



[12] 发明专利申请公开说明书

[21] 申请号 02809113.2

[43] 公开日 2004年9月1日

[11] 公开号 CN 1526106A

[22] 申请日 2002.3.26 [21] 申请号 02809113.2

[30] 优先权

[32] 2001. 3. 26 [33] US [31] 60/278,930

[32] 2001. 7. 25 [33] US [31] 09/916,243

[86] 国际申请 PCT/US2002/009528 2002.3.26

[87] 国际公布 WO2002/077904 英 2002.10.3

[85] 进入国家阶段日期 2003.10.29

[71] 申请人 美国网上搜索公司

地址 美国加利福尼亚

[72] 发明人 大卫·瓦克泰尔

[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

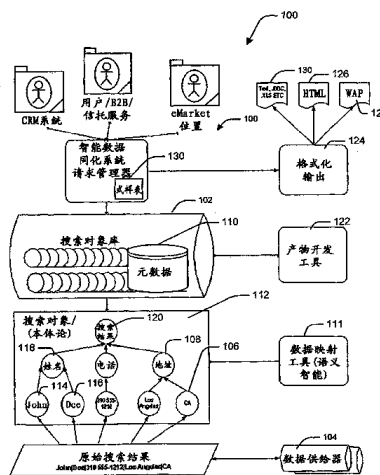
代理人 李玲

权利要求书 11 页 说明书 32 页 附图 18 页

[54] 发明名称 用于智能数据同化的方法和装置

[57] 摘要

本发明涉及一种智能数据同化系统，包括本体论描述、工作流和逻辑搜索对象。使用描述原子数据对象和语义对象的本体论，逻辑搜索对象可操作地连接到外部和内部数据供给器并返回搜索结果。通过工作流将语义对象组成合更大的语义结构以创建返回称为数据产物的搜索结果的个性化服务。通过能响应来自不同类型的数据客户端的服务请求的应用服务器来存取服务。图形用户界面提供创建逻辑搜索对象并将逻辑搜索对象集合成工作流和服务。



- 1、一种用于履行数据服务请求的方法，所述方法包括：
提供数据服务的本体论描述；
经第一通信链路，可操作地将第一逻辑搜索对象连接到数据供给器；
通过所述第一逻辑搜索对象，由所述数据服务请求生成搜索请求；
经所述通信链路，由所述第一逻辑搜索对象将所述搜索请求传送给所述数据供给器；
响应所述搜索请求，经所述通信链路，由所述第一逻辑搜索对象从所述数据供给器接收数据集；以及
使用所述本体论描述，由所述第一逻辑搜索对象生成知识实例。
- 2、如权利要求 1 所述的方法，进一步包括，所述第一通信链路与数据库服务器通信。
- 3、如权利要求 1 所述的方法，所述第一通信链路与 FTP 服务器通信。
- 4、如权利要求 1 所述的方法，进一步包括，所述第一通信链路与 Web 服务器通信。
- 5、如权利要求 1 所述的方法，进一步包括，所述第一通信链路 with 文件系统通信。
- 6、如权利要求 1 所述的方法，进一步包括，所述第一通信链路 with 人类数据供给器通信。
- 7、如权利要求 1 所述的方法，进一步包括，所述第一通信链路 with 专用于所述数据供给器的通信协议通信。
- 8、如权利要求 1 所述的方法，进一步包括，所述数据服务请求包括在 XML 文档中。
- 9、如权利要求 1 所述的方法，进一步包括：

可操作地将第一工作流连接到所述逻辑搜索对象；以及
由所述第一逻辑搜索对象将所述知识实例传送给所述第一工作流。

10、如权利要求 9 所述的方法，进一步包括：

可操作地将第二逻辑搜索对象连接到所述第一工作流；以及
由所述第一工作流封装所述第一和第二逻辑搜索对象间的所述本体论关系。

11、如权利要求 9 所述的方法，进一步包括：

由应用服务器，经第二通信链路，将所述数据客户端可操作地连接到所述第一工作流，以及

由所述应用服务器，经所述第二通信链路，从所述数据客户端接收数据服务请求消息，所述数据服务请求消息包括所述数据服务请求；

由所述应用服务器将所述数据服务请求消息传送给所述第一工作流；以及

由所述第一工作流将所述数据服务请求消息传送给所述逻辑搜索对象。

12、如权利要求 10 所述的方法，进一步包括，所述第二通信链路使用 SMTP 通信。

13、如权利要求 10 所述的方法，进一步包括，所述第二通信链路使用 JMS 通信。

14、如权利要求 10 所述的方法，进一步包括，所述第二通信链路使用 HTTP 通信。

15、如权利要求 10 所述的方法，进一步包括，所述第二通信链路使用 RMI 通信。

16、如权利要求 9 所述的方法，其中由所述第一工作流指定所述逻辑搜索对象。

17、如权利要求 11 所述的方法，其中使用所述服务请求消息，由所述应用服务器指定所述第一工作流。

18、如权利要求 10 所述的方法，进一步包括：

提供格式器；以及

由所述格式器将封装在所述知识实例中的所述数据集格式化为由所述数据客户端请求的格式。

19、如权利要求 9 所述的方法，进一步包括，将第二工作流可操作地连接到所述第一工作流。

20、一种用于经通信链路，由软件对象存取数据供给器的方法，包括：

由所述软件对象，从第二软件对象接收搜索请求消息文档；

通过所述软件对象，由所述搜索请求消息文档生成用于所述数据供给器的数据请求；

经所述通信链路，由所述软件对象将所述数据请求传送给所述数据供给器；

经所述通信链路，由所述软件对象从所述数据供给器接收数据集；以及

通过所述软件对象，由所述数据集生成语义对象。

21、如权利要求 20 所述的方法，其中通过所述软件对象，由所述数据集生成语义对象进一步包括：

可操作地将解析器适配器连接到所述软件对象；

提供所述数据集的解析器语义描述，用于由所述解析器适配器使用；

提供语义对象语义描述；

通过所述解析器适配器，使用所述解析器语义描述，从所述数据集生成抽取数据；以及

根据所述语义对象语义描述，通过所述解析器适配器，使用所抽取的数据，生成所述语义对象。

22、如权利要求 20 所述的方法，其中通过所述软件对象，由所述搜索请求生成用于所述数据供给器的数据请求进一步包括：

可操作地将请求编译器连接到所述软件对象；

可操作地将本机对象连接到所述请求编译器；

由所述本机对象封装用于所述数据供给器的数据请求的实施细节；

提供包括描述由所述请求编译器使用的数据结构的本体论信息的本机语义描述以便编译用于所述数据供给器的所述数据请求；

由所述请求编译器将所述搜索请求传送到所述本机对象；以及
通过所述本机对象，使用所述本机语义描述，由所述搜索请求生成所述数据请求。

23、一种用来履行数据服务请求的数据处理系统，包括：

处理器；以及

存储器，可操作地连接到所述处理器，并具有在此存储器中存储的程序指令，所述处理器可操作地执行所述程序指令，所述程序指令包括：

提供数据服务的本体论描述；

经第一通信链路，可操作地将第一逻辑搜索对象连接到数据供给器；

通过所述第一逻辑搜索对象，由所述数据服务请求生成搜索请求；

经所述通信链路，由所述第一逻辑搜索对象将所述搜索请求传送给所述数据供给器；

响应所述搜索请求，经所述通信链路，由所述第一逻辑搜索对象从所述数据供给器接收数据集；以及

使用所述本体论描述，由所述第一逻辑搜索对象生成知识实例。

24、如权利要求 23 所述的数据处理系统，其中所述第一通信链路与数据库服务器通信。

25、如权利要求 23 所述的数据处理系统，其中所述第一通信链路与 FTP 服务器通信。

26、如权利要求 23 所述的数据处理系统，其中所述第一通信链

路与 Web 服务器通信。

27、如权利要求 23 所述的数据处理系统，其中所述第一通信链路与文件系统通信。

28、如权利要求 23 所述的数据处理系统，其中所述第一通信链路与人类数据供给器通信。

29、如权利要求 23 所述的数据处理系统，其中所述第一通信链路与专用于所述数据供给器的通信协议通信。

30、如权利要求 23 所述的数据处理系统，其中所述数据服务请求包括在 XML 文档中。

31、如权利要求 23 所述的数据处理系统，所述程序指令进一步包括：

可操作地将第一工作流连接到所述逻辑搜索对象；以及
由所述第一逻辑搜索对象将所述知识实例传送给所述第一工作流。

32、如权利要求 31 所述的数据处理系统，所述程序指令进一步包括：

可操作地将第二逻辑搜索对象连接到所述第一工作流；以及
由所述第一工作流封装所述第一和第二逻辑搜索对象间的所述本体论关系。

33、如权利要求 31 所述的数据处理系统，所述程序指令进一步包括：

由应用服务器，经第二通信链路，将数据客户端可操作地连接到所述第一工作流，以及

由所述应用服务器，经所述第二通信链路，从所述数据客户端接收数据服务请求消息，所述数据服务请求消息包括所述数据服务请求；

由所述应用服务器将所述数据服务请求消息传送给所述第一工作流；以及

由所述第一工作流将所述数据服务请求消息传送给所述逻辑搜

索对象。

34、如权利要求 32 所述的数据处理系统，其中所述第二通信链路使用 SMTP 通信。

35、如权利要求 32 所述的数据处理系统，其中所述第二通信链路使用 JMS 通信。

36、如权利要求 32 所述的数据处理系统，其中所述第二通信链路使用 HTTP 通信。

37、如权利要求 32 所述的数据处理系统，其中所述第二通信链路使用 RMI 通信。

38、如权利要求 32 所述的数据处理系统，其中由所述第一 workflow 指定所述逻辑搜索对象。

39、如权利要求 33 所述的数据处理系统 1，其中使用所述服务请求消息，由所述应用服务器指定所述第一 workflow。

40、如权利要求 32 所述的数据处理系统，所述程序指令进一步包括：

提供格式器；以及

由所述格式器将封装在所述知识实例中的所述数据集格式化为由所述数据客户端请求的格式。

41、如权利要求 31 所述的数据处理系统，所述程序指令进一步包括，将第二 workflow 可操作地连接到所述第一 workflow。

42、一种用来经通信链路，存取数据供给器的数据处理系统，包括：

处理器；以及

存储器，可操作地连接到所述处理器，并具有在此存储器中存储的程序指令，所述处理器用来执行所述程序指令，所述程序指令包括：

由软件对象接收搜索请求消息文档；

通过所述软件对象，由所述搜索请求消息文档生成用于所述数据供给器的数据请求；

经所述通信链路，由所述软件对象将所述数据请求传送给所述数据供给器；

经所述通信链路，由所述软件对象从所述数据供给器接收数据集；以及

通过所述软件对象，由所述数据集生成语义对象。

43、如权利要求 42 所述的数据处理系统，其中用于通过所述软件对象，由所述数据集生成语义对象的所述程序指令进一步包括：

可操作地将解析器适配器连接到所述软件对象；

提供所述数据集的解析器语义描述，用于由所述解析器适配器使用；

提供语义对象语义描述；

通过所述解析器适配器，使用所述解析器语义描述，从所述数据集生成抽取数据；以及

根据所述语义对象语义描述，通过所述解析器适配器，使用所抽取的数据，生成所述语义对象。

44、如权利要求 42 所述的数据处理系统，其中用于通过所述软件对象，由所述搜索请求生成用于所述数据供给器的数据请求的所述程序指令进一步包括：

可操作地将请求编译器连接到所述软件对象；

可操作地将本机对象连接到所述请求编译器；

由所述本机对象封装用于所述数据供给器的数据请求的实施细节；

提供包括描述由所述请求编译器使用的数据结构的本体论信息的本机语义描述以便编译用于所述数据供给器的所述数据请求；

由所述请求编译器将所述搜索请求传送到所述本机对象；以及

通过所述本机对象，使用所述本机语义描述，由所述搜索请求生成所述数据请求。

45、一种计算机可读介质，包括有用于由计算机执行的程序指令，所述计算机程序指令使计算机履行数据服务请求，所述程序指令

包括:

提供数据服务的本体论描述;

经第一通信链路,可操作地将第一逻辑搜索对象连接到数据供给器;

通过所述第一逻辑搜索对象,由所述数据服务请求生成搜索请求;

经所述通信链路,由所述第一逻辑搜索对象将所述搜索请求传送给所述数据供给器;

响应所述搜索请求,经所述通信链路,由所述第一逻辑搜索对象从所述数据供给器接收数据集;以及

使用所述本体论描述,由所述第一逻辑搜索对象生成知识实例。

46、如权利要求 45 所述的计算机可读介质,其中所述第一通信链路与数据库服务器通信。

47、如权利要求 45 所述的计算机可读介质,其中所述第一通信链路与 FTP 服务器通信。

48、如权利要求 45 所述的计算机可读介质,进一步包括,所述第一通信链路与 Web 服务器通信。

49、如权利要求 45 所述的计算机可读介质,其中所述第一通信链路与文件系统通信。

50、如权利要求 45 所述的计算机可读介质,其中所述第一通信链路与人类数据供给器通信。

51、如权利要求 45 所述的计算机可读介质,其中所述第一通信链路与专用于所述数据供给器的通信协议通信。

52、如权利要求 45 所述的计算机可读介质,其中所述数据服务请求包括在 XML 文档中。

53、如权利要求 45 所述的计算机可读介质,所述程序指令进一步包括:

可操作地将第一工作流连接到所述逻辑搜索对象;以及

由所述第一逻辑搜索对象将所述知识实例传送给所述第一 workflow。

54、如权利要求 53 所述的计算机可读介质，所述程序指令进一步包括：

可操作地将第二逻辑搜索对象连接到所述第一 workflow；以及
由所述第一 workflow 封装所述第一和第二逻辑搜索对象间的所述本体论关系。

55、如权利要求 53 所述的计算机可读介质，所述程序指令进一步包括：

由应用服务器，经第二通信链路，将数据客户端可操作地连接到所述第一 workflow，以及

由所述应用服务器，经所述第二通信链路，从所述数据客户端接收数据服务请求消息，所述数据服务请求消息包括所述数据服务请求；

由所述应用服务器将所述数据服务请求消息传送给所述第一 workflow；以及

由所述第一 workflow 将所述数据服务请求消息传送给所述逻辑搜索对象。

56、如权利要求 54 所述的计算机可读介质，其中所述第二通信链路使用 SMTP 通信。

57、如权利要求 54 所述的计算机可读介质，其中所述第二通信链路使用 JMS 通信。

58、如权利要求 54 所述的计算机可读介质，其中所述第二通信链路使用 HTTP 通信。

59、如权利要求 54 所述的计算机可读介质，其中所述第二通信链路使用 RMI 通信。

60、如权利要求 54 所述的计算机可读介质，其中由所述第一 workflow 指定所述逻辑搜索对象。

61、如权利要求 54 所述的计算机可读介质，其中使用所述服务

请求消息，由所述应用服务器指定所述第一工作流。

62、如权利要求 54 所述的计算机可读介质，所述程序指令进一步包括：

提供格式器；以及

由所述格式器将封装在所述知识实例中的所述数据集格式化为由所述数据客户端请求的格式。

63、如权利要求 53 所述的计算机可读介质，所述程序指令进一步包括，将第二工作流可操作地连接到所述第一工作流。

64、一种计算机可读介质，包括有用于由计算机执行的程序指令，所述计算机程序指令使计算机经通信链路，存取数据供给器，所述程序指令包括：

由软件对象，接收搜索请求消息文档；

通过所述软件对象，由所述搜索请求消息文档生成用于所述数据供给器的数据请求；

经所述通信链路，由所述软件对象将所述数据请求传送给所述数据供给器；

经所述通信链路，由所述软件对象从所述数据供给器接收数据集；以及

通过所述软件对象，由所述数据集生成语义对象。

65、如权利要求 64 所述的计算机可读介质，其中用于通过所述软件对象，由所述数据集生成语义对象的所述程序指令进一步包括：

可操作地将解析器适配器连接到所述软件对象；

提供所述数据集的解析器语义描述，用于由所述解析器适配器使用；

提供语义对象语义描述；

通过所述解析器适配器，使用所述解析器语义描述，从所述数据集生成抽取数据；以及

根据所述语义对象语义描述，通过所述解析器适配器，使用所抽取的数据，生成所述语义对象。

66、如权利要求 64 所述的计算机可读介质，其中用于通过所述软件对象，由所述搜索请求生成用于所述数据供给器的数据请求的所述程序指令进一步包括：

可操作地将请求编译器连接到所述软件对象；

可操作地将本机对象连接到所述请求编译器；

由所述本机对象封装用于所述数据供给器的数据请求的实施细节；

提供包括描述由所述请求编译器使用的数据结构的本体论信息的本机语义描述以便编译用于所述数据供给器的所述数据请求；

由所述请求编译器将所述搜索请求传送到所述本机对象；以及

通过所述本机对象，使用所述本机语义描述，由所述搜索请求生成所述数据请求。

用于智能数据同化的方法和装置

技术领域

本发明一般涉及连接到计算机网络的服务器领域，以及更准确地说，涉及用作搜索引擎的服务器。

背景技术

Web 服务程序是由可操作地连接到 Internet 的计算机系统提供的软件应用程序。Web 服务程序从 Web 客户端接收请求并以用文档标记语言，诸如超文本标记语言（HTML）编写的文档的形式提供响应。第一 Web 服务器操作为文件系统的连接以及所提供的静态 HTML 文档存储为计算机系统内的文件。

