

[12] 发明专利申请公开说明书

[21] 申请号 99123169.4

[43]公开日 2000年5月17日

[11]公开号 CN 1253336A

[22]申请日 1999.10.27 [21]申请号 99123169.4

[30]优先权

[32]1998.10.30 [33]US [31]09/182,637

[71]申请人 朗迅科技公司

地址 美国新泽西州

[72]发明人 菲利浦·威廉姆·吉里斯

[74]专利代理机构 中国国际贸易促进委员会专利商标事务所

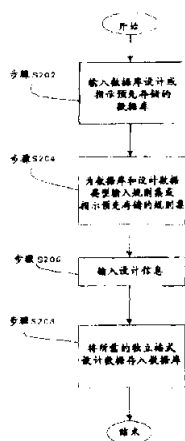
代理人 鄢 迅

权利要求书 4 页 说明书 10 页 附图页数 5 页

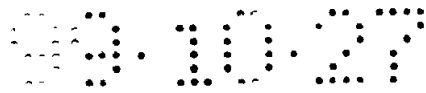
[54]发明名称 将数据作为流动信息存储的方法和装置

[57]摘要

一种设计信息系统,以方便可访问的平台独立形式接收设计信息并将其存储在数据库中。设计信息系统能以若干不同格式中的任何一种格式接收信息并将该信息转换为某种合适的格式以便存储。随后可由某个程序或应用从该数据库中抽取该设计信息以便构造某个所需的对象。存储设计信息以便在多平台之间方便转换,并继续用于新的开发平台和环境。

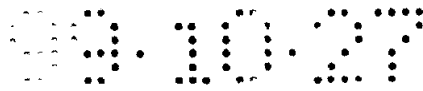


ISSN 1008-4274



权 利 要 求 书

1. 一种存储独立格式设计信息的方法，包括步骤：
 - (a) 提供具有为某类设计信息构造的数据库格式的一个数据库；
 - (b) 提供为该数据库和某个对象类型配置的规则集；
 - (c) 输入该对象类型的一个对象，其中，对象具有设计信息；并且
 - (d) 根据该规则集，将设计信息存储在数据库中。
2. 权利要求1的方法，其中
步骤(c) 输入具有设计信息的对象，该信息包括专用格式信息和独立格式信息，并且
步骤(d) 根据该规则集将独立格式信息存入数据库。
3. 权利要求2的方法，其中步骤(d) 不存储专用格式的信息。
4. 权利要求1的方法，其中设计信息的类型包括消息集、有限状态机和基于规则的集合中的一种类型。
5. 权利要求1的方法，其中，对象是图形设计图、文档、伪码和程序代码之一。
6. 权利要求1的方法，其中步骤(a) 访问一个预先存储的数据库。
7. 权利要求1的方法，其中步骤(a) 包括接收说明该数据库结构的用户输入的步骤。
8. 权利要求1的方法，其中步骤(b) 访问预先存储的规则集。
9. 权利要求1的方法，其中步骤(b) 包括接收说明该规则集的用户输入的步骤。
10. 一种设计信息系统，包括：
 - 一个存储器设备；
 - 至少一个输入设备；以及
 - 一个处理器，访问具有为某类设计信息构造的数据库格式的一个数据库，访问为该数据库和某个对象类型配置的一个规则集，接收该对象类型的一个对象，其中该对象具有设计信息，并且根据存储器中



的该规则集将设计信息存储在数据库中。

11. 权利要求10的设计信息系统，其中处理器接收具有设计信息的对象，该设计信息包括专用格式信息和独立格式信息，并且根据存储器设备中的规则集将独立格式信息存入数据库。

12. 权利要求11的设计信息系统，其中，处理器不存储专用格式信息。

13. 权利要求11的设计信息系统，其中，设计信息的类型包括消息集、有限状态机和基于规则的集合中的一种类型。

14. 权利要求10的设计信息系统，其中，对象是图形设计图、文档、伪码和程序代码之一。

15. 权利要求10的设计信息系统，其中，处理器访问某个预先存储的数据库。

16. 权利要求10的设计信息系统，进一步包括：
至少一个第一输入设备和第二输入设备；并且其中
处理器通过第一输入设备输入对象，并且根据由第二输入设备接收到的用户输入，在存储器设备中建立数据库。

17. 权利要求10的设计信息系统，其中，处理器访问某个预先存储的规则集。

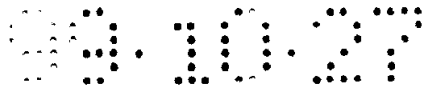
18. 权利要求10的设计信息系统，进一步包括：
至少一个第一输入设备和第二输入设备；并且其中
处理器通过第一输入设备输入对象，并且根据由第二输入设备接收到的用户输入建立规则集。

