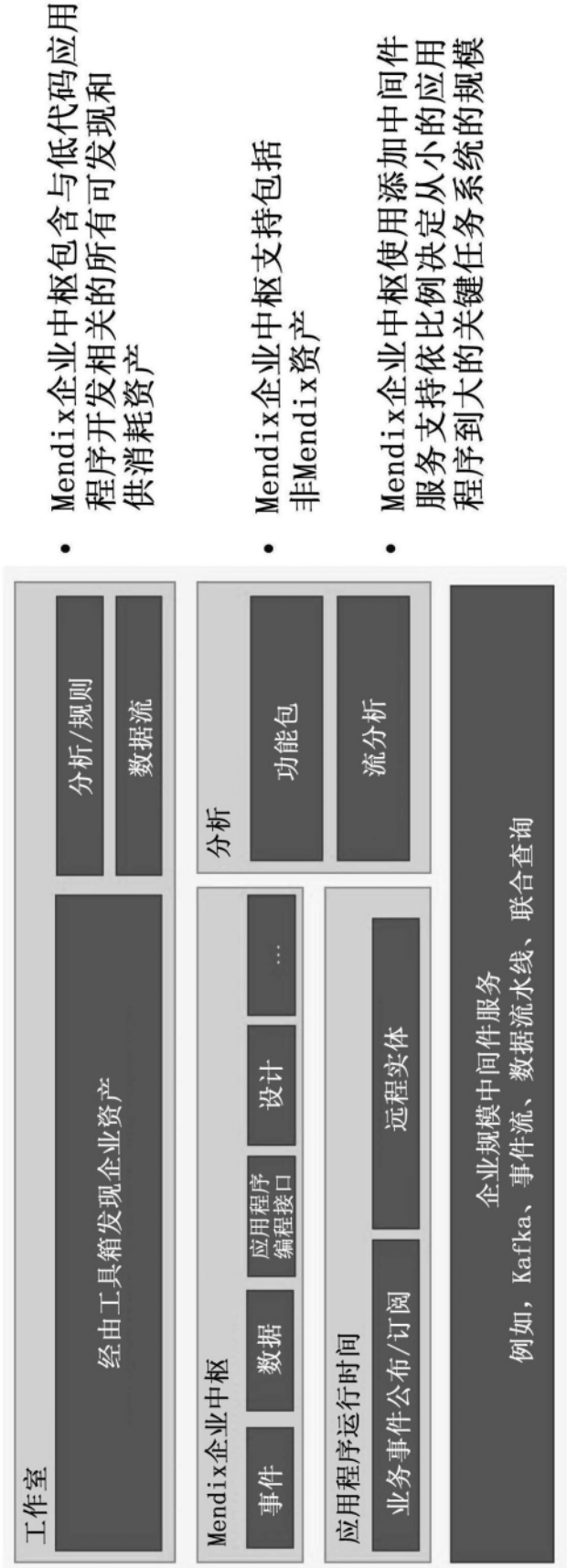


图18