可用将用户请求接收为输入参数的程序快速替换静态 HTML 文档。这些程序通常用各种解释过程语言，诸如 Perl 以及惯用过程语言，诸如 C 编写。

使用文档生成程序还引入新的性能。能从任何源获得数据并将数据用来创建文档。例如，能查询数据库以及在文档内使用该结果。此外，在文档生成程序的执行期间，文档生成程序能产生副作用。例如，典型的副作用包括产生和存储跟踪统计，用于由 Web 站点的主机以后使用和分析。

使用文档生成程序在提供 Web 服务程序的计算机系统内产生瓶颈。每个文档生成程序在其自己的过程中运行，以及所需过程的恒定产生和破坏产生不可接受的总需求。这些总需求导致产生能调用文档生成程序，而不创建新过程的应用服务程序。

在一种这种应用服务程序，应用服务程序包括解释程序以及用专用编程语言编写文档生成程序。在专用编程语言是过程语言的情况下，应用服务程序读取并运行文档生成程序而不调用新过程。在专用

编程语言是面向对象的语言，诸如 Java 的情况下，每个文档生成程序是称为小服务程序的软件对象。在这种情况下，Web 应用程序根据需要，调用小服务程序以及小服务程序处理剩余的文档生成功能。

以前开发的 Web 服务程序集中在产生更有效地使用计算资源以便通过 Web 站点产生越小越小的垂直薄片 (vertical slice)。例如，一次调用的小服务程序包括获得对应于客户请求的数据所必需的所有必要的逻辑，将所获得的数据安排为文档，以及将有关客户/小服务程序交互作用的任何统计报告回主机系统。通过 Web 站点的越来越小的垂直薄片导致不可升级 Web 服务程序体系结构，因为用于获得数据、生成文档以及产生副作用的所有逻辑均包括在单块软件对象中，从而使重新使用特定的软件对象或将其结合到另一系统或软件对象的能力破灭。

因此，对可升级 Web 应用程序体系结构来说，需要具有 Web 服务程序的组成部分的高度可复用性的特征。本发明满足这种需要。

发明内容

在本发明的一个方面中，提供用于实现数据服务请求的方法。该方法包括提供数据服务的本体论描述，第一逻辑搜索对象经第一通信链路，连接到数据供给器。第一逻辑搜索对象经通信链路，将由第一逻辑搜索对象由数据服务请求生成的搜索请求传送给数据供给器。响应搜索请求，第一逻辑搜索对象从数据供给器，经通信链路接收数据集。使用本体论描述，第一逻辑搜索对象由数据集生成知识实例。

在本发明的另一方面中，使用于实现数据服务请求的方法适合于包括数据库服务器、FTP 服务器、Web 服务器、文件系统的多种数据供给器以及人体数据供给器。另外，该方法适合于专用于数据供给器的通信协议。

在本发明的另一方面中，数据服务请求包括在 XML 文档。

在本发明的另一方面中，用于实现数据服务请求的方法进一步包括提供可操作地连接到逻辑搜索对象的第一工作流以及通过第一逻

辑搜索对象将知识实例传送给第一工作流。

在本发明的另一方面中，用于实现数据服务请求的方法进一步包括提供可操作地连接到第一工作流的第二逻辑搜索对象，以及第一工作流封装第一和第二逻辑搜索对象间的本体论关系。

在本发明的另一方面中，用于实现数据服务请求的方法进一步包括提供经第二通信链路，可操作地将数据客户端连接到第一工作流的应用服务器。应用服务器经第二通信链路，从数据客户端接收具有包含在数据服务请求消息中的数据服务请求的数据服务请求消息。应用服务器将数据服务请求消息传送给第一工作流以及第一工作流将数据服务请求消息传送给逻辑搜索对象。

在本发明的另一方面中，第二通信链路适合于使用 SMTP、JMS、HTTP 或 RMI 通信。

在本发明的另一方面中，用于实现数据服务请求的方法进一步包括提供格式化程序以及通过格式化程序将封装在所述知识实例中的数据集格式化为由数据客户端请求的格式。

在本发明的另一方面中，用于实现数据服务请求的方法进一步包括提供可操作地连接到第一工作流的第二工作流。

在本发明的另一方面中，数据处理系统用来经通信链路，访问数据供给器。数据处理系统包括处理器和可操作地连接到处理器并具有存储在此的程序指令的存储器。可操作处理器来执行程序指令。程序指令包括由软件对象接收搜索请求消息文档以及通过软件对象，由搜索请求消息文档生成用于数据供给器的数据请求。经通信链路，软件对象将数据请求传送给数据供给器。软件对象经通信链路，从数据供给器接收数据集，并由该数据集生成语义对象。

在本发明的另一方面中，用来经通信链路访问数据供给器的数据处理系统包括解析器适配器，可操作地连接到软件对象。提供数据集的解析器语义描述和语义对象语义描述，由解析器适配器使用。解析器适配器使用解析器语义描述，由数据集生成所抽取的数据。然后解析器适配器使用所抽取的数据，根据语义对象语义描述，生成语义

对象。

在本发明的另一方面中，适合经通信链路访问数据供给器的数据处理系统包括可操作地连接到软件对象的请求编译器。通过封装用于数据供给器的数据请求的实现的细节的本机对象，将本机对象可操作地连接到请求编译器上。所提供的本机语义描述包括描述由请求编译器使用来编译用于数据供给器的数据请求的数据结构的本体论信息。请求编译器将搜索请求传送给本机对象，以及本机对象使用本机语义描述，由搜索请求生成数据请求。

附图说明

参考下述的描述和附图，将更容易理解本发明的这些和其他特征、方面和优点，其中：

图 1 是根据本发明的实施例，智能数据同化系统的配置图；

图 2 是适合用作智能数据同化系统主机的通用计算机的硬件体系结构图；

图 3 是示例说明根据本发明的智能数据同化系统的实施例的体系结构图；

图 4 是在根据本发明的智能数据同化系统本体论的一个实施例内的层次图；

图 5 是根据本发明，描述在本体论中，语义结构和它们的相关的逻辑搜索对象间的本体论关系的示例性实施例的图；

图 6 是描述根据本发明，使用逻辑搜索对象的本体论关系，如何创建服务类的示例性实施例的图；

图 7a 和 7b 是描述在元数据存储器内如何描述语义对象的图；

图 8 是根据本发明，描述示例性工作流的实施例的操作的图；

图 9 是根据本发明，描述在工作流过程的实施例中的交互作用的协作图；

图 10 根据本发明，描述在工作流内使用的逻辑搜索对象的示例性实施例；

图 11 是根据本发明，描述一般逻辑搜索对象的实施例的分层体系结构，以及如何使用该分层体系结构来生成具有各种连接性能的逻辑搜索对象；

图 12 是根据本发明，示例说明当由智能数据同化系统的实施例接收请求信息时发生的事件的顺序的时序图；

图 13 根据本发明，示例说明采用逻辑搜索对象的智能数据同化系统；

图 14 是根据本发明的示例性服务请求消息的 XML 文档实施例；

图 15 是根据本发明的示例性状态消息的 XML 文档实施例；

图 16 是根据本发明的示例性取消消息的 XML 文档实施例；

图 17 是根据本发明的示例性输出消息的 XML 文档实施例；

图 18 是根据本发明的示例性响应消息的 XML 文档实施例。

具体实施方式

图 1 是根据本发明，智能数据同化系统的配置图。智能数据同化系统主机 10 经适合于通信的通信链路 12，使用各种传输协议，诸如传输控制协议/Internet 协议（TCP/IP），可操作地连接到通信网络 14，诸如 Internet。逻辑搜索对象主机通过适合于使用 TCP/IP 通信的通信链路 20，可操作地连接到通信网络上。数据供给器主机可操作地连接到数据库，但不限于数据供给器数据库 24。数据客户端主机 18 经适合于使用 TCP/IP 协议通信的通信链路 16，可操作地连接到智能数据同化系统主机。

在操作中，由数据客户端主机提供的数据客户端 26 将服务请求消息 27 传送给由智能数据同化系统主机提供的智能数据同化系统 101。服务请求消息包括智能数据同化系统用来生成查询 29 并将其传送给由数据供给器主机提供的数据供给器服务器 28 的搜索参数。数据供给器服务器通过生成数据集 31 并将其传送回智能数据同化系统来响应查询。智能数据同化系统增加所返回的数据的本体论实例并将

该数据集格式化为传送给数据客户端作为对服务请求消息的响应的格式化数据结果 33。

图 2 是适合于用作智能数据同化系统主机的通用计算机的硬件体系结构图。包括中央处理单元 (CPU) 310、超高速缓存 320 以及总线接口 330 的处理器 300 经系统总线 335, 可操作地连接到主存储器 340 和 I/O 控制单元 345。经 I/O 局部总线 350, I/O 控制单元可操作地连接到磁盘存储器控制器 395 和网络控制器 380。

磁盘存储器控制器可操作地连接到磁盘存储器设备 325。将用于实现智能数据同化系统的计算机程序指令 397 存储在磁盘存储设备上直到微处理器检索计算机程序指令并将它们存储在主存储器中为止。然后, 微处理器执行存储在主存储器中的计算机程序指令以便实现智能数据同化系统的特征。

网络控制器可操作地连接到通信设备 396 上。通信设备适合允许由通用计算机提供的智能数据同化系统经诸如 Internet 的计算机网络, 与在计算机网络上的另一软件对象通信。

图 3 是根据本发明, 示例说明智能数据同化系统的实施例的体系结构图。智能数据同化系统通过应用程序服务器 (未示出) 的服务, 连接到计算机网络上。智能数据同化系统是提供机制和子系统来创建和操作封装用来执行对来自多个数据供给器 104 的数据的搜索的知识和能力的逻辑搜索对象 (LSO) 的知识库 102 的面向对象的基于部件的服务器平台。经通信网络, 通过应用程序服务器, 由数据客户端访问智能数据同化系统。

使用各种方法, 将数据供给器逻辑连接到智能数据同化系统上。数据供给器提供包括来自数据供给器的各种数据产物, 诸如数据聚合器、搜索机器人或代理、基于 HTML 的搜索引擎、室内数据库或能提供机制以抽取电子信息的其他机制的数据集。另外, 数据供给器能是从智能数据同化系统接收请求以找出和返回不可电子访问的数据集的人为代理。例如, 可使用私人调查员来收集用于数据集, 诸如确认人的物理位置的数据。

LSO160 能重复使用、能构造并自带能建立与由数据供给器提供的特定数据集的连接的软件组成部分。此外，LSO 包括用于限制或过滤数据集并递送最终信息作为用可扩展标记语言（XML）书写的文档，或作为在此称为语义对象 108 的智能数据对象。另外，LSO 向其他软件组成部分展示其能力和信息，以便由那些软件组成部分使用。

将 LSO 结束定义的存储在数据库 110 中，作为元数据描述。在运行时，当调用特定 LSO 的特定实例时，由 LSO 的元数据定义创建 LSO 实例。然而，如果需要的话，智能数据同化系统有能力预先例示 LSO。

为向用户提供操作 LSO 结果的能力，智能数据同化系统提供数据映射工具以便将语义智能增加到原子数据上。通过提取（组成）各个字段为明确定义的本体论 112，智能数据同化系统提供允许用户操纵封装数据的摘要视图。例如智能数据同化系统能将称为“名字”114 的 LSO 和称为“姓”116 的 LSO 组合成称为“名字”的语义对象 118，然后，将具有称为“Address”108 的另一语义对象的“名字”组合成称为“Person”120 的语义对象。该语义结构允许智能遍历以及识别任何可用信息。

通过显示有关连接性的所有它们的性能，诸如本体论数据和成本，LSO 为逻辑搜索对象子系统提供机制来提供用于定义可由数据客户端，从智能数据同化系统请求的服务的工具 122。服务生成包括由数据客户端所请求的数据的数据产物形式的结果。通过使用图形用户界面，设计者浏览在知识库中的所有可用 LSO 和语义对象并操作 LSO 以及选择所需 LSO 或语义对象并将 LSO 和语义对象放入 workflow 中。

每个 workflow 通常表示单个服务提供。通过将一或多个 LSO 放入 workflow 中，设计者创建产生服务的 workflow。这些 workflow 的一些特征包括并行运行多个 LSO、定义故障 LSO，以及定义一个 LSO 的输出能是另一对象的输入的偶然性的能力。

使用本体论设计程序 122 来将来自数据供给器的数据的数据映射创建成智能数据同化系统本体论。通过该过程，执行数据的智能提取以便创建由 LSO 使用、保持在智能数据同化系统知识库中的语义对象。

为创建智能数据同化系统服务，使用产物配置器 122 来浏览和选择语义对象以及应用 workflow 定义。产物配置器提供加强的生产力特征，诸如拖放功能性和范例样式。

产物配置器为封装在语义对象中的信息提供表面。这允许设计者将数据操作为单个原子数据或高提取的复杂数据。封装在 LSO 中的数据可映射为单个语义对象，从而设计者能将表示特定 LSO 的那个语义对象放入 workflow 中。然而，不由 LSO 限制语义对象，产物配置器允许设计者来操纵横跨多个 LSO 的甚至更高提取的语义对象，从而产物配置器导出对指定服务，哪些 LSO 将插入 workflow 中，以及如果有的话，在封装在语义对象中的 LSO 间存在什么限制。

通过封装有关连接性、成本、查询机制以及本体论的信息的 LSO，产物配置器能操作提取数据以及将这些项放入 workflow 中。

除选择并将语义对象放在 workflow 中的能力外，产物配置器还为设计者提供配置有关由服务生成的数据产物的其他属性，诸如输入显示属性、输入规则、结果可扩展样式语言 (XSL) 样式表、以及定义数据产物的其他属性的能力。

当完成服务定义时，设计者具有使用现场或模拟数据，实时反复测试和提高服务，以及实时查看结果的能力，以及当满足完成该服务时，将服务保存到智能数据同化系统元数据储存库中，从而使服务实时配置。

智能数据同化系统巩固在 workflow 中执行的所有 LSO 的结果以及将结果数据表示为能以那种形式递送给数据客户端的 XML 文档。在一个实施例中，智能数据同化系统重新格式化 XML 文档以便使用 XSL 样式表 125，创建格式化输出 124。这些 XSL 样式表提供再现格式化数据，包括以 HTML 126、XML、无线应用协议 (WAP) 128、

Rich-Text 格式 (RTF)、Word (DOC)、WordPerfect (WPS)，或其他格式 130 的结果的模拟递送。

智能数据同化系统本体论便于多个计算机系统间交换数据，而与各个系统技术、信息体系结构和应用域无关，并允许创建将原子数据字段组合到允许构造由智能数据同化系统提供的服务的称为语义对象的智能编译块中。

