



(12) 发明专利

(10) 授权公告号 CN 101203906 B

(45) 授权公告日 2011.07.06

(21) 申请号 200680019192.4

US 2002/0095286 A1, 2002.07.18, 全文.

(22) 申请日 2006.05.08

US 6434524 B1, 2002.08.13, 全文.

(30) 优先权数据

11/142, 196 2005.05.31 US

EP 0911808 A1, 1999.04.28, 全文.

Danilo Mirkovic et al.. Practical

Plug-and-Play Dialogue Management.

(85) PCT申请进入国家阶段日

《PROCEEDINGS OF PACLING'05: 6th MEETING OF THE PACIFIC ASSOCIATION FOR COMPUTATIONAL LINGUISTICS》. 2005, 1-8.

2007.11.30

(86) PCT申请的申请数据

PCT/US2006/017800 2006.05.08

Manny Rayner et al.. Plug and Play

(87) PCT申请的公布数据

W02006/130314 EN 2006.12.07

Speech Understanding. 《ANNUAL MEETING OF THE ACL, PROCEEDINGS OF THE SECOND SIGDIAL WORKSHOP ON DISCOURSE AND DIALOGUE》. 2001, 第16卷 1-10.

(73) 专利权人 罗伯特·博世公司

Fuliang Weng et al.. A Conversational Dialogue System for Cognitively Overloaded Users. 《PROCEEDINGS OF INTERSPEECH 2004: 8th INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING (ICSLP)》. 2004, 233-236.

地址 美国伊利诺斯

审查员 康丹丹

(72) 发明人 D·米尔科维奇 L·卡维多

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 王英

(51) Int. Cl.

G10L 15/26(2006.01)

(56) 对比文件

CN 1455357 A, 2003.11.12, 全文.

US 2001/0041980 A1, 2001.11.15, 全文.

CN 1389852 A, 2003.01.08, 全文.

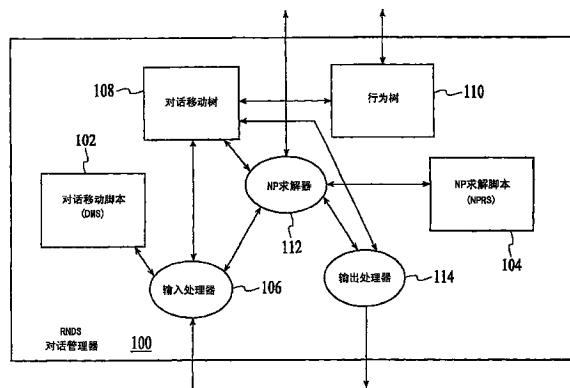
权利要求书 3 页 说明书 14 页 附图 8 页

(54) 发明名称

使用脚本进行对话管理的方法、设备和系统

(57) 摘要

描述了中性表示对话系统和方法（“RNDS”），包括基于信息状态更新方法的多应用、多设备口语对话系统。RNDS 包括对话系统的中性表示核心组件，其提供对诸如对话移动模型和指称求解这样的例程的脚本化域专用扩展，提供专用语义表示和相关联例程的轻易替换，以及到用于语言理解（即语音识别和解析）和语言生成的外部组件以及到域专用知识源的干净接口。RNDS 还允许与设备群的无缝交互。



1. 一种对话使能方法,包括 :

对从声信号生成的输入模式进行接收,所述输入模式包括对来自讲话者的对话贡献的描述;

使用对话移动脚本对所述描述和所述描述的至少一个参数进行识别,其中,所述对话移动脚本对应于设备和应用中的至少一个;

使用所述对话移动脚本将所述描述映射到对话移动,所述对话移动对应于所识别的参数并且独立于所述设备和应用;

使用所述描述中的名词短语对与所述描述对应的设备和应用中的至少一个进行识别;

选择与所识别的设备和应用中的至少一个对应的名词短语求解脚本,并且通过所选择的名词短语求解脚本访问名词短语求解规则;以及

使用所述名词短语求解规则对所述描述进行翻译,并且生成与所述设备和所述应用中的至少一个对应的查询。

2. 如权利要求 1 所述的方法,还包括 :

通过对所述对话移动脚本进行重新编程,使得所述对话移动适合于在多个不同域和多个不同应用中的至少一个上使用。

3. 如权利要求 1 所述的方法,还包括 :

响应于所述查询,从至少一个数据库中检索由来自所述对话贡献的对象描述所指称的至少一个对象。

4. 如权利要求 3 所述的方法,其中,所述对象描述包括名词短语。

5. 如权利要求 1 所述的方法,还包括 :

通过对所述名词短语求解脚本进行重新编程,使得所述名词短语求解规则中的至少一个规则适合于在多个不同域和多个不同应用中的至少一个上使用。

6. 如权利要求 1 所述的方法,其中,所述名词短语求解脚本包括知识库查询构造规则,所述知识库查询构造规则是可编程的,以对于多个设备中的每个设备定义如何将名词短语对象映射到查询。

7. 如权利要求 3 所述的方法,其中,生成所述查询包括将所述描述中的名词短语翻译成所述对象。

8. 如权利要求 1 所述的方法,其中,所述描述是基于结构化语言的描述,其包括所述对话贡献的语义描述、句法描述、音位信息、话语级别信息以及主题分类中的至少一个。

9. 如权利要求 1 所述的方法,其中,所述映射还包括对所述对话贡献的至少一个参数进行识别。

10. 如权利要求 1 所述的方法,还包括 :

对所述描述中的变量和所述对话移动的组件之间的匹配进行评分。

11. 一种对话使能系统,包括 :

基于信息状态更新方法的对话管理器,其包括多个对话移动和一个对应于设备和应用中至少一个的对话移动脚本,所述对话管理器对从声信号产生的输入模式进行接收,所述输入模式包括对来自讲话者的对话贡献的描述,所述对话移动脚本对所述描述和所述描述的至少一个参数进行识别,所述对话移动脚本将所述描述映射到所述多个对话移动中与所

识别的参数对应的一个对话移动；

知识管理器，其连接到所述对话管理器和至少一个数据库；以及

设备管理器，其连接到所述对话管理器，其中，所述对话管理器还包括名词短语求解器，其连接到多个名词短语求解脚本和多个规则，其中，每个名词短语求解脚本对应于所述设备和所述应用中的至少一个。

12. 如权利要求 11 所述的系统，其中，通过对所述对话移动脚本进行重新编程，使得所述多个对话移动中的至少一个适合于在至少一个不同的域和应用上使用。

13. 如权利要求 11 所述的系统，其中，所述名词短语求解器使用名词短语求解脚本来访问所述规则，并且对所识别的描述进行翻译，以生成对应于所述设备和应用中至少一个的查询。

14. 如权利要求 11 所述的系统，其中，通过对所述名词短语求解脚本进行重新编程，使得所述规则适合于在至少一个不同的域和应用上使用。

15. 如权利要求 13 所述的系统，其中，所述名词短语求解脚本包括知识库查询构造规则，所述知识库查询构造规则是可编程的，以对于所述设备中的每个设备定义如何将名词短语对象映射到查询。

16. 如权利要求 13 所述的系统，其中，所述知识管理器将所述查询传送到所述至少一个数据库。

17. 如权利要求 16 所述的系统，其中，响应于所述查询，所述对话管理器从所述数据库接收所述对话贡献的至少一个对象。

18. 如权利要求 11 所述的系统，还包括多个设备，其中，将每个设备通过符合设备应用编程接口的每个设备的描述连接到所述设备管理器，其中，每个设备描述包括行为模型、设备专用对话移动脚本、设备专用名词短语求解脚本、以及实现更多设备专用功能的代码中的至少一个。

19. 如权利要求 18 所述的系统，其中，所述实现更多设备专用功能的代码包括设备专用对话移动。

20. 如权利要求 18 所述的系统，其中，所述行为模型包括对应于对话移动脚本的变量的变量。

21. 如权利要求 19 所述的系统，其中，响应于将所述设备连接到所述设备管理器，所述对话管理器对来自所述设备应用编程接口的所述行为模型、设备专用对话移动脚本、设备专用名词短语求解脚本以及设备专用对话移动的信息进行接收和存储。

22. 如权利要求 11 所述的系统，还包括被连接到所述对话管理器的输入的语音识别模块和语言理解模块中的至少一个。

23. 如权利要求 11 所述的系统，还包括被连接到所述对话管理器的输出的语言发生模块和文本到语音模块中的至少一个。

24. 一种对话使能设备，包括：

用于对从声信号生成的输入模式进行接收的装置，所述输入模式包括对来自讲话者的对话贡献的描述；

用于使用对话移动脚本对所述描述和所述描述的至少一个参数进行识别的装置，其中，所述对话移动脚本对应于设备和应用中的至少一个；

用于使用所述对话移动脚本将所述描述映射到对话移动的装置,所述对话移动对应于所识别的参数并且独立于所述设备和应用;

用于使用所述描述中的名词短语对与所述描述对应的设备和应用中的至少一个进行识别的装置;

用于选择与所识别的设备和应用中的至少一个对应的名词短语求解脚本,并且通过所选择的名词短语求解脚本访问名词短语求解规则的装置;以及

用于使用所述名词短语求解规则对所述描述进行翻译,并且生成与所述设备和所述应用中的至少一个对应的查询的装置。

使用脚本进行对话管理的方法、设备和系统

技术领域

[0001] 本公开一般涉及对话管理，并且更具体地，涉及多应用、多设备口语对话系统的管理。

[0002] 背景技术

[0003] 对话包括参与者之间会话的语言以及由参与者为会话而构建的共享中心语境（例如，在后续会话中指称的“它”是指在会话中较早所描述的某物）。会话的参与者可以是人、机器、或者人和机器的任何组合。对话管理包括相对于共享语境对讲话者的话语进行解释，以及用于对对话参与者之间的交互进行管理的技术和策略。已经为诸如机器人设备的多模态控制、语音使能教学系统、与汽车内设备会话交互这样的应用开发了面向行为的对话系统。典型的对话系统结构包括像语音识别器、语言解析器 (parser)、语言发生器、语音合成器、以及对话管理器 (“DM”) 的各种组件。该对话系统还可以包括到诸如本体 (ontology) 或知识库 (“KB”) 的外部专用组件的连接以及对话使能设备。对话系统的例子见下列文献：(i)Lemon, O., A. Gruenstein, S. Peters (2002), “ Collaborative activities and multi-tasking in dialogue systems ” , Traitement Automatique des Langues (TAL) , 43 (2) ; (ii)Clark, B., J. Fry, M. Ginzton, S. Peters, H. Pon-Barry, Z. Thomsen-Grey (2001), “ Automated tutoring dialogues for training in shipboard damage control ” , SIGdial ; 和 (iii)Weng, F., L. Cavedon, B. Raghunathan, D. Mirkovic, H. Cheng, H. Schrnidt, H. Bratt, R. Mishra, S. Peters, L. Zhao, S. Upson, L. Shriberg, C. Bergmann (2004), “ A conversational dialogue system for cognitively overloaded users (poster) ” , INTERSPEECH.

