



(12) 发明专利

(10) 授权公告号 CN 101405695 B

(45) 授权公告日 2013. 07. 24

(21) 申请号 200780009953. 2

G06F 17/00 (2006. 01)

(22) 申请日 2007. 02. 15

(56) 对比文件

(30) 优先权数据

11/385, 124 2006. 03. 21 US

CN 1313558 A, 2001. 09. 19, 全文.

US 2004139359 A1, 2004. 07. 15, 全文.

CN 1497444 A, 2004. 05. 19, 全文.

(85) PCT申请进入国家阶段日

2008. 09. 22

审查员 徐卫锋

(86) PCT申请的申请数据

PCT/US2007/004194 2007. 02. 15

(87) PCT申请的公布数据

W02007/108879 EN 2007. 09. 27

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 J·泽曼 G·A·琼斯

S·J·H·肯特 S·J·库克

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 陈斌

(51) Int. Cl.

G06F 9/44 (2006. 01)

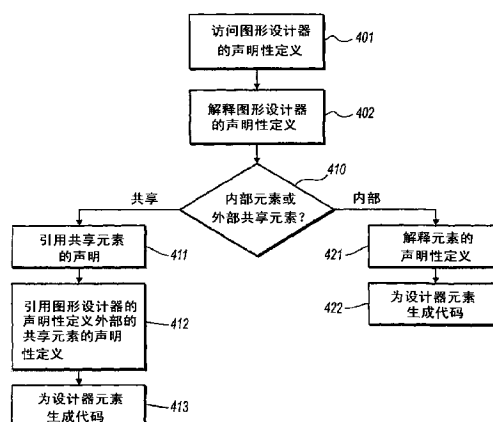
权利要求书2页 说明书6页 附图8页

(54) 发明名称

启用图形设计器重用的声明性定义

(57) 摘要

图形建模工具的一个或多个共享组件在多个图形建模工具之间的重用。图形设计器或图形建模工具, 允许构造其中对象、对象特性和对象相互关系可被可视化的模型的图形表示。因此, 可用其它图形建模工具的各部分支持和填充各种图形建模工具。



1. 一种用于在图形设计器内实现共享元素的计算机实现的方法,包括:

评估图形设计器的多个元素,至少一个元素在图形设计器内部定义,至少一个元素是共享元素;

标识至少一个共享元素并且标识至少一个非共享元素,每个标识都基于每个所标识元素的声明;

解释包括至少一个共享元素的声明的图形设计器的声明性定义,

所述共享元素包括域模型的组件,所述域模型包括域类、域类之间

的关系和枚举元素中的至少一个,以及

所述共享元素出现在模式定义中;

生成遵守所述声明性定义所定义的规范的指令序列;

引用在所述图形设计器的声明性定义的外部的所述共享元素的声明性定义的动作,所述引用动作不包括所述共享元素的完整的声明性定义;以及

使用所述图形设计器的声明性定义中的所述共享元素的声明,并使用所述共享元素的外部声明性定义,在所述图形设计器内实现所述共享元素的动作。

2. 如权利要求 1 所述的计算机实现的方法,其特征在于,还包括:

解释在所述图形设计器的声明性定义内部的非共享元素的声明性定义的动作。

3. 如权利要求 2 所述的计算机实现的方法,其特征在于,还包括:

使用非共享对象类型的声明性定义来实现所述非共享元素的动作。

4. 如权利要求 1 所述的计算机实现的方法,其特征在于,所述共享元素是第一类元素中的第一共享元素,所述方法还包括:

在所述图形设计器的声明性定义中检测第二类元素的第二共享元素的声明的动作;

引用在所述图形设计器的声明性定义外部的所述第二共享元素的声明性定义的动作;以及

使用所述图形设计器的声明性定义中的所述第二共享元素的声明,并使用所述第二共享元素的外部声明性定义,在所述图形设计器中实现所述第二共享元素的动作。

5. 如权利要求 1 所述的计算机实现的方法,其特征在于,所述共享元素是第一共享元素且所述图形设计器是第一图形设计器,所述方法还包括:

解释包括至少一个第二共享元素的声明的第二图形设计器的声明性定义的动作;

引用在所述第二图形设计器的声明性定义外部的所述第二共享元素的声明性定义的动作;以及

使用所述第二图形设计器的声明性定义中的所述第二共享元素的声明,并使用所述第二共享元素的外部声明性定义,来实现所述第二共享元素的动作。

6. 如权利要求 4 所述的计算机实现的方法,其特征在于,所述第一共享元素和所述第二共享元素的声明性定义是从共享库数据结构访问的。

7. 如权利要求 1 所述的计算机实现的方法,其特征在于,所述共享元素是第一共享元素且所述图形设计器是第一图形设计器,所述方法还包括:

解释包括至少一个第二共享元素的声明性定义的第二图形设计器的声明性定义的动作;以及

使用所述第二共享元素的声明性定义来实现所述第二共享元素的动作。

8. 如权利要求 7 所述的计算机实现的方法,其特征在于,所述第一共享元素和所述第二共享元素的声明性定义是从所述第二图形设计器内部的共享库数据结构访问的。

9. 如权利要求 1 所述的计算机实现的方法,其特征在于,所述共享元素包括含有域类、域类之间的关系和枚举元素中的至少一个的域模型的组件。

10. 如权利要求 1 所述的计算机实现的方法,其特征在于,所述共享元素包括含有形状、连接器、或装饰器元素中的至少一个的图形记法。

11. 如权利要求 1 所述的计算机实现的方法,其特征在于,所述图形设计器的声明性定义包括映射定义的声明性定义。

12. 一种用于在图形设计器内实现共享元素的系统,包括:

用于评估图形设计器的多个元素的装置,至少一个元素在图形设计器内部定义,至少一个元素是共享元素;

用于标识至少一个共享元素并且标识至少一个非共享元素的装置,每个标识都基于每个所标识元素的声明;

用于解释包括至少一个共享元素的声明的所述图形设计器的声明性定义的装置,

所述共享元素包括域模型的组件,所述域模型包括域类、域类之间的关系和枚举元素中的至少一个,以及

所述共享元素出现在模式定义中;

用于生成遵守所述声明性定义所定义的规范的指令序列的装置;

用于引用在所述图形设计器的声明性定义的外部的所述共享元素的声明性定义的装置,所述引用不包括所述共享元素的完整的声明性定义;以及

用于使用所述图形设计器的声明性定义中的所述共享元素的声明,并使用所述共享元素的外部声明性定义,在所述图形设计器内实现所述共享元素的装置。

启用图形设计器重用的声明性定义

[0001] 背景

[0002] 计算技术对人类进步贡献巨大。计算系统有众多物理形式,包括台式计算机、膝上型计算机、个人数字助理、电话、甚至常规上不被认为是计算系统的设备,诸如例如冰箱和汽车。因此,计算系统可以是具有任何类型的处理器和存储器的任何设备或系统。

