中国人民大学
RENMIN UNIVERSITY OF CHINA

WAMDM
Innovative Data Systems Research

2006
ANNUAL REPORT

网络与移动数据管理实验室

Lab of Web&Mobile Data Management

2007 年 1 月

中国人民大学
RENMIN UNIVERSITY OF CHINA

WAMDM
Innovative Data Systems Research

2006 ANNUAL REPORT

网络与移动数据管理实验室

Lab of Web&Mobile Data Management

2007 年 1 月

序

当 2006 年即将过去的时候，我和我的学生们讲，我们是否应该总结些什么，总结过去，展望未来，或许对我们自己，对他人，对社会都是一种责任，一种鼓舞，一种鞭策。经过近一个月的努力，我们终于有了手头的这部集子，算是对过去一年的一个交代，也是对未来一年的一个期盼。

过去五六年间，我们的研究工作始终围绕数据库技术与网络计算与移动计算环境的结合。因此实验室的名字为“网络与移动数据管理”（Web and Mobile Data management, WAMDM）。实验室的研究风格秉承萨师煊、王珊教授所一贯倡导的学术研究与系统开发并重的传统，以创新数据管理系统的研究为目标。EMC 公司信息安全部首席科学家 Burt Kaliski 博士来实验室访问后，认为我们的研究方法是一种保持研究不脱离实际的有效方法，并将在 EMC 实验室借鉴同样的做法（“I was particularly impressed by your approach of having two tracks – research and system – in order to ensure that the research is actually applied. We will be doing that as well at EMC Labs.）

过去的一年我们坚持了每周的学术讨论，力求捕捉新的技术发展脉络，为我们的创新研究寻找新的方向。本年度报告的第一部分即汇集这方面的一些报告。归纳起来我们认为数据库技术将面临变革，其中心或许是我们第一篇文章所说的“未来数据管理技术将由数据库管理到数据空间管理，由服务于企业计算到服务于人的计算。过去三四十年我们为企业造了一个成功的软件，未来十年我们将为社会大众的需要创造一个全新的软件”。与之相关的众多技术将等待我们去研究，如数据空间的管理，网络数据的集成，场合感应的数据管理，可信数据的管理，基于 Flash 的 DBMS，等等。这里只是抛砖引玉，希望能够带动相关的研究。

本年度报告的第二部分汇集了我们一年当中所发表的一些论文，分为 Deep Web 数据集成，Web 数据抽取，XML 数据管理，Ontology 数据管理，公路网的移动对象管理等。在以上学术研究的同时，我们着力开发了诸多实验系统，得到企业界关注。如基于 Deep Web 数据集成技术的 JobTong（工作通），Native XML 数据库系统 OrientX，基于手机数据管理的 PhoneDB 等，得到国内外同行的好评。我们先后与联想、中创软件、华为等企业建立良好的合作关系，进行技术转移。

实验室一贯重视国际学术的交流，我们先后与美国 IBM TJ Watson 研究中心，法国 INRIA，希腊 NTU Athens 开展合作交流，并得到中法先进技术项目和中希国际科技合作项目的支持。

2006 年 11 月 19 日至 22 日，我们应邀参加了在德国举办的“Dagstuhl Seminar on XQuery Implementation Paradigms”。所有参加者都是受邀参加，且在本领域有实实在在的系统研究。会议主席德国慕尼黑理工大学的 Torsten Grust 教授在发来的邀请中特别指出”Your native XML database system OrientX is clearly recognized as a highly significant contribution in this research area and the seminar organizers are looking forward to your attendance(你们的 Native XML 数据库系统 OrientX 在本研究领域被认为具有突出的贡献，会议组织方希望你的参加)”。参加完此次会议使我对创新研究有了新的感悟。其实科学研究也有上游和下游之分，站在科研下游谈创新是比较困难的。要创新必须首先进入占据科研上游的学术团体，当真正成为科研上游的一员时，其实不创新也是比较困难的。

过去的一年是收获的一年，我们将继续努力，希望在新的一年里有更大的收获。

在此谨以此集感谢来自学校方方面面的支持，感谢国家自然基金委和 863 计划的资助，感谢所有关心和支持过我们的人们。

孟小峰

2007 年 1 月 22 日于北京

目录

数据管理前沿技术报告

发表论文精选

数据管理前沿技术报告

从数据库到数据空间，从服务于企业到服务于大众

——From Database to DataSpace, From Enterprise to People

孟小峰
网络与移动数据管理实验室
中国人民大学信息学院

引言

当代数据的三个典型特点使得传统关系数据库捉襟见肘、疲于应付。第一是海量，全球的数据量在以指数的趋势迅猛增长，据保守估计，目前每年全球至少将产生 15 亿 TB 的新数据产生。第二是共享，互联网和通讯设备的普及使人们享受在他人的数据带来的好处，数据库之间因此也建立起越来越密切的联系。第三是多样化，现在数据已不再是关系模型下纯粹的结构化的文本数据，图片、音频、视频乃至非结构化的文档都大量的涌入到人们的应用中来。数据库的研究者和制造商们并非无视这些事实，他们在功能和性能上仍在不断地丰富完善，不断地修补着这架越来越难以驾驭的马车。毕竟目前数据库在数据管理中仍旧占有主导地位，但这却不能表明它处在越来越尴尬的境地和最终会被代替的命运。其实上面提到的三个特点只是数据发展中表面现象，我们是根据这些特点继续维护，延续着“头疼治头、脚疼医脚”的这种治标不治本的补救措施？还是另辟蹊径，寻求一种新的数据管理技术在根本上进行大胆地变革？答案是不言而喻的，因为我们面临则者前所未有的新的变革和新的需求。