19. 一种包括在用于存储设计信息的计算机可读介质中的计算机程序，包括：

第一访问源代码段，访问具有为某类设计信息构造的数据库格式的一个数据库；

第二访问源代码段，访问为该数据库和某个对象类型配置的规则集；

输入源代码段，输入该对象类型的一个对象，其中对象具有设计



信息；以及

存储源代码段，根据该规则集将设计信息存入数据库。

20. 权利要求19的计算机程序，其中

输入源代码段输入具有设计信息的对象，设计信息包含专用格式信息和独立格式信息；并且

存储源代码段根据规则集将独立格式信息存储在数据库中。

21. 权利要求20的计算机程序，其中存储源代码段不存储专用格式信息。

22. 权利要求19的计算机程序，其中，设计信息的类型包括消息集、有限状态机和基于规则的集合中的一种类型。

23. 权利要求19的计算机程序，其中，对象是图形设计图、文档、伪码和程序代码之一。

24. 权利要求19的计算机程序，其中，第一访问源代码访问某个预先存储的数据库。

25. 权利要求19的计算机程序，其中，第二访问源代码访问某个预先存储的规则集。

26. 一个计算机数据信号，包括：

第一访问信号段，访问具有为某类设计信息构造的数据库格式的一个数据库；

第二访问信号段，访问为该数据库和某个对象类型配置的规则集；

输入信号段，输入该对象类型的一个对象，其中对象具有设计信息；以及

存储信号段，根据该规则集将设计信息存入数据库。

27. 权利要求26的计算机数据信号，其中

输入信号段输入具有设计信息的对象，设计信息包含专用格式信息和独立格式信息；并且

存储信号段根据规则集将独立格式信息存储在数据库中。

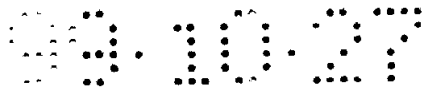
28. 权利要求27的计算机数据信号，其中存储信号段不存储专用格式信息。

29. 权利要求26的计算机数据信号，其中，设计信息的类型包括消息集、有限状态机和基于规则的集合中的一种类型。

30. 权利要求26的计算机数据信号，其中，对象是图形设计图、文档、伪码和程序代码之一。

31. 权利要求26的计算机数据信号，其中，第一访问信号段访问某个预先存储的数据库。

32. 权利要求26的计算机数据信号，其中，第二访问信号段访问某个预先存储的规则集。



说明书

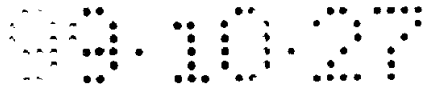
将数据作为流动信息存储的方法和装置

本发明涉及存储设计信息的改进。更具体地说，涉及按照便于携带、易于操作的形式存储信息的优点，适用于便利或自动转换为所需的格式和对象。

许多软件程序全部或部分基于某种基本类型的熟知设计数据，例如消息集、有限状态机（FSM）和基于规则的集合。这些基本设计数据在电信软件中是非常普遍的。通常，这个设计数据并没有被正式说明，而是利用程序设计语言手工硬编码，再被编译成由某个给定的处理机和操作系统执行的机器代码。设计数据或信息也在系统需求、文档或公开的标准中被说明，然后手工翻译成某个具体应用程序的源代码。目前的实际工作要求重用不同应用环境中的信息，因为写一个适合新环境的新的源代码是一个困难和费用很高的过程。

可以将设计数据或信息定义为说明某个计算机系统环境及其属性和相互联系的对象的数据。对象的例子包括代码模块、全球网（WWW）网页、包含仅用在该对象中的数据的设计规格说明、输入/输出界面、图形用户界面（GUI）和手工页。可以产生的对象的类型和变化是非常广泛的。由于对象对某个特定环境没有依赖性，因此，用在标准界面中的软件对象也可用于各种不同的应用。因此使用该标准界面的计算机可以利用设计信息来开发各种对象或计算机应用程序，只要所访问的信息不依赖某个特定的对象或环境。

设计信息可以多年保持其有用性，从当前到未来的环境。在其整个生命周期中可以按许多方式重用设计信息，并且涉及许多产品发布和生产线。因此，其生存期可能比任何单个产品更长，因为它可以适合于变化的技术和需求。保留和重用设计数据有利于更快地提交高质量的软件，在产品需求、文档和软件之间提供实时同步，并对不断变化的环境作出快速响应。



然而，当前重用数据的方法和技术损害了这些优点。目前设计数据的存储主要采用人工可读的形式，需要费用很高的人工干预，以便将信息转换成软件和其他产品，这就降低了软件的开发速度，影响了产品需求、文档和软件之间的同步，降低了对环境变化的响应速度。

如果能以某种适合于自动操作的形式来存储信息，就可以在设计数据的检索和重用中得到显著的进步。当前的应用把数据存储商业数据库中以便于检索，但目前的存储技术不允许处理这些信息以用于不同的应用环境。例如，语言学技术将设计信息描述为定制语言，并利用该定制语言来改写设计数据。然而，定制语言的建立及其修改是相当复杂和昂贵的，这是由于缺少一种能够利用市场上可买到的数据库来存储设计数据的简化技术。