[0003] 计算技术支持的一种常见的应用程序是图形建模工具(也被称为“图形设计器”)。图形建模工具便于图形模型的有序且高效的构造。图形模型包括相互关联的对象的集合的可视表示。图形模型可被各个领域以不同的方式使用。例如,工作流程、组织结构图、电子线路、软件架构、软件流程图等每个都可以使用图形模型来表示。可能有数千个其中图形建模工具可以有所用处的不同的应用程序。在每一情况下,对象的类型和相互关系可以大不相同。即使是在应用程序的单个域中,对如何显示对象和相互关系可以有明确的偏好。例如,一家银行可能偏爱一种图形建模样式以用于表示经济交易,而另一家银行可能将经济交易表示为不同的图形建模样式。

[0004] 构建图形建模工具是具有挑战性且耗时的任务。典型图形设计器的构建可以涉及实现图形记法的设计表面的实现的软件开发器、允许用户将元素拖放到设计表面上的工具箱、用于表示设计表面上的对象的特性的机制——这些特性表示关于用户正试图建模的问题的有意义的信息、以及像在模型数据中导航的资源管理等其它用户界面(UI)元素。

[0005] 在考虑对于不同的图形设计的巨大数量的域和偏好时,为每一域和偏好都构建图形建模工具的任务看来是不祥的。

[0006] 简要概述

[0007] 图形设计器或图形建模工具,允许构造包括各元素的模型的图形表示,其中一些元素是可视化的。尽管并非所需,但本发明的各实施例涉及一个或多个共享元素在多个图形设计器之间的共享,从而准许在给定图形设计器中使用更大的一组元素。一个位置可以具有一个元素的声明性定义,而该元素可以通过引用共享元素的声明性定义而不必明确地包括该共享元素的声明性定义而被合并到图形设计器的声明性定义中。

[0008] 提供本概述以便以简化的形式介绍将在以下详细描述中进一步描述的一些概念。本概述不旨在标识所要求保护的主题的关键特征或必要特征,也不旨在用于帮助确定所要求保护的主题的范围。

[0009] 附图简述

[0010] 使用附图以更具体地描述本发明的各实施例。可以理解,这些附图仅描述本发明的各典型实施例,从而不被认为是对其范围的限制,将通过使用附图用附加特征和细节来描述和解释这些实施例,附图中:

[0011] 图1示出本发明的原理的各实施例可以在其中操作的计算系统;

[0012] 图2示出显示图形设计器的用户界面;

[0013] 图3示出包括其中共享图形设计器的各元素的多个图形设计器的声明性定义的环境;

[0014] 图4示出根据本发明的原理的一实施例的用于实现具有共享元素的图形设计器

的方法的流程图；

[0015] 图 5 示出图形设计器的用户界面；

[0016] 图 6 示出图 5 的用户界面的图形树；

[0017] 图 7 示出另一图形设计器的用户界面；以及

[0018] 图 8 示出通用共享元素库的分层结构。

[0019] 详细描述

[0020] 本发明的各实施例涉及图形建模工具的声明性地定义的元素在多个图形建模工具中的重用。首先，将参考图 1 描述其中可使用本发明的原理的示例计算系统。然后，将参考后续附图进一步详细描述本发明的原理。

[0021] 本发明的各实施例可以包括含有各种计算机硬件的专用或通用计算机，这将在以下做出进一步讨论。图 1 示出可用于实现本发明的特征的示例计算系统 100 的示意图。所述计算系统只是这种合适的计算系统的一个示例，而并不旨在对本发明使用范围或功能提出任何限制。也不应该把本发明解释为对图 1 中示出的任一组件或其组合有任何依赖性要求。

[0022] 计算系统现在越来越多地采用各种形式。计算系统例如可以是手持式设备、电器、膝上型计算机、台式计算机、大型机、分布式计算系统、或甚至是常规上不被认为是计算系统的设备。在本说明书和权利要求书中，术语“计算系统”被广义地定义为包括包含至少一个处理器以及其上能含有可由处理器执行的计算机可执行指令的存储器的任何设备或系统（或其组合）。存储器可采取任何形式，且可取决于计算系统的本质和形式。计算系统可分布在网络环境上，且可包括多个组成计算系统。

[0023] 参考图 1，在其最基本配置中，计算系统 100 一般包括至少一个处理单元 102 和存储器 104。存储器 104 可以是系统存储器，它可以是易失性的、非易失性的或两者的某种组合。易失性存储器的示例包括随机存取存储器（RAM）。非易失性存储器的示例包括只读存储器（ROM）、闪存等。术语“存储器”也可在此处用来指示诸如物理存储介质等非易失性大容量存储。这样的存储可以是可移动或不可移动的，且可包括（但不限于）PCMCIA 卡、磁盘和光盘、磁带等。

[0024] 如此处所使用的，术语“模块”或“组件”可以指在计算系统上执行的软件对象或例程。此处描述的不同的组件、模块、引擎和服务可被实现为在计算系统上执行的对象或进程（例如，作为分开的线程）。尽管此处描述的系统和方法可用软件实现，但用硬件以及软件和硬件的组合的实现也是可能的且已被想到。

[0025] 在以下描述中，参考由一个或多个计算系统执行的动作来描述本发明的各实施例。如果这些动作是用软件实现的，则相关联计算系统中执行该动作的一个或多个处理器响应于执行了计算可执行指令而引导计算系统的操作。这一操作的一个示例涉及对数据的操纵。计算机可执行指令（和所操纵的数据）可被存储在计算系统 100 的存储器 104 中。

[0026] 计算系统 100 也可包含允许计算系统 100 例如通过网络 110 与其它计算系统通信的通信信道 108。通信信道 108 是通信介质的示例。通信介质一般以诸如载波或其它传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或其它数据，并且包括任何信息传递介质。作为示例而非限制，通信介质包括有线介质，诸如有线网络或直接线连接，以及无线介质，诸如声学、无线电、红外线和其它无线介质。如此处所用的术语计算机可

读介质既包括存储介质又包括通信介质。

[0027] 本发明的范围内的各实施例也包括用于承载或其上储存有计算机可执行指令或数据结构的计算机可读介质。这样的计算机可读介质可以是可由通用或专用计算机访问的任何可用介质。作为示例而非限制,这样的计算机可读介质可包括物理存储和/或存储器介质,诸如 RAM、ROM、EEPROM、CD-ROM 或其它光盘存储、磁盘存储或其它磁存储设备、或可用于承载或存储计算机可执行指令或数据结构形式的所需程序代码装置且可由通用或专用计算机访问的任何其它介质。当信息通过网络或另一通信连接(硬连线、无线或硬连线或无线的组合)传输或提供给计算机时,该计算机将该连接完全视为计算机可读介质。因此,任何这样的连接被适当地称为计算机可读介质。以上的组合也应包括在计算机可读介质的范围之内。