首先未来数据管理的主体将从单纯企业需求转向更为丰富的个人数据管理需求。

纽约时报著名的专栏作家托马斯·弗里曼在他的畅销书《世界是平的》一书中有这么一段话：

“我想全球划分为三个主要纪元。全球化 1.0 自 1492 年，持续到大约 1800 年。全球化 2.0 大概从 1800 年持续至 2000 年，中间曾经被大萧条及两次大战打断。2000 年世界进入了一个新纪元：全球化 3.0。世界从小缩成微小，竞赛场也铲平了。”

对此他又进一步解释道：

“在“1.0”，推动全球化的力量来自国家，在“2.0”，推动力来自企业，在“3.0”，推动力则来自个人。个人的力量大增，不但能直接进行全球合作，也能参与全球竞逐，利器即是软件，是各式各样的电脑程序，加上全球光纤网络的问世，使天涯若比邻。……”

从他的观点里我们可以得出，进入 21 世纪以后，随着个人电脑和互联网的普及，个人的影响力的提升使得在过去以企业为主导的模式逐渐地向以个人为主导的模式演变。

在过去的三十多年里，数据库技术主要服务于企业计算，我们几乎为企业的数据库管理开发了近乎完美的 DBMS。数据库作为当前最成熟的系统软件之一，已经成为了现代计算机信息系统和计算机应用系统的基础和核心。数据库也从最初的层次、网状数据库演变到了今天的关系数据库，为大家熟悉的 Oracle、DB2 和 SQL Server 等商业关系数据库已经广泛应用于各行各业。在很多人眼里看来，似乎一切都是如此的完美，所有的数据管理问题都会在这里得到答案。然而事实并非如此。

进入二十一世纪，我们忽然发现管理者世界上最大、最丰富的数据集合，而且主要为个人服务的 Google，MSN，Yahoo 均不使用传统 DBMS，而是另辟蹊径去寻找能更好满足个人数据管理需要的方法。

不可否认数据库技术在过去三十年里为推动企业数据管理的发展所做出的无法替代的贡献，并将继续发挥其应有的作用。但在世界进入全球化 3.0 后，推动力正在由企业转变到个人，因此可以断定新的数据管理技术将由服务于企业的管理而过渡到个人的管理需求上面，那么数据管理技术将在服务于人的管理中起到什么样的核心作用？

其次新的计算机科学问题将是如何解决计算性能和计算成本不断改善，但人可用的时间和精力却恒定不变这一矛盾现象。

大家都知道计算机领域中的摩尔定律，它的一个广义的解读是这样的：计算机的性能随着时间呈指数级的增长；同时，计算的成本则会随时间呈指数级的下降。这一定律随着计算机领域的飞速发展得到了越来越有确凿的证明：CPU 的速度、内外存的容量在迅速增长，相对的是它们的价格却一路下跌。

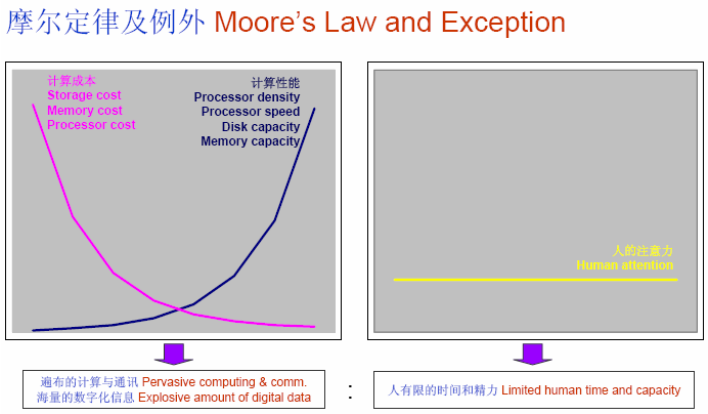


图 1.1 摩尔定律及例外

这样一个发展趋势的必然结果，就是计算和通讯会越来越普及，以至于数据量以难以想象的速度急剧膨胀，有人把这种现象称作是全球性的数据爆炸。也就是说目前计算技术的发展是使人更加应接不暇加速到来的信息，而没有丝毫减轻人们的负担。

据保守估计，目前每年全球至少将产生15亿TB的新数据。另一方面，在数据管理中却有一样东西是基本维持不变的--那就是人的注意力和人能够用在计算上的时间：每个人的总的寿命以及每一天用在工作中的时间在近千年中几乎没有太大的改变。于是作为数据管理技术的研究者和数据管理系统的使用者，我们发现正处于一对看起来很难调解的矛盾之中：一方面是遍布和汹涌而来的海量数据，另一方面则是人有限的时间和精力。这使得数据管理技术特别是传统的(关系)数据库管理技术面临越来越多的挑战，我们自己造的这块“巨石”已经压得我们喘不过气来。什么样的数据管理技术可以化解这对矛盾，如何使我们在这场人与数据的大战中占得先机？

第三，数据是未来计算的核心。

我们计算机领域的人一直把速度作为计算的核心，所以孜孜不倦地追求提高计算机的速度和效率。正如 30 年前我们一直在抱怨天气预报、机器翻译的质量不好是因为计算机的性能不够好。然而事实是，到现在计算机的速度已经提升了上百万倍甚至上亿倍，但机器翻译并没有像预料的那样取得具体的突破性进展，天气预报一样该不准确还是不准确。这里有一个有趣的例子。在 2005 年的一次 NIST(美国国家标准与技术局)举办的机器自动翻译大赛中，最终结果让人大跌眼镜，冠军竟然是仅开始研究三年而且是首次参加的这次比赛的 Google。这个结果让该领域的专家们“伤心不已”。事实上，令 Google 获胜的“统计式”翻译算法，其基础是统计与分析某一单词在这一语言环境中被运用的概率与位置，来寻找词汇的排列规则；而另一种“很有前途”的热门算法，“类比式”算法，则是分析数以亿计的现成的翻译作品，