因此，系统需要这样一种技术，在便于得到的数据库中存储设计信息，并使这些设计信息能用在多种用途中，而不依赖于应用的类型，其中，设计信息便于得到并可处理为所需的形式。

本发明提供了一种设计信息系统，该系统采用适合于方便地将设计信息自动转换为软件产品的一种形式来准备和存储设计信息。该系统包括一个不依赖于某个应用的数据库，采用用户定义的形式来存储设计信息，允许随后多次使用，例如程序设计代码、建立文档和画图。该信息系统包括一个处理机，访问具有为某类设计信息构造的数据库格式的数据库，访问为该数据库和某个对象类型配置的一组规则，接收该对象类型中的一个对象，其中，该对象具有设计信息，并根据存储器中的规则集将这些设计信息存在数据库中。

此后，这些设计信息在其整个生命周期中可以用在许多方面，从产品发布、部件系统到生产线。其生存期可以比任何单个产品或部件更长，因为它适应了变化的技术和需求。因此，该设计信息数据库能更快地提交更高质量的软件，在产品需求、文档和软件之间提供实时同步，并快速响应变化的环境。

将参考下面的附图详细描述本发明，其中，相同的数字代表相同的部件：

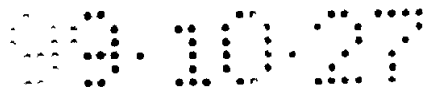


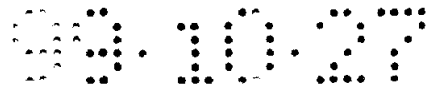
图1说明根据本发明最佳实施例的一个设计信息存储系统；
图2是一个流程图，说明根据本发明的设计信息存储方法的步骤；
图3说明利用有限状态机（FSM）设计信息的一个图形设计示意图；
图4说明一个示例性数据库的用户定义字段；
图5给出存放在图4的示例性数据库中的格式独立设计信息；以及
图6是引用具有FSM设计信息的需求文档的一个例子。

图1给出根据本发明最佳实施例的一个设计信息存储系统5。系统5包括连接到输入设备30和输出设备40的计算机10。计算机10包括连接到随机存取存储器（RAM）54、只读存储器（ROM）56和大规模存储设备50的中央处理机（CPU）52。输入设备30允许和计算机10进行人机对话和数据输入。因此输入设备30包括输入部件，例如键盘、鼠标、屏幕、扫描仪、磁盘驱动器和用于向计算机10输入数据的其他已知设备。输出设备40接收计算机10的输出数据。输出设备包含但不局限于打印机和屏幕。另外，输入设备30和输出设备40都可以是一台调制解调器、网络连接或脱机存储器，借助传播信号80和90与计算机10通信。

将设计信息数据库60存放在大规模存储设备50中，该设备可以是硬盘、软盘、光盘等。存储器50还存储计算机程序70，该程序在运行时指示计算机10执行根据本发明的设计信息存储方法。换句话说，无论是由存储器50装入，还是由输入设备30输入到计算机10，计算机程序70都把计算机10转换为实现本发明的专用机。更具体来说，本发明方法的每一步至少将部分计算机10转换为实现该步的专用计算机模块。

本发明的其他实施例包括固件实施例和硬件实施例，其中，本发明的方法被程序设计进入固件（例如EPROM，PROM或PLA）或者完全构造为硬件。构造本发明这样的固件和硬件实施例应该是利用已知技术的一般技术人员的一般技术。

此外，这里所公开的本发明也可以采取制品形式，例如计算机可用媒介，包括其中包含的计算机可读程序代码，其中，计算机可读



代码使计算机10执行本发明的方法。

数据库60是关系型的或面向对象的，正如熟练技术人员所理解的一样，并且由用户定义以存储所需设计信息。即，可以根据输入设计信息的类型或该信息的某种预期使用来设计和编辑数据库60的结构。最好使用市场上可买到的数据库管理程序来构造数据库60，例如微软的Access软件。

本发明对不同类型的设计数据进行格式化，例如从某种输入结构（例如图形设计图，程序设计代码，数据库记录表或需求文档）解码得到的消息集、有限状态机（FSM）和基于规则的集合，使之成为一种可访问的用户定义结构，以便至少编码成为一种不同的输出结构，例如图形设计图，程序设计代码，数据库记录表或需求文档。图形设计图是一个代表设计信息的图，该图的示例包括流程图、过程路径图、框图、业务图、网络图、设施管理图和工业过程图。需求文档描述某个系统或产品的设计和操作性、协议、功能和需求。