[0028] 计算机可执行指令包括例如,使通用计算机、专用计算机、或专用处理设备执行某一功能或某组功能的指令和数据。尽管用对结构特征和/或方法动作专用的语言描述了本主题,但可以理解,所附权利要求书中定义的主题不必限于在此所述的具体特征或动作。相反,在此所述的具体特征和动作是作为实现权利要求的示例形式公开的。

[0029] 图 2 示出显示图形设计器的用户界面 200。在本说明书和权利要求书中,“图形设计器”或“图形建模工具”是准许构造一个或多个可视化对象和相互关系的图形模型的任何应用程序。在一些情况下,单个应用程序可以支持多个图形设计器。如前所述,不同的应用程序域和偏好可以保证这些应用程序域和偏好的不同的对象类、相互关系和可视化。本发明的原理准许在多个图形设计器之间共享声明性地定义的元素。因此,术语“图形设计器”和“图形建模工具”是绝不限于现将参考图 2 描述的示例的广义术语。

[0030] 用户界面 200 包括实现图形记法的设计表面 201、允许用户将元素拖放到设计表面上的工具箱 (Toolbox) 窗格 202、用于表示设计表面上的对象的特性的特性 (Properties) 窗格 203 和在模型数据中导航的导航窗格 204。

[0031] 图形设计器 200 可被用来构造各种模型。然而,每一图形设计器 200 都将准许这些模型的某些类型的对象类、相互关系和可视化。以声明性方式使用声明性定义来定义图形设计器。取代使用提供给计算机以定义图形设计器的详细指令列表,图形设计器声明性定义指定一组声明性定义,每一个都定义图形设计器的一相关联的元素。“元素”可以是图形设计器的任何方面,包括例如该图形设计器的对象类、相互关系、可视化、或该图形设计器的任何其它方面(可视化的或非可视化的)。声明性定义的目的是成为描述图形设计器的行为的最小的、高效的方法,从而省略所有不相关的细节。在某一将来的观点上,声明性定义可由解释器或代码生成器来解释以生成遵守该声明性定义所定义的规范的指令序列。表示图形设计器的声明性定义的一种方法是使用可扩展标记语言 (XML) 文档。然而,声明性定义也可以用二进制或其它非文本形式来表示。可通过仅仅更改描述该元素的声明性定义,而非更改实现图形设计器的细节的可执行代码来更改图形设计器的一给定元素的实现。

[0032] 如刚才所述,图形设计器的声明定义可以是可扩展标记语言 (XML) 文档的形式。在一实施例中,图形设计器的声明性定义可以包括对特定问题域专用的并且因而定义专用对象、其特性和各对象之间的关系类型的对象类形式的各域模型组件。取决于问题域,这些元素可以有相当大的变化。例如,如果图形设计器被用在财务应用程序中,则对象类可以包

括现金、支付、发票等等。如果图形设计器被用在时间安排应用程序中,则对象类可以包括任务、完成依赖性顺序等等。如果用户构建设计器来对计算机网络的结构进行建模,则该用户可以定义具有像“Operating System(操作系统)”和“Storage Capacity(存储能力)”等特性和像“Network Connection(网络连接)”等关系的被称为“Server(服务器)”的对象。

[0033] 域模型可以包括三种元素:域类、域类之间的关系(其自身也具有域类的特性)和枚举。这些元素的每一个都可以出现在模式定义中,且可以通过从另一设计器引用该元素而被独立地重用。

[0034] 图形设计器声明性定义还可以包括可被用来可视化设计模型的对象和关系的记法元素。例如,在计算机联网域中,图形设计器可以使用图标来可视化“Server”对象并将“Server”对象的名字显示为该图标旁边的文本。可使用到另一“Server”对象的箭头来可视化网络连接。

[0035] 记法定义可以指定三种元素:形状、连接器和装饰器(decorator)。例如,形状可以是几何形状(使用像‘矩形’等预定义的几何图形)、其中对象被表示为图像的图像形状或具有丰富行为的复杂形状。连接器是连接两个形状的元素。对于连接器,设计器构建器可以指定该连接器可将哪一类型的形状连接到哪一其它类型的形状。装饰器给形状和连接器添加文本或图标。这些元素中的每一个还可以出现在图形设计的声明性定义的模式定义中。

[0036] 声明性定义还可以包括映射定义。映射定义指定域模型的哪些元素与记法的哪些元素相关联。例如,在上述示例中,图形设计器可以将对于“Server”的域对象映射到被用来可视化“Server”对象的形状。

[0037] 映射定义指定四种元素:TextDecoratorMap(文本装饰器映射)、IconDecoratorMap(图标装饰器映射)、ShapeMap(形状映射)和ConnectorMap(连接器映射)。TextDecoratorMap或IconDecoratorMap将显示于装饰器中的值映射到对象的值。ShapeMap将形状自身映射到一个或多个对象。该映射取决于形状的类型。每一ShapeMap可以包含用于该形状的DecoratorMap的列表。ConnectorMap将连接器映射到两个域对象之间的关系。每一ConnectorMap可以包含用于该连接器的DecoratorMap的列表。再一次,这些元素中的每一个可以出现在图形设计声明性定义的模式定义文件中。

[0038] 图3示意性示出包括其中有共享的元素的多个图形设计器的声明性定义的软件架构环境300。特别地,在该示例中,第一图形设计器声明性定义310包括元素311A到311J以及可能的由水平省略号311K表示的其它元素。第二图形设计器声明性定义320包括元素321A到321H以及可能的由水平省略号321J表示的其它元素。还可能有由水平省略号330所表示的图形设计器的其它声明性定义。

[0039] 图形设计器声明性定义包括其各组成元素的一些声明性定义。例如,第一图形设计器声明性定义310包括对于元素311A、311B、311D、311E、311F、311G和311I的声明性定义。第二图形设计器声明性定义320包括对于元素321B、321C、321E、321F和321G的声明性定义。

[0040] 图形设计器声明性定义还可以包括对于这些元素的某一些的声明,而不包括该元素的经表达的定义。例如,第一图形设计器声明性定义310包括元素311C、311H和311J的

声明,该声明采用了对共享元素 341A、341C 和 341B 的对应声明性定义的对应引用 312、314 和 313 的形式。第二图形设计器声明性定义 320 包括对于元素 321A、321D 和 321H 的声明,该声明采用了对共享元素 341B、341C 和 341D 的对应声明性定义的对应引用 322、323 和 324 的形式。

[0041] 因此,通过仅仅引用共享元素而不必包括该元素的完整的声明性定义,可以在各种不同的图形设计器中重用该共享元素的声明性定义。在一实施例中,共享元素的声明性定义可以被包括在共享元素的库 340 中。例如,在这一情况下,该库被示为包括对应元素的声明性定义 341A 到 341D 以及可能的由垂直省略号 341E 表示的其它声明性定义。库 340 可以在所有图形设计器声明性定义的外部。然而,库 340 还可以被全部包含在单个图形设计器声明性定义中,或者甚至分布在多个图形设计器声明性定义中。共享元素可以是例如,1) 包括域类、域类之间的关系和枚举元素中的至少一个的域模型的组件,2) 包括形状、连接器、或装饰器元素的至少一个的图形记法,和 / 或 3) 映射定义。