当需要翻译新的语句时，在现有的数据中搜索与之最相似的语句，来进行翻译——搜索和海量的数据分析，无论是哪一种，都是 Google 的专长。一句话，Google 制胜的法宝是其所多年积累的海量数据。这说明了计算的核心已不再是速度，而是数据，未来的世界承载在数据之上!但如果我们没有合适的数据管理技术来使用这些海量而噪杂的数据，对我们来说反而会是一场灾难。那究竟什么样的数据管理技术可以帮助我们驾驭这些数据呢？

归纳上述的三点我们不难看出，未来先进计算的核心是数据，而数据管理的主体不再是企业计算，而是围绕人的计算。着力解决人的时间和精力的问题将是我们面临的新的科学问题。

要解决这个问题，首先让我们看一下对应于个人管理现实中的数据，计算机中的数据管理主要面临的挑战：

- 数据的纷繁复杂。不管是结构化的还是非结构化，不管是文本，声音，图片，视频等等一切数据都是要管理的对象。
- 数据之间的逻辑关系。现实中之所以井然有序，在于人们对任何对象之间建立了逻辑关系，那么计算机中的数据如何进行关联是需要关心的重要问题。
- 数据的演化。现实中的对象会自我变化，会和其他的对象联合产生变化，那么对应计算机中的数据就是它们的自我演化问题。
- 数据的场合感应。要减轻人的时间和精力，就要能够主动地依其所处的场合（context）给出恰当的信息感应（awareness），减轻其处理数据的负担。
- 数据的可信性。解决人的管理问题，必须保证数据的真实，可信，和隐私保护等问题。

传统而严格的 DBMS 在这些现实面前将无能为力,这也促使我们去寻求一种新的数据管理技术，甚至可以进一步看作一种新的数据管理理念，这就是——数据空间（DataSpace）。基于以上分析，我们这里大胆地预测：未来数据管理技术将由数据库管理到数据空间管理，由服务于企业计算到服务于大众的计算。过去三四十年我们为企业打造了一个成功的软件，未来十年我们将为大众的需要创造造一个全新的软件。

数据空间---数据管理新概念

近年来，随着因特网的迅猛发展，web信息量急剧膨胀，日益成为一个巨大的数据库，对于这个实实在在的信息库，人们不知道其信息量的多少，不知道信息的存放位置，不知道信息的格式。这些海量的信息分布在世界各地的无数台计算机设备上，格式多样、内容丰富，有的与个人有关，有的与企业有关。这种数据信息存在方式的新特性，使人们对于数据资源的存储、访问等出现了新的特点：

- web日益成为一个信息闭包。web信息库中信息量的急剧增长，几乎包括了人们所需要任何信息，这一发展趋势对于传统的数据管理方法和技术形成了冲击，甚至改变了人们解决问题的思维方式，人们对于数据的价值和利用这种价值的方法有了新的认识。
- 传统的数据库技术不能满足新的数据管理的需要。Google、百度等网站对数据处理方式的改变也说明了这一特点，于是人们提出了一个新的概念：数据空间（DataSpace）。数据空间是存储个人、群组和企业信息的理想方式。

到底什么是数据空间，我们如何能够像传统的数据库一样，清晰的勾画出数据空间的内涵？

传统数据库的各种数据存储方式，关系也好，XML 也好，无不强调一个格式，总是先有一个格式，然后使数据服从于这个格式，如此才能存储数据，进而提供查询等服务。但是任何形式的数据，其核心都是数据本身，形式只是一种载体，如果将数据限制于某种形式之中，多少显得有些许被动，所以就是一种“被动”的方式，也就是说如果你有一份不同格式的数据要想存储于数据库中，必须将其转化为数据库中数据的存储格式。因此，对于这种格式性很强的存储，可以称之为**“先有格式，后有数据”**。

数据空间不同，从它的名字可以看出，它与数据库不同并且强调的是一个 Space，Space 是什么？是空间，广阔的宇宙是一个 Space，是个 ObjectSpace，不管这些 Object 在其中如何排列，如何组织，只要是属于这个 Space 的就是符合要求的。同样，数据空间是一个满是数据的空间，数据在其中如何组织都可以，表也罢，XML 也罢，文本也罢，只要你是数据的一种载体，你就可以存在于这个 Space 中，对数据的组织排放不做任何要求，正如[9]中所说：“一个数据空间应该包含与某个组织或个体相关的一切信息，无论这些信息是以何种形式存储、存放于何处”。这样一来，无论你有一份怎样格式的数据，XML 文档也好，文本文档也好，都可以存储于数据空间中，并且通过数据空间来对其进行掌控，这可以称之为**“淡化形式，凸现数据”**。

这里简单概括一下数据空间的特性：

数据空间与实体相对应：数据空间是有所属的，与实体一一对应，一个人可以有一个数据空间，一个组可以有一个数据空间，一个企业可以有它的数据空间。数据项是数据空间的基本元素。数据空间是数据项的集合。数据项是与数据空间所对应实体相关的信息单元。一个数据项可以是一封电子邮件，保存下来的一个网页、一个文件等，也可以是一个传统数据库表。数据空间中的数据项一定是对于实体有意义的。有用性也是定义数据空间的边界，这种有用性可以是现实的，也可以是潜在的。

数据空间具有空间和时间特性：从空间上，数据空间的数据分布存放在许多位置；从时间上，数据空间中的数据也随着实体的发展而不断变化，一些新的数据项会加入进来，同时一些不再具有应用价值的数据项会消失。数据空间的大小是动态变化的，随着实体的进化，数据空间会不断进化，通过数据挖掘、自适应等技术，数据质量会不断提高，包含的信息量会不断增强。

实体数据空间交叉重叠：由于数据空间是与实体是对应的，不同实体所对应的数据空间是有重叠的，一个数据项可能即属于实体一的数据空间，又属于实体二的数据空间。

个人数据空间将是数据空间的主要存在和应用形式：在过去的30年中，数据库应用的主要对象是企业，可以预见，在未来的数据管理领域，数据管理将会转移到为人服务，为提高人的生活质量和效率服务，个人数据管理将是未来数据管理技术研究和应用的主要对象。相应地，个人数据空间也将是数据空间研究的主要对象。