本体论包括术语词汇以及这些术语所表示的意思的说明。智能数据同化系统本体论提供用来构造有意义的更高级知识的一组有根据的结构。选择智能数据同化系统本体论中的基本术语以便定义和规定基本原理和区别。所选择的基本术语形成完全集，使用正式技术定义与另一个的关系。这些正式定义的关系为所选择的术语提供语义基础。包括在智能数据同化系统本体论中的这些关系允许表达特定域知识，而不包括特定域术语。

智能数据同化系统本体论包括抽象定义。例如，由基本原子或语义项、名字、中间名字以及姓构造人的语义。使用定义来创建本体论的知识实例，例如，当在运行时间时，用由 LSO 返回的特定语义和原子实例来增加本体论。

定义和知识实例的寿命可改变。包括在智能数据同化系统本体论中的定义是持久的、大量重复使用，并在智能数据同化系统中具有全局范畴。运行时间创建的知识实例是瞬时或持久的。在智能数据同化系统的环境中，请求通常是瞬时的，因为知识存在于本体论中仅特定请求实例的持续时间，然后丢弃。例如，从数据供给器检索数据，使用运行时间知识实例，增加对应于工作流中的 LSO 的本体论的部分，以及仅处理在特定请求实例中的创建的知识实例以及只要处理过，从本体论丢弃知识实例。在根据本发明的智能数据同化系统的一个实施例中，知识实例是持久的，以便以后可从智能数据同化系统查询知识实例并同化成另一服务请求。因此，如上所述，持久知识实例具有全局范畴。

使用智能数据同化系统的商业代理适合并能传送特定域知识。

商业代理通过使用在智能数据同化系统本体论中定义的基本术语和关系，以及通过基本术语的这些定义来完成此操作，智能数据同化系统允许组合（提取）基本术语以形成更高级智能结构。

在根据本发明的智能数据同化系统的一个实施例中，智能数据同化系统本体论存储在元数据库中作为 XML 定义并使用其 XML 定义，在许多不同层中表示。

智能数据同化系统加强所有数据客户端的安全性。用于执行安全性的基本机制是对照智能数据同化系统中的 Java 命名和目录接口 (JNDI) 树的名字和密码验证。通过协议方法，诸如 Java 消息业务 (JMS)，远程方法调用 (RMI)，或超文本传输协议 (HTTP)，发送给智能数据同化系统的消息具有所提供的名字和密码验证层安全性。智能数据同化系统还支持使用数字签名。发送给服务器的数据可包括能对照由请求和所存储的公钥生成的校验和核实的数字签名。智能数据同化系统还支持数据加密。

在处理层 (processing tier)，只要智能数据同化系统已经接受服务请求消息，智能数据同化系统保密相对于用于那个用户的可用服务的服务请求。客户端不必访问对他们来说不可用的服务。

智能数据同化系统还实施将结果返回给始发数据客户端。这种类型的安全性限制访问搜索结果并允许由提供相应的服务请求的那些数据客户端访问。

智能数据同化系统支持允许用于任何指定请求的多个通知的通知机制。该特征允许智能数据同化系统通过将结果发送给一个或所有用户的 email、便携电话、web 页等等来响应服务请求。

智能数据同化系统实施与特定商业过程有关的商业规则。将这些商业规则抽象成广义的机制，通过该机制，智能数据同化系统有条件地评价和确定适当的动作。智能数据同化系统支持在智能数据同化系统的各个层，执行规则验证的能力。

智能数据同化系统规则体系结构是开放的，以便存在期望返回表示任何规则成功或失败的布尔值的定义好的出口点。通过采用任何

商业过程来实现这些规则，其是人工干预，适当的算法，或现有规则引擎。

将智能数据同化系统体系结构设计成利用当前企业采用的标准。在根据本发明的智能数据同化系统的一个实施例中，智能数据同化系统的组成部分支持用于开发多层企业应用程序的 Java 2 企业编译(J2EE)标准。J2EE 通过使它们基于标准化的、模块化的组成部分，通过提供对那些组成部分的整套服务，以及通过自动处理应用程序行为的许多细节，简化企业应用程序，而没有复杂的编程。智能数据同化系统的该实施例还采用 J2EE 标准的许多特征，诸如软件可移植性，用于数据库存取的 JDBC API，和即使在 Internet 应用程序中，保护数据的安全性模型。

在根据本发明的智能数据同化系统的一个实施例中，用于这些组成部分的基础平台是 J2EE 可兼容应用服务器。合适的应用服务器是 BEA Weblogic 应用服务器。BEA 的 Weblogic 应用服务器的权能包括分布式事务管理器，Java 消息业务 (JMS)，支持可扩展标记语言(XML)，以及 J2EE 标准支持。

在根据本发明的智能数据同化系统的一个实施例中，智能数据同化系统使用 BEA Weblogic 方法集成(Process Integrator)(WLPI)来实现 workflow 任务。WLPI 为智能数据同化系统提供设计和使结合数据供给器应用程序、搜索服务和人为干预的商业过程自动化的能力。

在根据本发明的智能数据同化系统的实施例中，Oracle 数据库管理系统(DBMS) 提供具有持久权能的智能数据同化系统。

图 4 是在根据本发明的智能数据同化系统本体论的一个实施例的示例性部分中的层次图。智能数据同化系统本体论管理三层数据抽取。最初，将在数据集中的原始数据字段 140 从数据供给器位置映射到原子对象 (atomic object)，诸如人的原子对象 146 和信用原子对象 (credit atomic object) 148 中。原子对象将类型和基本命名信息增加到数据字段中。通过语义层 142 中的语义对象抽象，创建包括在原子对象中的有关数据字段的意义的智能 (intelligence)。

智能数据同化系统将数据字段视为可由智能数据同化系统检索的任何数据集的最基本元素。在本体论定义前，这些数据字段通常包括非语义上下文或智能。能识别字符或字节串，数字，小数点和日期；然而，应用无意义的定义来提供数据的理解。例如，在数据字段的位置 1 至 9 中，可包括值“561929975”，但典型的数据同化系统不理解这是社会安全号、序号还是产品号。通过智能数据同化系统将语义上下文与数据字段关联产生原子对象。

原子对象具有与特定类型的值有关的语义上下文。因此，能将数据字段转换成原子对象。例如，该值可以是字符串“John 10/9/2000 21:23.234 8A56FB”并在格式中包括语义信息：名字（First Name），日期（Date），和 ID 查询结果（Result of Query ID）。

语义对象是在智能数据同化系统本体论定义的，具有特定商业意义的原子对象的组合。智能（Intelligence）是通过这些高级组合创建。例如，能组合下述的原子对象：名字 147，中间首字母 151，和姓 149 以提供“人名”语义对象 150。另外，能组合下述原子对象：地址 153，城市 155，州 157，和邮政编码 159 以创建“人的地址”语义对象 152。随后，能通过将“人名”和“人的地址”组合来编译“人”语义对象 156 来创建更高级语义对象。用类似的方式，使用信用文件原子对象 148 来创建信用分数语义对象 154。然后使用信用分数语义对象来创建更高级的信用风险语义对象 158。在智能数据同化系统本体论的数据服务层 144 中组合语义对象以创建由智能数据同化系统内的其他子系统使用的高级数据提取。

图 5 是描述在根据本发明的本体论中，语义结构与它们相关的 LSO 间的本体论关系的示例性实施例的图。示例性实施例描述法人实体类 200 和两个子类、人 218 和公司 202。这些类的每一个包括封装根据合法所定义的，可存在法人实体类的例示的原理的子类。在人类的情况下，适当的类是住处类 220 以及对公司来说，适合类是公司地址类 204。这些类，街道地址 212，城市 210，邮政编码 208，和州 206 通过多个继承，即，住处和公司地址类，具有多个父类。

街道类进一步包括两种方法，方法 1 214 和方法 2 216，用于检索街道地址。能通过由将请求发送给住处或公司地址类的例示的对象寻址和使用这些方法以个性化用于检索街道地址的数据供给器。例如，用于有关个人的住处的数据请求可包括对来自纳税记录的街道地址的请求。对有关同一个人的住处信息的另一请求可包括对从汽车登记记录获得的街道地址的请求。在任何一种情况下，数据供给器是不同的而且可使用不同的方法来从不同的数据供给器请求和接收数据。

在操作中，住处类或公司类的示例分别答复对返回个人或公司的法人地址的请求。例如，当系统接收适当请求时，如果请求查找个人的合法地址，其调用将示例适当的 LSO 来提取所请求的数据的工作流。这些对象的每一个联系数据供给器并请求和从数据供给器接收适当的数据。将调用与该工作流有关的本体论的实例并且当已经检索到数据时，在示例的本体论中的任何节点，（诸如该例子中的住处），能代表用于本体论中的那个特节点的数据的基本集合。因此，具有能集合所接收的数据的能力的住处对象能代表个人地址。

图 6 是根据本发明，描述使用 LSO 的本体论关系如何创建服务类的示例性实施例。使用数据服务类，有选择地遍历 LSO 间的本体论关系。通过本体论的这种遍历产生查询的答复。例如，示例性服务类，Service 1 222，封装查询 "Where does John (a person) live?"。使用方法 1 214、城市 210，邮政编码 208，和州 206，该 Service 1 类的示例产生从个人 218 到住处 220，到街道 212 的本体论关系的遍历。这种遍历用环绕适当的对象的实线表示。

在另一例子中，示例性服务类，Service 2 224，封装查询 "Where can ABC corporation (a legal entity) be served?"。使用方法 2 216，Service 2 类的示例产生从法人实体 200，到公司 202，到公司地址 220，到街道 212，到城市 210，到邮政编码 208，和到州 206 的本体论关系的遍历。这种遍历用环绕适当的对象的虚线表示。

示例性服务通过用不同方式存取关系来开发 LSO 间的本体论

关系。 Service 1 封装有关个人地址的查询以及 Service 2 封装有关服务的公司位置的查询。在任何一种情况下，服务必须确定街道地址和每个服务最终使用同一街区，城市，邮政编码，和州对象类。然而，所获得的每个服务存取来自不同父类的这些对象。

语义对象结构不限于特定 LSO 以及由此的特定数据集。智能数据同化系统能由语义对象外推在多个工作流中 LSO 和 LSO 的适当的位置。智能数据同化系统本体论和所包括的语义对象的定义保存在智能数据同化系统的元数据存储库中并为全局内部数据字典的一部分。

在执行时间，元数据存储库为由 LSO 使用的结构数据提供持久性。搜索结构包括：数据供给器的标识，数据字段到智能数据同化系统已知的原子对象的本体论的映射结果，以及数据字段到数据供给器已知的原子对象的本体论的请求映射。最后，搜索结构包括当在执行时间，执行搜索时，参考 LSO 来使用。

图 7a 和 7b 是描述在元数据存储库，如何描述语义对象的图。语义对象可包括其他语义对象和各种类型的最终原子对象。例如，语义对象个人 400 包括另一语义对象地址 402，其包括州 404，城市 406，邮政编码 408，和街道号 410 原子对象。个人语义对象还包括原子对象名字 412 和姓 414。

图 7b 是在根据本发明的智能数据同化系统的实施例，使用存储在元数据存储库中的语义描述符，构造图 7 的语义对象的描述。类属描述符，诸如"城市"数据描述符 420 是包括集合方法的数据描述符的具体实现。数据描述符描述数据的名称，数据类型和数据顺序。由于仅存取数据，数据描述符接口仅公开用于存取数据供给器的获得方法。语义描述符，诸如"个人"语义描述符 416 公开检索个人语义描述符所包括的子元素的方法。这些子元素能是其他语义描述符，诸如"地址"语义描述符 418 或数据描述符诸如"名字"数据描述符 428 和"姓"数据描述符 430。在智能数据同化系统定义的接口具有实现接口的功能性的具体类。例如，用类属数据描述符实现数据描述符。该

类提供在数据描述符接口中未公开的设置方法。在初始化期间，创建描述符的组成部分使用具体类来创建设定值。

下面是描述符列表：

DataDescriptor-> 公开用于名字，类型，和顺序的获得方法

GenericDescriptor-> 包括设置方法的 **DataDescriptor** 的具体实现

PositionalDescriptor-> 公开用于位置和长度的获得方法

GenericPositionalDescriptor-> 包括设置方法的 **PositionalDescriptor** 的具体实现

SemanticDescriptor-> 获得元素（elements）、更多元素（hasMoreElements）、下一元素（getNextElement）的方法

GenericSemanticDescriptor-> 包括设置方法的 **SemanticDescriptor** 的具体实现

SemanticListDescriptor-> 公开获得原子长度（GetAtomicLength）的方法

GenericSemanticListDescriptor-> 包括设置方法的 **SemanticListDescriptor** 的具体实现

智能数据同化系统模型使如何表示数据抽象化并更关注如何使数据片彼此相关上。智能数据同化系统本体论不用改变，而不禁止另一数据供给器本体论。这提供可扩展性，因为智能数据同化系统和数据供给器松散连接。智能数据同化系统本体论构造定义搜索结果定义的语义上下文。

智能数据同化系统采用管理执行与服务有关的服务请求的工作流机。工作流机是管理工作流内以及具有 0 之后的起始节点的每个工作流，或端节点后的许多任务节点的任务的状态机。

在定义服务的过程中设计者识别包括在服务中的所有 LSO。然

后，将 LSO 与 workflow 中的任务节点相关联，由此实现定义指定服务。

当智能数据同化系统接收用于实现特定服务的请求时，智能数据同化系统将服务请求映射到特定 workflow，示例 workflow，合并由 workflow 合并的最终数据集和调用用于 workflow 的任何分析或商业规则处理。在根据本发明的智能数据同化系统的一个实施例中，通知请求者数据集可用。在另一实施例中，存储数据集（此处指持续地），便于由请求者以后检索。在另一实施例中，将最终数据集递送给请求者。

智能数据同化系统 workflow 机制不仅提供包括 LSO 的任务节点的顺序执行，而且提供将并行执行的 LSO 和创建 LSO 间临时关系的能力。例如特定 LSO 的输出可以是另一 LSO 的输入。另外，智能数据同化系统 workflow 机制允许 workflow 中的任务节点与能包括服务或其他商业过程的其他 workflow 关联，并由此创建使用嵌套 workflow 来生成重新使用的服务和 workflow 的不同类的机制。

图 8 是根据本发明，描述示例性 workflow 图的实施例的操作的图。在智能数据同化系统的一个实施例中，所使用的工作流处理环境是由 BEA Systems, Inc. of San Jose California, USA 提供的 Weblogic Process Integrator (Weblogic 过程综合器 WLPI)。使用 Weblogic 过程综合器 (WLPI) 工作室软件工具或通过创建工作流的 XML 表示来指定 workflow。在前描述的元数据存储包括用来构造 workflow 的特定实例的数据，该 workflow 用来实现服务请求。因此，元数据存储管理实现服务请求的 workflow 定义和 LSO 定义。

workflow 600 从服务请求处理机 602 接收以 XML 文档形式的服务请求。在步骤 604，workflow 验证请求。示例性验证包括使用内部判定规则，确定请求是否包括足够的信息来路由和完成请求。如果请求无效，结果返回到结果文档 608 中，而不执行任何搜索任务。如果所述请求有效，那么在节点 610 初始化搜索任务来履行请求。workflow 一次能启动多个任务。在该示例性 workflow 中，一次启动任务 A 612 和任务 B 614。任务 A 使用 LSO A 616 来连接到外部数据供给器 A

618 并从其检索数据。同时，任务 B 使用 LSO B 620 来连接到外部数据供给器 B 622 并从其检索数据。在合并过程 624 中合并从两个 LSO 返回的结果。根据如前所述的智能数据同化系统本体论，产生结果合并。

只要已经同化数据，智能数据同化系统考虑另外过滤数据集。例如，可将可选"opt-out"规则应用于合并结果上，允许个人或实体从数据集排除个人或实体不希望传播的个人信息。opt-out 规则是响应数据请求，防止某些信息发布到服务对象的内部规则。在根据本发明的 workflows 的实施例中，在即时搜索的情况中，在工作流过程结束时，将结果返回给服务请求处理机 608。在即时搜索中，数据客户端发送搜索请求，同时期望在短的时间周期内接收到搜索响应，在该短的时间周期内，数据客户端仍然连接到智能数据同化系统。在根据本发明的搜索请求的另一实施例中，数据客户端传送搜索请求，同时期望返回智能数据同化系统以便在以后的时间检索搜索结果。