[0004] 对话系统的 DM 是有助于对话参与者之间交互的监管模块。使用行为模型 (Activity Model) 的对话系统特指一种被称为“面向行为的对话”的对话类型，其是关于正在由用户和机器、计算机、以及 / 或者机器人 (联合) 开展的行为的对话。在用户或者讲话者发起系统中，DM 指示通过解释和后端系统响应对从一个组件到另一个组件的输入话语的处理。在处理中，例如，DM 对输入话语的信息输入进行检测和处理，并且生成系统输出。可以与不同解析器和语言发生组件一起使用 DM。通过行为模型 (“AM”), 即所公布的设备能力规范和它们与语言处理的关系，对与外部设备的交互进行调解。然而，由于在不同应用上对话移动 (dialogue move) 需求的变化、与语言解析器和其它组件接口中表示法的变化、以及具有领域专用方面的某些处理（例如，指称求解 (reference resolution)），所以对新领域的定制一般需要一些显著的规划努力。

[0005] 常规的对话管理系统的范围从广泛商业应用但是更受约束的基于语音扩展标记语言 (“VXML”) 的对话建模机制到基于对信息状态进行更新的 TrindiKit 方法的语义模型。虽然为特定的领域设计并且实现了许多对话系统，但是将这些系统应用到新的领域需要相当大的工程。相反地，基于 VXML 的对话管理基础结构允许对新的领域灵活实现基于语音的对话系统，但是仅提供对对话建模中许多问题的肤浅解决方法。

[0006] 通过引用合并

[0007] 这里通过引用将本说明书中所提到的每个出版物和 / 或发明申请全部合并到本申请中,如同专门并且单独指定每个独立的出版物和 / 或发明申请通过引用合并到本申请中。

附图说明

[0008] 图 1 是根据实施例的中性表示对话系统 (“RNDS”) 对话管理器 (“DM”) (“RNDS DM”) 的方框图;

[0009] 图 2 是根据实施例的 RNDS 的方框图;

[0010] 图 3 是根据实施例的用于使用对话移动脚本 (“DMS”) 将对话贡献 (dialogue contribution) 的描述映射到对话移动的流程图;

[0011] 图 4 是根据实施例的用于使用名词短语求解脚本 (“NPRS”) 生成相应于设备的数据库查询 (名词短语查询) 的流程图;

[0012] 图 5 是根据实施例用于 MP3 设备的示例行为模型 (“AM”) 的部分;

[0013] 图 6 是根据实施例用于 MP3 设备的“play”命令的示例 DMS;

[0014] 图 7 是根据实施例的对于“the song Vertigo by U2”的示例名词短语查询 (“NP 查询”) 对象;

[0015] 图 8 是根据实施例的名词短语求解规则的例子。

[0016] 在附图中,相同的参考标号代表相同的或者非常类似的组件或行为。为了对任何特定组件或行为的讨论进行标识,在参考标号中最高位数字是首次引入该组件的图号(例如,关于图 1 首次对组件 100 进行引入和讨论)。

具体实施方式

[0017] 基于信息状态更新方法,所描述的中性表示 (representation-neutral) 对话系统和方法包括多应用、多设备口语对话系统。在这里共同被称为中性表示对话系统 (“RNDS”) 的中性表示对话系统和方法包括对话系统的中性表示核心组件,其提供对诸如对话移动模型和指称求解这样的例程的脚本化的域专用扩展。RNDS 支持专用语义表示和相关例程的置换、到用于语言理解 (即语音识别和解析) 和语言生成的外部组件以及到域专用知识源的干净接口,从而减少或者消除重新编码任何核心软件组件的任何需求。如下所述, RNDS 还允许与设备群的无缝交互。

[0018] RNDS 的脚本包括对话移动脚本 (“DMS”)、行为模型和名词短语求解脚本 (“NPRS”),提供了将 RNDS 轻松定制到新的对话域和应用中的能力。此外,RNDS 构成了“即插即用”对话管理结构的基础,其中,设备应用编程接口 (“API”) 对所定制的对话移动、行为模型、知识库、以及到核心处理的域专用扩展 (例如指称求解) 进行封装。这种即插即用对话管理使得能够进行多设备对话管理,允许将新的对话使能设备动态添加到现存的多设备对话系统。

[0019] 实施例的 RNDS 通过使用 DMS 实现对对话管理的信息状态更新 方法。DMS 包括一般对话处理与对“对话移动”进行轻松扩展或定制的能力的组合,其对从用户到新域的对话贡献进行处理。特别地,这构成了 RNDS 中多设备对话管理的基础,其中,每个设备提供其自己的 DMS (“设备专用 DMS”),该 DMS 包括从用户话语到设备专用对话移动的映射。实施例

的其它脚本（例如，NPRS）允许对其它对话处理的定制，例如，名词短语求解（“NP 求解”）（即，将自然语言中的对象描述映射到数据库中的实际对象）以及行为模型，其中，行为模型对由对话系统控制的对话使能设备的能力提供描述。

[0020] 将 RNDS 脚本与来自核心对话管理结构的其它信息（例如，行为模型（“AM”））进行组合，以便在 RNDS 中提供即插即用能力。实施例的即插即用 RNDS 允许对新的设备进行对话使能和与 DM 进行动态注册，而不需要对 DM 进行重启或重新编译。例如，在汽车工业的语境中，这允许将用于汽车的新设备作为对话使能实体出售，随后将该新设备添加到诸如小汽车或卡车的车辆中的现存对话系统内。

[0021] 通过对 DM 中现存的对话管理体系结构进行扩展，实施例的 RNDS 还支持多设备对话管理。在 RNDS 下扩展的对话管理对相关的数据结构进行扩展，从而使得同时对多个对话使能设备进行管理。所扩展的对话管理还对诸如 NP 求解的内部处理进行扩展，以使扩展的处理适合多设备交互。特别地，这种处理可以取绝于所选择的设备，而设备识别自身使用 NP 求解。

[0022] 因此，RNDS 提供了强大的实际核心对话管理处理与域专用信息的易于脚本化的定义的组合，所述域专用信息的定义诸如对话移动和 NP 求解映射。因此，RNDS 允许将 DM 轻松定制到新的域和应用，并且还提供用于即插即用多设备对话管理的工具。RNDS 还提供明确的多设备对话管理处理，为了信息状态更新对话管理对核心对话管理体系结构进行扩展，从而能够对与多个设备的同时交互进行管理。

[0023] 在汽车应用语境中具有实际重要性的是，与车辆的会话对话需要与许多设备进行交互，并且自然状态的交互需要与不同设备的无缝对话管理。此外，随着与汽车内设备基于语音的对话交互成为现实，新设备或者具有升级功能的设备可能出现在市场上并且是对话使能的。使用这些新的或者被升级的设备应该不需要具有内装 DM 的车辆的所有者对他们的系统进行更新。因此，RNDS 的动态即插即用多设备对话管理是实现添加新的对话使能设备或者增强常规对话管理体系结构内现存设备对话能力的必要组成部分。

[0024] 在下列说明中，引入了许多特定细节，以便提供对 RNDS 的实施例的彻底理解，并且能够对 RNDS 的实施例进行说明。然而，本领域的相关技术人员将认识到，可以不需要一个或多个特定细节来实现这些实施例，或者用其它组件、系统等来实现这些实施例。在其它例子中，没有示出众所周知的结构或操作或者没有对其进行详细描述，以便避免使所公开实施例的各个方面变得模糊。

[0025] 图 1 是根据实施例的中性表示对话系统（“RNDS”）对话管理器（“DM”）（“RNDS DM”）100 的方框图。RNDS DM 100 包括多个系统或组件和设备专用输入，其包括但不限于对话移动脚本（“DMS”）102、名词短语求解脚本（“NPRS”）104、行为模块、输入处理器 106、对话移动树（“RMT”）108、行为树（“AT”）110、名词短语求解器（“NP 求解器”）112 以及输出处理器 114 中的至少一个。可以将 RNDS DM 的组件 102-114 按照一种或多种组合进行连接，以适合于容纳该 RNDS DM 的系统的实施例。

[0026] 示例实施例的 RNDS DM 100 包括输入处理器 106，其从 RNDSDM 外部的一个或多个系统接收输入。将输入处理器 106 连接到 DMS102、DMT 108 和 NP 求解器 112。将 NPRS 104 连接到 NP 求解器 112，但是不限于此。将实施例的 DMT 108 连接到 AT 110。还将 NP 求解器 112 连接到 DMT 108 和输出处理器 114。