图2说明根据本发明将设计信息存储在数据库中的操作。在步骤S202，数据库60被输入存储器50中。利用数据库管理程序，用户输入数据库60的数据库设计或识别某个预先存储的数据库设计，并根据该数据库设计输入一个空数据库60以便为某个特定类型的设计数据存储信息。数据库设计是数据库60的结构轮廓。因此，如果该设计信息采用FSM形式，则该数据库将具有专门对应FSM的结构，其字段包括当前状态、下一状态、触发移动到下一状态的事件以及在该事件发生时所采取的动作。

在步骤S204，用户输入专门为该数据库和这类设计数据配置的一个规则集，或指定一个预先存储的规则集。该规则集定义了计算机10如何识别该输入结构中的信息，以便CPU 52知道在数据库中的什么地方输入信息。规则集将取决于特定的输入结构和设计数据的类型。例如，FSM流程图将有一个不同于需求文档乃至FSM框图的规则集，因为它们具有不同的输入结构。即，FSM的流程图包括每个状态的几何图形以及在几何图形中列出的定义操作的事件和动作。需求文档和框



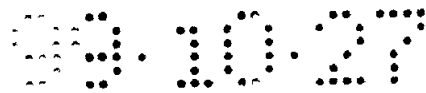
图一般不利用每个状态、事件和动作的形状。

在步骤S206，输入专用格式和独立格式的设计信息。可以上述许多不同结构中的任何一种结构接收设计信息，并且设计信息可以包含设计信息元素以及关于设计信息的结构之间关系的信息元素。当设计信息是一个图形设计图时，输入设备最好是一种能在市场上买到的阅读器，例如磁盘驱动器，并且通常包括至少一个键盘或扫描仪。当设计信息是一个消息集时，则输入设备是一个市场上能买到的文本或表格阅读器。消息集是众所周知的一种设计信息结构，例如一组命令，其中的每个命令有其相关的属性、参数和数值。此外，当设计信息是一个基于规则的集合时，输入设备是市场上能买到的一种图形设计阅读器或文本阅读器。

专用格式和独立格式的设计信息都是由计算机10读入的。然而，专用格式的信息在评估时被过滤掉，不存储在数据库60中。专用格式信息包括对于输入结构的排列是重要的、但与设计的功能和目标无关的数据。如果设计信息作为计算机程序被接收，例如，有关注释、关键字、格式以及其他特定语言特性的数据都被放弃，使得计算机程序的基本设计信息可以按数据库60所说明的独立格式的形式被保留。如果设计信息是作为需求文档被接收的，则大部分文本是独立格式的设计信息。专用格式的信息都被忽略掉，例如页号、段号、字体、标记设置和分页。如果设计信息是作为一个图形设计图被接收的，则诸如方框的标识、位置和形状、连线的方向和长度、以及页号都被忽略掉。

在步骤S208，所需的独立格式设计信息被存在用户定义的数据库中，如下所述，并且可以被不同的程序和应用查询、使用和重用。

现在将参照最佳设施例的具体示例来详细介绍本发明的操作。图3是一个图形设计图300，该图使用众所周知的有限状态机（FSM）的符号表示法，被表达为一系列的状态和动作。图300表示一台苏打水销售机的状态，并且起初位于空闲状态S1。当用户选择一瓶苏打水时，该机转移到“Need75”状态S2，并等待投入钱币。一旦接收到一枚25分的硬币，该机转移到“Need50”状态S3，并等待投入更多的硬币。一



一旦接收到第二枚25分的硬币，该机将转移到“Need25”状态S4。苏打水销售机接收第三枚25分的硬币，然后发放一罐苏打水。因此，在FSM符号表示法中，状态用椭圆形表示，而动作则用状态之间的带标号矢量表示。

如步骤S202所示，用户选择一个预先存储的数据库60，或设计并输入数据库60。图4说明数据库60，用户可以建立该数据库，用以存放用FSM符号表示法表示的设计数据。该数据库60所选择的字段是当前状态、事件、下一状态和事件动作。“当前状态”识别某个给定例子的处理过程中所在的位置。“下一状态”字段识别所在的下一位置。

“事件”字段识别从“当前状态”转移到“下一状态”所需的条件。从某个特定状态中发生的事件也可以触发在“事件动作”字段中识别的一个动作。

接着，如步骤S204所示，用户选择一个预先存储的规则集或一个规则集，指示计算机10如何从FSM处理图中识别形状和文本，并将数据输入到数据库60对应的字段中。例如，用户可以为FSM符号表示法表示的设计数据建立以下规则集：

(规则1) 将椭圆形中的数据识别为一个状态；

(规则2) 将矢量识别为状态转移；

(规则3) 将矢量标识符识别为事件；

(规则4) 将矢量标识符后面的文本识别为事件动作；

(规则5) 将矢量的基底识别为当前状态；

(规则6) 将箭头点识别为下一状态；

(规则7) 将带阴影的椭圆识别为初始当前状态；并且

(规则8) 将进入带阴影椭圆的箭头点识别为下一状态并退出子程序。

如步骤S206所示，输入设计数据。应该知道，不需要以图3所示的形状和矢量的形式输入设计数据，而可以采用计算机机器语言的形式输入，这些语言指示如何以图3所示的形式显示设计数据，例如，作为一个状态转移表。一般技术人员应该理解如何利用等价的计算机机器