图 9 是描述在根据本发明的 workflows 的实施例中的交互作用的协作图。所描述的工作流的实施例包括在工作流内可单独或共同使用的几个可选过程。

智能数据同化系统支持在单个请求中，组合起来实现多个服务的能力。这提供能将与行式项目有关的整个顺序发送给智能数据同化系统作为单个请求的机制。当智能数据同化系统接收搜索服务请求消息，服务请求处理机部件创建一个实现请求。

对存在于消息中的每个请求项，服务请求处理机 700 在履行线路表 765 (fulfillment line table) 中创建履行线路行 (fulfillment line row) 702 并调用将履行线路 ID 传递为参数的工作流实例 704。

将运行 workflow 的状态存储在履行线路中。当 workflow 实例结束时，将搜索结果存储在履行线路中。对非实时请求来说，智能数据同化系统将消息发送给 workflow 机来异步地产生 workflow 实例。与履行线路一起维护所有状态信息以便能回答任何以后的状态请求。通过履行线路，智能数据同化系统识别 workflow 实例是否仍在处理、已经失败，还

是正等待用户输入或手动搜索结果。

调用 workflow 实例包括消息 703，消息 703 包括以 XML 文档形式的履行线路 ID 和请求参数。workflow 实例接收消息并开始数据收集过程 706。

在起动通知节点 708 期间，workflow 实例将通知 710 发送给 workflow 状态机 711。workflow 状态管理器将更新消息 719 发送给履行线路以便履行线路的查询将表示 workflow 实例正对数据请求起作用。

workflow 实例包括在等待将从外部取消请求处理机 714 发送给 workflow 实例的外部异常中断消息 716 的 workflow 实例中的异常中断节点 712。如果接收到异常中断消息 716，异常中断节点结束 workflow 实例的数据收集过程 718 并将异常中断消息 720 传送给 workflow 状态管理器。作为响应，workflow 状态管理器 720 更新履行线路以表示已经中止 workflow 实例。

在节点 712，workflow 实例校验输入参数来确定是否适当地形成用于数据供给器的请求的参数值。如果未正确形成输入参数值，向异常处理器发出异常。为在失败（运行时间异常）的情况下，维持适当的错误状态，每个 workflow 实例具有异常处理机 715。如果均错误，诸如执行 LSO 出现异常，执行异常处理机并将 workflow 实例的状态设置成“FAILED”。在 workflow 实例失败的情况下，异常处理机将发生失败的消息 717 发送给 workflow 状态管理器，以便能在履行线路中更新 workflow 实例的状态。

workflow 实例能顺序地或并行调用多个 LSO。在该例子中，在步骤 722 调用第一 LSO 728 并将以 XML 文档 724 的形式的调用消息传送给 LSO。第一 LSO 将数据请求消息 730 传送给数据供给器 732 以及从数据供给器接收数据消息 734。LSO 通过将所请求的数据消息返回给 workflow 实例来响应。

workflow 实例确定在步骤 738 中是否正执行复杂的搜索。如果搜索是简单的搜索并且仅将调用第一 LSO，那么 workflow 实例在步骤 740 结束其搜索过程并将状态完成消息 742 发送给 workflow 状态管理器。如

果 workflow 实例完成该搜索，将更新结果 XML 文档 764 发送给履行线路。如果 workflow 状态管理器确定未正常完成 workflow 实例，其向智能数据同化系统通知器 762 发布异常 760。

如果请求复杂的搜索，诸如使用几个 LSO 来查询单独的数据库或 LSO 将从手工过程来收集数据，workflow 实例调用第二 LSO 744 并以 XML 文档的形式将第二语义上下文消息 736 发送给第二 LSO (未示出)。使用来自智能数据同化系统的客户端可请求将在相同对话中执行的履行作为请求，如实时搜索的情况。为便于此，智能数据同化系具有构造成同时处理请求的服务。对不能立即履行的搜索（如在手工搜索的情况下），智能数据同化系统具有能构造成异步处理的服务。因此，智能数据同化系统具有使完成 workflow 的响应消息同步并在响应消息中包括最终输出的能力。当请求异步服务时，智能数据同化系统响应数据客户端，而与 workflow 的完成无关，对此想法，数据客户端将“稍后返回”来请求结果。

workflow 实例能调用其他 workflow 实例。为实现此操作，workflow 实例调用 746 子 workflow 实例 750 并将调用消息 748 以 XML 文档的形式传送给子 workflow 实例。子 workflow 实例与 workflow 实例无关地操作。然而，workflow 实例能等待 746 子 workflow 实例结束处理。workflow 实例可将异常消息以 XML 文档 754 的形式发送给 workflow 实例，如果子 workflow 实例在执行期间遇到问题的话。当子 workflow 实例完成处理时，子 workflow 以 XML 文档的形式，将 workflow 结束消息 752 发送给 workflow 实例。workflow 实例接收 756 结束 workflow 消息并将包含在结束 workflow 消息中的数据合并 758 到语义对象中，用于由智能数据同化系统进一步处理。

能直接操作服务元数据存储或通过产物构造工具来完成 workflow 的定义。因此，通过定义包括在特定服务中的用于 LSO 的处理流程，实现 workflow 定义。智能数据同化系统将此视为通过一组判定和 LSO 的服务请求。

能开发各个 workflow 模型来支持同步或异步搜索并将最终输出合

并到定义好的结构中。可将商业规则或适当的算法应用于搜索结果上来支持 workflow 环境的特定属性。

workflow 机对 workflow 中的暂停提供支持以便执行手动干预。通过该能力，能执行不能经自动属性的搜索以及结果信息返回到 workflow 中并与自动搜索合并。

智能数据同化系统提供在完成 workflow 实例后，触发的通知机制。通知调度器利用提供 e-Mail、传真传输（传真）、电报、寻呼、DBMS 存储和 JMS 功能性的智能数据同化系统服务。使用智能数据同化系统的客户应用程序将通知信息提供为服务请求消息的一部分。智能数据同化系统接受用于任何服务请求的多个通知方法。

e-Mail 服务将格式化的 e-Mail 消息传送给包括在服务请求消息中的、具有完成搜索的标识符的 e-Mail 地址，以及有关如何获得结果的消息。e-Mail 服务还能格式化搜索结果并将它们包括在消息体中。

电话通知服务经寻呼机、传真和无线电话提供通知。

对提供存取 DBMS 的系统来说，智能数据同化系统用表示完成服务请求的值更新数据库表。

其他系统可通过将消息张贴到消息队列主题上来请求通知。这是通过智能数据同化系统的 JMS 服务来实现的。客户端实现接听张贴到队列的消息的机制。在接受消息后，数据客户端执行请求管理活动并将输出请求提交给智能数据同化系统。

图 10 根据本发明，描述在 workflow 中使用的 LSO 的示例性实施例。LSO 是包括一套组件的自带软件对象，当结合在一起时，封装该能力以便连接到数据供给器，响应请求，通过抽取和解析由数据供给器返回的数据，生成知识实例，以及将知识实例递送给本体论，用于由请求的当前实例使用以便能将数据合并到结果集中。

LSO 是可重使用的组成部分并理解对数据供给器的各种存取方法、做出搜索请求的协议，以及将所返回的数据转换成包括由智能数据同化系统使用的原子和语义结构的知识实例。这些特征形成

LSO500 的几个层的划分：翻译层；数据采集层以及连接层。通过称为适配器的软件对象实现这些层。例示 LSO 包括翻译适配器 502、采集适配器 504 和连接适配器 506。

翻译层包括编译数据供给器 508 的数据请求，然后解析所返回的数据流 510 的软件对象。请求编译器软件对象 512 理解用数据供给器定义的方式，如何构造该请求。例如，数据供给器可请求将发送给数据库的结构化查询语言(SQL)语句、将发送给统一资源定位器(URL)的查询串，或适当的查询协议。响应解析器软件对象 514 能识别数据正以其返回的格式，以及将那个数据翻译成可由智能数据同化系统使用的有意义的语义上下文并增加表示包括在 LSO 中的语义和原子对象的运行时间实例。例如，数据供给器可返回 SQL 记录集或字符分隔文本文件，在任何一种情况下，响应解析器将解析记录集或字符分隔文本文件并根据 LSO 所指定的，生成必要的原子对象。

请求编译器通过在协议层编译请求，抽象请求信息的结构。例如，请求编译器任务创建能立即输送到连接适配器的内容以便执行。例子请求编译器是用于编译用于在 HTTP 通信链路上传输的请求的 HTTP 请求编译器以及用于在 Telnet 通信链路上传输请求的 Telnet 请求编译器。

请求编译器对语义对象起作用以创建请求查询。其结合描述符信息创建定制请求串。请求编译器能利用不同方案来实现此操作。在根据本发明的请求编译器的实施例中，请求编译器利用本机对象和本机描述符。本机对象保存请求如何看待特定协议和数据供给器的实现细节。因此，本机对象是描述表示搜索请求的本机查看的对象的接口。例如，HTTP 请求编译器利用 URL 本机对象。URL 本机对象隐藏构造 HTTP 协议使用的 URL 查询串的细节。请求编译器利用执行该工作的从属类。然后，请求编译器变为象容器类来执行动作或控制器。

本机描述符包括描述数据结构的本体论信息。例如，如果发送给数据供给器的请求需要固定长度的分组，那么本机描述符包括用来

创建固定长度请求的数据结构的描述。在本机描述符中使用先前描述的语义描述符来描述本体论信息。

响应解析器抽象来自数据供给器的数据流和本体论对象的结构解析。响应解析器集中在数据流返回的格式上，而不是检索数据流的协议。例如，如果数据流是字符分隔、二进制、位置等等，解析器将处理那些复杂性并构造用来增加包括在 LSO 中的智能数据同化本体论的运行时间实例的原子和语义对象。

响应解析器生成诸如原子或语义对象的本体论对象。为实现此，解析器通过解析器描述符的语义描述符迭代将说明的新结构。当遇到新的数据结构时，从数据流抽取数据以及创建本体论对象并将其增加到智能数据同化本体论的运行时间实例中。解析器描述符包括描述利用语义描述符的数据结构的本体论信息。当便于将数据转换成串数组或令牌时，还使用流和串令牌器(tokenizer)。

数据采集层包括用来实施搜索的数据采集适配器软件对象。数据采集适配器管理将参数转化成搜索请求 516，将该请求提供给连接层中的适当的连接协议，获得结果 518，管理将数据流转化成语义 XML 结构并将该结果返回到智能数据同化系统中。

在操作中，数据采集适配器初始化请求编译器、响应解析器，以及连接适配器。只要那个操作发生，适配器具有与这些组成部分的每一个交互作用的任务以便编译请求，将查询发送给数据供给器以及解析应答和增加具有由应答生成的知识实例的本体论。例如，采集适配器在初始化时生成请求编译器，响应解析器以及连接适配器。数据采集适配器通过在请求编译器上执行方法调用来“编译”请求。数据采集适配器从请求编译器获得输出并将该响应输送给连接适配器。当连接适配器返回时，诸如输入流的输出输送到响应解析器，在此处，由输入流产生语义和原子对象。然后，将语义和原子对象传递回增加智能数据同化本体论的 LSO。

连接层包括使用各种通信和数据供给器存取协议，用于将数据分组 520 传送到外部数据供给器 521 和并从其接收数据分组 522 的连

接适配器。智能数据同化系统能使用各种协议，诸如 HTTP、各种 RDBMS 协议，诸如 Oracle Net 或 SQL Net，以及适合于特定数据供给器的其他协议，连接到外部数据供给器。智能数据同化系统理解到经 HTTP 的请求可需要建立插件或对数据库的请求可需要生成可用的特定驱动机。LSO 的连接层提供该能力。然后，将请求传送给数据供给器，在其执行它并获得搜索结果。

连接适配器隐藏在数据流中找出的传输特定头部。连接适配器还连接来自数据供给器的响应。例如，如果搜索产生三个结果，那么，返回的数据流应当包括连接在一起的三个结果。

连接适配器在连接适配器环境和外部数据供给器间建立链接。例如，连接适配器已知外部数据供给器 IP 地址、端口号以及在连接数据供给器中需要什么信息以便能发送数据。连接适配器利用各种从属类来完成连接适配器的任务。来自外部数据供给器的响应包括在输入流中。在根据本发明的连接适配器的一个实施例中，在运行时间前，建立对数据供给器的连接以便连接适配器能将请求数据发送给外部数据供给器，而不重新与数据供给器建立用于每个请求的连接。

在 LSO 的每个层，有多个可用数据供给器接口组成部分，以及实现数据供给器接口的特定变形的每个数据供给器接口组成部分。在根据本发明的 LSO 的实施例中，使用提供具有在运行时间，将数据供给器接口组成部分的各种组合布置到 LSO 的能力的 LSO 的企业 Java Bean (EJB) 打包，实现 LSO。

因为 LSO 具有在运行时间确定的 LSO 的配置参数，不由单个 LSO 限制在智能数据同化系统本体论中定义的对象。因此，语义对象能横跨多个 LSO，并具有当调用时，语义对象是智能的，足以连接、抽取和同化来自多个异类数据供给器的数据的能力。语义对象横跨多个 LSO 的能力允许智能数据同化系统的用户操作表示来自多个数据供给器的原子数据或抽象的同化数据的单个结构。

图 11 是根据本发明，描述类属 LSO 的实施例的分层体系结构以及如何使用该分层体系结构来生成具有不同连接能力的不同 LSO

的图。在图 10 中已经描述过层中的对象间的操作。如前所述，LSO 包括翻译层 502、采集适配器层 504 以及连接适配器层 506。如前所述，翻译层通过采集适配器层，通过连接适配器层，通过连接层，向下将 workflow 任务数据请求传送给数据供给器。数据供给器通过访问数据库 503 来响应数据请求以满足数据请求以及将满足该请求的数据传送给 LSO。然后，LSO 重新格式化所请求的数据并编译包括在由智能数据同化系统使用的本体论中的前面所述的语义对象以响应来自外部数据客户端的数据请求，如前所述。

在根据本发明的 LSO 的一个实施例中，采集层包括封装在前描述的数据存取对象 528，诸如在前描述的请求编译器和响应解析器的 EJB 打包子层 526。连接适配器层包括可操作地连接到高级协议子层 532 的语言特定的协议子层 530。高级协议子层位于低级协议子层 534 的上面。

在适合连接到以数据库 501 的形式的外部数据供给器的数据库 LSO 的实施例中，数据库 LSO 包括数据库数据存取对象 505，其包括 Java 数据库连接 (JDBC) API 以执行 SQL 语句，从而允许数据库 LSO 与 SQL 兼容的数据库交互作用。JDBC 数据存取对象使用在语言特定的协议子层中的 JDBC 连接库 507 以及在高级协议子层中的网络化数据库驱动器 509，使用物理通信介质，诸如固定网络链路 511 或电话拨号链接 513，通过 TCP/IP 低级协议子层 547 来连接到外部数据库上。

在适合于连接到以 Web 服务器 539 的形式的外部数据供给器的 Web 服务器 LSO 的实施例中，Web 服务器 LSO 包括用于在适合于使用 HTTP 通信的通信网络上连接到数据供给器的 HTTP 数据存取对象 515。HTTP 数据存取对象使用在语言特定协议子层中的 HTTP 连接库 517 以及在高级协议子层中的 HTTP 驱动器 API 519 以便使用物理通信介质，诸如固定网络链路 511 或电话拨号链路 513，通过 TCP/IP 低级协议子层 547，连接到外部 Web 服务器。

在适合于连接到以 FTP 文件服务器 541 的形式的外部数据供给

器的 FTP 服务器 LSO 的实施例中，FTP 服务器 LSO 包括用于在适合于使用 FTP 通信的通信网络上连接到数据供给器的 FTP 数据存取对象 521。FTP 数据存取对象使用语言特定协议子层中的 FTP 连接库 523 以及高级协议子层中的 FTP 驱动器 API525 以便使用物理通信介质，诸如通过 TCP/IP 低级协议子层 547 的固定网络链路 511 或电话拨号链路 513，连接到外部 FTP 服务器上。