[0027] 实施例的 DMS 102 和 NPRS 104 对到 RNDS DM 100 的输入命令进行识别，并且将该命令翻译成为了检索该命令的对象的查询。特别地，DMS 102 对输入命令（例如，“play”，或者其它命令，像问题等）与该输入命令的参数（例如，对象的名称、描述等，如播放“X”）一起进行识别，并且一旦识别了所接收的命令，NPRS 104 的规则就对命令进行翻译或者指定一个知识库或数据库查询，以便从数据库（未示出）中检索该命令的实际对象（命令“Play X”的对象“X”）。因为 取决于设备的语境，相同的字串可能具有两种完全不同的意义，所以将命令 / 描述翻译成特定的查询可以是依设备而定的。因此，RNDS DM 100 指定一个查询，该查询包括用户已请求对象的类型信息 (song(歌曲)) 以及任何特定的约束（例如，name(名称)、artist(艺术家) 等）。

[0028] 作为包括 RNDS DM 100 的对话系统的例子，图 2 是根据实施例的中性表示对话系统（“RNDS”）200 的方框图。RNDS 200 包括 RNDS DM 100，RNDS DM 100 连接到一个或多个其它组件，以适合于 RNDS 200 和 / 或容纳或包含 RNDS 200 的系统的配置。如上文关于图 1 以及本文其它地方所描述的，RNDS DM 100 包括 DMS 102 和 NPRS 104。除 RNDS DM 100 之外，实施例的 RNDS 200 还包括一个或多个语音识别（“SR”）模块或系统 202、语言理解模块 204、知识管理模块 206、知识源 208、语言发生（“LG”）模块 210、以及文本到语音（“TTS”）模块 212。仅在口语对话系统中包括 SR 模块 202 和 TTS 模块 212。RNDS 200 还可以包括应用管理器（未示出）。

[0029] SR 模块 202 在一个或多个输入处接收声信号，并且输出具有诸如置信度评分的附加标签的词序列或阵列。声信号可以来自用户（讲话者）（例如，用户话语）。语言理解模块 204 也被称为自然语言理解（“NLU”）模块 204，将其连接到 SR 模块 202，并且在一个或多个输入处从 SR 模块 202 接收具有附加置信度的词序列或阵列。NLU 模块 204 输出结构化意义表示，该结构化意义表示可能基于在域内语言数据上所训练的统计模型和可用知识库。将 RNDS DM 100 连接到 NLU 模块 204，并且 RNDS DM 100 在一个或多个输入处接收结构化意义表示。RNDS DM 100 在语境中对输入表示进行解释，基于该语境及其知识库向 RNDS 200 的其它模块发布合适的指令。

[0030] 将知识管理模块 206 或者在这里被称为“知识管理器”（“KM”）206 连接到 RNDS DM 100，并且知识管理模块 206 通过对一般本体、域专用本体和用于任何有效域的数据库进行管理，来作为到不同知识源或知识库的关口。RNDS DM 100 还在不同模块之间对当前知识库进行更新和同步。

[0031] LG 模块 210 也被称为自然语言发生器（“NLG”）210，例如，将其连接到 RNDS DM 100 的输出处理器 114，并且 LG 模块 210 在一个或多个输入处对来自 RNDS DM 100 的输出进行接收。LG 模块 210 以简洁并且合乎文法的方式将所接收的内容组织为适合于 TTS 模块 212。TTS 模块 212 从 LG 模块 210 接收组织后的内容（例如，具有标记特征的词序列），并且产生语音波形。

[0032] 还使用设备管理器 220 以及一个或多个各自的设备 API 222 将 RNDS DM 100 的组件连接到一个或多个设备 250。实施例的设备管理器 220 将 RNDS DM 100 的 AT 110 连接到设备 API 222，设备 API 222 适合于 RNDS 200 配置的设备 250 的数目。每个设备 API 222 包括 AM 230、设备专用 DMS 232、设备专用 NPRS 234（也被称为 NP 求解语法）、以及对话管理器处理扩展（“DM 处理扩展”），但是不限于此。

[0033] 因此, RNDS DM 100 的 DMS 102 和 NPRS 104 增强了 RNDS DM100 的扩展性、定制和重用, 并且提供了多设备即插即用 RNDS 200 的基础。DMS 102 和 NPRS 104 允许通过为新的域和应用生成脚本对 RNDS 200 进行定制。脚本还允许对设备信息进行封装, 以支持 RNDS200 的即插即用能力。因此, RNDS DM 100 和 RNDS 200 提供了一种框架, 在该框架中, 可以轻易地将新的设备或者用于现存设备的对话能力添加到主系统而不破坏现存的体系结构。作为一个例子, RNDS DM 100 和 RNDS 200 的实施例可以是用于对车内电子组件进行对话控制的系统的组件, 所述系统例如娱乐系统、导航系统、和通讯设备, 但是实施例不限于车辆系统。

[0034] 作为包括 DMS 102 的操作的一个例子, 图 3 是根据实施例的使用 DMS 将对话贡献的描述映射 300 到对话移动的流程图。在从讲话者接收到声信号之后, 映射 300 开始接收 302 来自讲话者的对话贡献的结构化描述 302。该结构化描述可以包括一个或多个句法、语义和音位信息, 以及诸如主题分类的话语范围信息。选择 DMS 用于对该描述以及该描述的至少一个参数进行识别 304。DMS 对应于设备和应用的至少一个, 但是不限于此。DMS 将该描述映射 306 到对话移 动, 其中, 对话移动独立于设备和应用并且对应于所识别的参数。通过对 DMS 进行重新编程, 可以对对话移动进行定制或者使其适合于在多个域和 / 或应用上使用。

[0035] 返回包括 NPRS 104 操作的例子, 图 4 是根据实施例的使用 NPRS 生成对应于设备的数据库查询 (名词短语查询) 的流程图。查询的生成开始于对与来自讲话者的对话贡献的描述对应的设备和 / 或应用进行识别 402。该识别 402 使用该描述的名词短语, 但是不限于此。选择 404 对应于所识别设备和 / 或应用的名词短语求解脚本 (NPRS), 并且使用 NPRS 访问名词短语求解规则。使用该名词短语求解规则对该描述进行翻译 406, 并且生成对应于设备和 / 或描述的查询。查询一旦产生, 就将其提供给 RNDS 的数据库或知识库, 以从数据库检索对应于对话贡献的对象 (或者多个对象)。

[0036] 可以将 RNDS DM 100 和 / 或 RNDS 200 的组件连接到在这里的图中未示出的其它组件。虽然在这里通常使用术语“组件”, 但是应该理解, “组件” 包括如在本领域中已知术语的电路、组件、模块、和 / 或电路、组件和 / 或模块的任何组合。虽然可以将在这里示出的各个组件描述为设置在一起, 但是实施例不限于此; 各个可选实施例的 RNDS DM 100 和 / 或 RNDS 200 可以将由任何所描述的组件提供的一个或多个功能分布在任何数目和 / 或类型的组件、模块和 / 或电路中。

[0037] 虽然示出了包括 RNDS DM 100 和 / 或 RNDS 200 的每个组件之一, 但是各个可选实施例包括任何数目的这些组件和 / 或这些组件的任何组合, 其中, 以本领域已知的各种配置对这些组件中的每个进行连接。此外, 虽然将 RNDS DM 100 和 / 或 RNDS 200 的组件表示为单独的方框, 但是可以将这些方框中的某些或者全部统一集成在单独一个芯片上、分布在多个芯片或者主系统的多个组件上、以及 / 或者由算法的某些组合提供。通常, 这里使用的术语“处理器”是指任何逻辑处理单元, 例如, 一个或多个 CPU、数字信号处理器 (“DSP”)、专用集成电路 (“ASIC”) 等。

[0038] RNDS DM 100 和 / 或 RNDS 200 的实际配置根据主系统的组件、配置、功能和 / 或形成因数而定; 因此, 在 RNDS DM 100 和 / 或 RNDS 200 的组件之间示出的连接仅仅是示例性的, 并且不是要将 RNDSDM 100 和 / 或 RNDS 200 限制于所示的配置。可以在一个或多个处理器上运行的软件算法、固件和硬件的任何组合来实现 RNDS DM100 和 / 或 RNDS 200, 其

中,可以将软件存储在任何合适的计算机可读介质上,例如,将微码存储在半导体芯片中、计算机可读盘上、或者从服务器下载的以及本地存储在主设备上。

[0039] RNDS DM 100 和 / 或 RNDS 200 在程序或算法控制下可以在其它组件的任何组合之间进行连接,这些其它组件例如包括处理器、存储器设备、总线、控制器、输入 / 输出设备、通信系统以及显示器。所连接的数据库可以包括任何数目、类型和 / 或组合的存储器设备,存储器设备包括只读存储器 (“ROM”) 和随机访问存储器 (“RAM”),但是不限于此。

[0040] 因为 RNDS DM 100 的 DMS 102 允许不同设备根据特定设备的需求而使用 RNDS DM 100 的 Java 组件,所以它有助于在不同的设备(连接到RNDS DM 100 的输入和输出的设备)上使用相同的 Java 代码组件。因此,DMS 102 类似于用于编写对话移动的编程语言。为了在许多设备上使用对话移动,通过在比实际 Java 代码更高的级别上改编对话移动的 Java 代码,DMS 102 提供了一种高效地改编对话移动以在不同域或应用上使用对话移动的方式。因此,为了对来自特定设备的输入或发往特定设备的输出进行处理,可以通过对 DMS 102 的改变来对 RNDS DM 100 所需要的任何变化进行调整,而无需变化核心对话移动的 Java 代码。这样,DMS 102 允许在新的应用中使用对话移动,或者允许对新的设备作出命令,而不改变对话移动的核心 Java 代码。类似地,DMS 102 也允许为新的设备高效地生成新的应用。

