(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101681263 B
(45)授权公告日 2013.12.04

(21)申请号 200880015938.3
(22)申请日 2008.03.04
(30)优先权数据
07301047.2 2007.05.16 EP
(85)PCT申请进入国家阶段日
2009.11.13
(86)PCT申请的申请数据
PCT/EP2008/052608 2008.03.04
(87)PCT申请的公布数据
WO2008/138654 EN 2008.11.20
(73)专利权人 国际商业机器公司
地址 美国纽约阿芒克
(72)发明人 R·科克雷 B·波尔捷
(74)专利代理机构 北京市金杜律师事务所 11256
代理人 吴立明
(51)Int.Cl.
*G06F 9/44*(2006.01)
(56)对比文件
US 2006165123 A1,2006.07.27, 全文.
WO 03001343 A2,2003.01.03, 全文.
US 2003120621 A1,2003.06.26, 全文.
US 2005085937 A1,2005.04.21, 说明书第[0008]-[0010]段,第[0019]段,第[0021]-[0023]段,第[0025]段,第[0028]段,第[0030]-[0032]段,图1,图2,图4).
审查员 曾璇

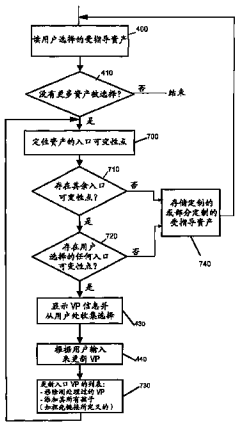
权利要求书2页 说明书9页 附图9页

(54)发明名称

用于开发基于软件资产的解决方案的一致方法和系统

(57)摘要

本发明涉及用于开发基于软件资产的解决方案的一致方法、系统和计算机程序。具体地,公开了一种用于消耗可复用软件资产的方法、计算机程序和系统,所述资产用元件和属性描述,所述资产包含至少一个可变元素(VP),可变元素包含至少一个变量。用户通过首先选择要消耗的资产在计算机上执行程序。对应于资产的决策树被遍历,每个决策点对应于可变元素。通过让用户输入以修改对应的可变元素的变量来处理决策点。存储修改的可变元素。决策点的依赖关系通过可变元素中的依赖关系属性来指示。程序可被用户主动停止或者当完全遍历决策树时停止。当决策树被部分遍历时,程序可在最近处理的决策点之后重启,从被部分修改的对应可变元素开始。




CN 101681263 B

1. 一种在计算机上执行的用于消耗可复用软件资产的方法，每个资产可以根据决策树来定制，所述资产使用包括元素和属性的结构化语言 RAS 来描述，一些元素是可变元素，每个可变元素包含至少一个变量元素，所述方法包括：

接收用户向计算机指示其想要消耗一个选定资产；

计算机遍历决策树，其中每个决策点作为所述选定资产的可变元素而被编码，所述遍历是通过读取所述选定资产的描述以及处理所述决策树的每个决策点来进行的，所述处理包含：

向用户提供与包括所述可变元素的所述至少一个变量元素的决策点相关联的可变元素；

从所述用户处收集对所述可变元素的修改；

相应地修改所述可变元素；

存储经过修改的资产描述。

2. 根据权利要求 1 所述的方法，其中遍历决策树的步骤在一轮处理中执行。

3. 根据权利要求 1 所述的方法，其中遍历决策树的步骤重复部分地执行，执行的次数等于遍历整个决策树所需的处理的轮数，这是通过：在每次部分遍历处理中，读取先前部分遍历处理中存储的所述经过修改的资产描述；以及开始处理在所述先前处理中最后处理的决策点之后的每个决策点。

4. 根据权利要求 3 所述的方法，其中遍历决策树的步骤还包括：

读取具有根标识属性的可变元素；

读取具有根依赖关系属性的所有可变元素；

读取依赖于根的可变元素的标识属性；

相继地读取具有依赖关系属性的所有可变元素，所述可变元素的标识属性已经读取。

5. 根据权利要求 1-4 中任一项所述的方法，其中在处理决策点

之后，部分遍历树的每一轮处理由用户主动停止。

6. 根据权利要求 1-4 中任一项所述的方法，其中使用可复用资产规范作为结构化语言来描述所述资产。

7. 根据权利要求 4 所述的方法，其中所述依赖关系属性是结构化语言 RAS“reference”属性。

8. 根据权利要求 1 的方法，其中用户提供、计算机向用户提供、以及计算机从用户处收集的步骤是通过图形用户界面完成的。

9. 根据权利要求 1 所述的方法，还包括以下初始步骤：

创建资产描述，对于相应资产决策树中的每个决策点，所述资产描述包括具有标识属性和依赖关系属性的元素，所述依赖关系属性包含所述决策点在所述资产决策树中所依赖的元素的标识属性。

10. 根据权利要求 1-4 中任一项所述的方法，其中从用户处收集修改的步骤包括从可变元素的所有变量元素中选择至少一个变量元素。

11. 根据权利要求 1-4 中任一项所述的方法，其中从用户处收集修改的步骤包括从用户处收集新的变量元素以及随附代码元素。

12. 根据权利要求 1-4 中任一项所述的方法，还包括：

在完成对所述决策树的遍历之后，将所述经过修改的资产描述应用于所述资产的内容，以创建定制的资产。

13. 一种包括适于执行根据权利要求 1-12 中任一项所述的方法的装置的系统。

# 用于开发基于软件资产的解决方案的一致方法和系统

## 技术领域

[0001] 本发明主要涉及开发软件解决方案；更具体地，本发明应用于开发基于软件资产的解决方案的方法。

[0002] 背景技术

[0003] 软件资产是为了在后续上下文中重复应用而被创建或完成的一个或多个相关产品的集合。资产的消费者是从解决方案业务建模、分析（使用的资产是模型）和设计到应用开发（使用的资产是代码段）的任何类型的 IT 或业务过程解决方案的架构师或设计师。

[0004] IT 行业具有标准化的软件资产封装。对象管理组 (OMG)(OMG 是 Object Management Group 公司在美国和 / 或其他国家的注册商标或商标）的可复用资产规范 (RAS) 是关于可复用软件资产的结构、内容和描述的一套指导和建议（见 2005 年 11 月的版本 2.2)。RAS 标识具有一致封装的软件资产的种类。RAS 支持对资产的定制，这对软件资产消费者而言是一个关键点。在 RAS 中，资产是产品 (artifact) 的容器，可变性点 (variability point) 是产品中预期将由消费者进行调整以适应目标解决方案的点。利用 RAS 的可变性支持，例如可以仅消耗资产的子集。

[0005] 2003 年 3 月 6 日公布的专利申请 US 2003/0046282 公开了资产检索模块，其用于根据用户的输入，从资产库有选择地检索软件资产的子集。本专利申请的目的还在于基于消费者的定制决策来提供促进访问软件资产的步骤的解决方案。但是，本专利申请的目的更关注于资产的创建和消耗，而不是其访问或搜索。

[0006] 在支持软件资产封装和访问的基础上，需要提供对可复用软件 资产的消耗的完全支持。可以使用 RAS 来制定用于安装、定制和使用软件资产的规则。在 RAS 中，资产的使用 (Usage) 部分包含用于安装、定制和使用资产的规则。虽然使用文本描述了将由用户或工具执行以便使用资产的活动，但是该自由形式的文本没有提供足够的结构化指导。

[0007] 为了消耗资产，用户还需要考虑其想要实现的基于资产的解决方案。在 Nokia 研究中心的 Alexandre Ran 和 Juha Kuusela 发表于 1996IEEE，Proceedings of IWSSD-8 的论文 Design Decision Trees 中，提出将设计模式的分级组织应用于设计决策树，该设计决策树是对较早决策施加的设计决策约束的部分排序。一般而言，决策树可以帮助用户通过一系列决策达到最终目标，即，树的一个叶子。通过应用与软件资产集合相对应的设计模式，软件资产消费者可以得到决策树。一旦决定了好的设计路径（即，基于资产的解决方案），消费者必须使用诸如 RAS 的库来执行对目标解决方案的每个软件资产的定制。