语言代替规则集中的形状和矢量，以便实现本发明。然而，为了便于理解，利用形状和矢量来列出规则，而不是利用机器语言。

根据规则1和7，计算机10知道将带阴影椭圆中的数据“空闲”输入数据库60的第一个记录的“当前状态”字段，如图5所示。根据规则2、3和5，计算机10知道处理来自“空闲”状态S1的矢量的基底，读出该矢量的标识符“UserSelectsSoda”事件E1，并且将事件E1输入到数据库60的第一个记录的“事件”字段。根据规则6和1，计算机10将矢量终止于此的椭圆识别为“下一状态”，并将“Need75”输入到数据库60的第一个记录的“下一状态”字段。

根据规则5和1，计算机10还将椭圆中的数据“Need75”识别为下一“当前状态”并将数据输入到下一记录的“当前状态”字段。根据规则2、3和5，计算机10知道将来自“Need75”状态S2的矢量的基底处理为状态转移，将矢量标识符“gotquarter”识别为事件E2，并将事件E2输入到数据库60中当前状态为“Need75”的记录中的“事件”字段中。

按照这种方式继续处理，最终到达状态“Need25”。在计算机10根据规则1和5将“Need25”输入到数据库60的“当前状态”之后，根据规则2、3和5，计算机10将矢量标识符“gotquarter”识别为事件E4并将其输入到数据库60对应的“事件”字段中。根据规则4，计算机10知道输入事件E4后面的文本作为事件动作A4，并将“DispenseCan”输入到数据库60中其当前状态为“Need25”的记录的“事件动作”字段中。根据规则1和8，计算机知道状态标识符“空闲”是下一状态，将“空闲”输入到数据库60的“下一状态”字段中，如图5所示，并退出数据库输入子程序。一旦执行完该子程序，对该图的功能设计很关键的设计信息就被存储在数据库60中，如步骤S208和图5所示。

这类设计信息和结构的数据库设计和规则集只需被设计一次。即，一旦存储了图4的数据库设计和用于某个FSM图形设计图的规则集，计算机10就可以利用现有的数据库设计和规则集，从另一个FSM图形设计图中输入设计数据。其优点是，只需提供新的设计图，就能将新的

独立格式设计数据存储于数据库60中，因此，在不需重新输入数据库格式或规则集的情况下，允许用户存储信息。

图6是从一个识别专用格式和独立格式设计信息之间差别的需求文档中摘录的例子。一旦接收到该摘录，计算机程序70寻找例如关键字这样的标识符，这些关键字识别希望存储在数据库60中的独立格式设计信息。例如，图6所示的摘录描述了图3所示的图形设计图，并且利用了图4所示的数据库结构和字段。根据下面说明的规则6，在图6中，关键字“PROCESSING”之后的信息包括将作为图4所示的用户定义字段而被存储在数据库60中的设计信息。该信息被写成某种伪码，并可以方便地由具有一般熟练技术的程序员采用诸如Pascal、Fortran、C/C++、Lisp、Basic等程序设计语言进行编码。因此程序设计语言是本发明可以用来将设计信息存入数据库60的另一种表达形式。

如上所述，已经存储了如果图4所示的数据库60，则用户无需输入一个新的数据库。而可以在步骤S202中说明计算机10将利用以前存储的图4的数据库设计。

接着，如步骤S204所示，用户输入一个规则集，以便指示计算机10如何从FSM需求文档中识别文本，并将数据输入到数据库60的相应字段中。例如，一个可应用的规则集包括以下命令：

- (规则1) 将“from”和“to”之间的数据识别为当前状态；
- (规则2) 将“to”和“upon”之间的数据识别为一个状态；
- (规则3) 将“upon”后一直到“and”或“；”之前的数据识别为事件；
- (规则4) 将“and”后的数据识别为事件动作；
- (规则5) 将“.”识别为退出子程序；
- (规则6) 忽略关键字“PROCESSING:”之前的文本；并且
- (规则7) 处理关键字“PROCESSING:”之后的文本。

如步骤S206所示，采用众所周知的方式输入设计数据，例如字处理文档文件。根据规则6和7，计算机10忽略了需求文档中关键字“PROCESSING:”之前的所有特定格式信息，并开始处理该关键字

后面的文本。根据规则1，计算机10识别“from”和“to”之间的数据，并且知道将数据“空闲”输入到数据库60的第一个记录的“当前状态”字段中。根据规则2，计算机10识别“to”和“upon”之间的数据，并知道将数据“Need75”输入到数据库60的第一记录中的“下一状态”字段。根据规则3，计算机10识别“upon”到“;”之间的数据，并知道将数据“UserSelectSoda”输入到数据库60的第一个记录的“事件”字段。