综上，我们可以观察到数据空间本质上区别于传统的数据库（见表1 ）。

表1：数据空间与数据库的区别

数据空间	**传统数据库**
淡化形式，凸现数据	先有格式，后有数据
开放的，支持多种不同的数据源	支持有限的数据格式，封闭的
强调数据的可关联性和可演化性	关注数据的稳定性
具有*Pay-As-You-Go* 特性	需要的时候集中建成
面向实体需要	面向应用主题

因此，数据空间将是一个开放的系统，其中包含与实体有关的各种数据，对其进行管理的目的是提高实体的运行效率。数据空间涉及很多技术，如数据的获取、组织、存储、索引；任务的管理；场合感应；隐私保护等等。

新的机遇与挑战

数据信息的新特点，使人们开始重新审视和定位数据空间技术。目前的情况与30年前的情况类似，那时企业的集中数据管理需求催生了数据库技术，现在对于基于web的个体数据管理需求期待着数据空间技术研究的重大突破。数据空间技术为研究者提供了重大机遇。

数据空间研究日趋活跃：目前，人们对数据空间相关技术的研究日趋活跃。Jens-Peter Dittrich Marcos, Antonio Vaz Salles等人将数据空间理论应用于个人信息管理，作了大量工作，2005年，在VLDB2005 发表了一篇Demo Paper：iMemex—将个人从信息从枷锁中解放出来，实现了个人信息管理原型系统。其后进行了更深入的研究，分别在2006年VLDB和PIM Workshop发表了论文，系统阐述了个人数据空间管理的概念，提出了一个新的数据模型：*iMEMEX*，基于数据源视图的概念，建立了个人数据空间框架模型，并基于此实现了个人数据空间原型系统。

数据空间面临众多研究课题：尽管对于数据空间技术进行了一些研究，但是还不深入，在数据空间，还有很多挑战性的研究课题。

- 数据空间的理论基础

个人数据空间的研究涉及很多基础理论问题，如对数据空间中不同数据模型、数据关系和查询结果的理解，在传统数据库中，人们关注的是查询语言的表达能力，在具有上下文相关性的数据空间中，针对的是内容来自不同参与者的查询，这就面临许多问题，例如，我们如何检测具有不同语法结构但语义相同的查询。

- 数据空间中的数据关系和数据模型

传统的数据库管理系统大都是基于关系模型的，有完备的关系代数、关系演算理论。在数据空间中，传统的关系模型是否适用，网状模型和层次模型是否是更好的选择，相应于新的数据关系和数据模型特点，原来的理论是否适用；数据空间中也面临数据的一致性、安全性、并行性等问题，原来的并发处理策略、事务处理等理论是否适用；访问效率评价问题，和传统数据库不同，在数据空间中，影响访问效率的关键因素也不再是磁盘的I/0，那么这种情况下如何衡量访问效率，如何进行查询优化。

- 数据的存储和索引

面对如此巨大的个人数据量，在一台机器上进行数据的存储是不可能的，同时从安全性、可访问性等方面进行考虑，也不是很好的。而且Web数据是变动的，有的数据会随时消失，这样就要求我们对数据的安全性策略进行研究。

众多的研究课题为研究者提供了机遇。但是，与之结伴而来的是巨大的挑战。数据空间的挑战性来源于自身的特点。

- 由于数据空间的分布性，使得数据空间的数据组织、存储、索引、访问等都有新的特点。
- 数据空间中主体因素的作用。由于数据空间面向的实体千差万别，从而，依赖于实体的数据空间也有很多个性，因此对数据空间技术的研究，既要包括对通用数据空间模型和应用技术的研究，也要研究主体因素的作用。
- 数据的多样性给数据空间的数据输入带来挑战。数据类型愈来愈丰富，包括各种格式的电子文档、音频、视频、图像、移动数据等各种信息，数据空间必须建立一个

开放的能够适应各种数据格式的数据接口。

- 数据空间中的数据关系和任务管理。数据空间管理的目的是提高实体的效率，因此其不仅能够被动的显示主体所需要的信息，而且能够通过对数据关系的分析，自动的提醒主体应该做的的事情，这也是场合感应的基础，也是很有挑战性的研究课题。

此外，数据空间的实现模式是怎样的？如何评估数据空间管理系统的性能？等等，都是挑战性的问题。也正因为这些挑战的存在，愈发显现出它的魅力。相信随着研究的深入，数据空间及其管理技术，将成为新一代数据管理技术的核心，成为人们享受信息技术发展带来的巨大效益的基础平台。

综上所述，DataSpace 是一个“泛化”了的数据库，所‘泛’之处就在于数据形式的泛。这种泛化的 DB 从根本上来说还应该是 DB，只是相当于在传统 DB 之上又架设了一层，使得使用更加方便，但是同时也带来了挑战，正所谓便利多多，挑战多多。但是为了它所能带来的诱人便利，这些挑战还是值得一试的。

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，用户的感觉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。

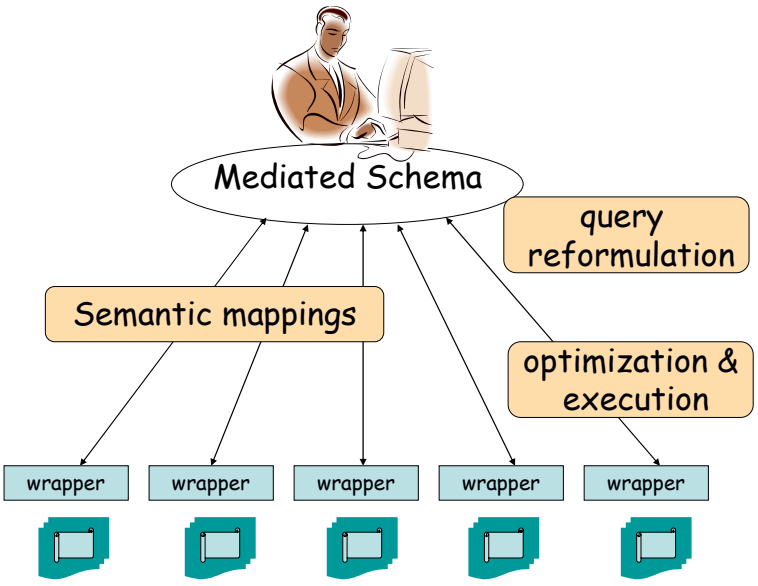


图 1

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的数据源，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={V1,V2,…,Vn},能否仅使用视图V1,V2,…,Vn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，才能达到比较良好的效果。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。