[0041] 参考图 1 和 2,可以与用于解析、NL 生成 (NLG) 等的不同组件一起使用 RNDS DM 100。RNDS DM 100 的实施例可以使用基于规则的头部驱动解析器 (例如,二阶 (Gemini) NLU 系统),其具有适合特定应用域的语法,其中,解析器自身进行语义标准化,返回直接对应于设备行为特定表示的语义“逻辑形式”。RNDS DM 100 的实施例还可以 使用三方统计解析器,其仅返回弱标准化的语义形式。

[0042] 实施例的 RNDS DM 100 使用信息状态模型或者更新方法 (见 Larsson, S. 和 D. Traum 2000, “information state and dialoguemangement in the TRINDI dialogue move engine toolkit”, NaturalLanguage Engineering, 6(3-4)), 以便维持对话语境。“信息状态模型”是一种建模方法,其对用于解释会话参与者的每个贡献的取决于语境的方法进行建模 (由参与者引入会话的对象和其它知识构成语境)。信息状态模型基于前述事物构建语境,使得不要求讲话者在会话的后段必须表达完整的想法。随后,使用对话语境对输入话语 (包括片段和修正) 进行解释,对名词短语 (“NP”) 进行求解,构建突出响应,对事件进行追踪等。还使用对话状态对语音识别器期望进行偏移并且改进 SR 性能 (见 Lemon, O. 和 A. Gruenstein, 2004, “ Multi-threaded contentfor robust conversational interfaces :context-sensitive speech-recognitionand interpretation of corrective fragments ” , Transactions onComputer-Human Interaction (ACM TOCHI) 511(3))。

[0043] 对话信息状态的两个中心 RNDS DM 100 组件是 DMT 108 和 AT110。DMT 108 表示对话的历史语境。将每个对话贡献分类为“对话移动”(例如,“Command”、“WhQuestion”、“WhAnswer”等),并且通过将其自身附加到 DMT 108 上合适的“有效”节点,在语境中对其进行解释。例如,“WhQuestion”附加到对应于“WhAnswer”节点的有效节点。DMT 108 的树结构专门支持多线程、多主题会话,其中,新的会话主题产生新的分支 (见 Lemon, O. , A. Gruenstein, S. Peters(2002). “ Collaborative activities and multi-tasking in dialogue systems ” , Traitement Automatique des Langues (TAL) , 43 (2))。不能将其自

身附加到最近有效节点的对话移动可以附加到另一个分支内的有效节点（对应于所恢复的会话），或者通过将其自身附加到根节点来打开新的分支（对应于新的会话线程）。DMT 108 还作为用于对片段、多话语构造和修正进行解释的语境，并且为诸如 NP 求解的任务提供论述结构。

[0044] AT 110 对关于对话的行为进行管理。当用户发布命令时，这通常导致创建新的行为并且将其添加到 AT 110。在可以将行为实际发送到设备来执行之前，RNDS 200 尝试对其进行“求解”，例如，对所有提及的 NP 进行求解或者产生子对话以得出进一步的信息。典型地，修正和校正（例如，“I meant/said...”）包括对现存行为表示进行编辑。在 AT110 上对行为执行进行监控，并且变化可以导致生成诸如关于失败或者成功完成任务的通知消息。

[0045] 如上所述，经由设备管理器 220 和设备 API 222 将 AT 110 连接到 RNDS 200 的设备 250 的 AM 230。AM 230 对设备专用信息进行封装，AM 230 是与 RNDS DM 100 接口的代理或设备的能力的声明性说明，并且 AM 230 包括语言信息，诸如从 predicate(谓词)/argument(变元) 结构到设备行为的映射。当给定具有丢失变元的用户命令时，标记为“required”的变元可以生成子对话。

[0046] 可以将设备封装方法以及特别是这里所描述的对话移动脚本语言和 NP 求解规则应用到初始域，例如，对 MP3 音乐播放器进行控制并且访问音乐数据库。图 5 是根据实施例用于 MP3 设备的示例行为模型（“AM”）530 的一部分。如下所述，该 AM 530 的“required”变元位置包括“Playable”，其对应于来自与该应用相关联的对象的相关联本体的类；“playable-object”是通过对对话移动进行匹配所填充的变量名，如下所述。

[0047] 对话管理器可以完全从程序上对对话移动进行编码（例如，用 Java）。对话管理器还可以包括对应于在面向行为对话中所找到的对话贡献类型的通用目对话移动（例如，“Command”、“WhQuestion”、“WhAnswer”等）库。随着将对话管理器应用到新的应用，将新的对话移动实现为适合于新的应用，或者对现存的对话移动进行提炼以便应用到新的应用。以这种方式实现多种应用。

[0048] 因此，典型地，将对话移动定制到新的域需要大量的编码或重新编码。此外，使用现有的具有宽覆盖范围语法的解析器、或者使用语料训练统计解析器需要对话管理器能够对新的输入语义形式进行处理。宽覆盖范围的需求指示从输入到正确对话移动的映射是可以轻易扩展的。对覆盖范围进行扩展的一种方法是相对于广泛的语言本体（例如，WordNet（见 Miller, G A., 1995, "WordNet :A lexical database for English" , Comm. of the ACM 38）或者其它知识库（见 Dzikovska, M., 2004, "A Practical Semantic Representation for Natural Language Parsing" , Ph. D. Thesis, University of Rochester）对语义信息进行标准化。然而，这仍然需要将输入形式映射到内部表示。

[0049] 实施例的用于编写 DMS 102 的对话脚本语言提升了对话移动的重用、增强了扩展性、并且处理多个域上的语义变化。实施例的对话脚本语言定义了对话移动的分等级定义，允许对现存对话移动的继承和重用，同时允许对特定域或设备的定制。实施例的对话脚本语言还定义了从输入语义形式到适当对话移动的直接映射。此外，对话脚本语言为信息更新提供了附加规则。另外，对话脚本语言提供了其它对话移动专用信息，例如，为消除疑义生成的输出说明、对所需信息的请求。

[0050] 使用可轻易扩展的 DMS 102 与诸如如上文所述使用本体或知识库的使用其它方法实现宽语义覆盖范围一致。然而,它另外提供了通用方法,用于将应用专用信息提供给 RNDS DM 100,将其定制到新的域,并且使能这里所描述的 RNDS200 的即插即用多设备体系结构。

[0051] 图 6 是根据实施例用于 MP3 设备的“play”命令的示例对话移动脚本 (“DMS”) 602。DMS 602 中的变量对应于用于相应设备的 AM 中的变量。特别地,用于 MP3 设备的 AM 包括具有相应 (所需要的)“_playable-object”变元的“play”操作。当输入语义形式与该 DMS602 中的“Input”模板匹配时,统一操作 (unification operation) 填充“_playable-object”变量,其对来自设备对象域的对象进行求解 (下文对其进一步描述),并且在从设备 AM 构建的行为中填充对应的空隙 (slot)。随后是对对话移动脚本语言的属性的进一步细节的说明。虽然该示例 DMS 602 是与特定类型的设备一起使用的特定命令,但是这里所描述的实施例不限于该命令或者该设备。

[0052] 实施例的对话移动脚本语言允许对对话移动的分等级说明和细化。示例 DMS 602 对应于“play”命令,并且继承更一般的“Command”对话移动。例如,在 Java 中实现“Command”对话移动,其中对应的 DMS 具有对实现该对话移动的 Java 类命名的字段。通过相同的通用代码来实现“play”命令移动,但是“play”命令移动为触发该移动指定了它自己的模式,并且定义了附加模式和适当的生成消息。通常,继承的深度是没有界限的 (例如,可以定义在非常专用语境中应用的“play”命令移动的子移动)。

[0053] DMS 的继承对于其特别有用的一种类型的移动是在设备上的信息查询移动。关于 MP3 数据库中音乐或者城市信息向导中餐馆的问题在结构上通常是类似的,即查询结构自身是 (相对) 域独立的。可以通过不同的对话移动 (对应于不同的设备或知识源) 处理每种类型的查询,但是可以从单独一个“Query”对话移动中继承每个“Input”集。

[0054] 可以在对话移动的抽象级别上应用的其它操作包括“重写规则”,在将输入形式与对话移动“Input”模板进行匹配之前使用该“重写规则”对输入形式进行转化,例如,将间接命令转化成直接命令,或者以适当的语义特征替代时间标记 (例如,“now”)。重写规则是独立于域 / 设备的,并且在对任何特定设备或对话移动进行选择之前将其应用于输入。

[0055] 实施例的 DMS 还经由语义模板支持对话移动的选择。DMS 的“Input”部分包括触发该特定对话移动的输入项目的列表。将这些模板与解析器的输出进行匹配 (所述解析器为,在示例 DMS 602 的情况下,在从与 MP3 播放器交互的用户的 Wizard of Oz 实验中收集的语料上训练的统计解析器 (见 Cheng, H., H. Bratt, R. Mishra, E. Shriberg, S. Upson, J. Chen, F. Weng, S. Peters, L. Cavedon, J. Niekrasz, 2004, “A Wizard of Oz framework for collecting spoken human-computer dialogs”, INTERSPEECH; 8th International Conference on Spoken Language Processing, Jeju Island, Korea))。在与“Input”模板进行匹配之前,可以以任何方式 (例如,使用本体,或者经由重写规则) 对解析后的形式进行标准化或处理。可以将“Input”模板附加到域专用对话移动或者通用移动 (并且是继承的)。

[0056] 在示例 DMS 602 中的“Input”模板的特定形式记法说明了对特定统计解析器的输出进行匹配的特定例子;不同的解析器输出形式记法将使用不同的形式记法,但是一般而言,可以将模板视为特征结构,并且匹配操作是有效的单向统一。将实施例的符号解释如