[0042] 图 4 示出根据本发明的原理的一实施例的用于在图形设计器声明性定义中重用共享元素的方法 400 的流程图。对于每一图形设计器声明性定义可以重复方法 400。将频繁参考图 3 中示出的示例架构 300 来描述方法 400。

[0043] 对于一给定的图形设计器(也被称为“图形建模工具”)的声明性定义,方法 400 访问该图形设计器的声明性定义(动作 401)。例如,访问图 3 的图形设计器声明性定义 310。随后解释图形设计的声明性定义(动作 402)。

[0044] 在解释时,解释组件可以评估该图形设计器的多个元素。更具体地,评估这些元素的每一个的声明性定义。评估元素的流程将取决于该元素是否定义在图形设计器声明性定义内部(如元素 311A、311B、311D、311E、311F、311G 和 311I 的情况),或该元素是否是定义在图形设计器声明性定义外部的共享元素(如元素 311C、311H 和 311J 的情况)而不同(判定框 410)。该元素可基于元素声明本身而被标识为共享的或非共享的元素。在本说明书和权利要求书中,组件的“声明”被定义为对声明性定义的引用,而不包括该声明性定义。

[0045] 如果该元素是外部共享元素(判定框 410 中的“共享”),则访问该元素的声明(动作 411),该声明包括对该元素的声明性定义的位置的引用。随后访问包含该声明性定义的位置,在该位置上引用了声明定义(动作 412)。随后使用图形设计器声明性定义外部的共享元素的声明性定义来为该共享设计器元素生成代码(动作 413)。例如,该代码可以使得该共享元素如果其是可视元素则被呈现在显示器上,和 / 或不管其是可视还是不可视元素都被实例化在存储器中。对于该设计器元素的代码生成可使用该声明性定义以任何方式实现。

[0046] 另一方面,如果该元素的声明性定义在图形设计器声明性定义的内部(判定框 410 中的“内部”),则使用解释组件来解释该声明性定义(动作 421),并为该设计器元素生成代码(动作 422)。在执行该代码时,使该元素被呈现和 / 或实例化在存储器中从而执行其定义的功能。

[0047] 已描述了本发明的原理和一些具体示例,现将参考图 5 到 8 描述更具体的示例。在第一图形设计器中,声明性定义被用在公司环境中的业务分析者应用程序中。由对应的图形设计器所构建的所得图形模型示于图 5 中。图 5 的用户界面示出众多像“Customer(顾客)”等的业务实体、其特性和彼此之间的关系。

[0048] 底层设计器模式定义在上述设计器中可视化的域类“Business Entity(业务实体)”。设计器最终用户正在使用该设计器创建像“Customer”等的对象。这里是域类“BusinessEntity”的定义。图6示出作为图形树的图形设计声明性定义。

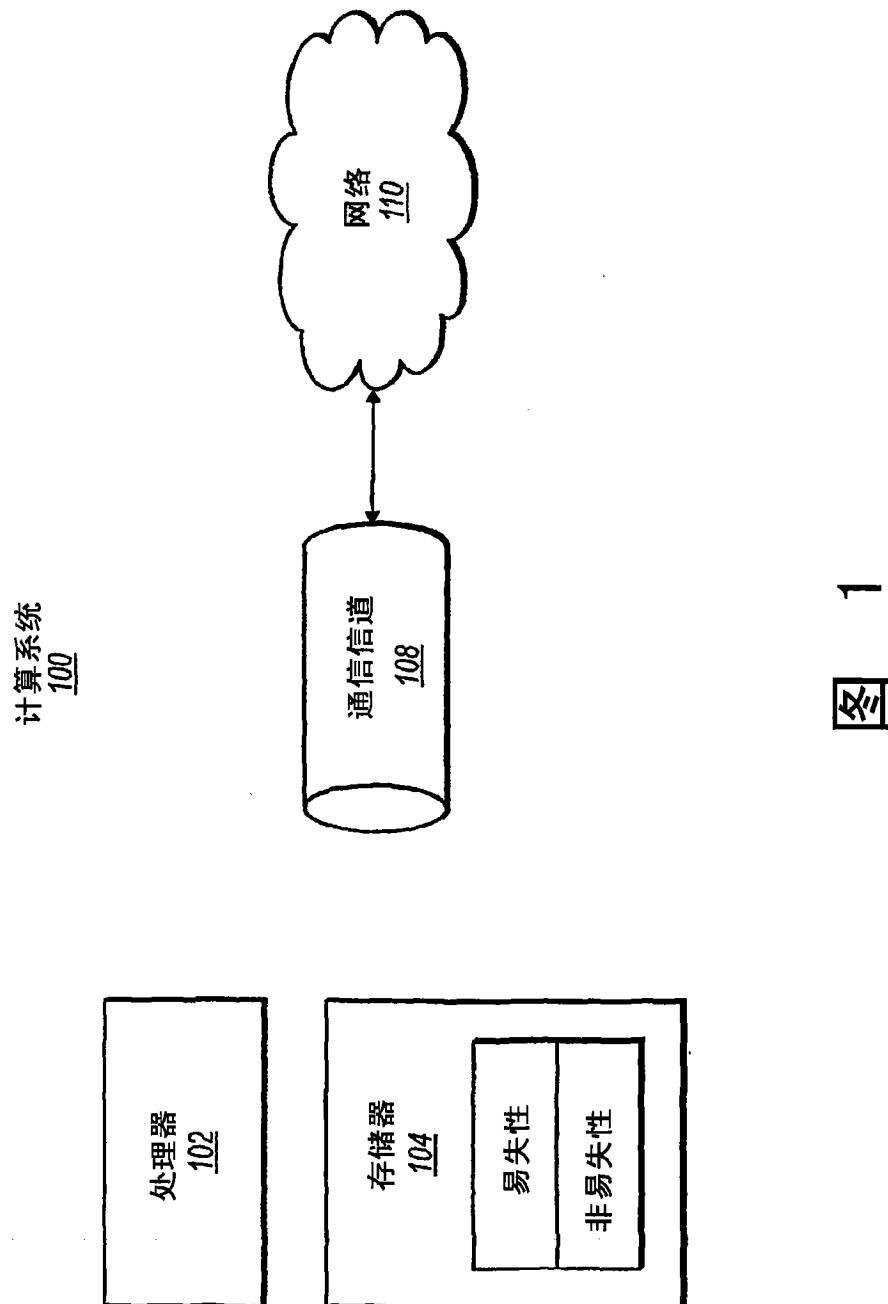
[0049] 在另一图形设计器中,业务分析者可以定义一个业务流程和在特定流程步骤中使用哪些业务实体。图7示出显示从第二图形设计器所得的图形模型的用户界面。