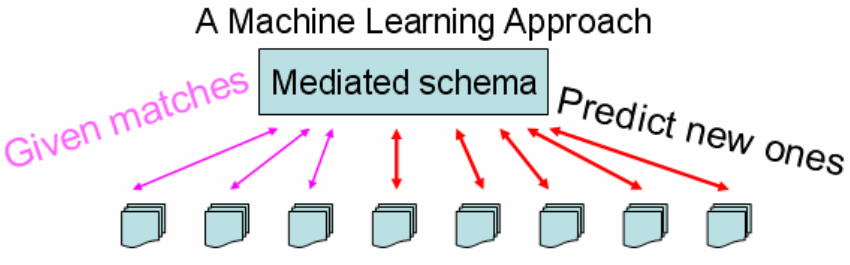


图 2

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。

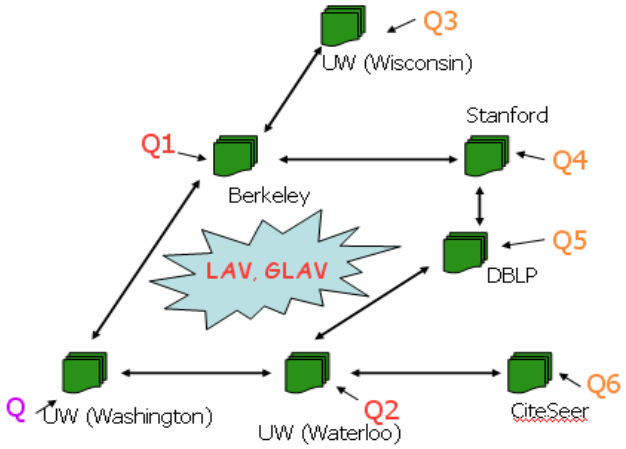


图 3

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“随需应变”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。

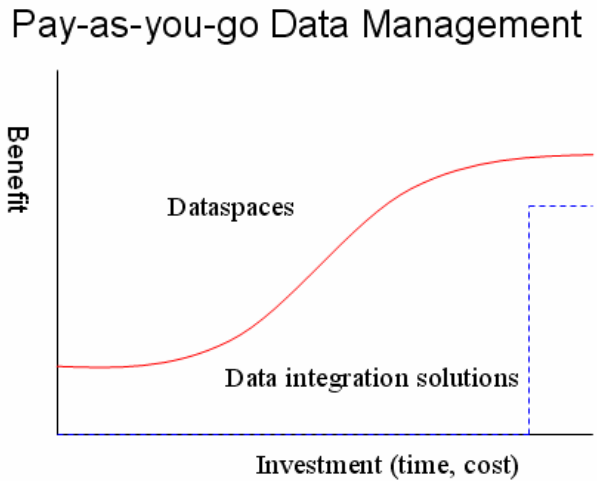
第一个因素是社会性的。数据集成的本质是人们合作和共享数据的问题。它包括找到合适的数据，使数据集成系统的用户相信这些数据的来源、正确性和安全性，并愿意共享它们。（这需要考虑到用户的想法，他们愿意共享这些数据可能是因为共享数据的便利性或是应用结果带来的好处）。还要使数据的拥有者相信，他们所有的关于共享数据的担心，包括私密性、系统的查询性能表现等等，都会被妥善解决。

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，研究者们的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，才能够给用户提供查询服务。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。



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。Dataspace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。

不确定性与数据血统(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和中它就是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。这种深入洞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赖的。

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。目前已经有了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数据，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, Fran¸cois Goasdou´e, Marie-Christine Rousse, 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. 0. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views. In Proceedings of ACM PODS, 1995.

Deep Web 数据集成问题研究

刘伟 （Web 组） 孟小峰 孟卫一

摘 要：随着 World Wide Web（WWW）的飞速发展，Deep Web 中蕴含了海量的可供访问的信息，并且还在迅速的增长。这些信息要通过查询接口在线访问其后端的 Web 数据库。尽管丰富的信息蕴藏在 Deep Web 中，由于 Deep Web 数据的异构性和动态性，有效地把这些信息加以利用是一件十分挑战性的工作。Deep Web 数据集成至今仍然是一个新兴的研究领域，其中包含有若干需要解决的问题。总体来看，在该领域已经开展了大量的研究工作，但各个方面发展并不均衡。本文提出了一个 Deep Web 数据集成的系统架构，依据这个系统架构对 Deep Web 数据集成领域中若干关键研究问题的现状进行了回顾总结，并对未来的研究发展方向作了较为深入的探讨分析。

1、引言

随着 World Wide Web 的飞速发展，其中蕴含了海量的信息可供我们利用。根据文献[1]最新的调查，目前整个 Web 超过了 200,000TB 的信息量，而且仍在快速的增长。在 Web 领域的研究目的在于发展新的技术可以有效地从 Web 中获取有用的信息。Web 中的信息主要通过网页的形式对外发布，而由文本和超链接构成的网页有其独特之处：数量惊人，信息丰富；由不同的个人或群体开发，形式与内容有很大的差异；分布在地球上 Internet 连接的每一个角落，这就造成了 Web 数据的异质性和缺乏结构性。正是由于这个原因，使得自动地从中获取有价值的信息和数据变成一件十分具有挑战性的任务。到目前为止，为了有效地利用 Web 上的信息，所采用的方法涉及了广泛的领域：数据挖掘、机器学习、自然语言处理、统计分析、数据库和信息检索等。

