

数据集成：历史、现状、未来

艾静 (Web 组)

引言：

本文主要部分是对论文《Data Integration: The Teenage Years》[1]的介绍，这篇论文是第32届VLDB会议(VLDB2006)上十年最佳论文的获奖发言，作者在文中总结了Data Integration这十几年来来的发展成果，在商业领域的一些相关产品，并提出了目前数据集成系统普遍存在的问题以及未来面临的挑战。

本文还对数据集成领域中的一些重要思想和几个热点问题做了更加详细的介绍，力争将数据集成这十几年来来的发展状况尽可能清晰地展现给读者。

一、背景介绍

近几十年来，计算机网络的飞速发展和信息化的推进，使得人类社会所积累的数据量已经超过了过去 5000 年的总和。数据的采集、存储、处理和传播的数量也与日俱增。企业或社会组织实现数据共享，可以使更多的人更充分地利用已有的数据资源，减少资料收集、数据采集等重复劳动和相应费用。

然而，这些为不同应用服务的信息都存储在许多不同的数据源之中，其管理系统也各不相同。为更有效地利用这些信息，需要从多个分布、异构和自治的数据源中集成数据，同时还需要保持数据在不同系统上的完整性和一致性。另外，必须向用户隐藏这些差异，提供给用户一个统一和透明的数据访问接口。研究的重点即在于确立一种具有普遍意义的、可操作性强的分布异构数据源的集成方法。

因此，如何对数据进行有效的集成管理已成为增强企业商业竞争力的必然选择，尤其是对于那些拥有多部门多数据源的大型企业来说，数据集成更是至关重要。因为每一个部门都会拥有自己的数据库，这些数据库可能是独立、异构且自治的，为了各部门间更好的合作和数据共享，并且为用户提供更好的搜索查询质量，建立一个完善的数据集成系统是极有应用价值而且尤为重要的。

二、Information Manifold：具有统一的查询借口！

1. 背景

1996年Alon Halevy、Anand Rajaraman、Joann Ordille三人合著的论文《Querying Heterogeneous Information Sources using Source Descriptions》[2]发表在VLDB国际会议上，2006年被评为VLDB十年最佳论文。

这篇论文提出了一个数据集成project——Information Manifold，Information Manifold和其他同类的project极大地促进了数据集成的发展，并导致了一系列数据集成系统商业产品的诞生。

2. 重要意义

Information Manifold的目的是为多数据源提供一个统一的查询接口。用户通过这个接口提交查询可以直接得到对多个数据源的查询结果，就像是对一个数据源进行查询一样。

请看这个查询的例子：找出由Woody Allen导演的在我所在的地区放映的电影的评论。

这是一个复杂的查询，要回答这个查询需要对三个Web站点（相当于数据库中的表）的内容进行连接：一个有演员和导演信息的电影网站；一个电影放映时间和地点的网站，以及一个影评站点。

如果用户不得不自己访问这三个Web站点，然后在三个站点上分别进行有关信息的查询（只能查询该站点的数据库支持的信息），再自己手动把这些信息连接起来，才能得到所需的信息，那么这种复杂度必定是不可忍受的。因此，数据集成研究工作的目标就是设计出一种合适的数据集成系统，它能够自动为用户完成这些操作，并且在可以接受的时间内返回查询的结果数据。至于这些结果信息是否来自多个自治而且异构的数据库，原来的形式是否各不相同，等等问题，都由系统来解决，用户的感受就是对单一数据库的简单查询。Information Manifold就是在这方面比较成功的范例。

3. 主要成果

Information Manifold 对data integration这十年来的发展的主要贡献就是论文里提出的对已知的数据源内容的描述方式（称为**source description，即源的描述**）。一个数据集成系统会给它的用户提供一种模式，用于用户提交他们的查询。其中典型的代表就是**中介模式（或称全局模式，mediated schema）**。用户提交的查询都是基于这个中介模式的，因此data integration系统必须预先建立好中介模式与数据源模式之间的**语义映射(semantic mappings)**。在这里，Information Manifold提出了一种著名的语义映射关系的构建方法，后来被称为**LAV(Local-as-View)**方法。有了模式间的映射关系，用户提交的基于中介模式的查询通过**查询重写(query reformulation)**转化成对于各数据源的可执行的一系列查询。现在多使用LAV视图进行查询重写，被称为**利用视图应答查询(Answering queries using views，简称AQUV)**。然后查询引擎再进行查询优化和执行。形象化描述如图1 所示。