[0050] 图7的用户界面示出一业务流程。该流程开始于标为‘Check Inventory(检查库存)’的步骤。该图表形状示出该流程步骤所涉及的众多业务实体,像‘Order(定单)’或‘Customer(顾客)’等。那些对象被定义在其它设计器的域模式中,且在该图形设计器中重用。

[0051] 图8示出包括定义其中共享的元素可被定义来供其它图形设计器使用的声明性定义的一部分的元素(具有父节点“Ds1Library(Ds1库)”)的图形设计声明性定义(示为具有父节点“Ds1”)的分层结构的示例。

[0052] 虽然参考图4描述了各个方法,但是本发明的原理延及包括其上具有一个或多个可执行指令的计算机可读介质的计算机程序产品,该指令在被服务器计算系统上的一个或多个处理器执行时使该计算系统执行这些方法中的全部或部分或任一个或两者。例如,参考图1,存储器104和通信信道108可以表示这种计算机可读介质的示例。存储器104表示物理存储和/或存储器介质形式的物理计算机可读介质的示例。本发明的原理还延及计算系统本身,其通过硬件、软件或硬件和软件的组合来配置以执行图4的方法的所有部分。

[0053] 因此,本发明的原理准许共享元素在多个图形设计器声明性定义之间重用。本发明可具体化为其它具体形式而不背离其精神或本质特征。所述实施例在所有方面都应被认为仅是说明性而非限制性的。从而,本发明的范围由所附权利要求书而非前述描述指示。落入权利要求书的等效方式的含义和范围内的所有改变应被权利要求书的范围涵盖。