[0008] 这种方法的主要缺点在于，因为资产提供者在一侧创建资产存储库而通常在文本文件中（包含编辑的文本、HTML、XML、SGML 等等）创建单独的文档，因此资产消费者建立基于软件资产的解决方案要依赖于两个分离的信息源，即，所消耗的资产和指导此消耗的决策树，这导致了两步的方法：浏览决策树并决定采取哪条路径，即选择通向目标的路径；第二步是必须沿着该路径来定制资产。需要为软件资产消费者提供一种构建目标软件解决方案而避免两步（路径决策、软件资产定制）过程的方法。

## 发明内容

[0009] 本发明的目的是为可复用软件资产的消费者提供一种用于设计其基于资产的解决方案并同时定制所复用的软件资产的方法。

[0010] 根据权利要求 1，本目的是通过一种在计算机上执行的用于消耗可复用软件资产的方法来实现的，其中每个资产可以根据决策树来 定制，所述资产使用包括元素和属性的结构化语言来描述，某些元素是可变元素（可变性点），每个可变元素包含至少一个变量元素（变量），所述方法包括：

[0011] - 用户向计算机指示其想要消耗一个选定资产；

[0012] - 计算机遍历决策树，其中每一决策点被编码为所选资产的可变元素，所述遍历是通过读取选定资产的描述以及通过以下方式来处理决策树的每个决策点：

[0013] - 为用户提供与包括所述可变元素的至少一个变量元素的决策点相关联的可变元素；

[0014] - 从用户收集对所述可变元素的修改；

[0015] - 相应地修改所述可变元素；

[0016] - 存储修改后的资产描述。

[0017] 本发明的目的还可以通过如权利要求 2-12 中任一项所述的用于消耗可复用软件资产的方法来实现。本发明的目的还可以通过根据权利要求 13 的包括编程代码指令的计算机程序产品来实现，当所述程序在计算机上执行时，编程代码指令用于执行根据权利要求 1-12 中任一项所述方法的步骤。本发明的目的还可以通过根据权利要求 14 所述的包括适于执行根据权利要求 1-12 中任一项所述方法的装置的系统来实现。

[0018] 按照本发明的方法，软件资产消费者是利用基础决策树来指导的，其中该基础决策树实现了要做出的消耗资产的决策之间的层级关系。决策树嵌入在资产中，并通过将每一决策点映射到资产中的可变性点而被编码。决策树分支（从一个决策点到下一个决策点的链接）由可变性点的依赖关系来表示。此方法允许在一个步骤中建立基于软件资产的解决方案。

[0019] 此解决方案的一个优点是，与传统树相反，它可以从非根位置进入决策树或者从非叶节点退出决策树。利用本发明的方法，可以在到达叶节点层级之前停止，同时记录决策。这支持高度灵活的工具。

[0020] 此解决方案的其他优点可以列举如下：

[0021] - 利用本发明的解决方案，消耗资产的规定性指导和资产本身是通过设计一起完成的，这允许：

[0022] ●规定性指导由资产的结构一致地驱动，不允许差异或不匹配：资产和指导是同步的。

[0023] ●资产提供者不必再向一侧的库提供资产而向另一侧的另一库提供资产指导（例如，文本文件），而是向“受指导资产 (guided asset)”数据库提供合二为一的材料（资产 + 指导）。

[0024] ●可变性点定义（包括可变性点依赖关系）反映了指导消费者消耗资产所需的决策树的布局。

[0025] - 指导对于所有规则而言一致地支持角色 (role-enabled)，这支持具有合作（包

括适当的移交）的端到端接合模型，这允许：

[0026] ●这使得存储部分填充的参数能与资产一起设置，反映当时做出的决策。

[0027] ●当他／她已做出与其角色和在消耗可定制资产的过程中相关的技能有关的决策时，资产消费者可停止。然后他／她可将产生的资产移交给团队中负责做出后续决策的下一消费者。此过程允许不同的角色合作解决原始资产中的可变性点。例如，业务分析师做出与业务过程有关的所有决策，然后他／她将修改的资产传递给能做出架构决策的软件架构师，等等。

[0028] ●在任何级别做出的决策也被自动记录，不需要在该侧有它们的记录。

[0029] -指导对于所有规则是而言一致地支持工具：

[0030] ●因为每个可变性点定义以标准格式（在我们的实现中的RAS）形式化地捕获，因此变量对用户的表示以及决策的过程可以通过工具来支持，不论当前决策是否处理需求、架构、业务建模、设计或代码。

[0031] ●使用相同的标准（在优选实施方式中，是RAS）来捕获资产内容和规定指导可以显著节约软件厂商的规模：单个工具框架可以在指导和资产消耗上支持专业人员。这使得能够根据一般性工具来消耗资产（与为每一资产类型开发专用的工具相反）。

[0032] -本发明生成多个参数路径，其仅受依赖关系的约束（一般而言，依赖关系产生森林而不是树）。本发明的树是动态的，并且通过消耗方法的每一消费者通路且根据消费者的专业目标被修改。

[0033] 可使用可复用资产规范v2.2(RAS)来实现本发明。重点注意，本发明独立于特定的标准或规范（如RAS），但是可用RAS实现。而且，这意味着可以通过支持（理解）RAS的任何工具或框架支撑，包括IBM Rational Software Architect(RSA)，RSA是用于软件架构师和创建基于组件(SOA，J2EE和portal)的应用的模型驱动开发者的模型驱动的开发和静态分析工具。

## 附图说明

[0034] 图1所示为根据本发明的优选实施方式的创建“受指导资产”的数据库的逻辑框图；

[0035] 图2所示为根据本发明的优选实施方式的消耗“受指导资产”的逻辑框图；

[0036] 图3所示为根据本发明的优选实施方式的创建“受指导资产”的方法的流程图；

[0037] 图4所示为根据本发明的优选实施方式，消耗“受指导资产”的方法的流程图；

[0038] 图5(5A、5B、5C)所示为根据本发明的优选实施方式的“受指导资产”的一个例子的RAS表示；

[0039] 图6所示为图5描述的“受指导资产”的RAS表示中实现的决策树的图形表示；

[0040] 图7所示为根据本发明的另一实施方式，消耗包含多个入口点 的“受指导资产”的方法的流程图。

## 具体实施方式

[0041] 在优选实施方式中，本发明用RAS v2.2实现。

[0042] 图1所示为根据本发明的优选实施方式的创建“受指导资产”的数据库的逻辑框

图。为了使用户开发基于资产的解决方案，必须创建包含受指导的可复用软件资产的数据库。通常，资产数据库优选地是关系型数据库（所谓的资产库），其提供到数据的简单通路。在下文中，对数据库或库的使用不做区别。在库中，资产被用结构化语言（例如，RAS）描述为对象，其包括具有属性并链接到文件产品自身（它是模型、代码）的元件。图 1 描述了创建“受指导资产”的主要步骤，包括库中资产的修改的描述。“受指导资产”的描述包含关于怎样消耗资产的信息，该信息将为资产消费者提供逐步的指导。

[0043] 在优选实施方式中，创建“受指导资产”数据库的方法是用于消耗计算机 (160) 上执行的资产的软件应用 (120) 中的初始操作 (130,140)。这个操作 (130,140) 可由资产提供者本身执行或者由开发解决方案并向其 IT 专业人员提供受指导的软件资产的公司来执行。资产本身（模型、代码）位于资产库 (100) 中，并用作输入。创建“受指导资产”的流程图稍后将在本文中结合图 3 的描述进行说明。总体上，对于每个从资产库中读出的资产，根据资产消耗信息 (110) 检索 (130) 决策树，然后将决策树编码并嵌入 (140) 到资产中。产生的“受指导资产”存储在“受指导资产”库中 (150)。