按照这种方式继续处理，最终到达关键字“PROCESSING:”下的最后一行。根据规则1，计算机10识别“from”和“to”之间的数据，并且知道将数据“Need25”输入到数据库60的下一个记录的“当前状态”字段中。根据规则2，计算机10识别“to”和“upon”之间的数据，并知道将数据“空闲”输入到数据库60的相应的“下一状态”字段中。根据规则3，计算机10识别“upon”和“and”之间的数据，并知道将数据“GotQuarter”输入到数据库60的相应的“事件”字段。根据规则4，计算机10识别“and”之后的数据，并知道将数据“DispenseCan”输入到数据库60的相应的“事件动作”字段中。根据规则5，计算机10识别“.”并退出数据库输入子程序。一旦执行完该子程序，对该图的功能设计很重要的设计信息就被存放在数据库60中，如步骤S208和图6所示，而专用格式信息“go”、“from”、“to”等都被忽略掉。

根据图6所示的需求文档存储在数据库60中的设计信息与根据图3所示的设计输入图存储在数据库60中的设计信息是一致的。即，本发明应用了需求文档和设计输入图来提供相同的存储数据库60。因此，采用通用格式存储相同的设计信息，而不依赖于特定的表达方式。这些设计数据能够成为可重用的资产，容易适应未来开发的表达方式，因为去掉了那些无关重要的因素。数据库60可以被各种应用查询和使用，例如交互式的PC程序，包括系统GUI、仿真程序和全球网应用，并且可以被用来开发程序设计代码和文档。例如，可以将已存储的设计数据转换为其他有用的格式，利用相关未知申请号，题目为“将设计信息扩展到软件产品中的方法和装置”，申请日期和申请人与本主

题申请相同，在此全文引用作为参考。

虽然存储设计数据的方法是结合用FSM符号表示法并采用需求文档表示的设计数据进行描述的，但本发明的方法并不局限于这里所说明的这些表达形式或数据库设计。熟练技术人员通过以上的介绍应该很容易理解如何开发数据库设计和规则集来存储采用任何形式表示的设计数据，这些形式包括，但不局限于，FSM、消息集、基于规则的集合、程序代码、文档、表项数据和图形设计图。

还应该理解，数据库60可以具有任何类型的独立格式结构，包括关系型的和面向对象的结构。还应该很容易理解，可以用市场上买到的数据库程序或为特定数据库设计的表格输入系统直接修改数据库60。