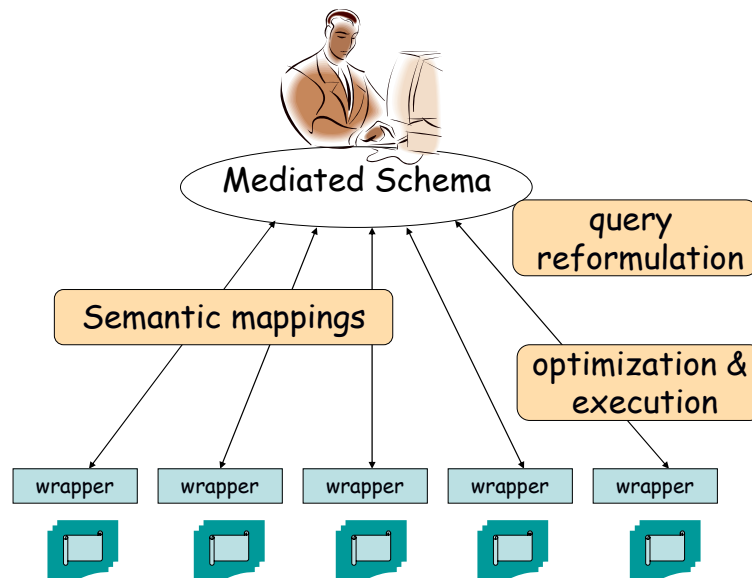


图 1

以下是一些重要内容（上面综述中的**黑体字**部分）的小专题，这些基本上概括了数据集成过去十年内的主要研究成果：

中介模式/全局模式(mediated schema):

中介模式是现在最典型的的数据集成方法,它通过提供一个统一的数据逻辑视图来隐藏底层的数据细节,使用户可以把集成的数据源看作一个统一的整体。

数据集成系统通过中介模式将各数据源的数据集成起来,而数据仍存储在各个局部数据源中,通过各数据源的**包装器(wrapper)**对数据进行转换使之符合中介模式。用户的查询是基于中介模式的,不必知道每个数据源的模式。中介器(**mediator**)将基于中介模式的一个查询转换为基于各局部数据源模式的一系列查询,交给查询引擎做优化并执行。对每个数据源进行的查询都会返回结果数据,中介器再对这些数据做连接和集成,最后将符合用户查询要求的信息返回给用户。

使用中介模式的数据集成方法解决了各数据源中数据的更新问题。因为当底层数据源发生变化时,只需要修改中介模式的虚拟逻辑视图就可以了,大大减少了数据集成系统的维护开销。

这种方法也弥补了数据仓库方法的不足,数据仓库方法必须将各数据源的所有数据都预先取到一个中心数据仓库里,当数据发生改变时,还要到底层数据源中再取一次,还要更新与这些变化了的数据的相关的那些数据,维护开销太大。

语义映射(**semantic mappings**):

这里指的是一种能够描述中介模式和数据源模式之间的语义关系的映射,它把多个数据源的模式通过映射关系集成到中介模式上。

这种映射关系就是我们前面提到的“**source description**”的主要组成部分。

语义映射关系的构建方法: **LAV**和**GAV**

目前,数据集成领域关于模式间映射关系构建的基本方法主要有两种:**GAV(Global-as-View)**方法和**LAV(Local-as-View)**方法。

GAV方法是将各本地数据源的局部视图映射到全局视图,即全局模式被描述为源模式上的一组视图。用户查询直接作用于定义在数据源模式上的全局视图。**GAV**方法的优点是查询效率比较高,缺点是用这种方法构建出来的映射关系的可扩展性较差,不适合数据源存在动态变化的情况。因为一旦有任何一个局部数据源发生改变,全局视图都必须进行修改,维护起来较困难,开销也比较大。**GAV**是较早以前提出的方法。

Information Manifold提出了一种新的、更适合数据源特点的语义映射关系构建方法,即**LAV**方法。**LAV**方法是将全局视图映射到各数据源上的本地局部视图,即各数据源模式被描述为全局模式上的视图。当用户提交某个查询时,中介系统通过整合不同的数据源视图决定如何应答查询。这种方法可看做利用视图回答查询。该方法的优点是映射关系的可扩展性好,适合于信息源变化比较大的情况,缺点是可能会造成“信息遗失”、信息查询效率低。

LAV方法有如下两个显而易见的好处:

第一,描述数据源变得更简单容易了。描述(即视图)只用描述本地数据库就可以了,不必再描述用户查询需要涉及到的其他的数据源和各数据源之间的关系。由于有这种特性,当有新的数据源要加入进来时,数据集成系统可以非常容易地适应,因为每个视图仅描述这个数据库的内容。在实际应用的数据集成系统中,往往要涉及到成百上千个数据源,而且经常需要去除旧的不用数据源,加入新的源,再做集成,所以这个容易更新再集成的特性是极其重要的,所以**LAV**方法是现在最流行的数据集成方法。

第二,对数据源的描述更加精确了。因为源的描述(**source description**)在视图定义语言的表达能力中起着最关键的作用,因为系统能够选取一个最小数量的数据源集合来回答一个特定的查询,所以比较节省时间和系统开销。

目前兴起的**GLAV(global-local-as-view)**映射方法是一种**GAV**和**LAV**方法相结合的产物,

它是由全局模式上的视图与各数据源上的视图相结合形成的。GLAV方法可以结合GAV和LAV的优势，能够为数据集成系统提供更具表达能力的语义映射。

查询重写(query reformulation):

数据集成系统为多数据源提供统一的接口，利用视图描述一个自治的、异构的数据源的集合。用户基于中介模式提交一个查询，数据集成系统通过源模式与中介模式之间的映射关系将该查询重写为数据源可接受的语法形式传给数据源，在随后的阶段基于数据源的查询被优化并执行。

利用视图应答查询(Answering queries using views, 简称AQUV)

也被称为利用视图重写查询(rewriting queries using views)，即给定一个数据库模式上的查询 q ，和同一数据库模式上的视图定义集 $V=\{V_1, V_2, \dots, V_n\}$ ，能否仅使用视图 V_1, V_2, \dots, V_n 获得对查询 Q 的应答[6]。

在使用LAV方法构建映射关系的数据集成系统中，各数据源模式是全局模式上的视图，数据源的内容由在中介模式上的视图来描述。因此可以将数据源看成是物化的视图(materialized views)，将视图定义看成是数据源描述(source description)。从而将在中介模式上构造的用户查询，重写为一系列的直接基于各数据源模式的查询[5]，这就是利用视图应答查询问题。

有时候我们不一定能得到与用户查询等价的重写查询，原因是物化视图越来越多，想全部覆盖这些视图是很困难的。在有些情况下，作为近似，我们可以找到最大包含集，它提供可用数据源上可能的最佳结果集。

因此查询重写分为两种类型：

相等的查询重写：重写的查询与原查询有相同的结果集，可以理解为等价的查询重写；

最大包含的查询重写：重写的查询是原查询的最大子集。

三、数据集成系统的发展建设

1. 模式间的映射关系的生成

模式和模式间的语义映射关系是数据集成系统的构建基础。

现在，建立“source description”已经迅速成为开发实际应用的数据集成系统的最主要的瓶颈。更准确地说，瓶颈是建立源模式与中介模式之间的语义映射关系。要创建这样的映射关系并且维护它们，需要专门的数据库专家来完成，而且，他们还必须同时具备丰富的商业知识，才能够理解需要进行匹配的模式所具有的意义。对于企业来说，聘请这样的专门人才来建立和维护数据库的模式匹配关系，代价肯定是比较大的。

有需求就有发展的动力。这促成了数据集成研究领域里的一个相当重要的分支：半自动化生成模式映射关系。一般地，完全自动地生成映射关系是一个几乎不可能完成的问题，因此研究努力的方向应该是创造出能够加速mapping的生成并且尽可能减少人工干预的工具。

在自动化生成模式匹配的研究领域中，现有的工作都是基于这样的思想：第一，用于建立模式之间的匹配的技术都基于那些模式本身所包含的线索，比如模式元素与数据值或属性值在语言上的相似性与重叠性，第二，据观察，这些方法没有一个是十分简单的，以后的数据集成系统的发展趋势必然是联合这一系列单独的技术，来创建模式之间的映射关系，才能达到比较好的效果。第三，一个重要的观察结果是，模式匹配的创建工作常常具有很大的重复性。例如，在做数据集成时，我们建立同一个域上的多个模式到同一个中介模式的映射

关系。因此，我们可以使用**机器学习算法**，这种方法是：先人工建立一个初步的模式映射关系，作为训练数据，然后对这些mapping做归纳，预言产生出其它那些未知的模式间的映射关系（见图2）。这些技术今天已经在商业领域中使用，并且带来了重要的商业价值和好处。