[0044] 在优选实施方式中，资产消耗信息作为文本文件（包含编辑的文本、HTML、XML、SGML 等）存储在“资产消耗指导信息”数据库中。这些文本文件不允许通过自动执行程序来从数据中自动提取决策树。这就是创建决策树 (130) 通常由有资格的操作者 (138) 通过用户界面（优选地为图形用户界面 (135)）来执行的原因。如果软件资产消耗指导信息以标准化的方式存储，则不再需要这一人 为干预。注意，在将来，资产提供者不再保持资产库和资产信息数据库，而是将仅仅处理一个被提供的“受指导资产”库。

[0045] 注意，导致创建“受指导资产数据库”的所有提及的程序 (130,135 和 140) 可在不同的计算机上执行。

[0046] 图 2 所示为根据本发明的优选实施方式的消耗“受指导资产”的逻辑框图。一旦构建了受指导资产库，则为了创建基于资产的解决方案，IT 专业人员（资产消费者）将按所指导的方式、通过执行用于消耗资产 (120) 的程序的部分 (230,240) 来定制软件资产，其可在相同的计算机 (160) 上或在不同计算机上执行。资产消费者将优选地通过图形用户界面 (200) 来访问受指导资产数据库 (150) 以创建定制的受指导资产 (220)。该程序将针对每一资产向用户显示 (230) 决策树。然后该程序将在决策树的每一决策点捕获 (240) 用户的输入，并将相应地定制受指导资产。注意，如本领域技术人员公知的，当资产被消耗时，最终解决方案（更新的模型或用于进一步使用的代码）的构建完成，完成解决方案的这个步骤不在本发明的范围之内。

[0047] 图 3 所示为根据本发明的优选实施方式的创建“受指导资产”的方法的流程图。在计算机 (160) 上执行用于创建受指导资产的程序 (130,140)，此后可以通过相同计算机 (160) 或不同计算机上执行用于消耗资产的程序 (230,240) 来消耗该资产。

[0048] 从资产库数据库 (100) 读取 (300) 第一资产。在优选实施方式中，资产以标准化的方式（例如使用 RAS 2.2）存储在数据库中。如果找到了资产（对测试 310 回答“是”），则从资产消耗指导信息数据库 (110) 中提取对应的决策树 (320)。

[0049] 标识 (330) 根决策点。它对应于对决策树分支的第一可能选择。下一步骤包括通过使用先前标识的决策点中可用的信息以及先前从资产消耗指导信息数据库 (110) 中提取的树 (320) 中所包含的信息，在资产中创建可变性点 (VP)，以及设置该 VP 的属性 (340)。

此信息包括：

[0050] ●决策点的名称：名称需要在资产中是唯一的，使得它唯一标识可变性点（决策点）；

[0051] ●描述：这是针对人（用户）的文本信息，描述在这个点做出的决策；

[0052] ●一组变量：决策点包括一组至少一个可能的选择或备选（在下文中称为变量），其已与资产一起封装。而且，如果资产消费者没有做改变，这些变量之一可以是被缺省使用的；

[0053] ●变量的类型：对一个决策点，仅某种类型的变量是可接受的。这些变量是可替换的（例如，模型、代码）；

[0054] ●标识变量间的共存规则的参数（例如变量逻辑：包含的、互斥的、可选的）；

[0055] ●工具：为了支持自动化和上下文，每一可变性（决策）点需要列出可处理它的软件工具（可用于做决策的工具）。图 5 示出可变性点的详细例子；

[0056] ●标识决策点在决策树中位置的参数：例如，其可以是树中祖先节点的名称。

[0057] 重复（对测试 350 回答 Yes）同一步骤 (340)，直到考察完树的所有决策点，即直到在决策树中再没有孩子（对测试 350 回答 No）。然后受指导资产被存储在受指导资产库 (150) 中，并且从资产库读取 (300) 下一资产。

[0058] 注意，只有当这些数据以标准化方式存储在资产消耗指导信息数据库 (110) 中时，被提取的决策树 (320) 和用于填充 VP(340) 属性的信息可以由用于消耗资产的程序 (120) 自动执行。如上所述，结合图 1 的说明，此信息目前通常不是标准化的，且操作者 (138) 可执行初始步骤：通过提供用户接口（优选地，图形用户界面 (135)）的程序、在资产消耗指导信息数据库 (110) 中、以使程序 (120) 可消耗的形式来创建信息。这个步骤也可在创建受指导资产的阶段期间同时执行。

[0059] 图 4 所示为根据本发明的优选实施方式的消耗“受指导资产”的方法的流程图。用于消耗资产的程序 (120) 的该部分在计算机 (160) 上执行，该计算机可以与用于创建受指导资产的计算机相同或不同。资产消费者（IT 专业人员）优选地通过图形用户界面（通常是向导）与该程序部分进行交互。用户知道哪个资产需要被包括在他正在开发的基于资产的解决方案中。将指导用户根据他正在开发的解决方案的细节来定制资产。

[0060] 第一步骤包括：通过图形用户界面显示用户可从中选择 (400) 资产的屏幕。当资产被选择时（测试 410 的结果为“是”），对应的决策树的根的可变性点被程序 (420) 定位。对应于根的 VP 描述被显示 (430)，并且用户通过修改 VP 的内容来提供输入，以便将修改引入根 VP，从而反映其定制选择。计算机按照用户的建议更新 (440)VP 描述。如果资产的下一决策点（测试 450 的结果为“是”）被用户选择（测试 460 的结果为“是”），则对其进行显示和更新 (430,440)，直到决策树被遍历，即访问了每一节点（测试 450 的结果为“否”）。注意，可以通过依赖于同一决策点而访问所有决策点或者通过相继地访问每一分支来遍历决策树，所有的选项都是可能的，即，可使用所有树遍历策略和算法，只要其符合决策点间的依赖关系关系。这是通过在一轮处理中执行方法来定制资产的第一选项。稍后在本文中参考图 7 详细说明的另一可能性是在多于一轮处理中定制一个资产。在这种情况下，当用户决定停止时（对测试 460 回答 No），第一轮可结束。这意味着定制的结果是部分消耗的资产，图 4 表示消耗受指导资产的第一轮处理。

[0061] 指导资产消费者输入其消耗资产的选择 (430),因为 :

[0062] - 决策点根据树的依赖关系而相继地显示 ;

[0063] - 在每一决策点,计算机显示 VP 和所有 VP 变量的描述。

[0064] 在步骤 440,通过输入消费者的选择来更新受指导资产。消费者的选择被用于映射该决策点的 VP。这记录了消费者做出的选择。消费者可改变 VP 描述中的任何内容 :根据变量间的共存规则来抑制或添加变量,修改变量的属性等。下面在本文中参考图 5 说明描述资 产的结构化语言。

[0065] 图 5 所示为根据本发明的优选实施方式的“受指导资产”的 RAS 表示,其全面地描述了决策点。请注意,这些细节只是为了说明本发明是怎样工作而给出的,而不是应当如何以任何结构化语言为基础 ( 这里是 RAS v2.2) 来实现本发明的限制性定义。此外,在优选实施方式中,没有提出对 RAS 2.2 的扩展,而是使用相同的元素和属性,但是其中的某些是在特定的语义中使用的。在此示例中,考虑包含支持服务配置和激活 (Service Configuration and Activation) 的一组应用组件的源代码实现的软件资产,如电信行业中所定义的,例如在 eTOM(对电信行业中的业务过程进行标准化的增强型电信操作图) 中。

[0066] 更精确地,该资产已按如下方式设计,即提出具有多个可能备选方案的服务构造实现 ( 在服务订单管理系统中),并且利用主要性能指标兼容性检查 ( 在服务激活管理系统中 ) 也具有可替换的实现。在电信行业中,业务过程是很稳定的,但是其步骤的实现将会根据使用的是内部能力还是外部能力而有所不同。在我们的例子中,如果资产提供者以这种方式设计它,如果资产消费者选择服务构造的特定备选方案,则他可以在几个外部订单创建实现 ( 在服务设计系统中 ) 之间进行选择。