整个 Web 看似杂乱无章，但如果按其所蕴涵信息的“深度”可以划分为 Surface Web 和 Deep Web 两大部分。Surface Web 是指通过超链接可以被传统搜索引擎索引到的页面的集合。在现实中，有大约 21.3%的页面由于缺乏被指向的超链接而没有被搜索引擎索引到，我们把这一部分页面也看作是 Surface Web 的范畴。而对于 Deep Web，目前还没有一个统一的定义，文献[2]中认为 Deep Web 是指 Web 中不能被传统的搜索引擎索引到的那部分内容，特别是指那些通过查询实时产生的动态页面，但随着搜索引擎爬虫（Crawler）能力的增强，使得 Deep Web 这一概念变得复杂不易界定，很难给出一个可以长期一致认同的定义。

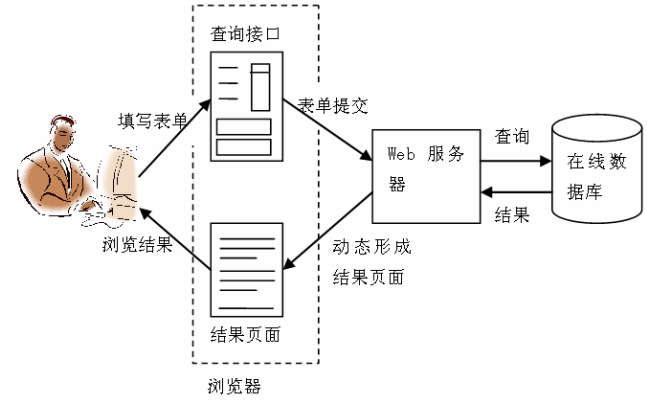


图 1 从 Web 数据库中获取数据的过程

在这里我们试图对 Deep Web 的范畴给出一个较为完整的描述：Deep Web 是指 Web 中可访问的在线数据库，这里简称为 Web 数据库或 WDB，其内容存储在真正的数据库中。这些内容只有在被查询时才会由 Web 服务器动态生成页面把结果返回给访问者（图 1），因此

没有超链接指向这些页面，这是和那些可以被直接访问的静态页面的根本区别。按照存储信息的结构化程度可以进一步划分为结构化信息、文档信息和非文本文件， 网上购物网站存储的信息属于结构化信息，新闻网站存储的信息属于文档信息，二者因结构化程度的不同对其查询所应用的技术也差别很大，而非文本文件，主要包括多媒体文件、图像文件、软件和特定格式的文档（比如 PDF 文件）。在一般的意义下，我们对 Deep Web 信息的获取更关注的是对结构化信息的获取，而不是文档或非文本文件。其原因不难理解，对结构化数据的集成更有意义，可以采用的技术也更丰富。Deep Web 数据集成也主要是指对结构化信息的集成。随着 Web 相关技术的日益成熟和 Deep Web 所蕴含信息量的快速增长，通过对 Web 数据库的访问逐渐成为获取信息的主要手段，而对 Deep Web 的研究也越来越受到人们的关注。

与Surface Web相比，Deep Web蕴藏了更加丰富，更加“专业”（专注于某一领域）的信息。在2000年7月，Brightplanet对Deep Web做了一次较为全面的宏观统计，发布了Deep Web的白皮书[1]（在该文中Brightplanet对Deep Web的定义主要指的是Web数据库），指出整个Web上大约有43,000-96,000个Web数据库，并从宏观上对Deep Web做了定量的调查统计，下面列出其中部分的调查结果：

- Deep Web 蕴含的信息量是 Surface Web 的 400-500 倍。
- 对 Deep Web 数据的访问量比 Surface Web 要高出 15%。
- Deep Web 蕴含的信息量比 Surface Web 的质量更高。
- Deep Web 的增长速度要远大于 Surface Web。
- 超过 50%的 Deep Web 的内容是特定于某个域的，即面向某个领域。
- 整个 Deep Web 覆盖了现实世界中的各个领域，比如商业、教育、政府等等。
- Deep Web 上 95%的信息是可以公开访问的，即免费获取。

整个Web是开放的、不断变化的，作为Web的组成部分，应该如何有效地评估当前整个Deep Web的规模，即当前Deep Web上Web数据库的数量以及变化情况。UIUC大学在 2004 年 4 月对整个Deep Web做了一次较为准确的估算[2]，推测整个Web上有 307000 个提供Web数据库的网站、450000 个Web数据库，比Brightplanet在 2000 年估计的 500000 个数据库网站的数目增长了 6 倍多。

Deep Web中的Web数据库不但数量众多，而且覆盖了现实世界的各个领域。一些专门的机构，象CompletePlanet和InvisibleWeb等，构建了Deep Web目录对按现实世界的领域对Deep Web的内容做了分类，主要包括(1)商业与经济(2)计算机与互联网(3)新闻媒体(4)娱乐等一共十几个分类。这只是宏观的分类，每个分类下面还有小的分类，比如科学可以继续分为社会科学与自然科学，而自然科学又可分为若干学科。在表1[2]中可以看出，尽管这些网站对Web数据库进行了细致的分类，但所列出的Web数据库仅仅只是整个Web数据库的很小的一个比例（即使最大的CompletePlanet也只有15.6%）。因此从宏观上对Web数据库按现实世界的领域分类做一个定量的分析是十分迫切而且必要的工作。

表 1　Deep Web 目录的覆盖率

	Web 数据库的数目	覆盖率
completeplanet.com	70000	15.6%
lii.org	14000	3.1%
turbo10.com	2300	0.5%
invisible-web.net	1000	0.2%