下,但是不限于此 :“#”表示词汇项,词汇项具有在“/”符号之后的词性标签 ;“|”表示可替换项 ;“?”表示可选变元 ;“_”表示来自行为模型的变量匹配 ;并且“*”代表与任何事物的匹配。因此,在示例 DMS602 中的对话移动与“play X”、“start X”、或者包括“play X”或“hearX”的间接命令(为方便起见,间接命令使用通用重写规则提取它们的嵌入语句)匹配;将要播放的对象标记为可选的(即,甚至当该变元丢失时,模板也匹配)。

[0057] 因为可以替换模板的形式和相应的匹配算法而不影响 RNDS DM 体系结构,所以 RNDS DM 100 是中性表示。这使得能够轻易地将解析器或 NLG 组件替换成使用不同表示的组件。例如,可以替换更加标准的特征结构表示和特征统一算法,而无需对 RNDS DM 代码的其它变化。

[0058] 当输入形式与对话移动的“Input”部分中的条目匹配时,这可能导致变量受限制;特别地,可以将变量限制为对应于来自 AM 的变量。例如,如果与示例 DMS 602 匹配的输入包括合适的“arg”变元,那么这为“_playable-object”提供了值;如果不存在“arg”,那么将该变量剩余为未填充(在该情况下,“Command”对话移动可以生成对信息的请求)。

[0059] 通常,由于一般存在多个脚本对话移动和在每个移动的“Input”部分中的多个条目,所以多种匹配是可能的。实施例的 RNDS DM 100 使用一般标准对每个可能的匹配进行评分(例如,当前语境的可用性;未求解信息的最小化)。然而,可选实施例可以包括概率方法,以便对来自多个标准的证明进行合并以选择适当的对话移动,其包括韵律信息和浅主题分类。

[0060] 实施例的对话脚本语言提供了指定附加规则的机制。这些附加规则确定了可以附加到 DMT 中现存有效节点的对话移动的类型。例如,示例 DMS 602 示出了可以将消除歧义的“WhQuetsion”或用于填充丢失变元的“WhQuestion”附加到“Command”节点(通常在更加抽象的级别上而不是在专用命令的级别上指定该附加规则)。在这些说明书之一内部,可以在“适当位置”脚注对话移动信息(如为“WhQuetsion :fill :play”移动所做的那样)。

[0061] DMS 还在“CloseOn”字段内对哪些邻近移动关闭对话移动(即,使对话移动无效,从而没有其它移动可以附加到其上)进行编码。关闭用于附加的节点有效地关闭了相应的会话线程(修正可以重新打开“Command”或“Query”节点)。节点还在特定周期之后自动关闭。

[0062] 自动生成大部分 RDNS 输出,例如,在通用目的对话移动中对其进行编码。然而,应用可以要求域和设备专用输出。也可以在实施例的 DMS 102 中对这些域和设备专用输出进行编码,由于这些输出将是系统响应,所以在“System”对话移动中对其进行编码。DMS 102 允许这些特定输出的任何表示,只要该表示与 RNDS DM 100 针对给定应用所接口的特定 NLG 系统 210 所使用表示匹配。以这种方式,因为在对话系统中使用不同的 NLG 组件(使用不同的表示)仅导致对 DMS 102 的修改,并且不需要对 RNDS DM 100 的核心进行修改,所以 RNDS DM 100 是中性表示。

[0063] 因此,RNDS 200 提供了强大的实际核心对话管理过程与域专用信息的轻松脚本化定义的合并,域专用信息例如对话移动和 NP 求解映射。这样,可以将 RNDS 200 定制到新的域和应用,并且 RNDS 200 还提供用于如上所述的即插即用多设备对话管理的工具。该即插即用能力是系统的重要特征,所述系统可以使它们的功能扩展而不脱机。典型地,即插即用包括将提供增强功能的新的组件添加到 RNDS 200,而不破坏现存的框架。实施例的 RNDS

200 通过使用说明语言实现即插即用环境以及对组件实现的封装,其中组件使用所述说明语言公告它们的能力。

[0064] 即插即用多设备对话管理支持将新的设备添加到 RNDS 200,其与 RNDS DM 100 一起作用,不必下载新的软件或者修改 RNDS DM100。当将新的设备插入 RNDS 200 中时,新的设备将其接口信息广播到 RNDS DM 100。例如,新的设备可以为 RNDS DM 100 提供其实现的功能(行为模型)、其处理的语言(设备理解的来自用户的事物类型)、任何设备专用 DMS 232、任何设备专用 NPRS 234(如果系统中存在对有关设备的对象的描述,那么设备告知 RNDS DM 100 如何将对象翻译成对设备 KB 的查询)、以及设备专用核心 Java 代码的信息。

[0065] 参考图 2,向 RNDS DM 100 注册的新设备对 RNDS DM 100 在管理与这些新设备的对话中使用的所有所需信息进行封装。该封装信息包括实施例中的四个组件,其中,这四个组件包括如上所述的 DMS232、对可通过对话访问的任何设备功能进行描述的 AM 230、设备专用本体和 / 或知识库、以及用于设备专用 NP 求解的规则 234。可选实施例的封装信息可以包括适合于设备和 RNDS DM 100 的附加的或者不同的信息。

[0066] 通过将新的 Java 类包括到适合于设备的封装信息中,还可以添加对话管理过程的对话专用实现,或者使用该专用实现重写通用实现。例如,可以添加对由新的设备引入的新交互形式进行处理的对话移动。然而,通常,组成封装信息的四个组件包括 RNDS DM 100 为了与设备进行对话所使用的全部设备专用信息,并且允许对话使能设备的动态即插即用。

[0067] 在提供多设备对话管理中,实施例的 RNDS DM 100 对 DMT 体系结构进行扩展,以允许将新的设备动态添加或者插入 RNDS 200。一旦添加,新的对话使能设备就将它们自己向 RNDS DM 100 注册,并且 DMT 108 中的节点与适当的专用设备相关联。同样,“当前设备”成为信息状态的组成部分,并且在该语境中对输入话语进行解释。

[0068] RNDS DM 100 还将设备选择(即,确定话语与那个设备相关联)实现为多设备对话管理的组件过程。RNDS DM 100 的设备选择判决过程包括词汇和语义信息、对话移动分类和论述(discourse)结构,以及到“当前设备”的偏离。判决过程还使用将 NP 与适当的设备本体进行相关的策略(例如,指称“song”将与关联于 MP3 设备但可能不关联于其它设备的本体内的类型匹配)。然而,该策略不一定解决所有的设备模糊,例如,电话设备(“get John on the phone”)以及导航服务(“how do I get to John’s house ?”)都可以使用地址本。

[0069] 因此,因为与 NP 求解有关的信息提供了与所提到的设备有关的重要线索,所以 RNDS DM 100 将判决选择和 NP 求解过程相关为相互依存的过程,但是如下所述,NP 求解实际上可以是完全设备专用的。这样,为了对设备进行识别,RNDS DM 100 执行浅 NP 分析(例如,将名词和合适的名字与关联于特定设备的本体类型和 KB 项进行匹配),并且随后使用下述的设备专用 NP 求解规则对 NP 进行完全求解。然而,可选实施例可以为这种分类任务使用其它特征(例如,浅主题分类技术)和 / 或概率方法。

[0070] 可以将大部分 NP 求解过程视为相当独立于域的(例如,首语重复求解)。然而, NP 求解的方面是同时取决于域和设备的。例如,在 MP3 播放器音乐播放的语境中,与当使用触摸屏多模态界面时相比,对短语“What’s this”进行不同解释。RNDS DM 100 以类似于

DMS 102 定制对话移动的方式,通过经由 NP 求解脚本语言和 NPRS 104 对在 RNDS DM 100 中为特定域 / 设备实现的核心 NP 求解能力进行适配定制,对此进行处理。NPRS 104 是有效的语境无关语法,其允许用户在当前对话信息状态和输入语义形式的语境下,对如何将 NP 对象映射到用于特定设备的知识库查询进行定义。特别地,对于 MP3 设备,例如,在“What’s this”语境中的“this”将被映射到一个查询,该查询返回当前正播放歌曲的名称。

[0071] RNDS DM 100 使用 NPRS 104 将 NP 翻译成实现基于约束的 KB 查询(在这里被称为“名词短语查询”或“NP 查询”)的 Java 对象。图 7 是根据实施例的对于“the song Vertigo by U2”的示例名词短语查询(“NP 查询”)对象 700。RNDS DM 100 的规则指定如何将在输入语义形式中指定的 NP 翻译成这种对象。RNDS DM 100 包括多个通用约束对象和相关联的变形,但是还可以将更多的对象包括为设备封装的组成部分,以便提供专用于该设备的任何新颖处理。例如,MP3 设备将需要对诸如“by artist”的限定词进行处理,并且知道将该构造翻译成 KB 的“artist”字段上适当的约束。

[0072] 当然,将信息从 NP 表示中提取出来的方式取决于输入的特定格式以及与设备相关的 KB 的结构。RNDS DM 100 使用基于规则的语言,来指定 NP(不管格式)如何映射到基于约束的查询对象,使用通用或设备专用框架构造操作。将这种规则用于处理同义词(即,通过将多个名词映射到相同的查询类型)以及指定来自复杂 NP 的复杂查询对象的构造。

[0073] 图 8 是根据实施例的名词短语求解(“NP 求解”)规则 800 的例子。从用于 MP3 设备的 NP 求解脚本中获得这些 NP 求解规则 800,但是不限于此。每个规则的左侧与来自在该特定应用中使用的统计解析器输出的构造匹配:符号“#”指示已将相应的词标记为头词;“/”之后的记号是所匹配项的词性(“POS”);并且大写的条目指明变量。每个规则的右侧指定如何为 KB 构造查询或约束:第一个项表示这是何种类型的约束(其确定使用何种构造过程);右侧的剩余部分指定进行测试的特定 KB 字段。