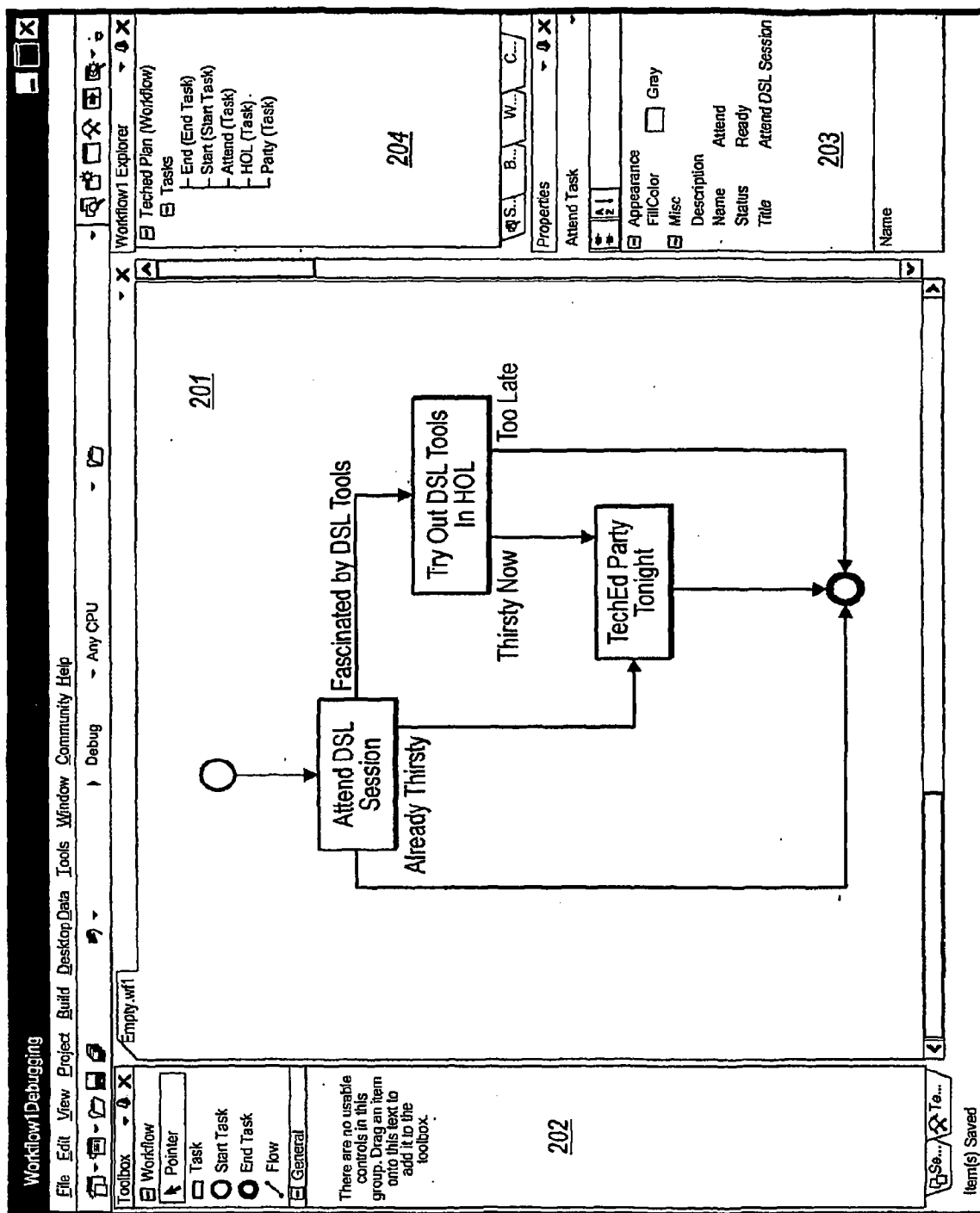


图 2

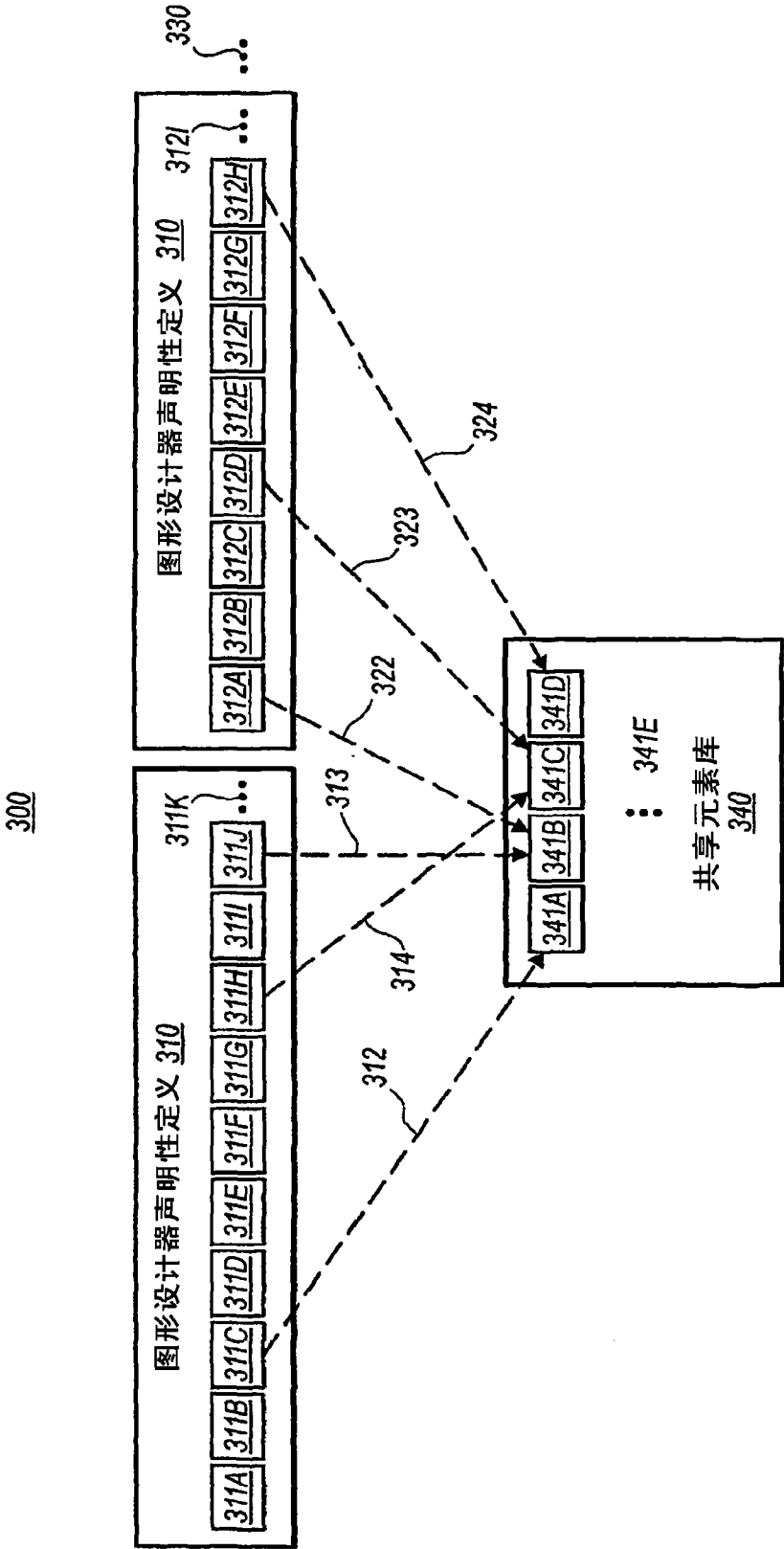


图 3