对 Deep Web 中信息的获取主要的途径是通过对网站中所提供的查询接口提交查询来获得，图 2 是 Amazon 网站提供的查询接口。每个查询接口支持在若干个属性上进行查询，比

[1] http://www.brightplanet.com/technology/DeepWeb.asp

如要查询某一本图书，可以根据书名、作者、价格等。这些属性就构成了查询接口的模式（Schema）信息。查询接口模式的大小是指属性的数目。查询接口顾名思义是外部访问 Web 数据库的门户，是从 Web 数据库中获取数据的主要途径，因此在 Web 数据库研究领域，对查询接口的模式信息的研究占有极其重要的地位。

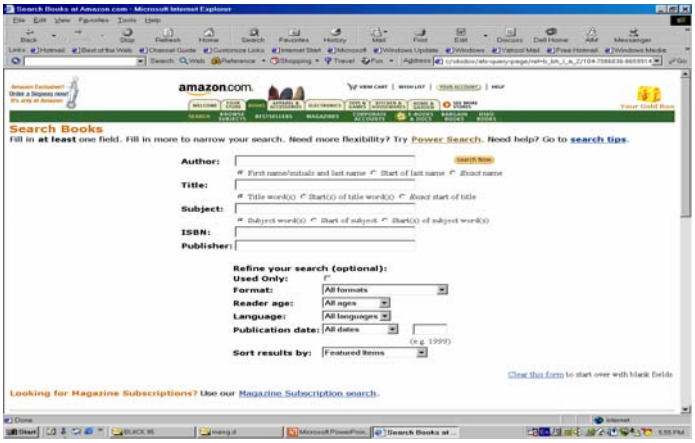


图 2 查询接口示例

对 Deep Web 信息的访问是通过在查询接口上提交查询，这和对搜索引擎的访问在某种程度上来说是相似的，但 Deep Web 数据和搜索引擎二者之间是有着很大区别的：

- 搜索引擎搜索结果是网页，而 Deep Web 中的搜索结果主要是结构化的数据。
- Web 数据库通常有复杂的接口，而搜索引擎的接口较为简单，一般是关键字搜索。
- 搜索引擎对结果的排序是根据搜索结果与所提交查询的相似性，Web 数据库则是根据结果中某个属性的值。

2、Deep Web 数据集成系统架构

自从 21 世纪以来，随着 Internet 飞速的发展和软硬件技术的日益成熟，从 Web 中自动获取有用的信息不再只是设想，Deep Web 也受到越来越多的研究者的关注，并且越来越多的相关研究成果发表。对 Deep Web 研究的根本目的是为了能够自动地获取利用自由分布在整个 Web 上的 Deep Web 中丰富的信息并加以集成。

虽然整个 Deep Web 中几乎包含了我们所需要的任何信息，但要想以手工的方式对其加以有效的利用在实际当中是一件非常困难的事情，而对 Deep Web 数据库的集成正是为了以尽可能自动的方式来完成对 Web 数据库中信息的有效利用。

图 3 给出了Deep Web数据集成的系统框架[2]，并依照这个框架对系统中各部分功能进行简要的描述。Deep Web数据成框架共分为两个大的模块：集成查询接口的生成和对集成查询接口上查询的处理。每个模块又分为若干子模块，分别完成特定的功能。

集成查询接口生成模块：该模块整体的功能是在 Web 中发现 Web 数据库并对其按领域进行分类，在每个分类上对所有查询接口集成，为用户提供一个统一的查询接口，使之可以同时向多个实际的查询接口提交查询，即达到同时访问属于同一领域的多个 Web 数据库的目的。该部分共有四个主要的子模块：Web 数据库的发现、查询接口模式的抽取、基于领域 Web 数据库的分类和查询接口集成。Web 数据库的发现是指从 Web 中发现具有一个真正 Web 数据库的网站，然后从中发现可访问这个 Web 数据库的查询接口；查询接口模式的抽取是对前一步获得的查询接口中所包含的属性进行分析和抽取，将一个查询接口分解成为一组属性的集合；Web 数据库的分类是指根据已得到的查询接口的属性信息确定其对应 Web

[2] Weiyi Meng Lecture Notes on Web Data Management, Binghamton University, 2005

数据库所属的领域，即按照领域对 Web 数据库进行分类；查询接口的集成是对属于同一个领域的查询接口进行集成，得到一个全局的查询接口，通过这个集成的查询接口可以达到同时访问多个本地的查询接口。