[0074] 参考 NP 求解规则 800,(1) 为“Song”类型对象构造简单查询;(2) 使用 MP3 专用约束构造过程,对标记为“this”(将其映射到当前正在播放的歌曲)的“Song”进行匹配;(3) 将“by X”补语映射到在“Artist”上的约束(对于歌曲或曲集),其中,“FRAME”变量指示所嵌入的 NP 可以是复杂的,并且应该对其自身进行分析;(4) 对“namedX”形式的嵌入语句进行处理;以及(5) 对其它可以限定歌曲查询的补语进行处理。由于一个规则应用的输出可以是另一个规则的输入,所以规则可以是递归的。

[0075] 虽然实施例的 NP 求解规则构造不取决于规则的特定语法或格式,但是可以为给定的新设备轻松地指定 NP 求解规则构造,并且将其与设备一起进行封装。由于具有 DMS 102,所以可以在设备专用脚本中对通用构造进行继承和重写,而可以将设备专用 NP 解释规则与给定设备一起进行封装。

[0076] 这里所描述的 RNDS 200 提供了对多设备会话、对话移动脚本、以及脚本功能的管理,以便对其他核心对话处理(诸如 NP 求解)进行扩展以覆盖域专用现象,其中,所述对话移动脚本有助于到新的域更大的可移植性,并且还使多设备对话成为可能。还对 RNDS 200 内部的数据结构和处理过程进行增强,以便对不同设备之间的会话切换进行处理。对这些特征进行打包,以便有助于对 API 进行编程,API 使能将新的设备动态“即插即用”到 RNDS DM 100 中。

[0077] 实施例的 RNDS 包括一种方法,该方法包含以下操作的至少一个:对从声信号产生

的包括来自讲话者的对话贡献的描述的输入模式进行接收；使用对话移动脚本 (DMS) 对该描述和该描述的至少一个参数进行识别，其中，DMS 对应于设备和应用中至少一个；并且使用 DMS 将该描述映射到对话移动，该对话移动对应于所识别的参数并且独立于设备和应用。

[0078] 实施例的方法还包括：通过对 DMS 重新编程，使对话移动进行改编，以使其适合于在多个不同域和多个不同应用的至少一个上使用。

[0079] 实施例的方法还包括以下操作的至少一个：使用该描述的名词短语对与该描述对应的设备和应用的至少一个进行识别；选择对应于所识别设备和应用中至少一个的名词短语求解脚本 (NPRS)，并且经由所选择的 NPRS 访问名词短语求解规则；以及使用名词短语求解规则对该描述进行翻译，并且生成对应于设备和描述中至少一个的查询。

[0080] 实施例的方法还包括：响应于该查询，从至少一个数据库中检索由来自对话贡献的对象描述所指称的至少一个对象。实施例的对象描述包括名词短语。

[0081] 实施例的方法还包括：通过对 NPRS 进行重新编程，对至少一个名词短语求解规则进行改编，以使其适合于在多个不同域和多个不同应用的至少一个上使用。

[0082] 实施例的 NPRS 包括知识库查询构造规则，可以对其进行编程以便定义如何为多个设备中的每个设备将名词短语对象映射到查询。

[0083] 实施例的生成查询包括将该描述的名词短语翻译成对象。

[0084] 实施例的该描述是基于结构化语言的描述，其包括对话贡献的语义描述、句法描述、音位信息、话语级别信息、以及主题分类中的至少一个。

[0085] 实施例的映射还包括对对话贡献的至少一个参数进行识别。

[0086] 实施例的方法还包括对该描述的变量和对话移动组件之间的匹 配进行评分。

[0087] 实施例的 RNDS 包括一种系统，该系统包含以下组件的至少一个：基于信息状态更新方法的对话管理器，其包括多个对话移动和对应于设备和应用中至少一个的对话移动脚本 (DMS)，该话管理器对从声信号生成的输入模式进行接收，该输入模式包括对来自讲话者的对话贡献的描述，该 DMS 对该描述和该描述的至少一个参数进行识别，该 DMS 将该描述映射到对应于所识别参数的一个对话移动；连接到对话管理器和至少一个数据库的知识管理器；以及连接到对话管理器的设备管理器。

[0088] 通过对 DMS 重新编程，对实施例的多个对话移动进行改编，以使其适合于在不同域和应用的至少一个上使用。

[0089] 实施例的对话管理器还包括连接到多个名词短语求解脚本 (NPRS) 和多个规则的名词短语求解器，其中，每个 NPRS 对应于设备和应用中的至少一个。

[0090] 实施例的名词短语求解器使用 NPRS 来访问规则，并且对所识别的描述进行翻译，以生成对应于设备和应用中至少一个的查询。

[0091] 通过对 NPRS 重新编程来改编实施例的规则，以使其适合于在至少一个不同域和应用上使用。

[0092] 实施例的 NPRS 包括知识库查询构造规则，可对其进行编程以便定义如何将名词短语对象映射到用于每个设备的查询。

[0093] 实施例的系统还包括至少一个数据库，其中，知识管理器将查询传送到数据库。响应于该查询，实施例的对话管理器从数据库接收对话贡献的至少一个对象。

[0094] 实施例的系统还包括多个设备,其中,将每个设备通过符合设备应用编程接口 (API) 的每个设备的描述连接到设备管理器,其中,每个设备描述包括行为模型、设备专用 DMS、设备专用 NPRS、以及实现更多设备专用功能的代码中的至少一个。实现更多设备专用功能的实施例的代码包括设备专用对话移动。实施例的行为模型包括对应于 DMS 变量的变量。

[0095] 响应于将设备连接到设备管理器,实施例的对话管理器对来自设备 API 的行为模型、设备专用 DMS、设备专用 NPRS、以及设备专用对话移动的信息进行接收和存储。

[0096] 实施例的系统还包括被连接到对话管理器输入的语音识别模块和语言理解模块中的至少一个。

[0097] 实施例的系统还包括被连接到对话管理器输出的语言发生模块和文本到语音模块中的至少一个。

[0098] 实施例的 RNDS 包括对话使能设备,其包括以下装置中的至少一个:用于对从声信号生成的输入模式进行接收的装置,所述输入模式包括对来自讲话者的对话贡献的描述;用于使用对话移动脚本 (DMS) 对该描述和该描述的至少一个参数进行识别的装置,其中, DMS 对应于设备和应用中的至少一个;以及用于使用 DMS 将描述映射到对话移动的装置,该对话移动对应于所识别的参数并且独立于设备和应用。

[0099] 实施例的 RNDS 包括计算机可读介质,该计算机可读介质包括可执行指令,当在处理系统中执行该指令时,通过以下操作管理对话:对从声信号生成的输入模式进行接收,所述输入模式包括对来自讲话者的对话贡献的描述;使用对话移动脚本 (DMS) 对该描述和该描述的至少一个参数进行识别,其中, DMS 对应于设备和应用中的至少一个;以及使用 DMS 将该描述映射到对话移动,该对话移动对应于所识别的参数并且独立于设备和应用。

[0100] 实施例的 RNDS 包括一种方法,该方法包含以下操作中的至少一个:基于信息状态更新方法提供对话管理系统中的多个通用对话脚本,其中,每个对话脚本包括对例程的扩展,其中,该扩展提供专用于域、设备和应用中至少一个的功能;对对话管理系统的输入进行接收,该输入包括对对话贡献的描述;以及使用至少一个通用对话脚本将所接收的输入映射到多个核心组件中的至少一个,其中,核心组件包括对话管理组件,其独立于域、设备和应用中的至少一个。

[0101] 实施例的通用对话脚本包括对话移动脚本 (DMS) 和名词短语求解脚本 (NPRS)。

[0102] 实施例的方法还包括以下操作:对另外的通用对话脚本进行接收,其中,另外的通用对话脚本对与对话管理系统交互的域、设备和应用中的至少一个进行扩展。实施例的另外的通用对话脚本是从连接到对话管理系统的新的设备和新的应用中的至少一个中接收的。

[0103] 可以将这里所描述的 RNDS 的方面实现为编程于各种电路中的功能,各种电路包括可编程逻辑器件 (PLD),诸如现场可编程门阵列 (FPGA)、可编程阵列逻辑 (PAL) 器件、电可编程逻辑、存储器器件、基于蜂窝的标准器件、以及专用集成电路 (ASIC)。用于实现 RNDS 的方面的一些其它可能包括:具有存储器的微控制器(例如电可擦除可编程只读存储器 (EEPROM))、嵌入式微处理器、固件、软件等。此外,可以将 RNDS 的方面嵌入微处理器,该微处理器具有基于软件的电路仿真、离散逻辑(顺序和组合)、用户定制器件、模糊(神经)逻辑、量子器件、以及以上类型器件的任何混合。当然,可以以多种组件类型提供以下器件技术,例如,如互补金属氧化物半导体 (CMOS) 的金属氧化物半导体场效应晶体管 (MOSFET) 技

术、如射极耦合逻辑 (ECL) 的双极技术、聚合技术 (硅共轭聚合体和金属共轭聚合体金属结构)、混合模拟和数字等。

[0104] 应该注意,可以将这里所公开的各个组件描述和表示 (或者代表) 为嵌入在各种计算机可读介质中的数据和 / 或指令。可以将该数据和 / 或指令嵌入在其中的计算机可读介质包括各种形式的非易失存储介质 (例如,光、磁或半导体存储介质) 和载波,但是不限于此,其中,可以使用载波通过无线、光、或者有线信号传输介质或者其任何组合传送该格式化数据和 / 或指令。通过载波对该数据和 / 或指令进行传送的例子包括在因特网和 / 或其它计算机网络上经由一个或多个数据传送协议 (例如,HTTP、FTP、SMTP 等) 进行传送 (上载、下载、电子邮件等),但是不限于此。当在计算机系统内经由一个或多个计算机可读介质进行接收时,可以通过计算机系统内与执行一个或多个其它计算机程序相结合的处理实体 (例如,一个或多个处理器) 对上述组件的这种基于数据和 / 或指令的表示进行处理。