在适合连接到以文件系统 543 的形式的外部数据供给器的文件 LSO 的实施例中，文件服务器 LSO 包括用于连接到数据供给器文件系统的文件数据存取对象 521。文件数据存取对象使用语言特定协议子层中的文件 I/O API523 以及高级协议子层中的文件系统 API525 以便连接到存储在物理介质，诸如软盘 551 或磁带 553 上的文件系统 543。

在适合连接到以人为参与者 545 的形式的外部数据供给器的手动输入 LSO 的实施例中，手动输入 LSO 包括用于管理手动输入数据的对列列表数据存取对象 521。手动输入存取对象使用在语言特定协议子层中的应用程序 537 以及以表单形式的用户界面和以高级协议子层中的屏幕 539 以通过手动记录输入装置，诸如屏幕和键盘 549 连接到人为参与者 545。

应用程序服务器通过从外部系统接受服务请求以及用履行过的服务请求响应，提供具有智能数据同化系统的接口。应用程序服务器还是确保可靠地执行智能数据同化系统接收的所有服务请求的那个过程。应用程序服务器接受请求有关由智能数据同化系统管理的状态或信息的信息响应的各种服务请求或初始化通过示例工作流的服务请求的服务请求。尽管 workflow 机制支持创建包括多个 LSO 的自动化 workflow 的能力，而没有任何人为干预，workflow 机制还支持挂起 workflow 并允许发生将数据送回系统，然后重新 workflow 执行的手动处理的能力。

通过智能数据同化系统公开的 API，将服务请求提交给智能数据同化系统。根据不同数据供给器，履行这些服务请求。只要履行服务请求，智能数据同化系统将可选地提供该结果，直到由提交服务请

求的专门系统请求为止。搜索结果能保留在智能数据同化系统中达可配置时间周期。

智能数据同化系统 API 结构建立在经各种传输机制接收 XML 消息的能力之上。在根据本发明的智能数据同化系统的一个实施例中，智能数据同化系统的处理层处理下述传输：

Java 消息服务 (JMS)

超文本传输协议 (HTTP)

远程消息接口 (RMI)

简单的邮件传输协议 (SMTP)

处理层接受根据每个传输协议发送的文件。每个文档包括含服务请求的 XML。智能数据同化系统提供 EJB 打包给提供经所请求的传输，编译 XML 消息和发送 XML 消息的接口的各个智能数据同化系统代理。构造配置描述符文件设置由打包组件利用的传输方法。

存取智能数据同化系统的服务的数据客户端能选择使用打包组件来便于编译和发送消息或可另外选择直接与智能数据同化系统代理通信。

发送给逻辑搜索对象应用服务器的每个消息产生从连接到智能数据同化系统的智能数据同化系统获得响应。响应的类型由提交消息的类型而定。在根据本发明的智能数据同化系统的一个实施例中，所使用的消息包括：服务请求、状态请求、服务信息请求以及取消请求。在智能数据同化系统的实施例中，由数据客户端传送给智能数据同化系统的消息作为 XML 文档。

每个消息包括含有助于由数据客户端的观看识别数据客户端的消息类型、消息源和消息的唯一性的属性的外部消息打包。

在下述表中描述基本消息要素：

message	打包整个智能数据同化系统消息的要素
message # client-id	包括识别消息的发送者的数据客户端标识符的属性
message # type	包括智能数据同化系统消息的类型的属性
data	包括消息的数据的要素

服务请求消息启动对智能数据同化系统中服务的请求。服务请求消息包括两个外部要素“message”和“data”。“message”要素包括识别消息的源点和类型的属性。

“data”要素包括有关所请求的何种服务的信息。服务请求可包括对服务的一个或多个请求，以及对每个请求的“product”要素。产物要素包括表示所请求的哪种服务的属性的数量。在“product”要素内，有表示用于请求的参数、格式化选项以及通知选项的多个标记。

下述表包括在服务请求消息中的要素的描述：

service-request	打包单个服务请求的要素
service-request#message-id	用来识别该请求消息的唯一标识符
service-item	表示单个服务请求项的要素
service-item#sku	表示已经请求何种服务的标识符
service-item#item	用来区别相同消息中的服务请求的标识符
parameter	表示参数的要素
parameter#name	参数名称
parameter#value	参数值
notification	表示通知选项的要素，可包括多个参数要素
notification#type	表示所请求的通知类型的属性
format	表示将用于该要素的格式化选项的要素。如果其在通知块中，仅将其应用于通知人
format#style	用来格式化该通知的式样表
result-format	包括与该消息的最终输出有关的选项的要素。包括前面描述过的格式要素
results	工作流用来增加结果数据的要素

现在参考图 14，示出了示例性的 XML 文档服务请求消息 100。如前所述，示例性 XML 文档服务请求消息包括外部消息要素 1101 和内部数据要素 1102。包括在数据要素中的是包括 service-item 要素

1106 的 `service-request` 要素 1104。 `service-request` 要素包括两个参数要素 1108 和 1110，定义名字和姓，作为搜索请求的一部分。数据要素进一步包括当准备好由服务请求所产生的数据产物时，指定用来通知数据客户端的通知类型的 `notification` 要素 1112。数据要素进一步包括 `result-format` 要素，用于指定数据产物 1114 的格式。 `results` 要素包括在数据要素中，作为用于响应服务请求，将发送给数据客户端的最终数据产物的位置存储器。

状态请求消息是用于智能数据同化系统的请求以便提供在前发送的服务请求的状态信息。

下表包括在状态请求消息中的要素的描述：

<code>status</code>	打包单个状态请求的要素
<code>status#message-id</code>	用来识别该状态消息的唯一标识符
<code>status-request</code>	表示单个状态请求项的要素
<code>status-request#item</code>	用来区别相同消息中的请求的标识符
<code>IDAS-id</code>	数据客户端需求状态信息的智能数据同化系统请求的标识符

图 15 是根据本发明，示例性状态消息的 XML 文档实施例。如前所述， `messages` 要素 1118 包括 `data` 要素 1120， `message` 要素包括表示这是状态消息的类型属性 1119。 `data` 要素包括含多个 `state-message` 要素 1124 的状态要素 1122，每个具有其自己的项和标识符属性。

取消请求消息是用于智能数据同化系统停止处理在前服务请求的请求。在智能数据同化系统有机会处理该取消请求消息前，已经结束将取消的处理是可能的。

下述表包括在取消请求中的要素的描述：

<code>cancel</code>	打包单个取消请求的要素
<code>cancel#message-id</code>	用来识别该取消消息的唯一标识符
<code>cancel-request</code>	表示单个取消请求项的要素
<code>cancel-request#item</code>	用来区别相同消息中的取消请求的标识符
<code>idas-id</code>	数据客户端正取消的智能数据同化系统请求的标识符

图 16 是根据本发明，示例性取消消息的 XML 文档实施例。示例性取消消息的 XML 文档实施例包括在前描述过的包括在前描述过的 data 要素 1128 的 message 要素 1126。message 要素的类型属性是表示这是取消消息的“Cancel”1127。data 要素包括含多个 cancel request 要素 1132 的 cancel 要素 1130。

由数据客户端将输出消息发送给智能数据同化系统来请求在前传送的服务请求的结果。下述表包括在输出请求消息中的要素的描述：

output	打包单个输出请求的要素
output#message-id	用来识别该输出消息的唯一标识符
output-request	表示单个输出请求项的要素
output-request#item	用来区别相同消息中的输出请求的标识符
idas-id	你希望从其检索数据的智能数据同化系统处理的标识符
format	表示将使用的格式化选项的要素
format#style	用来格式化输出的 XML 式样表

图 17 是根据本发明的示例性输出消息的 XML 文档实施例。示例性输出消息的 XML 文档包括含在前描述过的 data 要素 1136 的 message 要素 1134。data 要素包含 output-request 要素的 output 要素 1138。output-requests 要素包括识别用于数据客户端请求结果的服务请求的标识符属性 1141。format 要素 1142 指定使用式样属性 1144 的所请求数据的格式。

如果正确构造和格式化请求的话，响应消息表示对智能数据同化系统的请求所做出的响应。响应包括产物数据、搜索结果数据或状态数据。其还包括请求是否成功的指示器。

下述表包括在响应消息中的要素的描述：

message-type	表示消息类型。对响应消息来说，其总是 “Response”
service-response-message-id	用来识别这是对消息的响应的唯一标识符
response-item	来自对请求消息的响应的 line-id
result	表示响应的处理 “Success” 或 “Error” 的属性
request	响应的请求类型
idas-id	与响应有关的智能数据同化系统的标识符
response-value	包括用于那个特定响应的响应数据的任意标记

图 18 是根据本发明，示例性响应消息的 XML 文档实施例。示例性响应消息的 XML 文档实施例包括含在前描述的 data 要素 1148 的在前描述的 message 要素 1146。message 要素包括表示消息是响应消息的类型属性 1149。data 要素包括含 response 要素 1152 的 service-response 要素 1150。response 要素包括表示示例 workflow 任务来满足所属服务请求的日期和时间的 response-value 要素 1154。

选择消息包括对用户选择的响应。当从选项列表做出选择时发布，以及消息封装该结果。

下述表包括在选择消息中的要素的描述：

selection-results	包括选择结果数据的要素
selection-results#message-id	包括消息标识符的属性
additional-data-item	包括单个选择结果的要素
additional-data-item#item	表示项目数的属性
additional-data-item#idas-id	选择所引用的智能数据同化系统标识符
selected	表示选择项的要素
selected#item	给出所选择的项的属性

使用产物信息消息来请求有关由智能数据同化系统提供的一个或多个服务的信息。对产物信息的响应包括描述服务、可用格式化选项、用于输入参数的智能数据同化系统本体论以及用于所期望的输出

的智能数据同化系统本体论的信息。

在根据本发明的智能数据同化系统的实施例中，数据客户端经应用服务器访问智能数据同化系统的服务。应用服务器提用于在各种通信网络上与数据客户端通信的相容接口。应用服务器经多种通信网络，从数据客户端接收搜索请求消息，将搜索请求消息转化成用于由智能数据同化系统使用的相容内部格式，以及经通信网络协调对数据客户端的响应的传输。

图 12 是根据本发明，示例说明当由应用服务器的实施例接收请求消息时发生的事件顺序的时序图。应用服务器包括处理层 1000 和服务器层 (未示出)。处理层包括协议指定的消息提取器 1004、请求处理器 1010，请求事件翻译器 1014。数据客户端传送 1002 包括以 XML 文档形式的在前描述过的请求消息的协议指定的消息 1003。例如，数据客户端可以是 Web 浏览器以及协议指定的消息可以是经 Internet，使用 HTTP 传送的布告。在根据本发明的处理层的实施例中，所支持的协议包括 HTTP，SMTP，RMI，和 JMS。

由协议特定提取器 1004 接收协议指定的消息以及协议指定的提取器抽取 1006 所嵌入的 XML 文档请求消息 1008。将 XML 文档请求消息传送给请求处理器 1010。请求处理器解析 1009 XML 文档请求消息并抽取属性以便确定将示例的事件处理机的类。请求处理机然后用所抽取的属性和 XML 文档请求消息创建事件处理机。XML 文档请求消息 1012 然后上传到服务器层，在此处由请求事件翻译器将其分解成用于工作流的各种组件 1016。

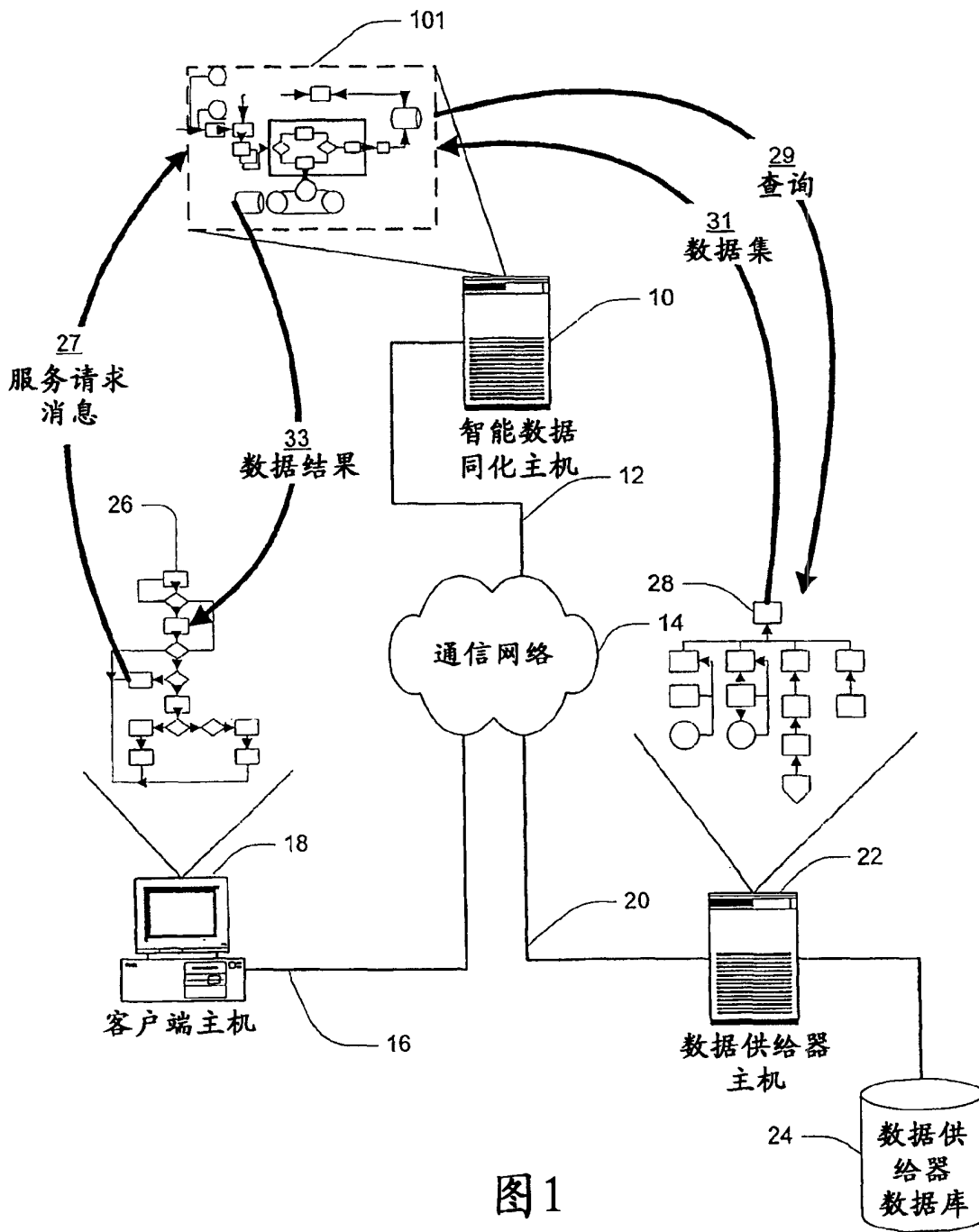
图 13 示例说明根据本发明，如何连接智能数据同化系统的实施例。应用服务器包括含控制智能数据同化系统的操作的服务器控制器的服务器层。响应通过数据客户端，传送给应用服务器的请求消息，由在前描述过的处理层 1000 内生成的多个服务器事件 902 启动智能数据同化系统内的操作。状态管理器 905 协调响应多个服务器事件的多个服务器事件处理机 908 的动作。反过来，服务器事件处理机使用在前描述过的工作流机 910 的服务来创建在前描述过的工作流的实

例。 workflow 包括调用在前描述的 LSO914 的任务节点 912。由存储于在前描述过的元数据存储 916 中的一组在前描述过的描述符定义 LSO 的属性。 LSO 包括在前描述过的翻译器 918，用于与 workflow 任务交互作用，以及在前描述过的采集适配器，用于编译查询和解析响应，以及在前描述过的连接适配器 922，用于连接到外部数据供给器。

使用适当的通信链路，诸如在前描述过的通信链路 924，928 和 932， LSO 实现到外部数据供给器诸如数据供给器 926，930，和 934 的连接。如前所述， LSO 收集数据集和创建封装该数据集的语义对象。 workflow 使用语义对象来将数据结果传送给结果数据库 938。