[0067] 图 5 说明了受指导资产的基于 RAS 的简化定义,资产的例子如上所述。为了清楚而简化这个定义。这个定义符合 RAS 2.2,如果不想遵循 RAS 2.2,则可以使用描述资产的任何其他语法选项。这个受指导资产定义通过执行图 1 和图 3 所述的方法而被创建。RAS 是使用元素 (<asset>,<classification>,<VariabilityPoint>,<artifact> 等 ) 来描述对象的结构化语言。每一元件具有属性 (id,描述,引用等 )。

[0068] 描述是 XML 文本,其在图 5A 中具有第一个 <asset> 标签而在图 5C 中具有最后的 </asset> 标签。各种产品之间,标签描述如较早结合图 1 和图 3 而说明的资产的各个可变性点及其变量。这些可变性点被映射到图 6 所示的资产的对应决策树的决策点。决策树的决策点 对应于当执行图 2 和图 4 所述的方法时资产消费者将必须选择的资产定制的不同选项。

[0069] 在图 5B 中,在两个产品上描述两个可变性点,在产品“TelcoService Configuration and Activation component( 电信服务配置和激活组件 )”中描述用于“Asset consumption( 资产消耗 )”的“root( 根 )”可变性点 (VP),以及在产品“Service Construction( 服务构造 )”中描述用于“Service Construction implementation( 服务构造实现 )”的“VP1”可变性点。

[0070] 在图 5B 中,根 VP 是如图 3 所示的资产定制的通常入口,并且不存在定制的可能备选方案,没有“变量”。属性“reference( 引用 )”描述根决策点在决策树中的位置 ( 它是空,因为根没有祖先 )。更一般地,这个“reference”属性将限定 VP 在决策树中的依赖关系。VP 的描述、变量的类型以及指导变量间共存的规则 ( 如果有的话 ) 由属性“context-id( 上

下文 id)”提供。根 VP 的“context-id”的值是“ctxt1”。由于与 RAS 2.2 的兼容性,所有 VP 的“context-id”的值列出在图 5A 中在全局 <classification> 标签的 <context> 标签下。对于“ctx1”,图 5A 中的描述仅包含名称和描述,因为没有变量。

[0071] 图 5B 示出,在 vp1 决策点处,资产消费者将具有定制的选择。vp 1VP 具有分别在产品“Service Construction implementation option 1(服务构造实现选项 1)”、“Service Construction implementationoption 2(服务构造实现选项 2)”和“Service Constructionimplementation option 3(服务构造实现选项 3)”中描述的 3 个变量。vp1VP 的属性“reference”是根 (reference =“root”),因为 vp1 在决策树中依赖于根。vp1 VP 的“context-id”的值是“ctx2”。对于“ctx2”,图 5A 中的描述包含名称、描述、变量类型和共存规则 (“互斥的”)。

[0072] 在图 5C 中,在 2 两个产品上描述两个可变性点,在产品“ServiceDesign System(服务设计系统)”中描述用于“External Order Creationimplementation(外部订单创建实现)”的“vp2”可变性点 (VP), 以及在产品“KPI Compliance Check(KPI 兼容性检查)”中描述用于“KPI Compliance Check implementation(KPI 兼容性检查实现)”的“VP3”可变性点。

[0073] 在图 5C 中,“vp2”VP 具有分别在产品“External Order Creationimplementation option 1(外部订单创建实现选项 1)”、“ExternalOrder Creation implementation option 2(外部订单创建实现选项 2)”中描述的 2 个变量。vp2VP 的属性“reference”是 vp1(reference =“vp1”决策树中的父节点)。vp2 VP 的“context-id”的值是“ctx3”。对于“ctx3”,图 5A 中的描述包含名称、描述、变量类型和共存规则 (“exclusive(互斥)”)。

[0074] 在图 5C 中,vp3VP 具有分别在产品“KPI Compliance Checkimplementation option 1(KPI 兼容性检查实现选项 1)”、“KPICompliance Check implementation option 2(KPI 兼容性检查实现选项 2)”中描述的 2 个变量。vp2VP 的属性“reference”是 vp1(reference =“root”决策树中的父节点)。vp2VP 的“context-id”的值是“ctx4”。对于“ctx4”,图 5A 中的描述包含名称、描述、变量类型和共存规则 (“exclusive(互斥)”)。

[0075] 如上所述,通过指向同一上下文,可变性点及其变量链接在一起,以及上下文定义包括细节,如组合或不组合变量 (在我们的例子中是互斥的),以及可接受的变量类型 (J2EE 企业档案文件,Java 档案文件,Java 文件)。

[0076] 可变性点“Service Construction implementation(服务构造实现)”和“KPI Compliance Check implementation(KPI 兼容性检查实现)”引用“Asset Consumption(资产消耗)”可变性点,它代表决策树的根。这反映了决策树的入口点。

[0077] 可变性点“External Order Creation implementation(外部订单创建实现)”引用“Service Construction implementation(服务构造实现)”可变性点。这反映了决策树中结构的依赖关系。

[0078] 这个优选实施方式使用 RAS v2.2,仅存在一个小例外。在 RAS 中,可变性点上的引用属性可用于提供更多的背景和说明 ;我们改变了 <variability-point>XML 标签的“reference”属性的语义,以捕获可变性点之间的祖先关系,反映决策树的结构,即必须考虑的决策点的顺序。

[0079] 图 6 所示为图 5 的 RAS 表示所示的受指导资产的决策树所隐含的依赖关系树的图

形表示 (600)。决策点是根、vp1、vp2、vp3。表示了以下的依赖关系 :vp2 依赖于 vp1 ;vp1 和 vp3 依赖于根。在优选实施方式中,依赖关系对应于受指导资产中 VP 的“reference”属性。对于每一决策点,依赖关系对决策树可能的分支编码。

[0080] 如果与一个资产相关的消耗指导信息 (110) 包含设计模型和实现模型,则在资产中有 2 个固有的逻辑决策树 :设计树和实现树。在受指导资产中,仅建立了一个决策树,即,每个资产有一个决策树。当资产中存在多个固有的逻辑树时,我们在开始处添加一个决策点 ( 可变性点 ),称为根。然后,根可变性点简单地询问用户他想走哪个固有逻辑树 ( 在我们的例子中,是他想消耗设计模型还是实现模型 )。这保证了总是有一个且仅一个决策树指导消耗受指导资产。用于消耗资产的方法然后在参数中生成多个路径,仅受依赖关系的约束。

[0081] 在图 6 中,树 610、620 和 630 示出消耗图 5 的受指导资产的 3 种方式。在 610 中,消费者选择服务构造实现 vp1 决策点的选项 1,它与其他变量 ( 选项 2 和选项 3) 是互斥的 ( 根据“ctx2”上下文的描述 )。然后,消费者在外部订单创建实现 vp2 决策点选择选项 2。

[0082] 对于消耗的资产 620,对受指导资产没有改变,因为用户已决定不消耗它。在 630 中,资产消费者在决策树的相继的决策点选择了不同的变量。

[0083] 用于消耗受指导资产的方法的扩展允许多于一轮处理,这在图 7 的流程图中描述。假设已执行消耗资产的第一轮处理 ( 图 4),且还未完全遍历受指导资产的决策树,即该树的所有决策点未被处理 (430,440) :未被处理的决策树的决策点的对应可变性点未被更新 (440)。对于第一轮处理之后的处理,消费者可重新进入决策树考察通过一个“入口点”,它可以是在以前通路中已处理过 (430,440) 的决策点或者是最近已访问的决策点之后 ( 在相关树中 ) 的决策点。在这个更一般的图 7 的流程图中,建议消费者从决策树点所有入口点中选择一个入口点,包括决策树的根,它是图 4 的流程图中第一轮处理的唯一入口点。