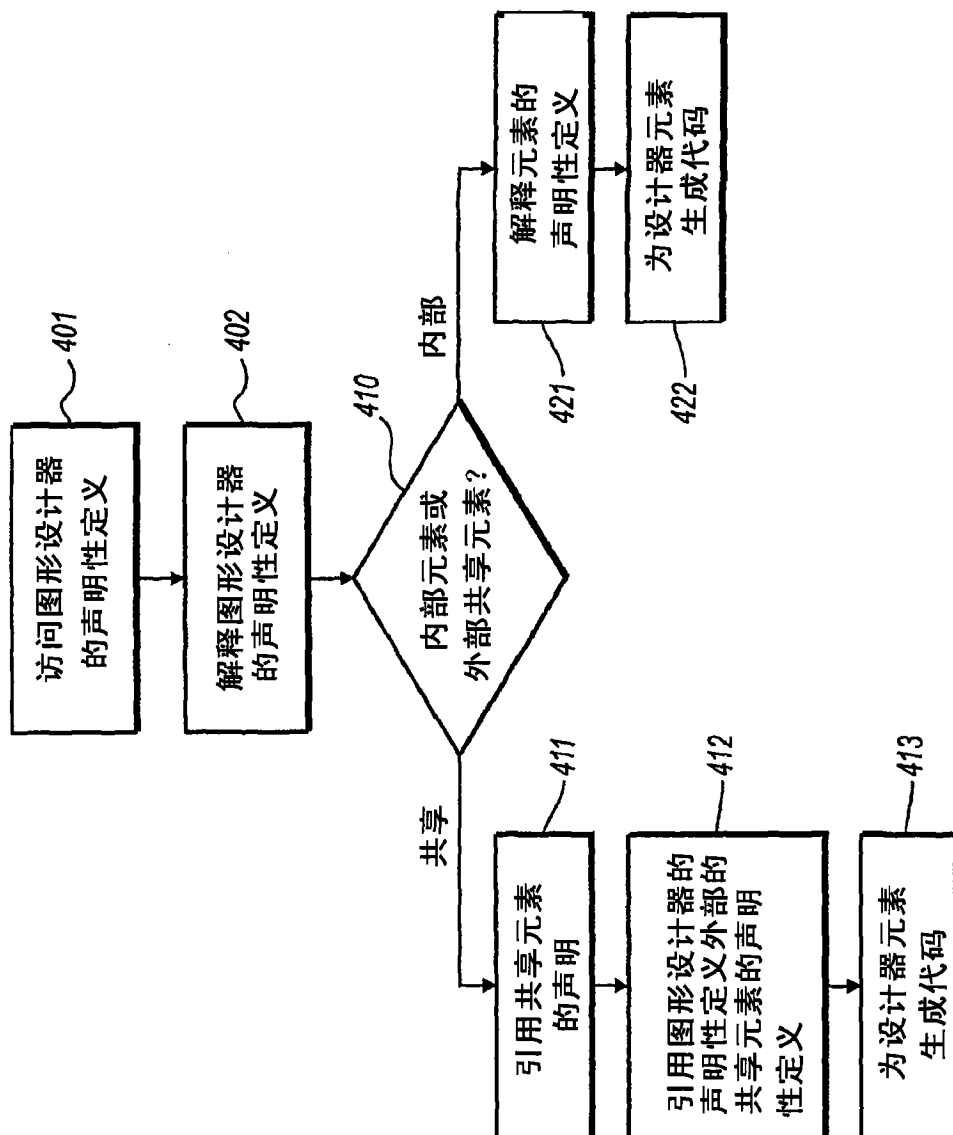


图 4

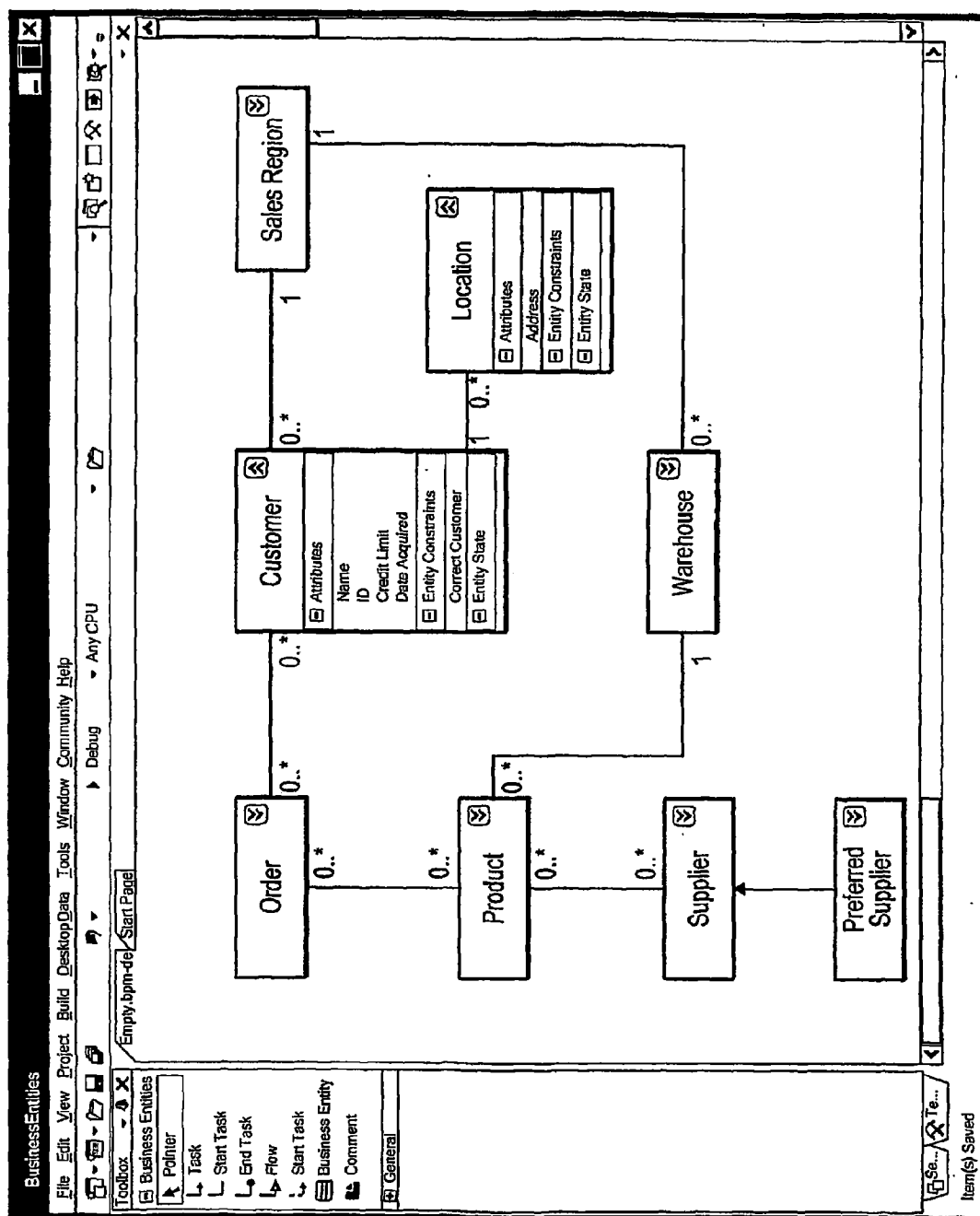


图 5

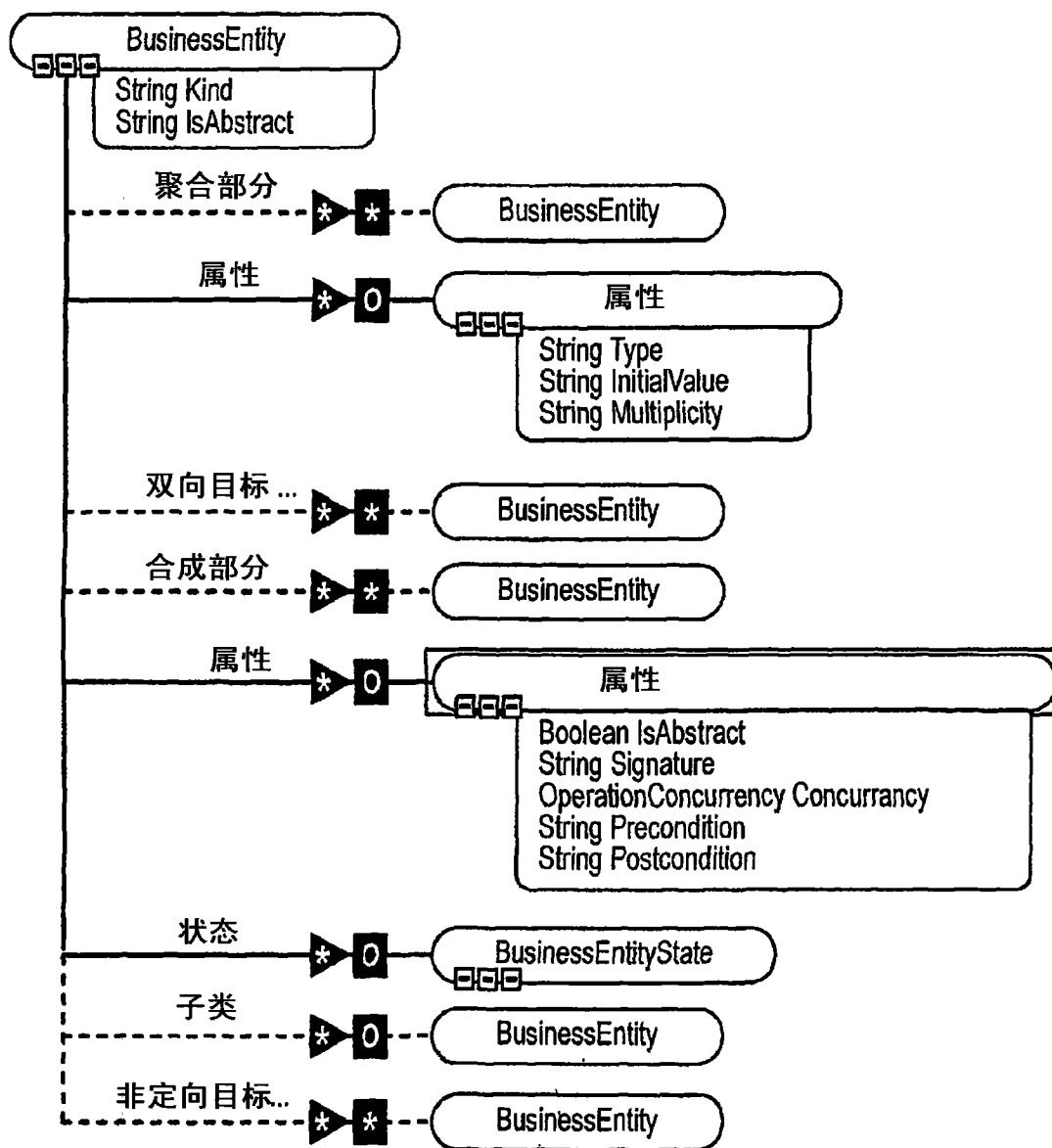