智能数据同化系统管理在结果数据库中，以 XML 格式的数据结果。产物定义识别数据结果将存在于结果数据库中的持续时间。在智能数据同化系统中继续存在的数据结果能用于归档目的。根据包括在请求消息中的格式标记，使用格式器 940 的 XSL 式样表，将数据结果再现成用于传送给数据客户端的适当格式。服务控制器接收由格式器产生的所再现的数据并将所再现的数据结果传送 942 给数据客户端，作为对数据客户端的服务请求消息的响应。

尽管在某些特定的实施例中已经描述了本发明，对本领域的技术人员来说，许多另外的改进和修改是显而易见的。因此，应理解到，除如特别描述的外，可实施本发明。因此，在各个方面，应当将本发明的实施例视为示例性和非限制性，由本申请支持的由权利要求以及权利要求的等效限定的本发明的范围，而不是上述描述。



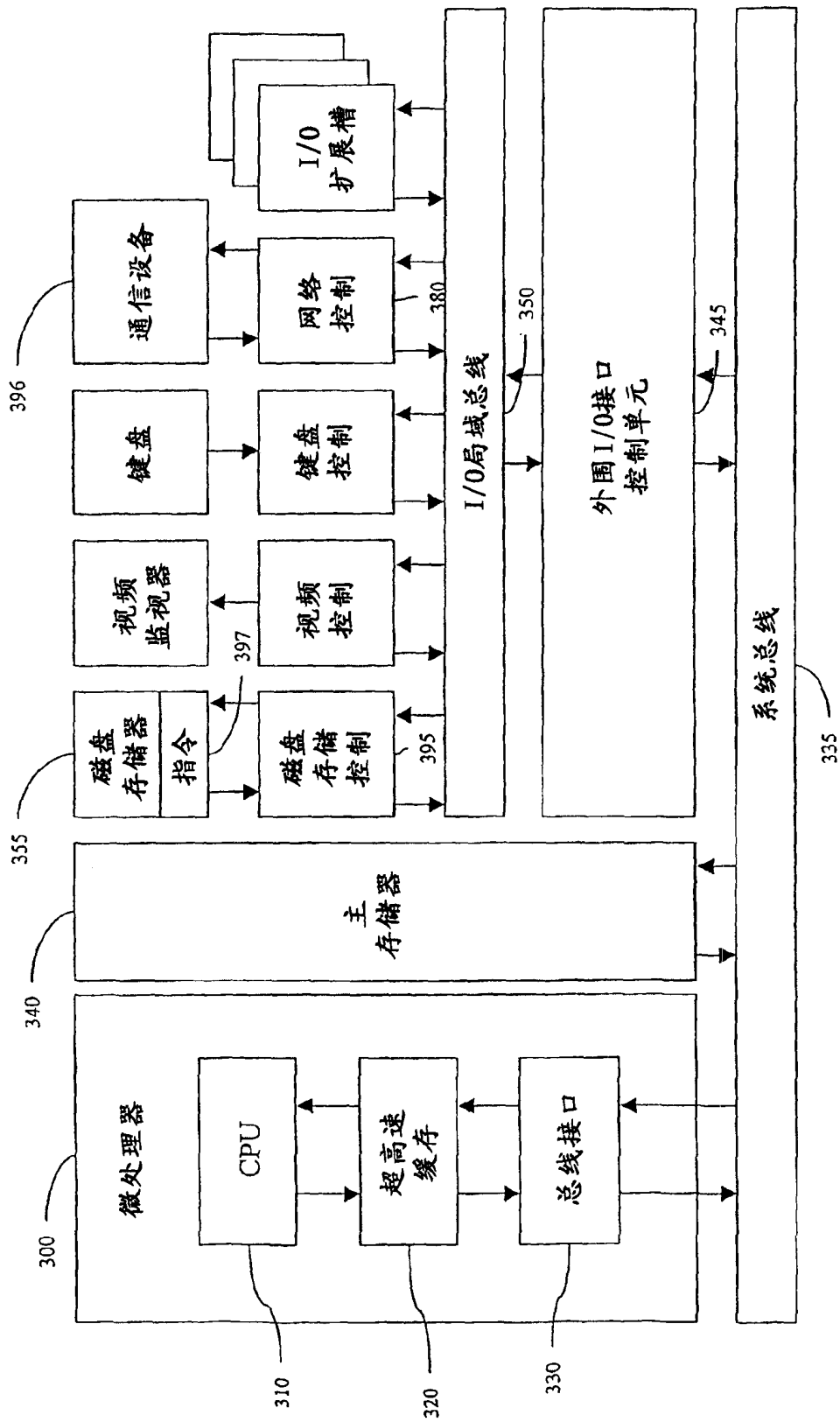


图2

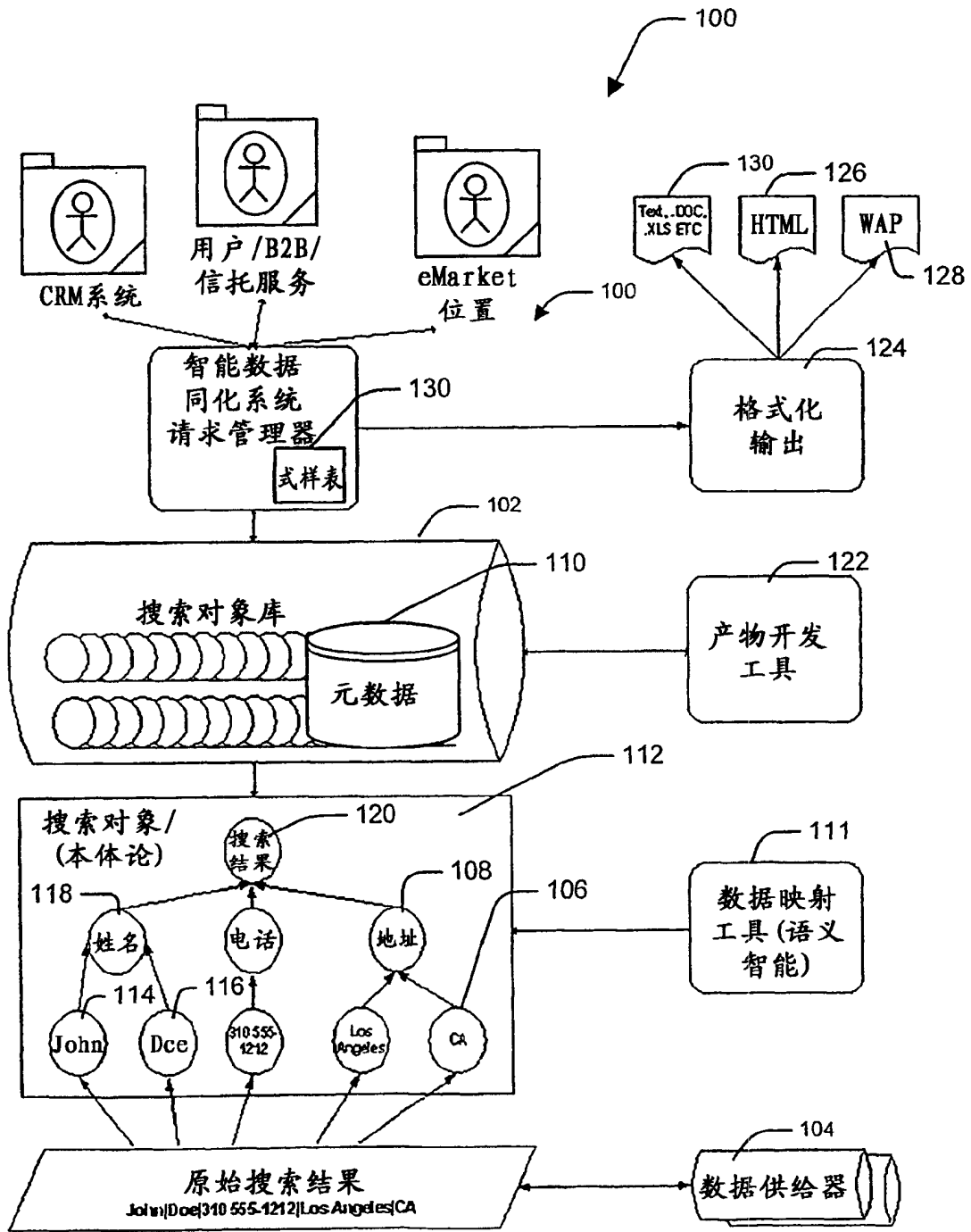


图 3

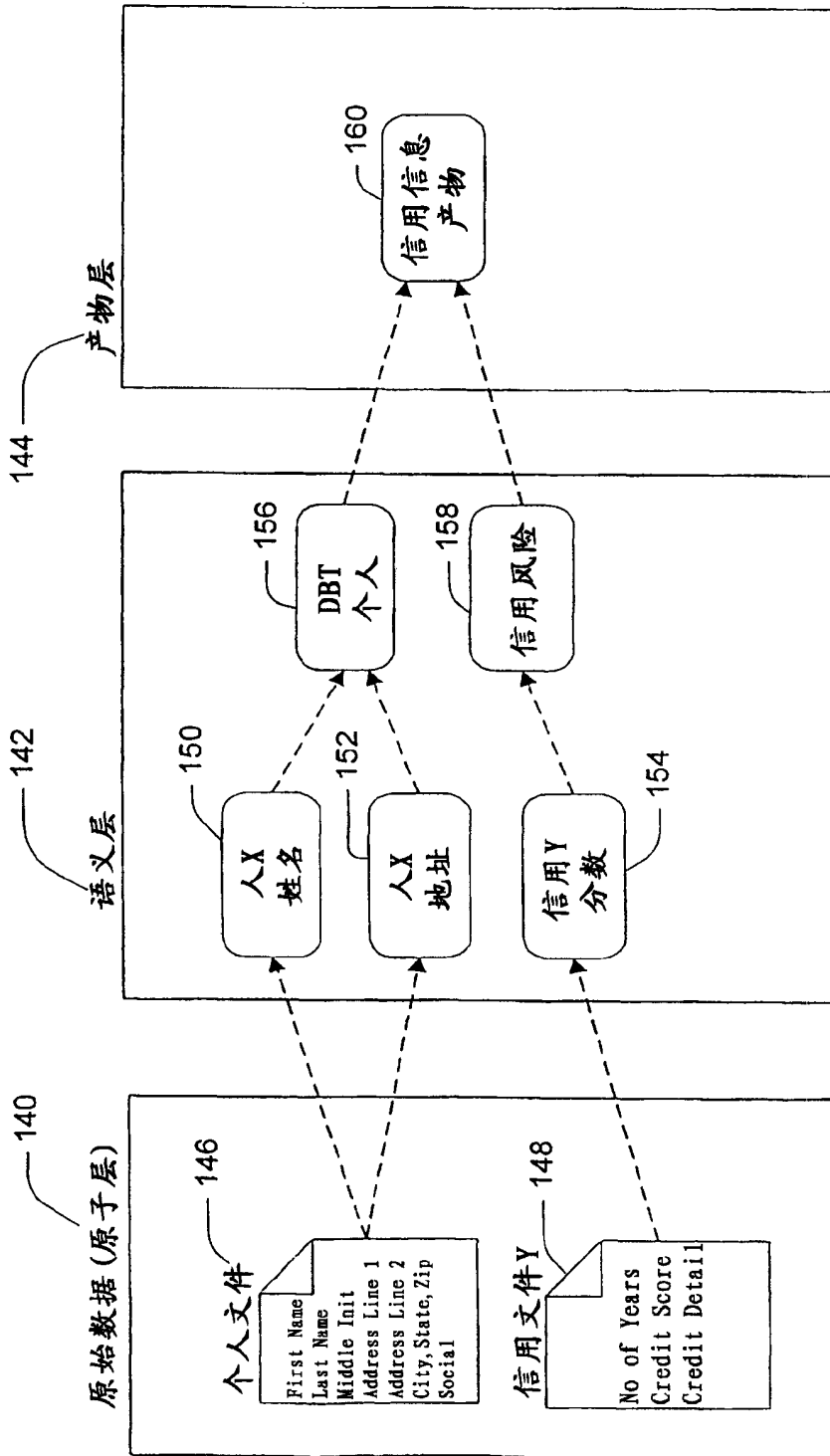


图4

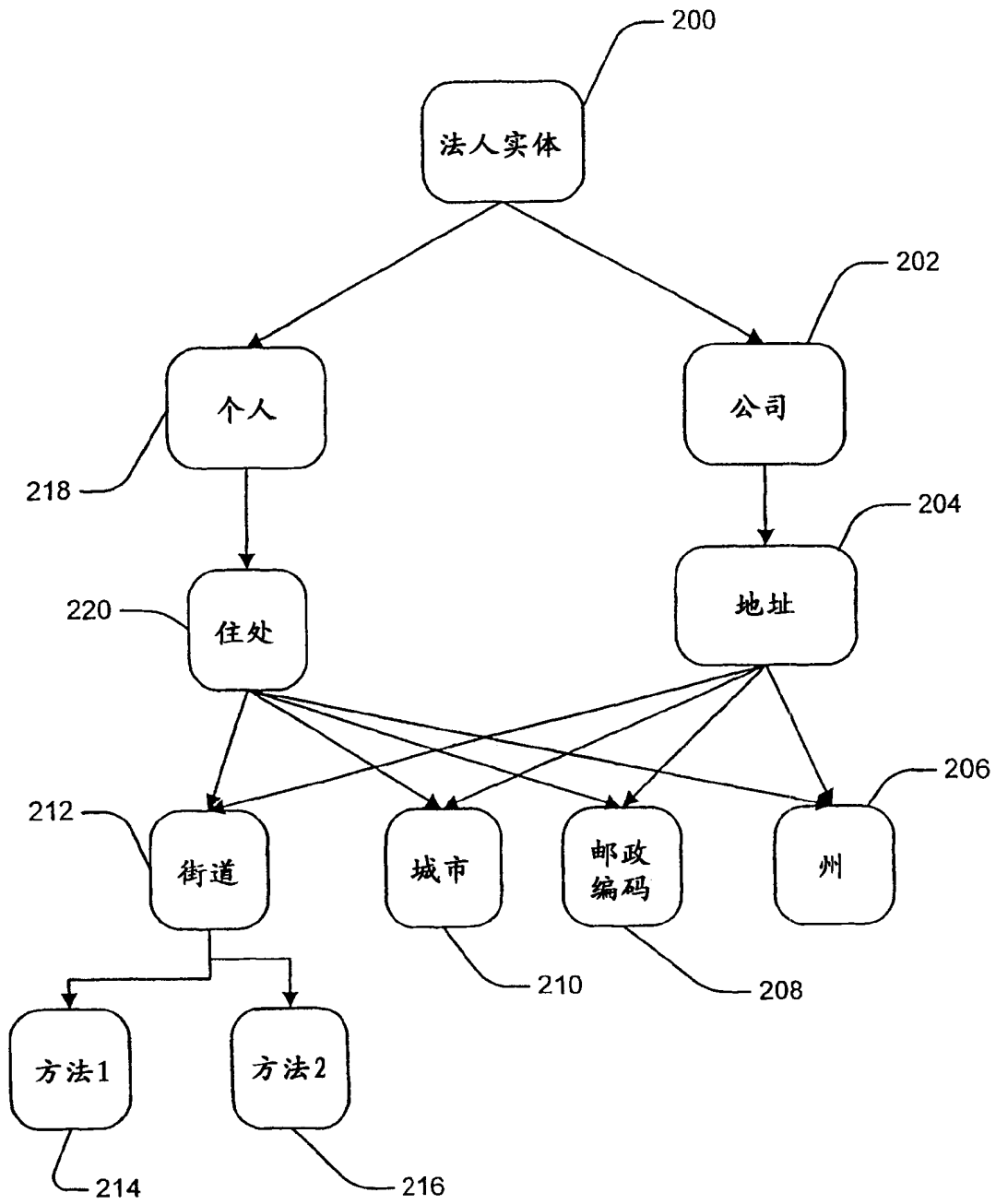


图5

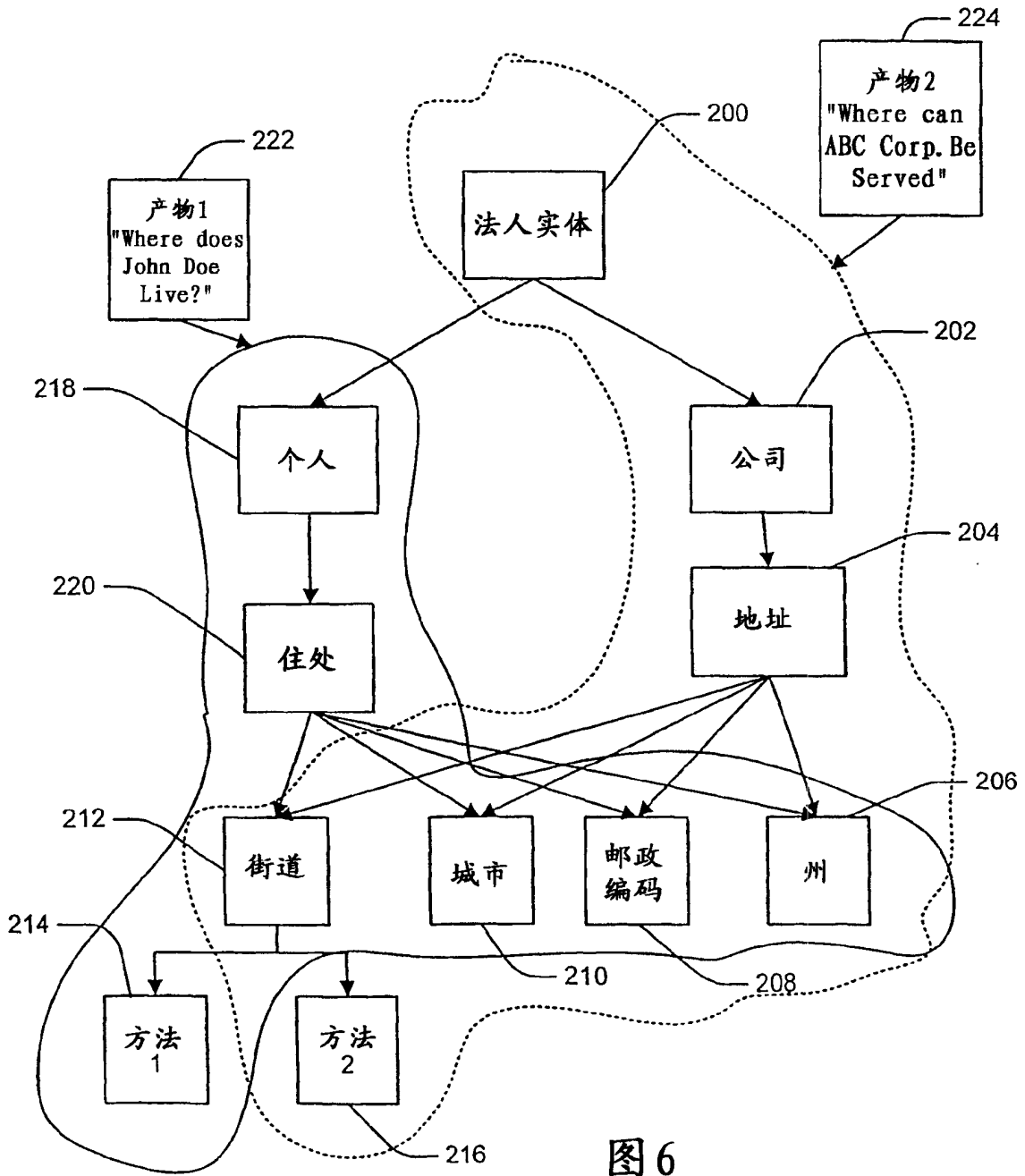


图6

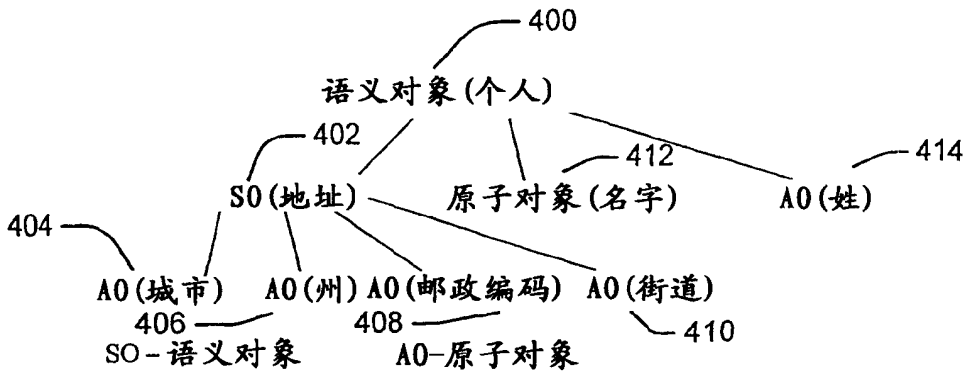


图7a

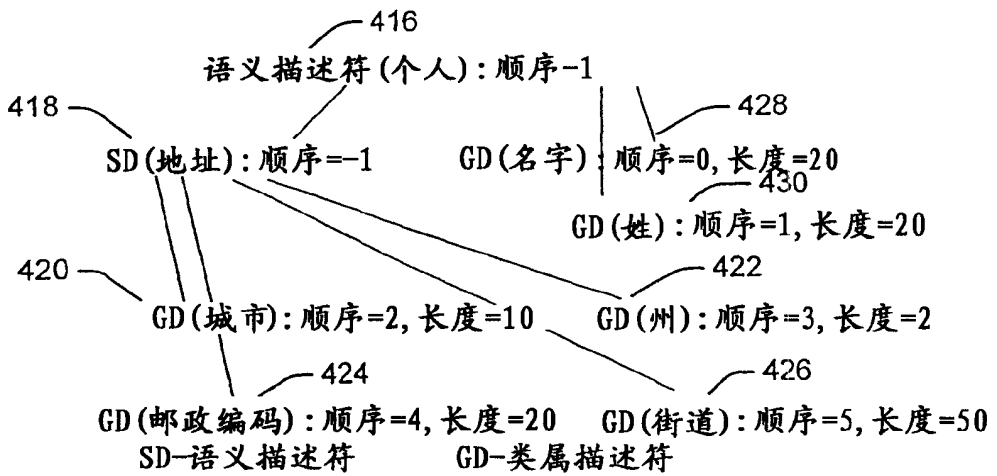


图7b

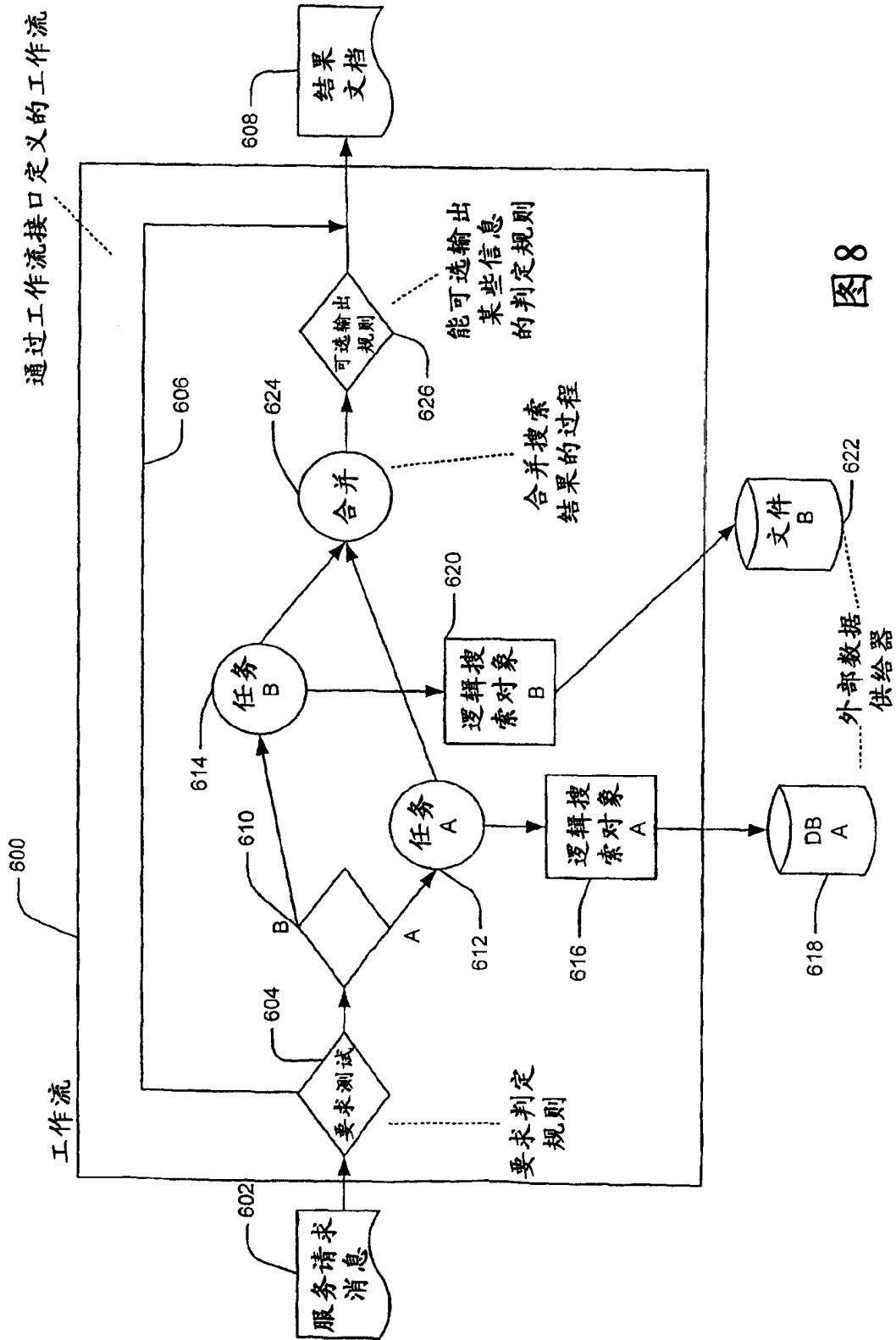


图 8

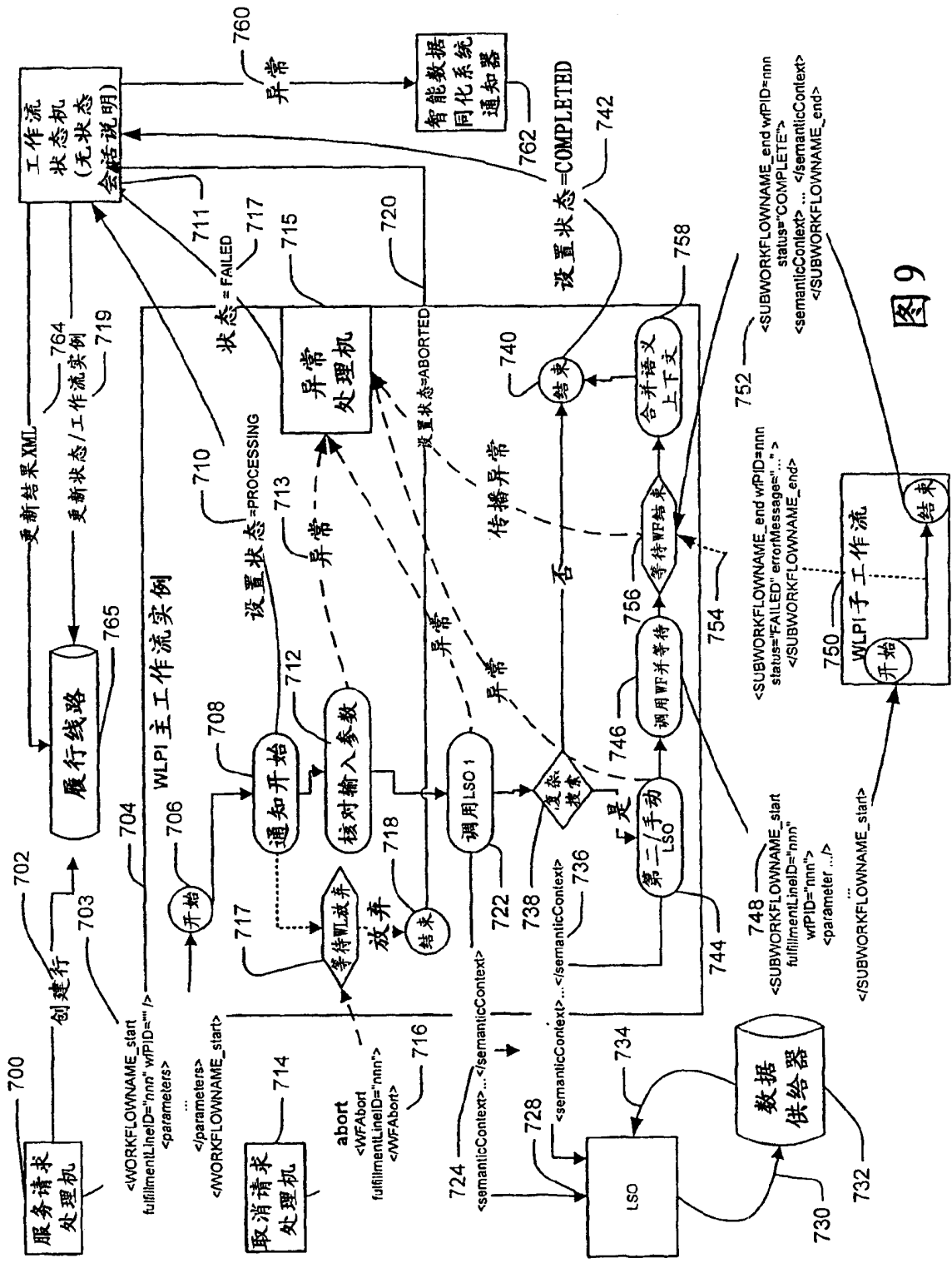


图9

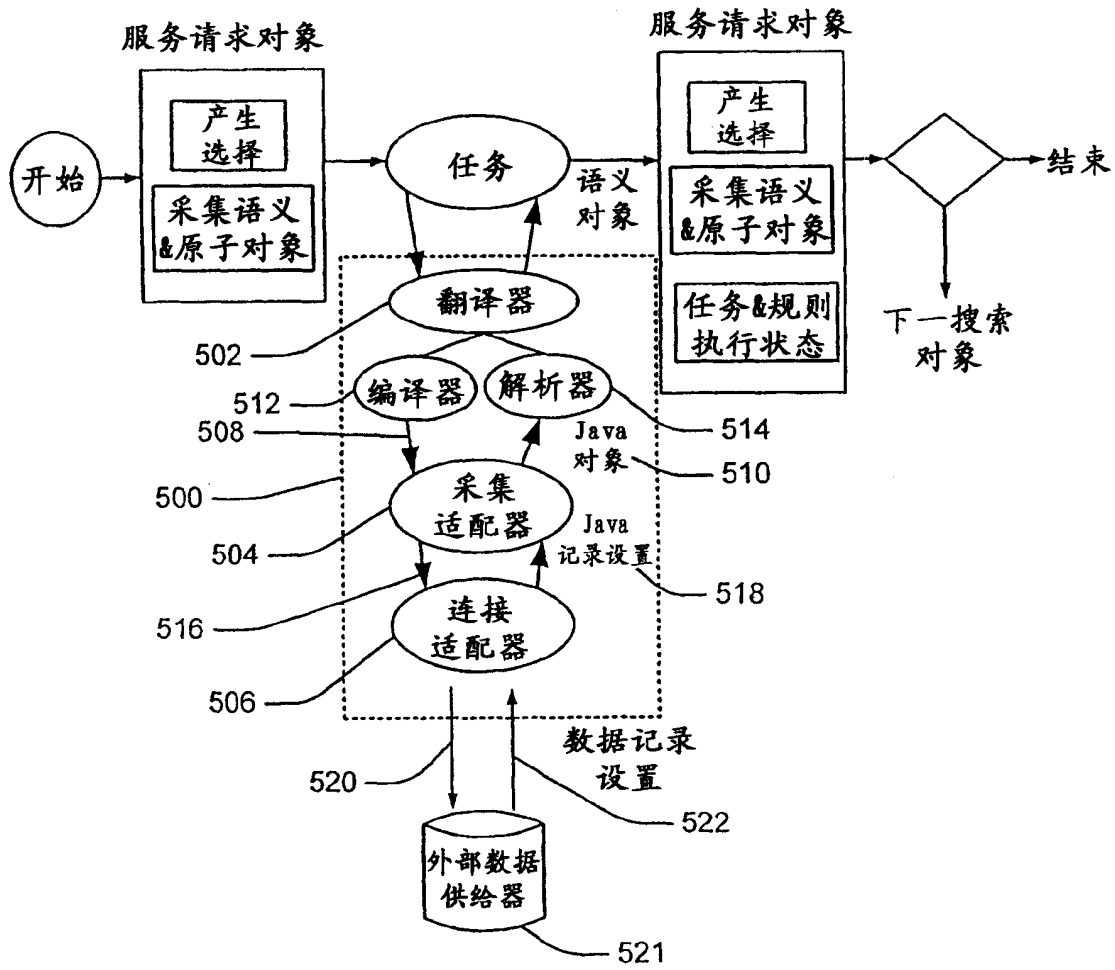
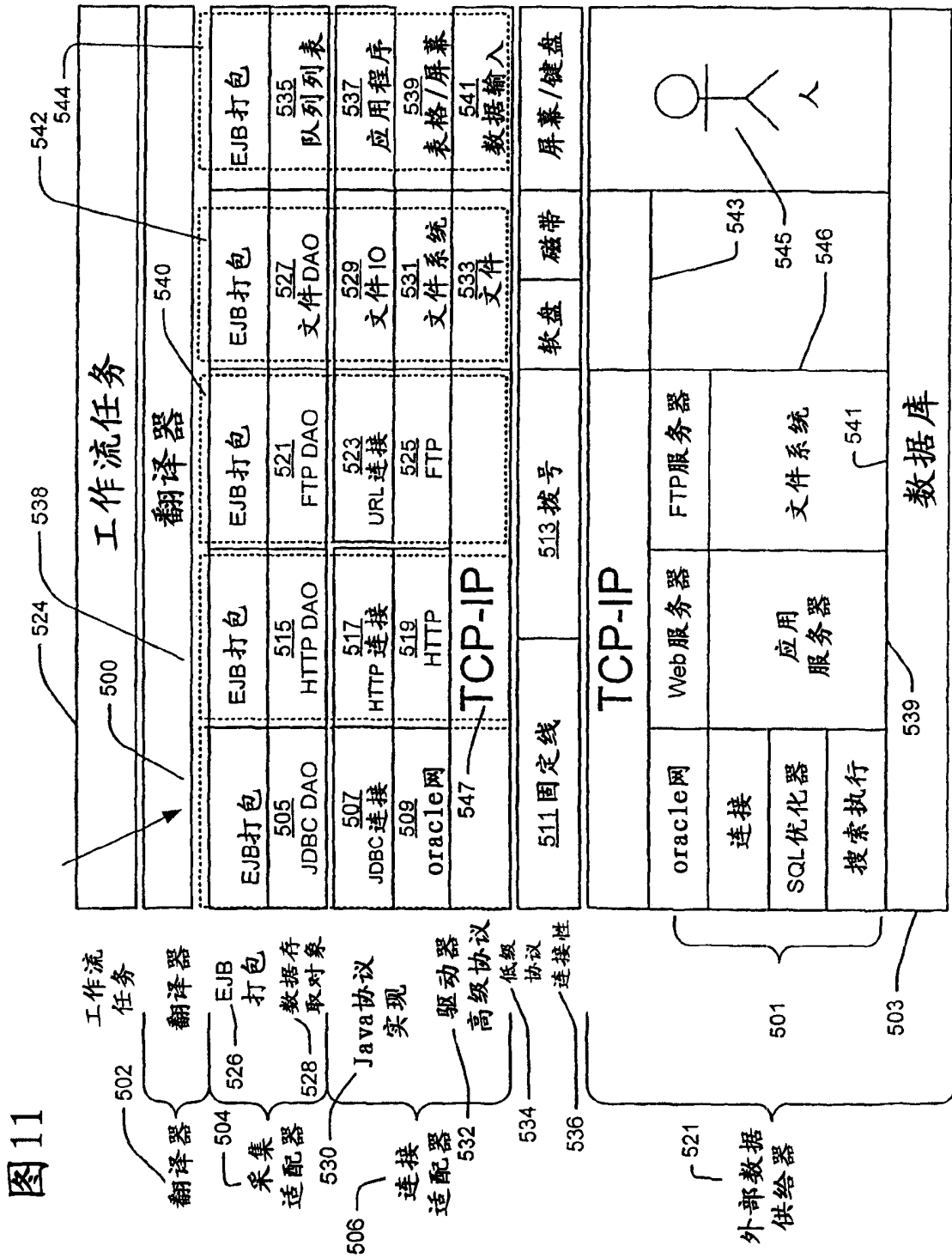


图 10



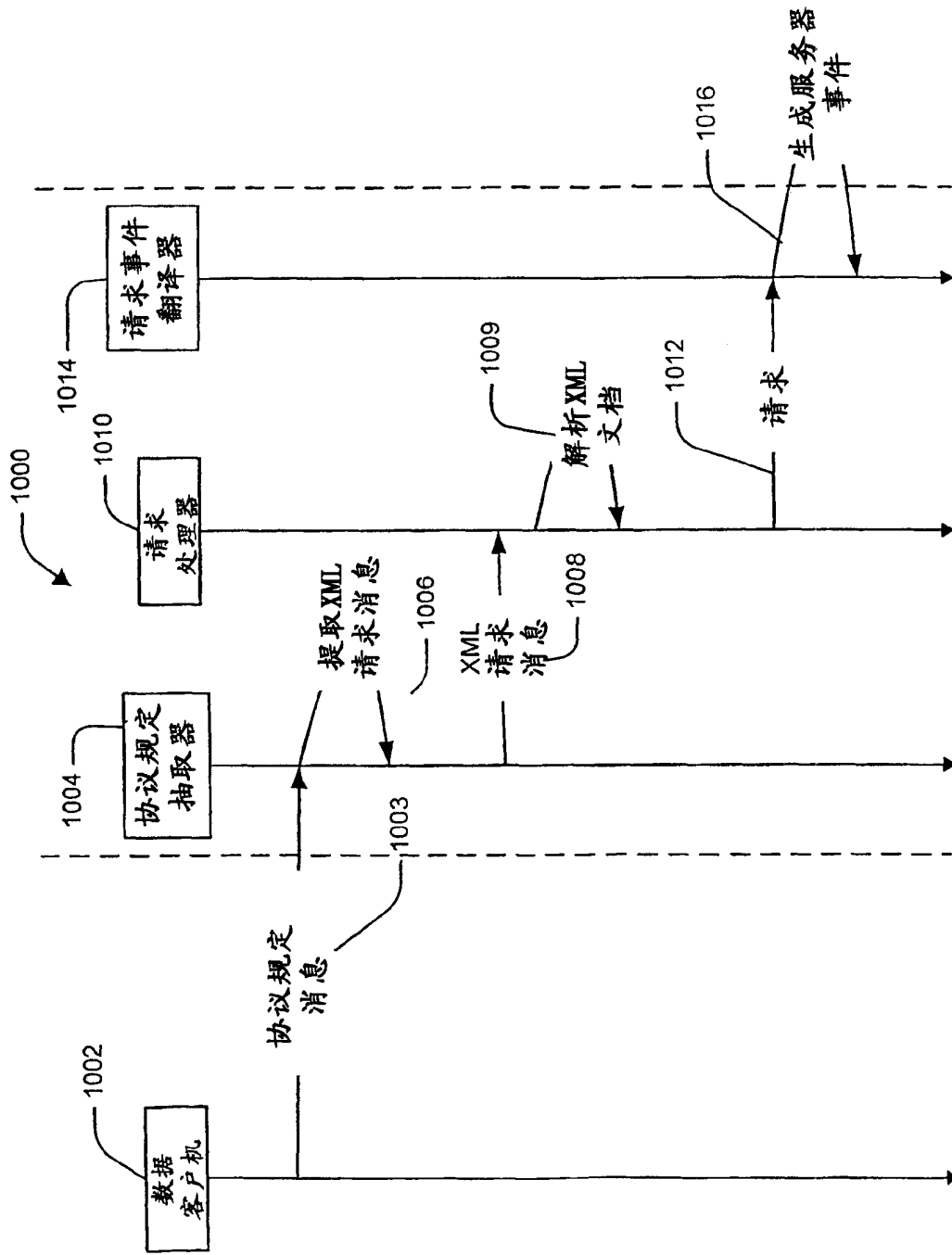


图12

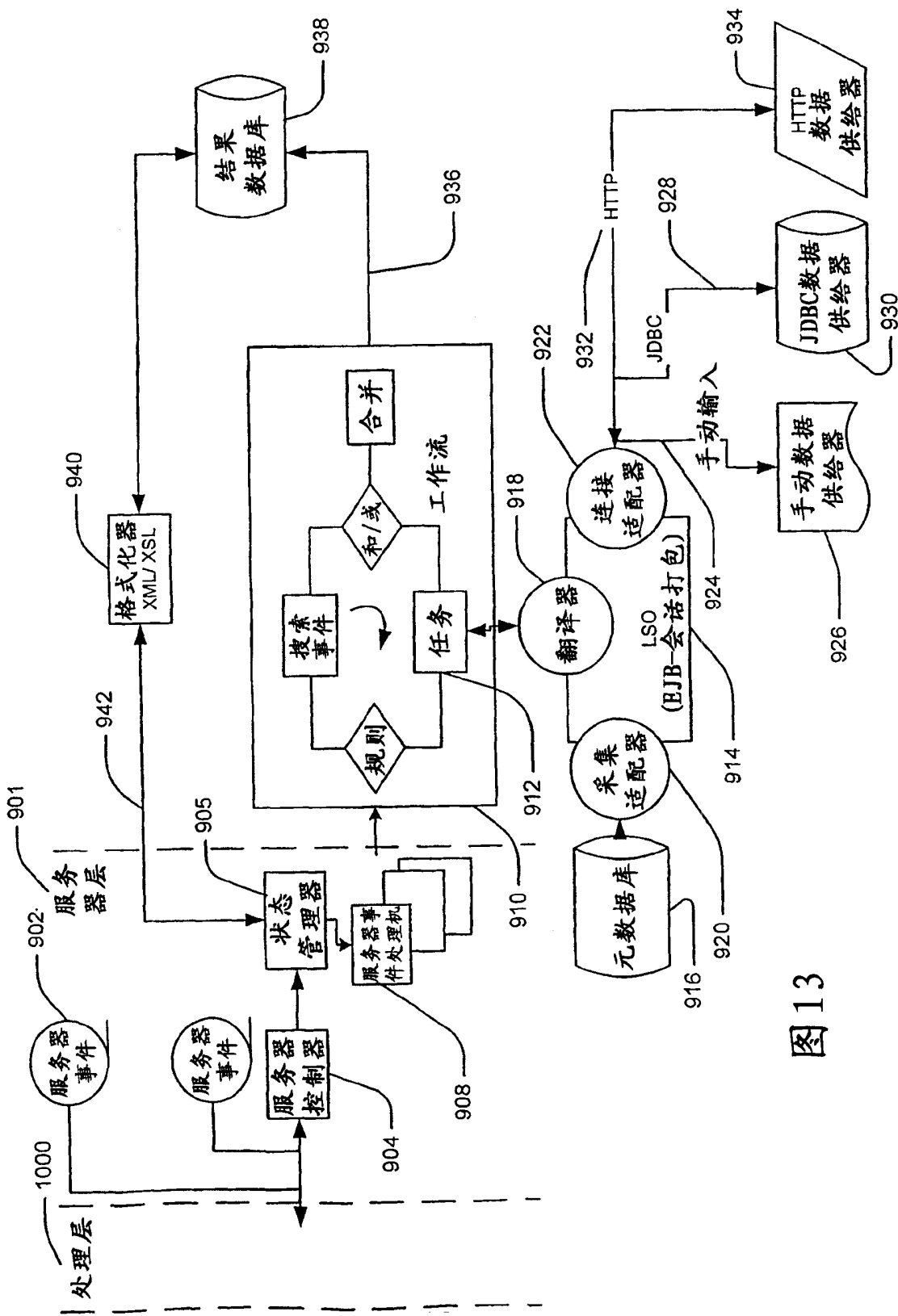


图13

采样服务请求消息

```

<?xml version="1.0" encoding="UTF-8"?>
<message type="ServiceRequest" source-id="com.searchco">
  <data>
    <service-request message-id="11001100">
      <service-item sku="CSRASearch" item="1">
        <parameter name="1"firstname1" value="John" />
        <parameter name="1"lastname1" value="Smith" />
        <notification type="email">
          <parameter name="address"
            value="John.Smith@company.com" />
          <format style="email" />
        </notification>
        <result-format>
          <format style="csra-results" />
        </result-format>
        <results>