[0084] 一旦计算机读取了用户选择的受指导资产 (400)( 测试 410 的结果为“是”),计算机定位资产的入口可变性点 (700) :如果不是所有决策点在以前的通过中已被处理 ( 测试 710 的结果为“是”),则要求消费者选择其中之一,并且如果选择了一个 ( 测试 720 的结果为“是”),则处理决策点,即要求用户输入其选择 (430),且计算机相应地更新资产中对应的可变性点 (440)。通过抑制刚处理的决策点的可变性点以及通过在资产的相关树中添加后续决策点,即决策树的孩子 (730),添加用于维护入口点列表的新步骤。这个列表被计算机读出,用于执行定位所有入口点的后续步骤 (700)。

[0085] 用于消耗受指导资产的程序可以结束,因为消费者不选择消耗资产 ( 测试 410 的结果为“否”)。如果计算机没发现任何入口点,因为不再有其余入口点 ( 测试 710 的结果为“否”),或者如果消费者不选择处理其余入口点 ( 测试 720 的结果为“否”),则所选资产的处理完成且计算机存储 (740) 资产,因为它已被处理,即资产被消费者完全消耗 ( 测试 710 的结果为“否”) 或部分消耗 ( 测试 720 的结果为“否”)。

[0086] 图 7 的流程图说明的过程的变化是 :当消费者仅消耗了受指导资产的一部分时 ( 测试 720 的结果为“否”),部分消耗的受指导资产被存储 ( 测试 720 的结果为“否”) 为新的受指导资产。其原因取决于资产怎样被消耗。通过选择对应 VP 的变量,消费者可处理 (430,440) 决策点。但是,在决策点,消费者可增加 (430,440) 对应的 VP,例如,通过添加新变量以及通过添加如图 5B 的受指导资产描述所示的 <reference value = filename.jar>

来添加对应的代码。生 成的已修改受指导资产可能对于其他消费者而言是有用的，以及在这种情况下，将增加的受指导资产的结果存储为新的“受指导资产”可能是有利的。增加的受指导资产的下一消费者将通过处理尚未处理的决策点来定制它。根据消费者的专业能力和偏好以及依据使用的结构化语言 (RAS)，可以有其他方式修改 VP(430)。

[0087] 决策点处理和对应的可变性点修改可以发生在资产创建时（缺省值），或者由其他消费者在资产消耗时。例如，架构师可能已对资产做出架构决策当设计者得到资产且必须做设计决策时。

[0088] 一般地，部分消耗的资产被存储 (740) 用于被同一用户、或者执行与资产消耗不同角色的其他用户（架构师、设计者、IT 专业人员、安全专家等）进一步消耗。

[0089] 图 7 所示为本发明的解决方案的最大优点 :资产消费者可以在他 / 她做出了与他 / 她在消耗可定制资产的过程中的角色和相关技能的决策时停止。然后他 / 她可将产生的资产移交给团队中负责做出后续决策的下一消费者。如上文所述，这个方法允许不同的专业人员合作解决原始资产中的可变性点，如业务分析师、软件架构师，作为设计者和实现安全的不同设计专家。