图 6

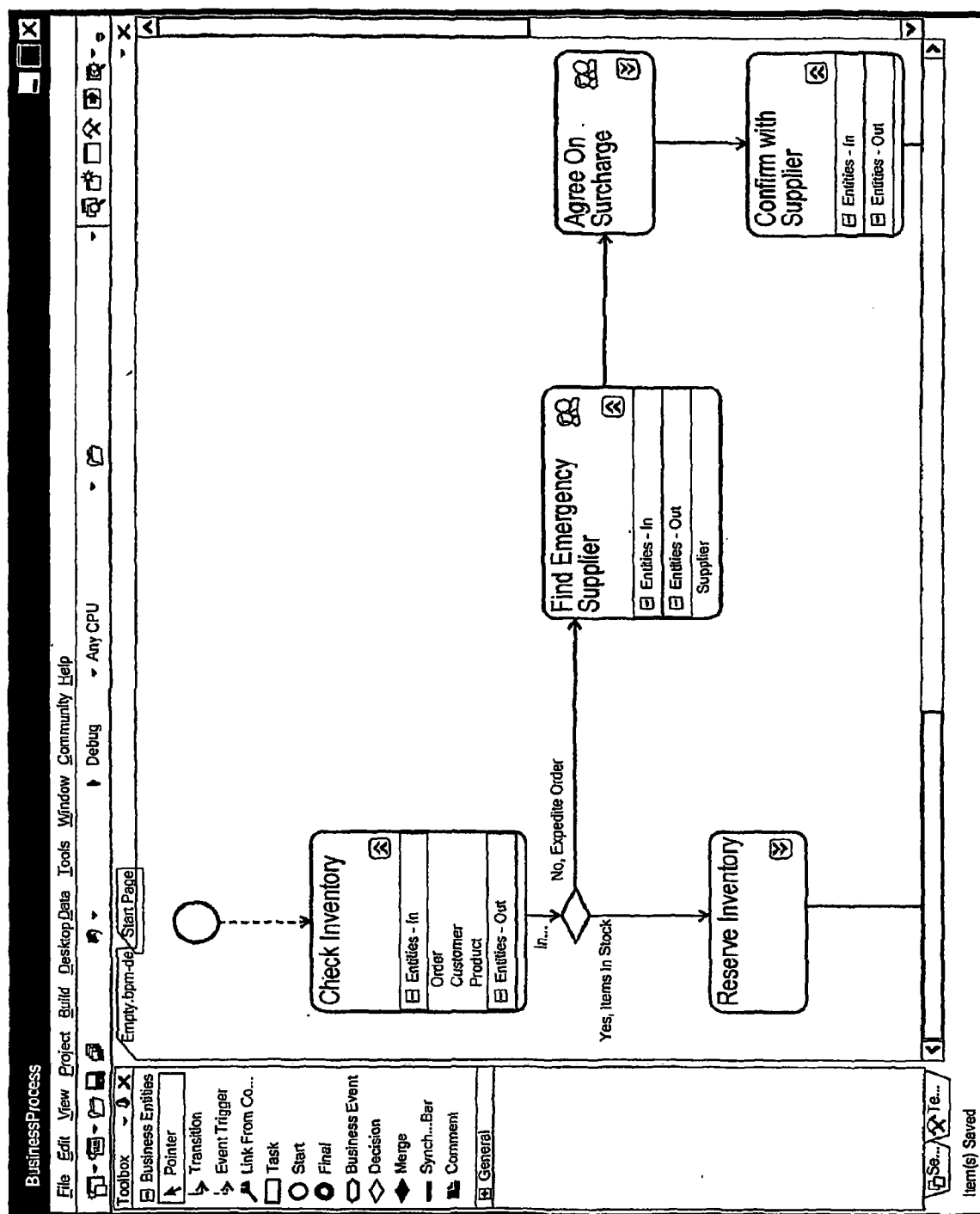


图 7

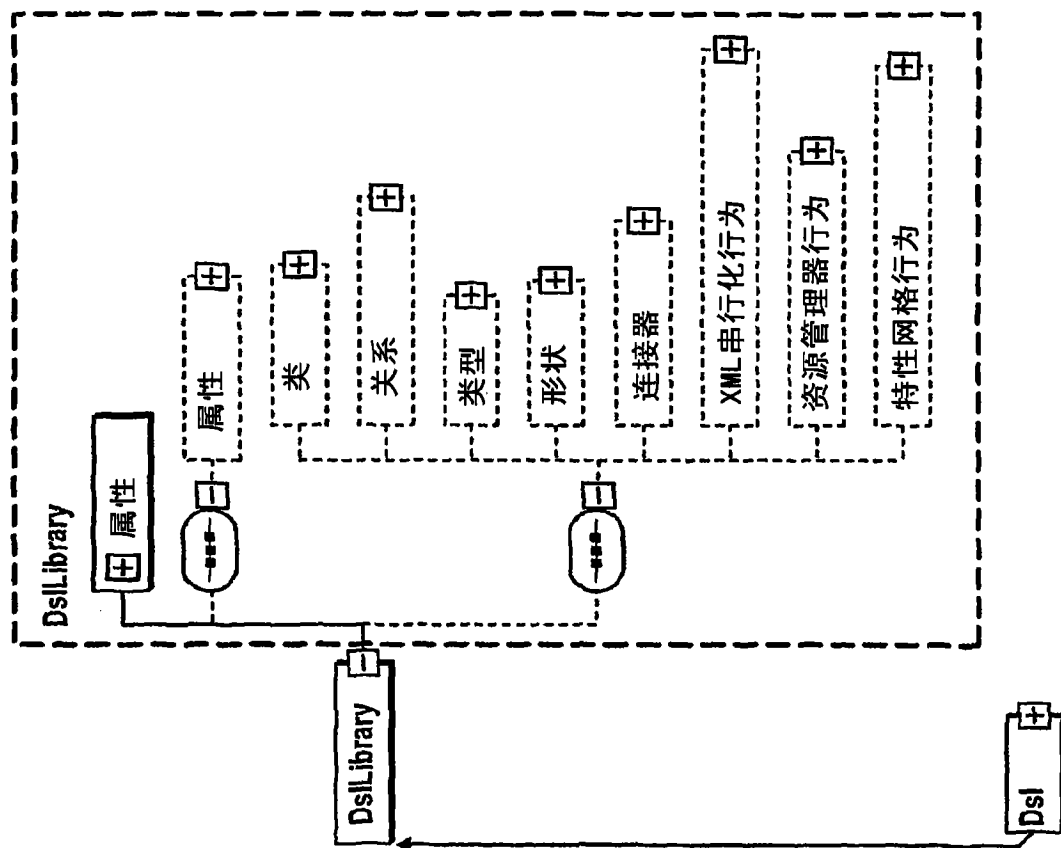


图 8