        </results>
      </service-item>
    </service-request>
  </data>
</message>

```

Diagram annotations (IDs) in the original image:

- 1100: points to the opening <message> tag.
- 1101: points to the opening <data> tag.
- 1104: points to the opening <service-request> tag.
- 1106: points to the opening <service-item> tag.
- 1108: points to the opening <parameter name="1"firstname1" value="John" /> tag.
- 1110: points to the opening <parameter name="1"lastname1" value="Smith" /> tag.
- 1112: points to the opening <notification type="email"> tag.
- 1114: points to the opening <result-format> tag.
- 1116: points to the opening <results> tag.

图 14

采样状态消息

```
<?xml version="1.0" encoding="UTF-8"?>
<message type="Status" source-id="com.searchco" >
  <data>
    <status message-id="1010010101">
      <status-request item "1" IDAS-id="394" />
      <status-request item "2" IDAS-id="488" />
      <status-request item "3" IDAS-id="506" />
      <status-request item "4" IDAS-id="13892192312" />
    </status>
  </data>
</message>
```

The diagram shows the following line numbers for the XML tags:

- Line 1118: Opening tag for `<message type="Status" source-id="com.searchco" >`
- Line 1100: Opening tag for `<data>`
- Line 1119: Opening tag for `<status message-id="1010010101">`
- Line 1122: Closing tag for `<status message-id="1010010101">`
- Line 1124: Closing tag for `<status-request item "1" IDAS-id="394" />`
- Line 1124: Closing tag for `<status-request item "2" IDAS-id="488" />`
- Line 1124: Closing tag for `<status-request item "3" IDAS-id="506" />`