[0090] 注意，即使在给定资产中只有一个树被编码，向用户提供多个入口点非常类似于提供多个树（子树）指导资产的消耗。

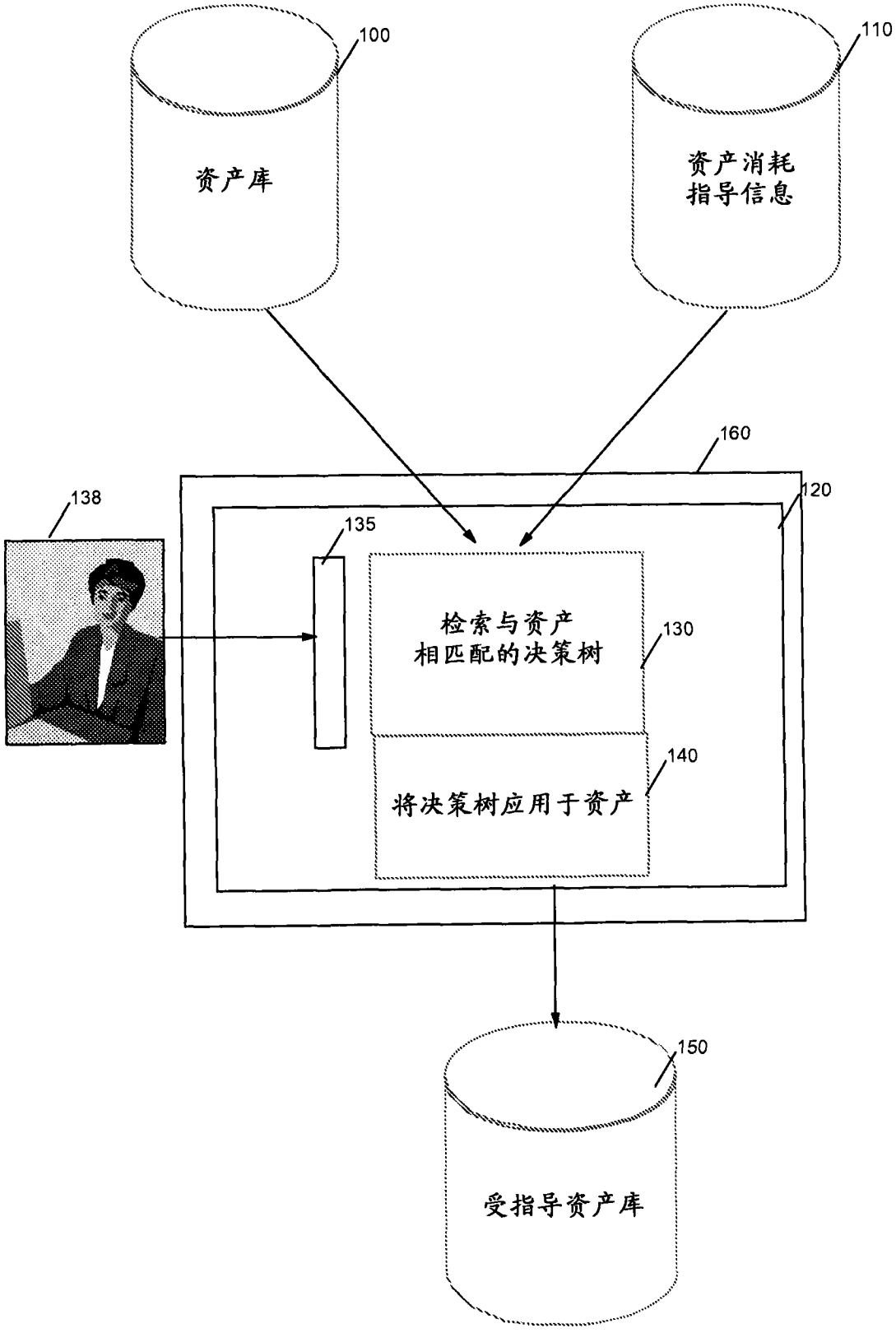

100
资产库
110
资产消耗
指导信息
160
120
138
135
检索与资产
相匹配的决策树
130
140
将决策树应用于资产
150
受指导资产库

图 1

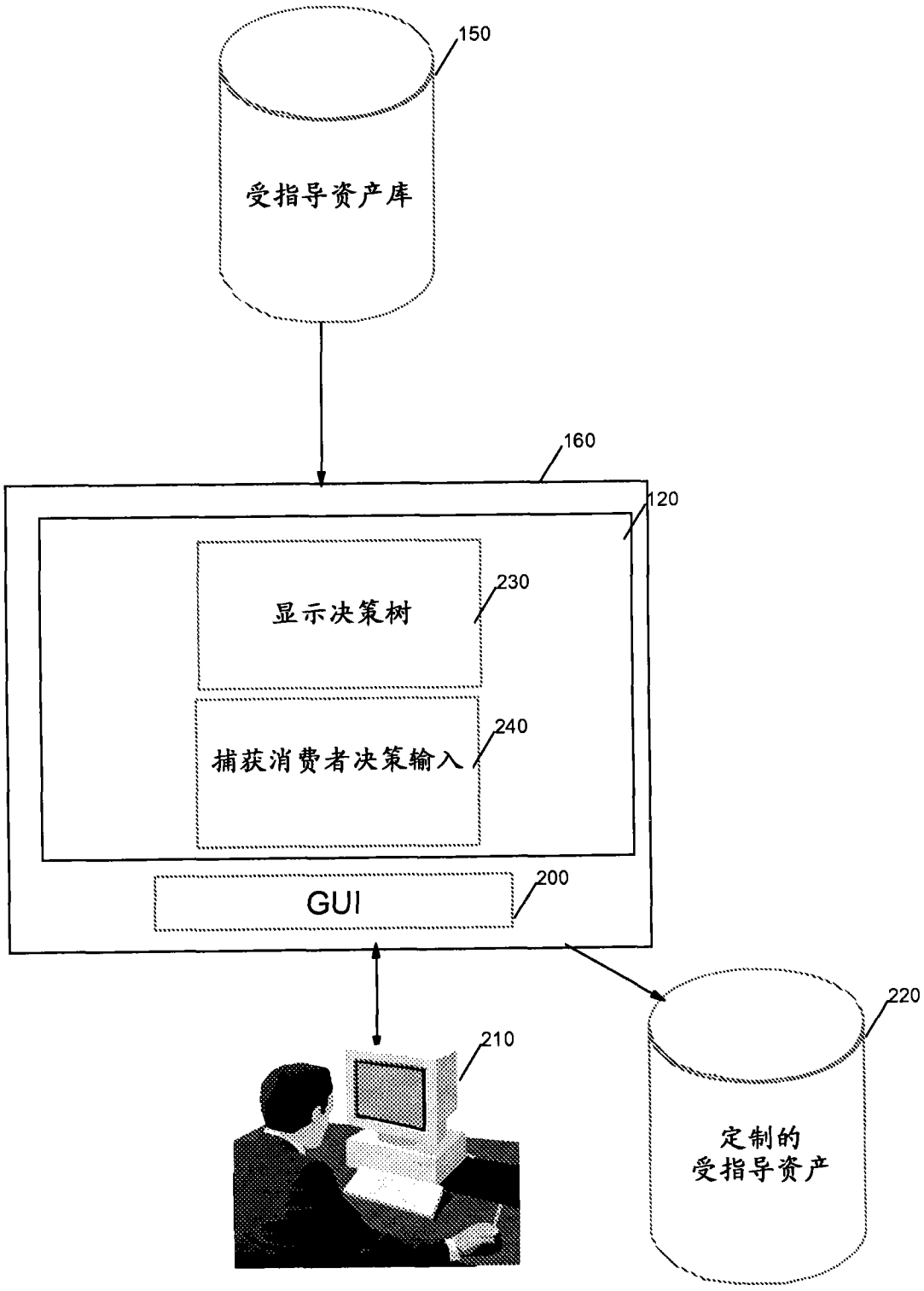

150
受指导资产库
160
120
显示决策树
230
捕获消费者决策输入
240
GUI
200
210
220
定制的
受指导资产

图 2

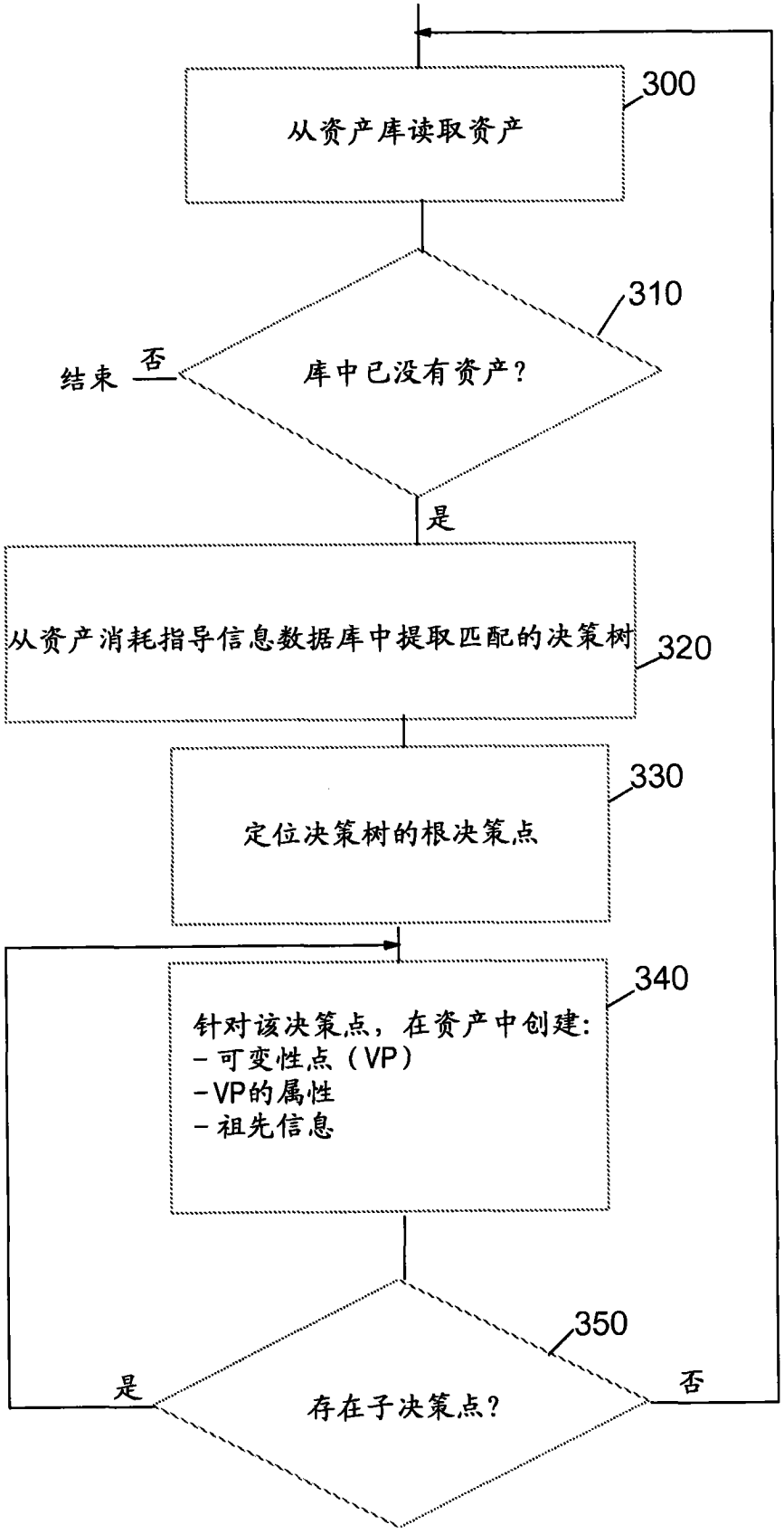

从资产库读取资产
300
库中已没有资产？
310
结束
否
是
从资产消耗指导信息数据库中提取匹配的决策树
320
定位决策树的根决策点
330
针对该决策点，在资产中创建:
- 可变性点（VP）
- VP的属性
- 祖先信息
340
存在子决策点？
350
是
否