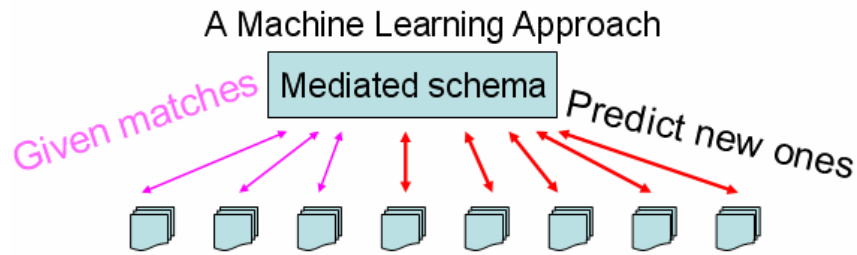


图 2

2. 适应性查询处理

一旦一个被提交给中介模式的查询已经被重写为一系列的面向各个数据源的查询，这些查询就需要被有效率地执行。尽管分布式数据管理中有许多技术在这里都很适用，但又有一些新的挑战出现了，主要是由于数据集成系统中的信息的动态特性决定的。

数据集成系统与传统的数据库系统不同，它的各个数据源具有自治性和异构性，各个数据源数据的可访问性以及传输速度是经常变化和不可预测的，执行引擎没有足够的信息来制定出一个好的查询计划。因此传统的停止-进行方式的查询处理不能很好地处理数据集成系统得查询。而能够在查询执行过程中动态调整查询计划的适应性查询处理是针对此类应用的最佳选择。适应性查询处理逐渐成为一项重要的技术。

3. XML

我们不能忽视XML在过去十年的数据集成发展史上所起的重要作用。

如今的Web数据库实质上就是一个巨大的异构数据库的集合，怎样为大量异构的数据提供某种统一的表示方法无疑是数据集成研究领域中的重要问题。这就要求我们找到一种标准、开放的数据结构来表示数据。而XML的出现无疑为异构数据源的集成带来了新的希望。

XML是互联网联合组织(W3C)设计并推荐的新一代可扩展标记语言，它是SGML的一个优化子集。它以一种开放的自我描述方式定义数据结构，在描述数据内容的同时能突出对结构的描述，从而体现出数据之间的关系。XML是一种半结构化的数据模型，它的很多特性使得它可以描述不规则的数据，能够集成来自不同数据源的数据，可以将多个应用程序所生成的数据纳入同一个XML文件。

实质上，XML没有解决任何语义集成的问题，那些数据源共享XML文件，然而这些文件的标签在这种应用之外就是毫无意义的。可是，用户看起来的效果是好象这些数据源里的数据真的被共享了一样，而且用户的操作也是像在一个真正的、数据共享的数据集成系统中进行的一样。现在XML对数据集成研究的推动力越来越重要了。

如果没有XML，集成系统就必须了解每个数据库描述数据的模式和规则，这几乎是不可能实现的。Web数据源中的数据表示形式的不同几乎是无法穷尽的，XML能够使不同来源的结构化的数据很容易地结合在一起。

从技术的角度来看，目前一些数据集成系统已经使用了XML作为它的基本数据模型，并且用XML查询语言（XQuery）作为数据库查询语言。要维护和支持这样的系统，数据集成系统的每一个方面都需要被扩展，使之具有支持和处理XML的能力。

主要的挑战是：XML的嵌套特性，而且XML是半结构化的语言。

Tsimmis Project首先阐述了半结构化数据在数据集成中的益处和重要作用。

4. P2P数据管理

点对点（peer-to-peer）文件共享系统的兴起，鼓舞了数据管理研究领域对P2P结构实现数据共享的兴趣。除了P2P模式的常规要求以外，研究者们还提供了P2P在数据集成环境下的两种附加的优点。

第一，在实际应用中，几个不同的组织要求共享数据，这种情况经常发生。但是这些组织中却没有一个想要担负起创建一个中介模式、维护它，并且为它建立和那些数据源模式之间的映射关系的责任。这怎么办呢？P2P结构为我们提供了一个非常好的解决办法。P2P结构提供的是一种真正的分布式管理共享数据的模式，每一个数据源仅仅需要提供它自己与它周围一系列邻居数据源的语义映射关系，其他更复杂的集成是系统依循着网络中的语义路（semantic paths）形成的。源的描述（source description）提供了研究P2P结构下的模式及其映射的建立的的基础。