- Line 1124: Closing tag for `<status-request item "4" IDAS-id="13892192312" />`
- Line 1124: Closing tag for `<status>`
- Line 1124: Closing tag for `<data>`
- Line 1124: Closing tag for `<message>`

图 15

采样取消消息

```
<?xml version="1.0"?> 1127
<message type="Cancel" source-id="com.searchco" > 1126
  <data> 1128
    <cancel message-id="1234567890"> 1130
      <cancel-request item "1" IDAS-id="1234567890" />
      <cancel-request item "2" IDAS-id="2345678901" /> 1132
    </cancel>
  </data>
</message>
```

图 16

采样输出请求消息

```
<?xml version="1.0"?>
<message type="OutputRequest" source-id="com.searchco" >
  <data> 1136
    <output message-id="1234567890" > 1138
      <output-request item="1" IDAS-id="1234567890" />
        <format style="pdf" /> 1140
      </output-request> 1142
    <output-request item "2" IDAS-id=" 2345678901" />
      <format style=" html" /> 1141
    </output-request> 1144
  </output>
</data>
</message>
```

图 17

采样响应消息

Sample of a message return when a service request has been accepted.

```
<?xml version="1.0"?>
<message type="Response" source-id="TDAS">
  <data>
    <service-response message-id="1010010101">
      <response item="1" result="Success"
        request="Order" IDAS-id="345234234">
        <response-value>Workflow started at 01:23pm
          September 4th </response-value >
      </response >
      <response item "2" result="Success"
        request=" Order" IDAS-id="345234234" >
        September 4th, 2001 </ response-value >
      </ response >
    </service-response>
  </data>
</ IDAS-message>
```

图 18