图 3

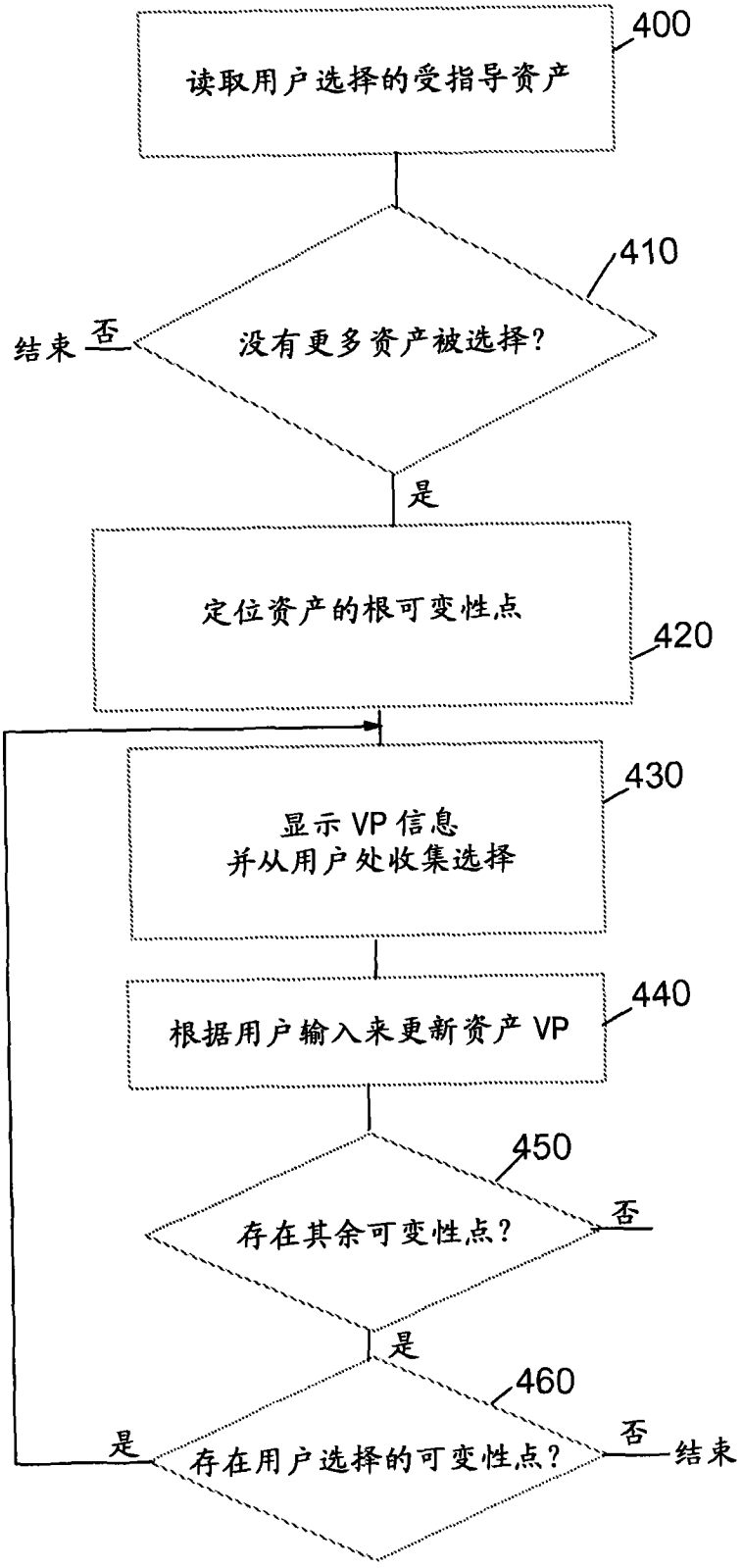

读取用户选择的受指导资产
400
没有更多资产被选择?
410
否
结束
是
定位资产的根可变性点
420
显示 VP 信息
并从用户处收集选择
430
根据用户输入来更新资产 VP
440
存在其余可变性点?
450
否
是
存在用户选择的可变性点?
460
否
结束
是

图 4

```
<asset>
 <classification>
  <context name="Generic Asset Consumption " id="ctxt1">
   <description value="Default root decision point for all assets being consumed."/>
  </context>
  <context name="Service Construction variants" id="ctxt2">
   <description value="Shipped with asset"/>
   <descriptorGroup name=" Service Construction variant details">
     <nodeDescriptor href="https://com.ibm.../.../classif/vp_types.xmi#EAR"/>
     <nodeDescriptor href="https://com.ibm.../.../classif/variant_logic.xmi#exclusive"/>
   </descriptorGroup>
  </context>
  <context name="External Order Creation implementation variants" id="ctxt3">
   <descriptorGroup name="External Order Creation variant details">
     <nodeDescriptor href="https://com.ibm.../.../classif/vp_types.xmi#JAR"/>
     <nodeDescriptor href="https://com.ibm.../.../classif/variant_logic.xmi#exclusive"/>
   </descriptorGroup>
  </context>
  <context name="KPI Compliance Check implementation variants" id="ctxt4">
   <descriptorGroup name="KPI Compliance Check variant details">
     <nodeDescriptor href="https://com.ibm.../.../classif/vp_types.xmi#Java"/>
     <nodeDescriptor href="https://com.ibm.../.../classif/variant_logic.xmi#exclusive"/>
   </descriptorGroup>
  </context>
 </classification>
...
```

图 5A

```
...
  <solution>
    <artifact name="Telco Service Configuration and Activation
component">
      <variabilityPoint id="root" name="Asset Consumption"
description="Are you sure you want to include this asset within your
target solution?" reference="" context-id="ctxt1"/>
      <artifact name="Service Order Management System">
        <artifact name="Service Construction">
          <variabilityPoint id="vp1" name="Service Construction
implementation" reference="root" context-id="ctxt2"/>
          <artifact>
            <reference value="ServiceConstructionEAR.ear"/>
          </artifact>
        </artifact>
        <artifact name="Service Construction implementation option 1">
          <artifact-context context-id="ctxt2">
          <reference value="ServiceConstructionEAR1.ear"/>
        </artifact>
        <artifact name="Service Construction implementation option 2">
          <artifact-context context-id="ctxt2">
          <reference value="ServiceConstructionEAR2.ear"/>
        </artifact>
        <artifact name="Service Construction implementation option 3">
          <artifact-context context-id="ctxt2">
          <reference value="ServiceConstructionEAR3.ear"/>
        </artifact>
      </artifact>
...
```

图 5B

```
...
    <artifact name="Service Design System">
      <artifact>
        <artifact>
          <reference value="ServiceDesignSystemEAR.ear"/>
          <variabilityPoint id="vp2" name="External Order Creation
implementation" reference="vp1" context-id="ctxt3">
          <artifact>
            <reference value="ExternalOrderCreation.jar "/>
          </artifact>
        </artifact>
        <artifact name="External Order Creation implementation option 1">
          <artifact-context context-id="ctxt3">
          <reference value="ExternalOrderCreation1.jar"/>
        </artifact>
        <artifact name="External Order Creation implementation option 2"> ... [
details omitted here ] ... </artifact>
      </artifact>
    </artifact>
    <artifact name="Service Activation Management System">
      <artifact name="KPI Compliance Check">
        <variabilityPoint id="vp3" name="KPI Compliance Check implementation"
reference="root" context-id="ctxt4"/>
        <artifact>
          <reference
value="com.ibm.eis.refimpl.telco.smof.KPIComplianceCheck.java"/>
        </artifact>
      </artifact>
        <artifact name=" KPI Compliance Check option 1"> ... [ details omitted
here ] ... </artifact>
        <artifact name=" KPI Compliance Check option 2"> ... [ details omitted
here ] ... </artifact>
    </artifact>
  </artifact>
 </solution>
</asset>
```