第二，设计一个单独的中介模式为一个数据集成系统服务，这有时候会比较难，而且一个单独的中介模式又比较难以将系统中全部的语义关系都表示清楚。请考虑一个科研合作环境下的数据共享问题，需要被共享的数据可能包括来自不同大学的科研成果，不同书籍上的信息，等等。数据的多样性和异构性，以及合作团体对于共享这些数据的需要，都是非常多样而且经常变化的，这些特性对于一个单独的中介模式来说，都是极其难以管理好的。但是P2P模式就不同了，在这种结构下[3]，没有一个单独的全局的中介模式，数据的共享只发生在网络上这个数据源的邻居数据源之间。

图3是P2P模式的一个形象化的示例。

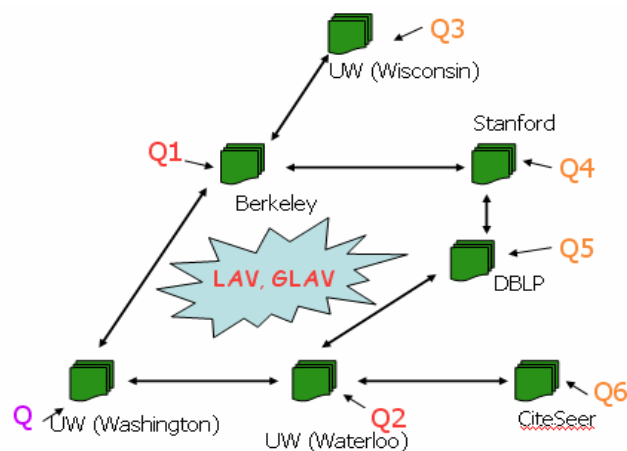


图 3

5. 人工智能的重要作用

数据集成在人工智能（AI）的领域里也是一个非常活跃的研究课题。在早期，数据集成在人工智能领域的应用被称为描述逻辑（Description Logics），它是知识表示的一个分支，能够描述数据源之间的关系。Information Manifold系统的中介模式就是基于典型的描述逻辑，它把描述逻辑的表达能力同数据库查询语言联合起来了。描述逻辑为中介模式的表示，还有语义查询的优化提供了更加灵活的机制。

机器学习在为数据集成系统半自动化地建立语义mapping这个领域扮演了一个非常重要的角色。我们可以预言，未来机器学习将会对数据集成有着越来越重要的影响。

四、企业信息集成

上个世纪九十年代末开始,数据集成从实验室里面“走”了出来,进入到了商业化领域中,成为现代化企业信息管理必不可少的应用技术。今天,这种工业被称为“企业信息集成”(Enterprise Information Integration,简称EII)。

现代企业对于数据集成的需求日益增长,试图找到一种用单一系统对企业的所有信息资产实现集成和管理的解决方案,从而达到有效地集成企业信息,对多个数据库统一管理的目的。

EII工具的出现解决了数据管理领域的一个非常让人头痛的问题——从多个数据源提取数据。它的根本思想是:为来自多个不同数据源的信息提供集成工具,这种工具无需首先把所有的数据从网上下载到本地的数据仓库里。这正是EII工具的优越和先进之处。

EII系统中,数据是“按需应变”地抽取的。查询经过优化、分段又被返回所有的数据源,而结果则被放入到数据源的虚拟视图,“虚拟”是从数据通常都是驻留在数据源的意义上来说的。EII工具是“访问”而不是“移动”数据。这就从根本上简化了分布数据的访问和集成[4]。

和任何新兴的产业一样,EII也面临着许多挑战,下面是具有代表性的一些:

水平 vs.垂直: 从商业角度来看,EII公司必须决定:是要建造一个能在任何应用环境下使用的水平平台,还是为某一个特殊的垂直方向制造特定的工具。这就是EII发展中的水平 vs.垂直(Horizontal vs. Vertical)问题。

垂直方法的观点是:用户更关心他们的全部问题能否都被解决,因此在解决方法中必定有一个“纵深”方向很适合解决某个用户的问题,所以我们要向下深入研究这个方面,不必特别关心这解决方法中的其它方面,以及它与解决方法的其它方面的整合。