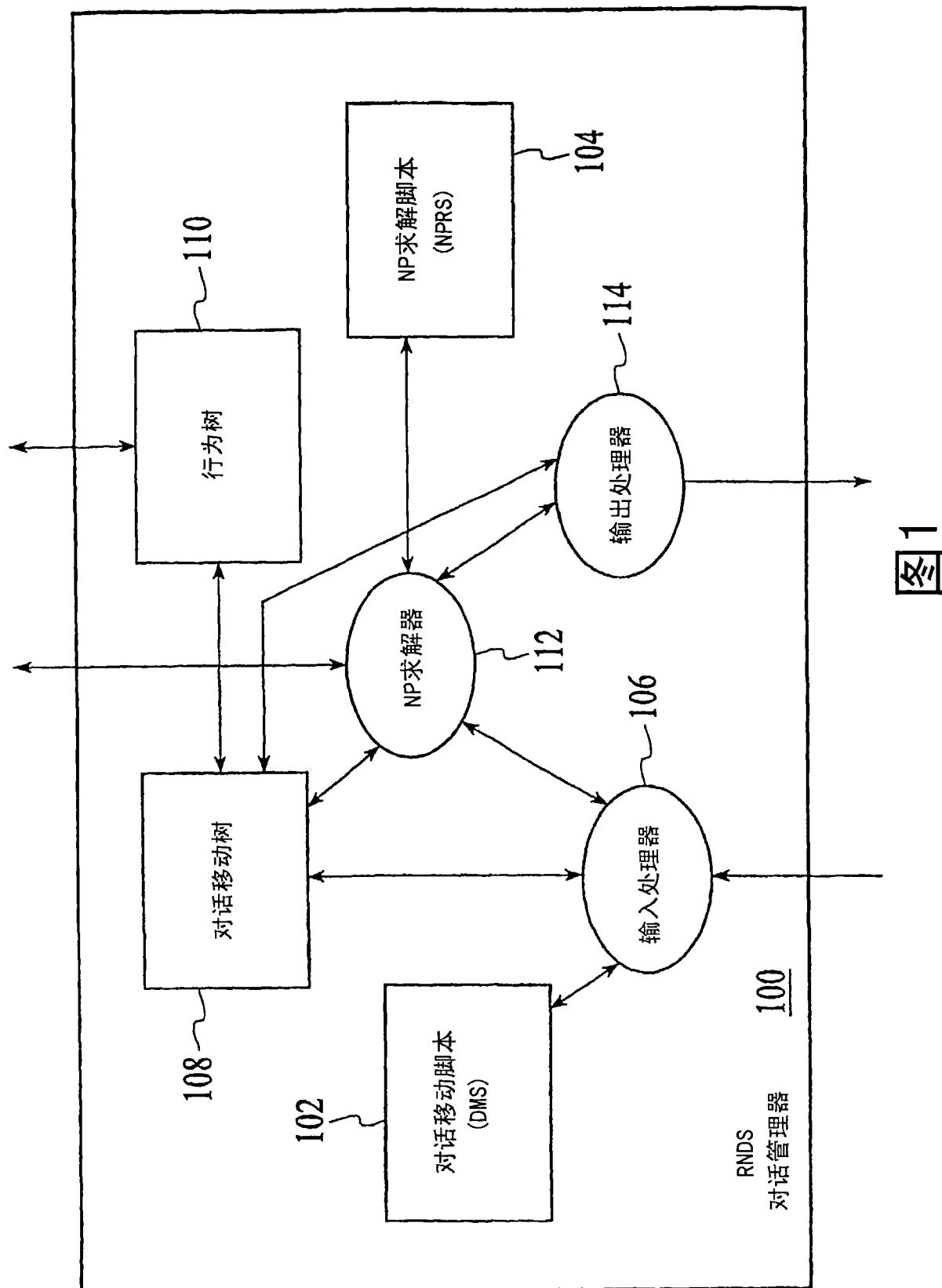
[0105] 除非上下文明确要求,否则,在整个说明书和权利要求书中,将词语“包括”、“包含”等解释为包含在内的意义,这与排除在外或者 穷尽的意义相对立;也就是说,其具有“包括,但不限于”的意义。使用单数或复数的单词分别包括复数或单数个。另外,单词“在这里”、“在此之下”、“上文”、“下文”以及具有类似意思的词是将该申请作为一个整体,而不是该申请的任何特定组成部分。当涉及两个或多个项目的组而使用词“或”时,该词覆盖了下列所有对该词的解释:该组内任何项目、该组内所有项目、以及该组内项目的任何组合。

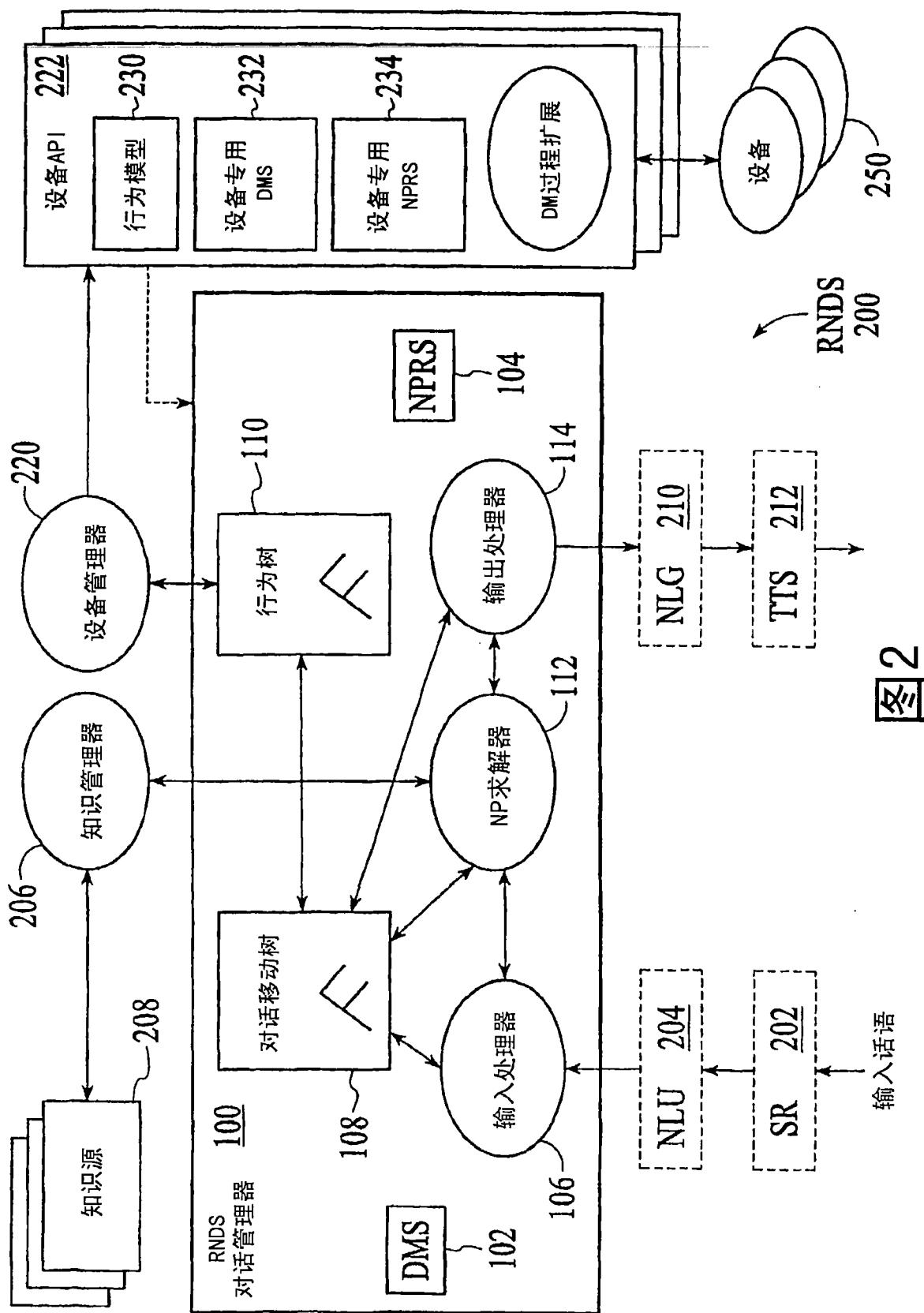
[0106] 上文对所说明的 RNDS 实施例的描述不是想要穷尽所有实施例或者将 RNDS 限制在所公开的确切形式中。虽然这里为了说明性目的对 RNDS 的特定实施例和例子进行了描述,但是如本相关领域的技术人员所认识到的,在 RNDS 范围内的各种等价修改是可能的。在这里所提供的 RNDS 教义可以应用到其它处理系统和方法,而不仅仅用于上述系统和方法。

[0107] 可以对上述各个实施例的元素和行为进行合并以提供更多的实施例。可以根据上述说明对 RNDS 进行这些和其它变化。

[0108] 通常,在后面的权利要求中,不应该将所使用的术语解释为将 RNDS 限制在说明书和权利要求中所公开的特定实施例,而应该将其解释为包括在权利要求下操作的所有处理系统。因此,公开的内容不是要对 RNDS 进行限制,相反,RNDS 的范围完全由权利要求确定。

[0109] 虽然在下文的某些权利要求形式中呈现了 RNDS 的某些方面,但是发明者在任何数目的权利要求形式中预期 RNDS 的各个方面。例如,虽然仅将 RNDS 的一个方面叙述为嵌入在机器可读介质中,但是可以将其它方面同样嵌入在机器可读介质中。因此,发明者保留了在提交申请之后添加额外权利要求的权利,以便为 RNDS 的其它方面寻求这种额外的权利要求形式。