图 5C

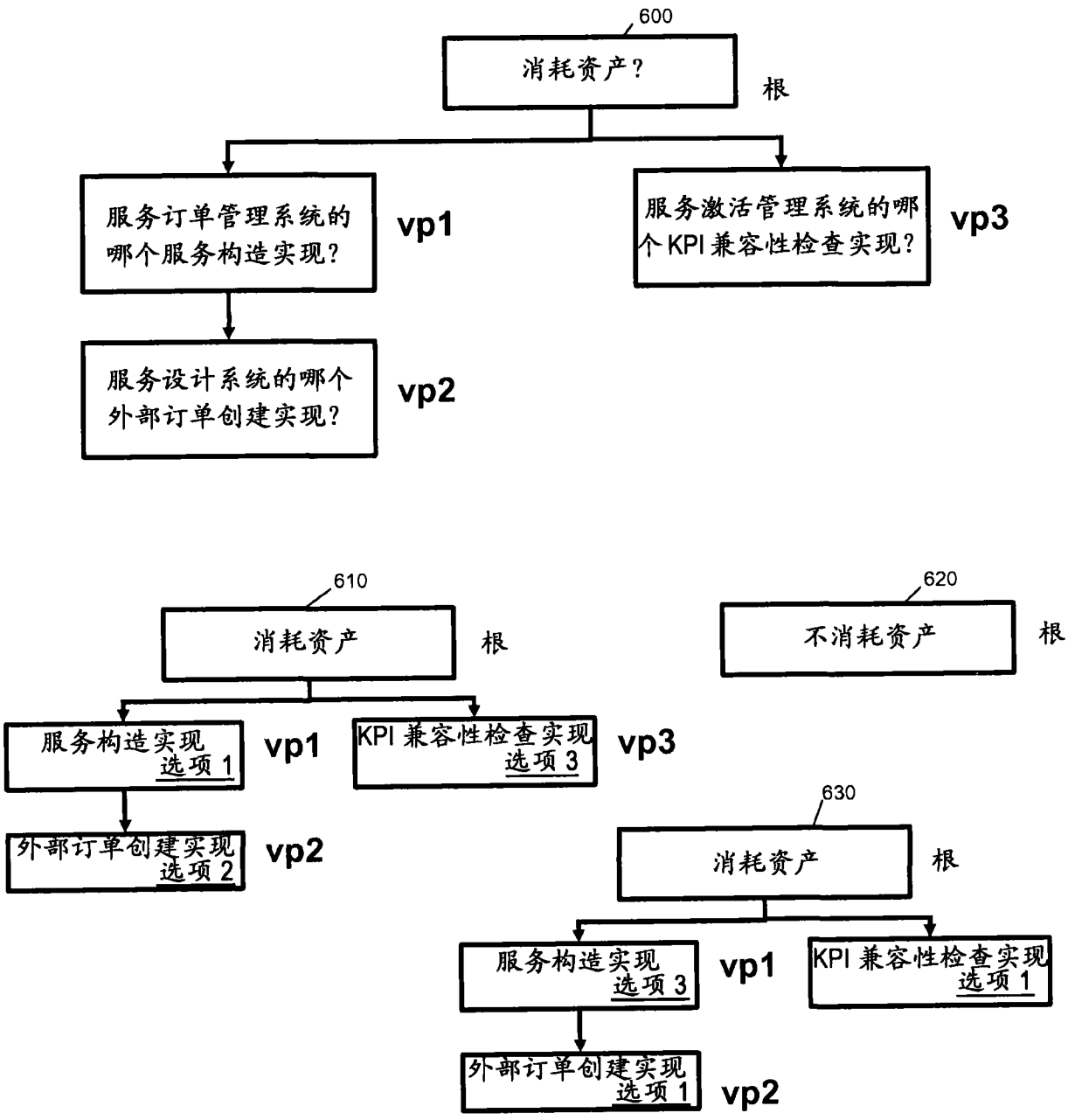

600
消耗资产？
根
服务订单管理系统的哪个服务构造实现？
vp1
服务激活管理系统的哪个KPI兼容性检查实现？
vp3
服务设计系统的哪个外部订单创建实现？
vp2
610
消耗资产
根
服务构造实现
选项 1
vp1
KPI 兼容性检查实现
选项 3
vp3
外部订单创建实现
选项 2
vp2
620
不消耗资产
根
630
消耗资产
根
服务构造实现
选项 3
vp1
KPI 兼容性检查实现
选项 1
外部订单创建实现
选项 1
vp2

图 6

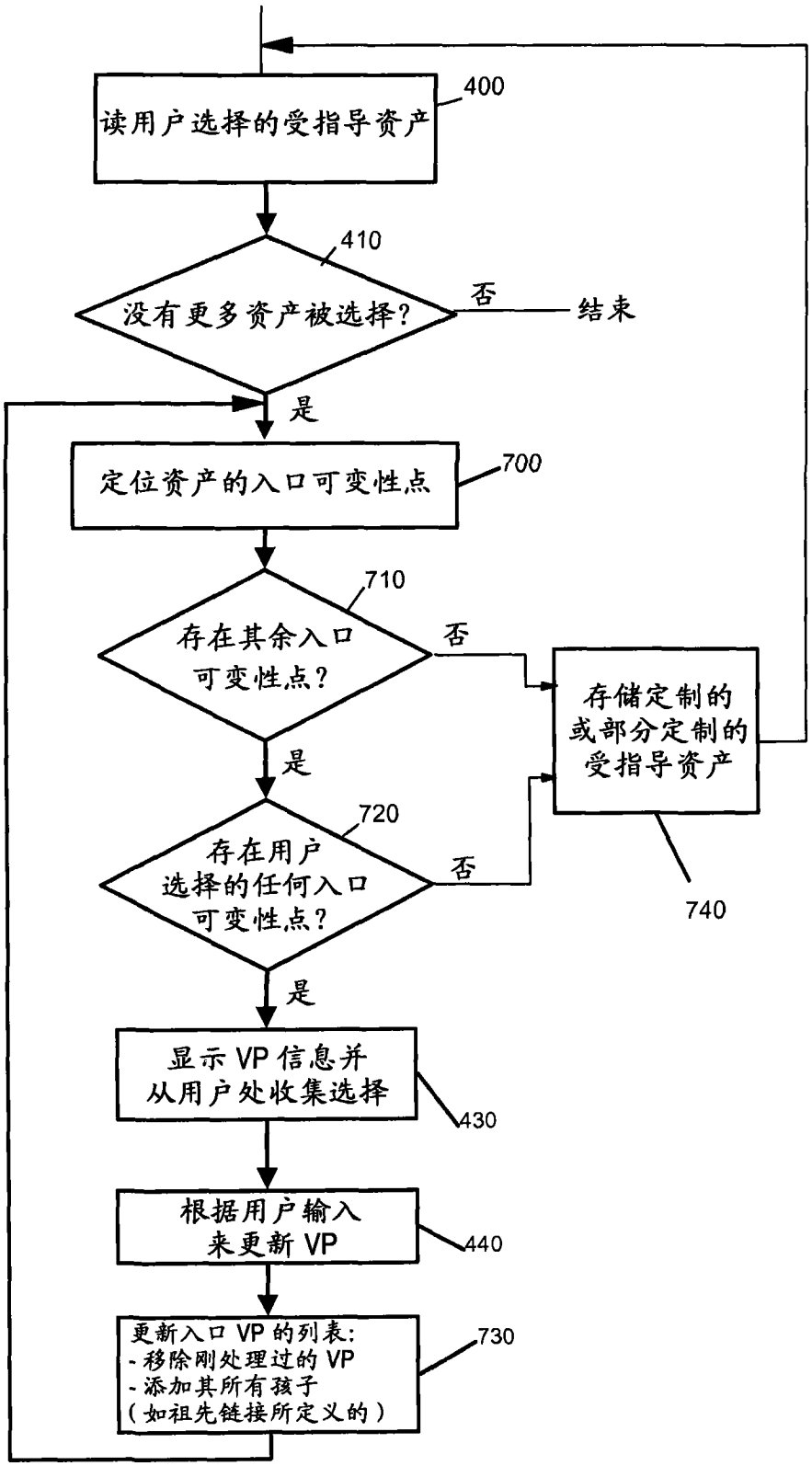

读用户选择的受指导资产
400
410
没有更多资产被选择?
否
结束
是
定位资产的入口可变性点
700
710
存在其余入口可变性点?
否
是
720
存在用户选择的任何入口可变性点?
否
是
存储定制的或部分定制的受指导资产
740
显示 VP 信息并从用户处收集选择
430
根据用户输入来更新 VP
440
更新入口 VP 的列表:
- 移除刚处理过的 VP
- 添加其所有孩子
(如祖先链接所定义的)
730

图 7