水平方法的观点是:系统的一般性使人很难断定哪一个“垂直”的方向是我们在解决方法里要特别关注的。所以建立一个通用性较好的“水平”平台更重要。

对于一个新建立的公司来说,这是一个在现有资源不足的情况下如何区分建设的优先次序的热点问题。

和EAI工具以及其他一些中间件的整合: 数据管理的中间件产品是一个非常复杂的问题,EII工具的出现又把这种复杂度加剧了。一个更为成熟的工具是企业应用集成(Enterprise Application Integration, EAI),它可以通过中间件作为粘合剂来连接企业内外各种业务相关的异构系统、应用以及数据源。

EAI的核心就是使用中间件连接企业应用,使应用更加便利,EII则更关注于集成数据和查询。然而,从某种意义上来说,数据是为了应用服务的,查询得到的数据是要放入其他的数据源的。事实上,要查询数据,最好使用EII工具;但是若要更新数据,那么就必须得求助于EAI工具。因此,EII和EAI工具的分离也许只是一个暂时性的问题。其他的产品包括数据清洗工具(data cleaning tools)和记录分析工具(reporting and analysis tools),这些工具与EII和EAI的结合将会有重大的进步。

尽管面临着这些挑战,还有激烈的竞争和因特网泡沫破裂后极其困难的商业环境,EII产业仍然存活了下来,今天它已经成为现代企业的一项不可缺少的技术。

除企业市场之外,数据集成在因特网搜索研究领域里也扮演着相当重要的角色。到2006年,大型的搜索公司(比如google)在集成来自Web上的多个数据源中的信息方面,取得了一定的进步。在这里,源的描述(source description)起了至关重要的作用:因为给无关数据源发送的巨大的查询量的开销是非常高的。因此数据源必须要被尽可能精确地描述。而且,垂直搜索(vertical search)关注于创造特殊的搜索引擎,集成来自于某一特定领域(如旅行、工作等领域)的多个deep web数据源上的数据。垂直搜索引擎产生于Web的早期(比如Junglee

and Netbot公司)。这些搜索引擎中也包含了复杂的源描述。

五、未来的挑战

几个基本因素决定了数据集成研究将面临着长期的挑战。

第一个因素是社会性的。数据集成的本质是人们合作和共享数据的问题。它包括找到合适的的数据,使数据集成系统的用户相信这些数据的来源、正确性和安全性,并愿意共享它们。(这需要考虑到用户的想法,他们愿意共享这些数据可能是因为共享数据的便利性或是应用结果带来的好处)。还要使数据的拥有者相信,他们所有的关于共享数据的担心,包括私密性、系统的查询性能表现等等,都会被妥善解决。

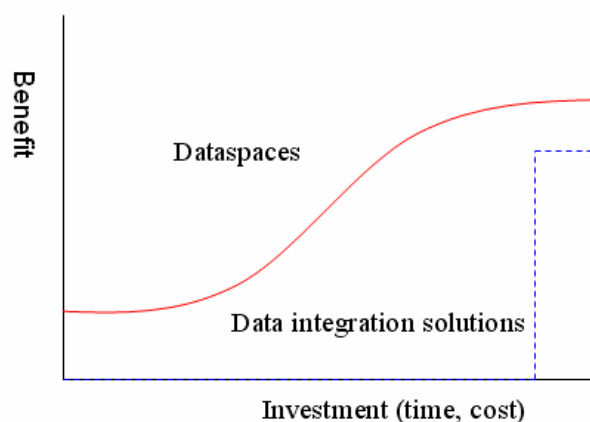
第二个因素是集成的复杂性。在很多应用环境下,人们并不清楚“数据集成”的意义是什么,也不知道如何对已经联合在一起的一堆数据进行操作。数据管理系统的设计者必须应该考虑到这种情况:用户的要求有时可能会导致这种预料不到的数据集成的复杂性。系统必须能够适应这种状况。

由于以上这些原因,数据集成被认为是一个和人工智能一样难的问题,甚至更难!因此,研究者们目标应该是以多种方案、不同角度创造能够使数据集成变得更加便利的工具。以下是几个目前比较流行的数据集成领域的创新与挑战:

数据空间(Dataspaces): Pay-as-you-go的数据管理模式

现在的数据库系统合数据集成系统的一个基本的缺点是:需要很长的建立时间。创建一个数据库系统,必须首先建立一个模式,然后向数据库中增添元组。等这些工作都完成以后,才能够给用户提供服务。创建一个数据集成系统,需要预先建立中介模式到数据源模式之间的语义关系,才能看得到数据源中的内容!但是现在,一种新的数据管理模式——Dataspaces出现了!它强调的是一种pay-as-you-go的数据管理模式:不需要任何的建立时间就能够给用户提供服务!随着时间的推移,用户的需求不断增加,dataspaces系统“增量式”地添加服务的内容,改进服务的质量,这个过程也是数据不断被集成的过程。因此dataspaces并不像data integration系统那样,先把数据集成好了,再给用户提供服务,而是“随需要随集成”的方法,即上面提到的“pay-as-you-go”方式。

Pay-as-you-go Data Management



例如,在dataspaces的最早期,只能提供一些最基本的,如数据源上的关键字查询之类的功能。Dataspaces使用一系列启发式的抽取规则,从本来完全互异的、毫无联系的数据项中析取出它们之间的关系,使用path query方法建立这些关联。最终,当两个数据源之间确

实需要更紧密的集成时，`dataspace`就可以自动创建它们两个之间的mapping。接下来的事情就是让人去修改并维护它了。

不确定性 & 数据血统(Uncertainty and lineage):

在数据集成研究领域，不确定数据的操作和数据血统的问题有很长的历史了。如果说管理不确定性数据和数据血统在传统数据库系统中似乎只是一个好的特点，那么在数据集成系统中它就成为一个必须具备的功能了。一般情况下，来自于多个数据源的数据都是不确定性的数据，它们彼此的形式都不一致。系统必须能够找出这些看似乱七八糟的数据中内在的联系和确定性。当系统不能自动找出这种确定性的时候，可以交由用户来考虑一下数据的血统（也叫数据沿袭），搜索引擎沿着用户的搜索过程把这些URL都提供给用户，因此用户能够通过分析URL理清数据的脉络，决定哪个搜索结果更值得深入探寻下去。通过对数据血统的分析，用户可以知道数据何时更新、如何计算以及从何处而来，这些帮助用户追溯数据产生的来源。这种深入调查数据来龙去脉的能力能够帮助用户断定哪个数据源是可信的。

重新使用人们的关注点(Reusing human attention):

若要在数据源上做更加紧密的语义集成，一个重要的原则就是：要重新利用用户的关注信息。一个简单而明显的例子就是，每一次用户使用`dataspace`系统进行查询，`dataspace`都能从中得到一条用户关注信息的语义线索。这样的线索可以从用户查询数据源时得到。当用户建立语义mapping，或者剪切数据，再把它粘贴到另一个地方，这些操作都能给系统提供很多用户关注点的信息。如果能够建立一个支持这些语义线索的系统，那么语义集成将会变得非常快。目前已经有一些重用用户关注信息的很成功的例子。

六、结束语

不久以前，数据集成还是只是实验室里的一个很好的想法和一块研究者好奇心的领域，但是今天，数据集成是一个必需品。现代经济是基于计算机网络的广泛的下部基础构造。

Thomas Friedman在他的座右铭里这样写到：世界是平的。在一个“平的”世界里，任何产品或服务，无论它们在世界的任何一个角落，都可以被联结起来，成为某一个特定应用或产品的组成部分。为了达到这个理想，数据需要在不同的服务提供商之间被合适地共享，用户要能够在合适的时间里找到自己想要的信息，无论这些数据存储在网络的什么地方。信息集成要成为这种下层基础构造的一部分，要发展成熟到可以融入大背景中，就像其他的到处可见的技术一样。在过去的十年里，实际的信息集成领域里的研究人员们已经取得了相当大的进步，现在我们正面临着更大的挑战！

参考文献:

- [1] Alon Halevy, Anand Rajaraman, Joann Ordille: Data Integration: The Teenage Years. VLDB06
- [2] Alon Y. Levy, Anand Rajaraman, Joann J. Ordille: Querying Heterogeneous Information Sources Using Source Descriptions. VLDB96
- [3] P. Adjiman, Philippe Chatalic, Francois Goasdoué, Marie-Christine Rouse, and Laurent Simon: Distributed reasoning in a peer-to-peer setting. In *ECAI*, pages 945–946, 2004.
- [4] AMT公共知识库: <http://www.amteam.org/>
- [5] S. Chaudhuri, R. Krishnamurthy, S. Potamianos, and K. Shim: Optimizing queries with

materialized views. In Proceedings of ICDE-95, 1995.

[6] A. Y. Levy, A. O. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views. In Proceedings of ACM PODS, 1995.