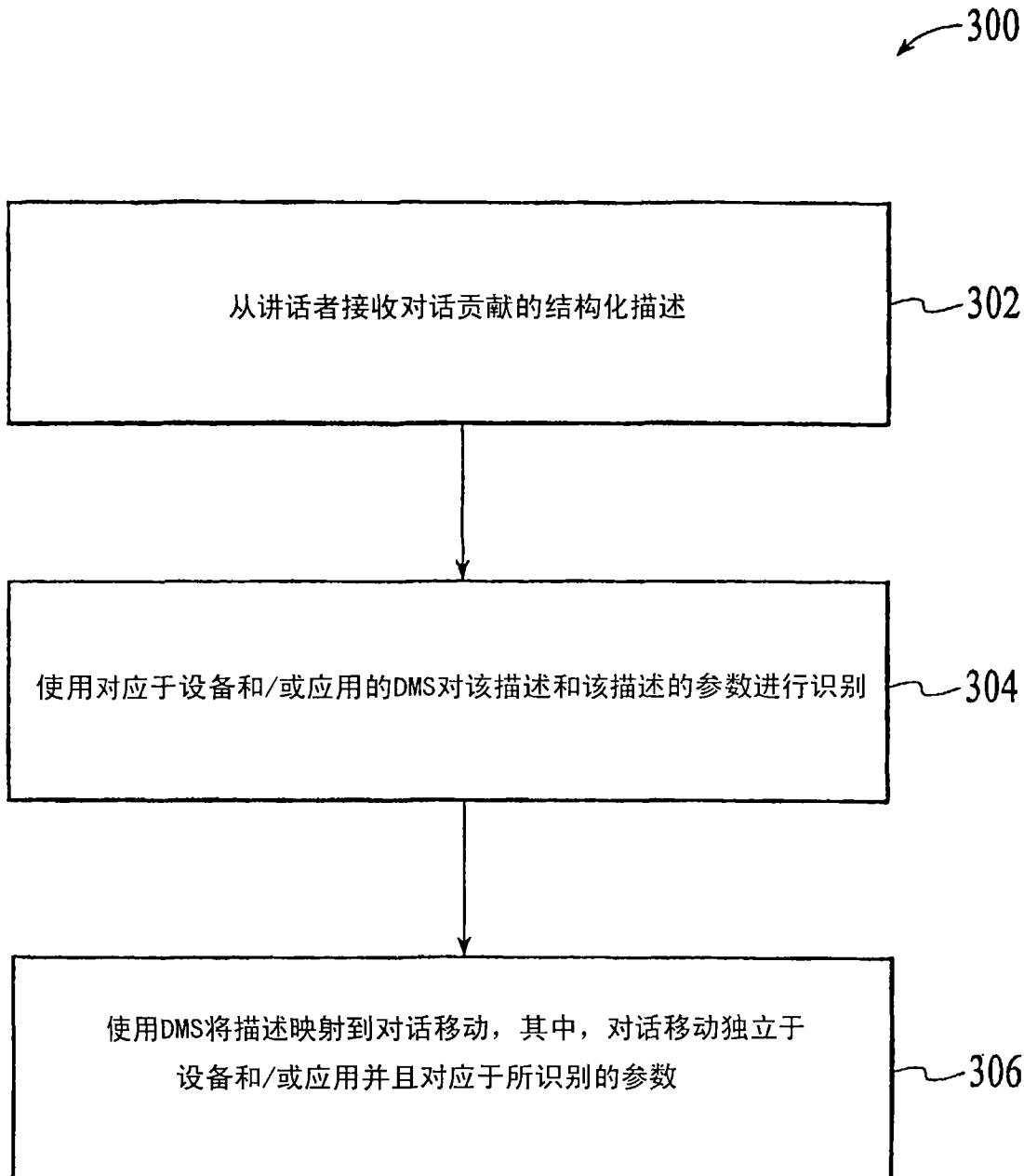


图3

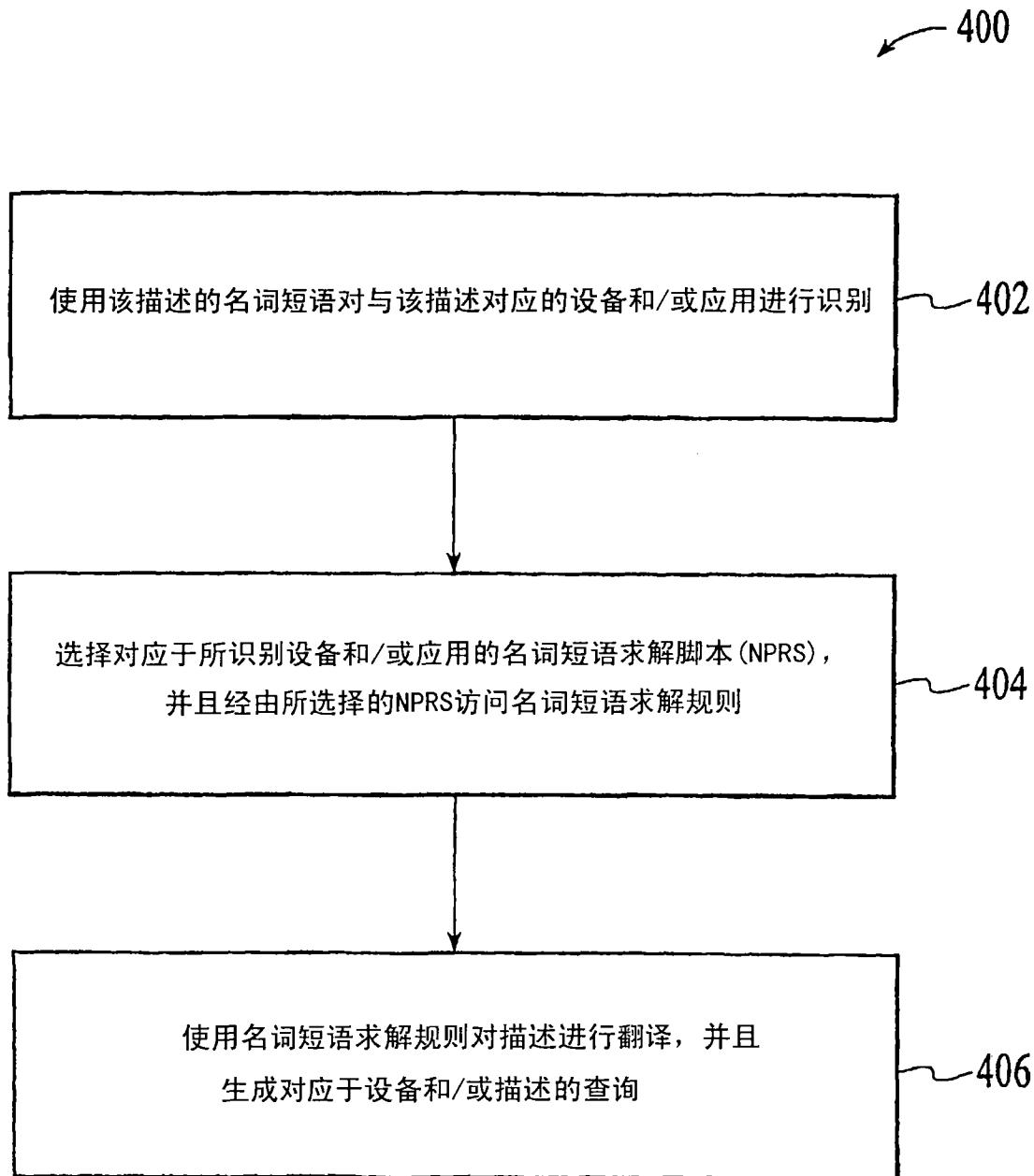


图4

530 ↗

```
Types {
    Playable;
    ...
}

Slots {
    Playable playable-object;
    ...
}

// 任务定义
taskdef<play, "play"> {
    DefinableSlots {
        required Playable playable-object;
        optional ... // 可选变元 (例如, volume)
    }
    ...
}
```

图5

602

```

User Commands: play {
    // 从通用Command对话移动继承
    Description "play something"
    Input {
        // "play/start X"
        "s ( features (mood (imperative ) )
            predicate (#play/vb| #start/vb) ,
            ? arglist (obj :_,?sbj :* ) ) "
            // " I want to play/hear X "
        "s ( features (mood (indirect ) ),
            predicate (#play/vb| #hear/vb) ,
            ?arglist (obj:_ playable-object; ,?sbj:*) ) "
            // 其他模板...
    }
    producing { // Questions
        System WHQuestion: disambiguate
        System WHQuestion: fill :play:_ playable-object; {
            Output { avs "(el / play
                            :question (ql /what)
                            :agent !) "
                    }...
            CloseOn System Report: play: playing {
                Output { avs "(el / play
                                :patient (pl / [song] )
                                :aspect continuous) "
                    }
                ...
            }
        ...
    }
}

```

图6

700
←

KbFrame : Device : MP3

Simple Constraint : Type = Song

Simple Constraint : Name = Vertigo

Frame Constraint : Artist =

KbFrame : Device : Device : MP3

Simple Constraint : Name = U2

图 7

800 ↙

1. #song --> Simple system:hasCategory : music:Song
"What *songs* do you have?"
2. #this/dt --> MP3Specific this music:Song
"What's *this*?"
3. ncomp(#by/in,sbj:FRAME) --> Frame music:songHasArtist|music:albumHasArtist
"Do you have anything *by* X?"
4. s(predicate(#called|#named),arglist(obj(#:WORD))) --> Simple system:hasName
"Do you have a song *called/named* X?"
5. ncomp(#on/in|#in/in|#from/in,spj:Frame) --> Frame music:hasSongList
"Play something *from* an album by X"

图 8