从上述的介绍和附图可知，本发明的概念可以很方便地应用于各类最佳实施例包括这里所介绍的。因此，在示例应用中所描述的本发明的范围，应该只受附加的权利要求书的限制。

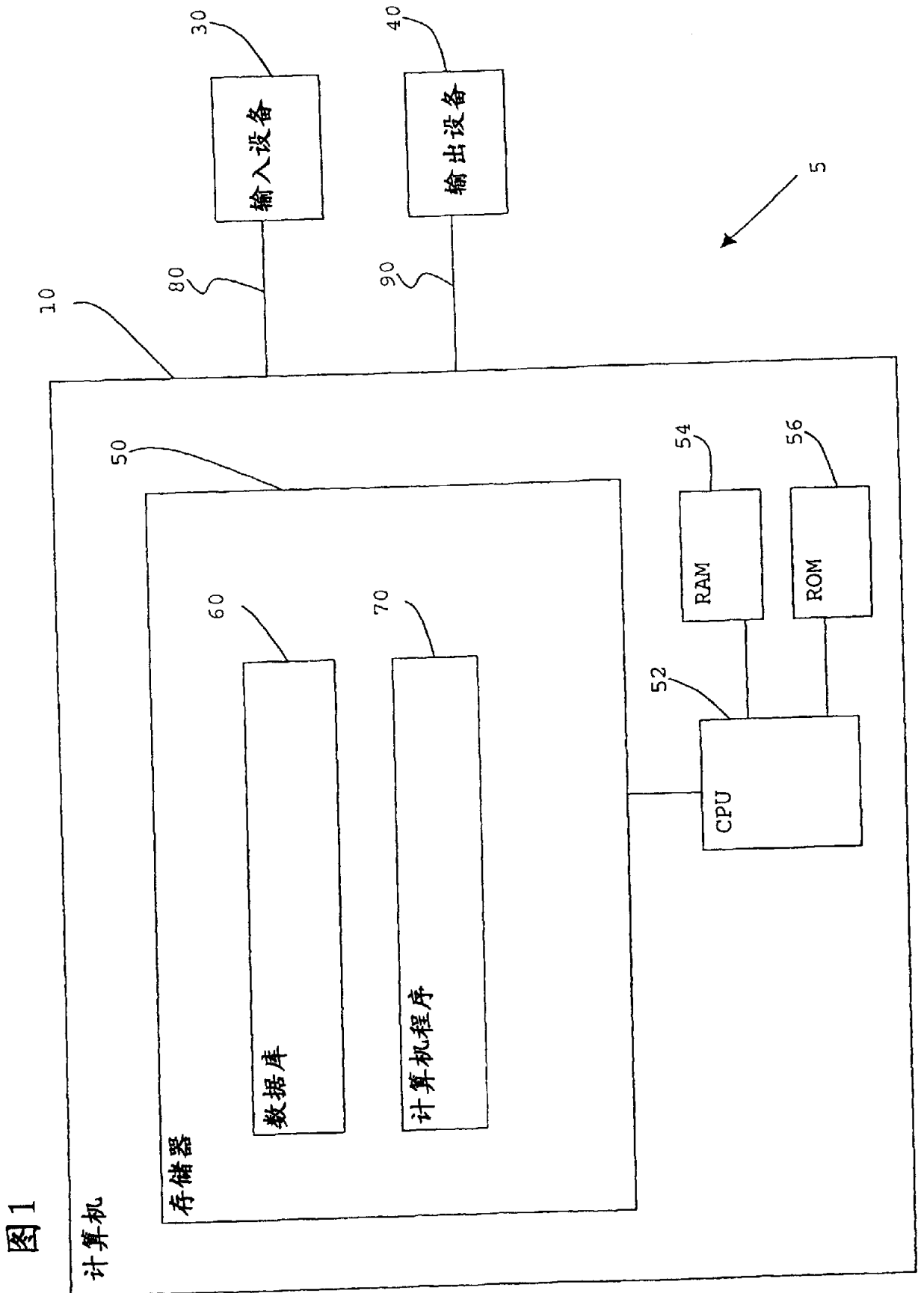


图1

计算机

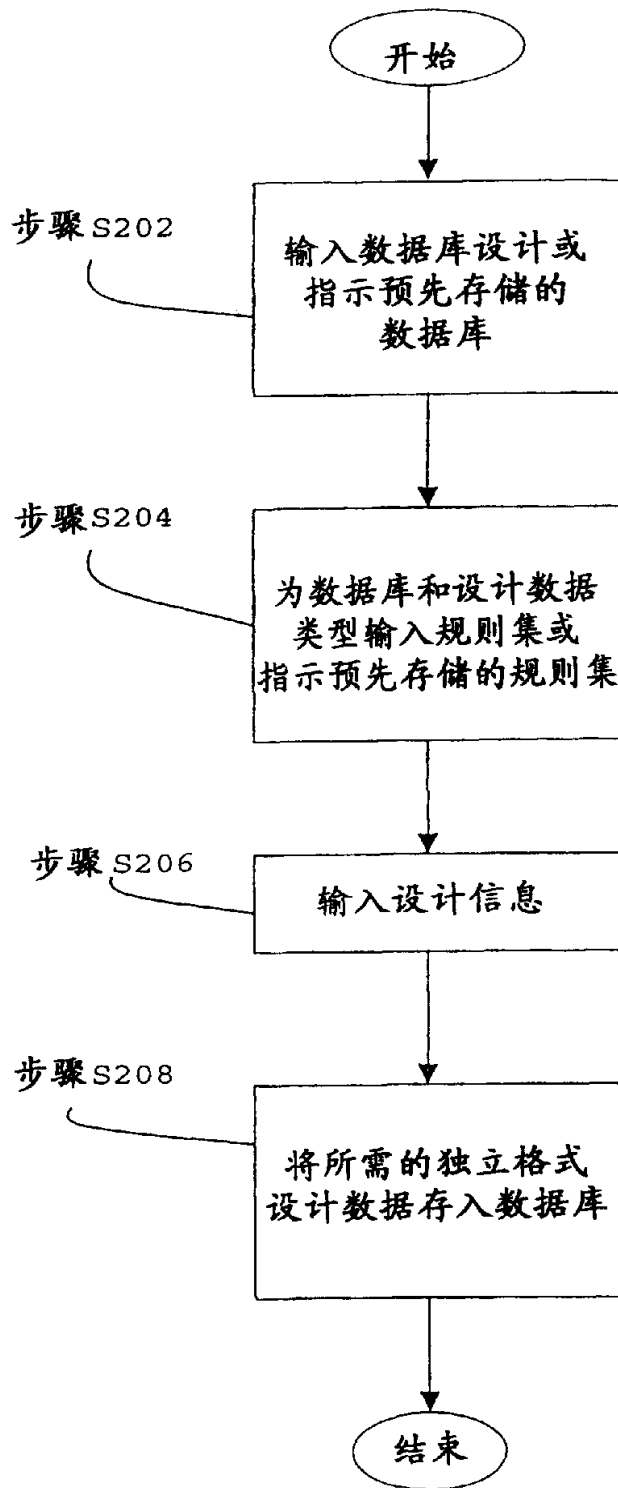
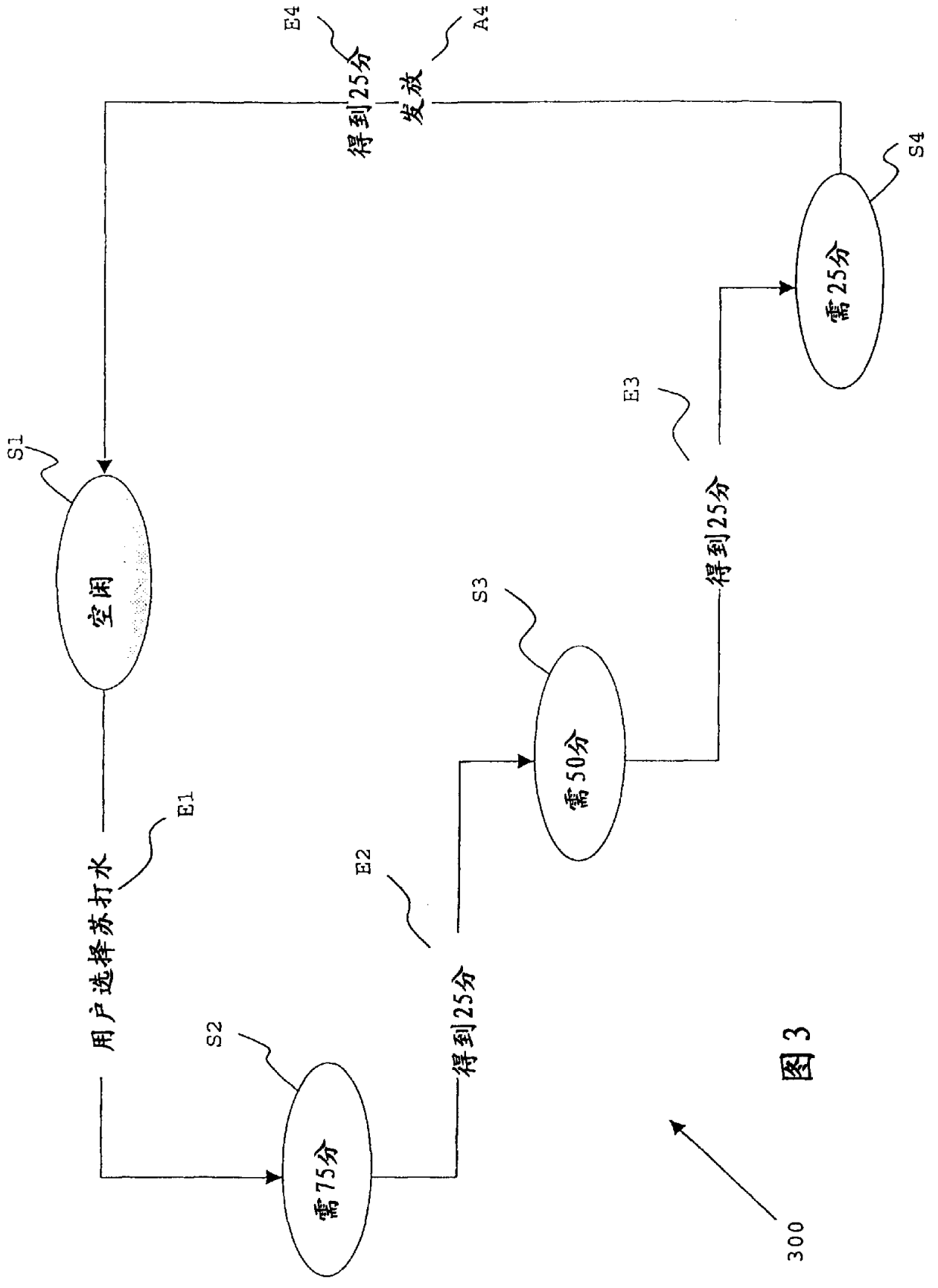


图 2



3

图3

300

60

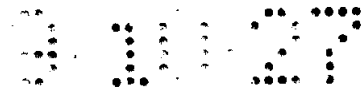
当前状态	事件	下一状态	事件动作

图 4

60

当前状态	事件	下一状态	事件动作
空闲	用户选择苏打水	需75分	
需75分	得到25分	需50分	
需50分	得到25分	需25分	
需25分	得到25分	空闲	发放一罐

图 5



需求: 苏打水机操作设计

简介: 本设计利用状态、事件和动作定义投币式苏打水销售机的操作过程

字段: 当前状态; 下一状态; 事件; 事件动作

处理:

若用户选择苏打水, 则从空闲到需75分;

若得到25分, 则从75分到需50分;

若得到25分, 则从需50分到需25分;

若得到25分, 则从需25分到空闲并发放一罐苏打水.

图6