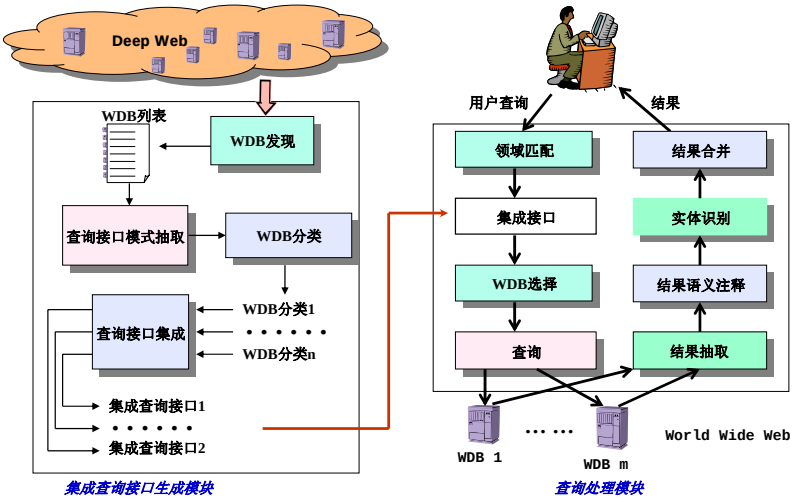


图 3 Deep Web 数据集成系统框架

查询处理模块：当用户在集成的查询接口上填写并提交查询，需要将该查询转化到对各个本地查询接口的查询，提交后各个 Web 数据库会返回符合查询的结果页面，从这些结果页面中将查询结果抽取出来并添加语义注释，由于各个 Web 数据库之间是自主性和异质性，因此所产生的结果数据格式也是各不相同，需要将这些格式各异的数据形成统一的格式，最终得到可被自动处理的数据格式。该部分包括领域的映射、Web 数据库的选择、查询分派、结果抽取、结果注释、实体识别和结果合并 7 个子模块。领域的映射是指根据使用者提交的查询自动为其选择合适的领域并把查询提交到该领域集成查询接口中；Web 数据库的选择是指从属于该领域的所有的 Web 数据库中选择出合适子集，使得既能够得到令人满意的查询结果，又可以最大限度降低所需花费的代价；查询分派是指将在集成查询接口提交的查询转化为要访问的 Web 数据库的各个本地查询接口上的查询；结果抽取是指从得到的查询结果页面中将查询结果尽量准确完全的抽取出来并形成下一步可处理的存储模式；结果注释是指对抽取的结果添加语义描述，即添加元数据信息；实体识别是指从不同 Web 数据库获得的结果中发现表示现实世界同一实体的数据，这一步是为了可以去掉结果中重复数据，即降低数据的冗余度；结果合并是指把从不同 Web 数据库获得的结果转化成统一的表现形式，以相同的模式进行存储。

3、集成查询接口的生成

为了能够同时访问多个 Web 数据库数据，在 Web 数据库集成系统中必须要提供一个统一的访问途经。每个 Web 数据库都提供了查询接口，我们需要把每个 Web 数据库的查询接口进行集成并得到一个统一的接口，该接口称为集成接口。通过在集成接口上提交查询，就达到了同时在多个 Web 数据库的查询接口提交查询的目的。为了得到这个集成接口，需要经历四个主要的步骤。首先要在 Web 上发现要集成的查询接口；其次对这些接口进行解析，获得它们的模式信息，即查询能力；第三要把它们按不同的领域分类；第四是把属于同一个领域的接口集成为一个统一的接口。

3.1 Web 数据库的发现

Web 数据库的发现是指在 Web 中发现可访问的 Web 数据库，完成这个功能主要分为两个步骤：1、找到 Web 数据库所在的网站；2、从获得的网站中发现能够对 Web 数据库查询的查询接口。比较全面而准确的把它们从 Web 中搜索出来是一件非常困难而又耗时的事情，其原因有三：首先由于目前 Web 中存在大约 450000 个可访问的 Web 数据库，这些自主的、相互独立的 Web 数据库分布在整个 Web 的各个角落，虽然对 Web 数据库做了搜集与整理，但从表 1 中可以看到只覆盖了全部 Web 数据库的很少一部分；其次 Web 是动态的、不断变化的，Web 数据库也是如此，不断有新的产生和旧的消失，即使现存的 Web 数据库内容和规模也处于不断变化之中；第三，查询接口在网页上都是以 Html 语言的 Form 元素所形成的表单的形式展现，但并不是说由 Form 元素所形成的表单都是查询接口，比如网站中用户的注册、BBS 讨论组、写发邮件，还有搜索引擎和元搜索引擎也都是表单的表现形式，要能够从中准确地识别出真正的 Web 数据库的查询接口。

对于第一步目前的解决途经有三种。第一种是从completeplanet.com和invisible-web.net这样的网站中获取，虽然不能找到所有的Web数据库，但这些Web数据库都已按领域作了分类，对于小规模的集成仍然是一个有效的方案；第二种是遍历Web中所有IP[2]，这种方案在理论上可以把所有的Web数据库完整地找出来，但目前大约有22亿3千万个有效的IP，逐个遍历显然代价过高，因此只能作为一种研究统计手段，比如估计整个Web上Web数据库的规模、Web数据库在各个域上比例分布等等；第三种是利用搜索引擎进行搜索，虽然搜索引擎不能获取Web数据库中的内容，但可以用来找到Web数据库所在网站，由于必须提交向搜索引擎提交查询，因此这种方案是基于某个领域的Web数据库的发现，也更加具有实际应用意义。其关键在于如何向搜索引擎提交有效的查询，使得含有Web数据库的网站尽可能多的出现在查询结果中，并使其排名尽量靠前。

第二步是从网站中找到可以向Web数据库提交查询的查询接口。由于查询接口和搜索引擎、元搜索引擎以及用户注册等都是以Html语言的Form元素表示，因此有两个问题需要解决：首先，通常一个网站包含上千甚至更多的页面，遍历所有页面找出显然代价太大；其次需要从所有Form元素中将查询接口准确的区分出来。在文献[2]中通过大量的观察提出了一个巧妙的办法来来解决这个问题，即从网站的主页开始以宽度优先遍历所形成的树，查询接口在这棵树中的深度不会超过5，而且94%的查询接口不会超过3，这样搜索空间就会大大降低。而对于第二个问题文献[3]基于查询接口的特征利用C4.5决策树实现了对查询接口的识别，其中主要分为两个步骤，首先是查询接口特征的产生；其次是在这些可以作为判断依据的特征之上利用C4.5算法得到一棵决策树，通过这棵决策树找出真正的查询接口。利用查询接口的特征作为判断依据是一种直观有效的解决途径，实验结果表明对于从Web中随机的数据集准确性只达到了87%，还有很大提升空间。在文献[3]中提出了一个判断页面中是否含有查询接口的一个简单方法。该方法共有三个简单的规则：首先页面中要有Form标签；其次Form标签中必须有Text输入控件；第三，至少出现一组关键词中的一个，像“查询”、“搜索”等等。这种方法在其实验中可以达到至少93%的准确性。但这些方法还有一些不完善的地方，首先它们还不能把代表Web数据库的查询接口与搜索引擎区的查询接口分开来，这就需要进一步总结这二者之间可区分的特征；另外该工作只是根据Form表单在页面中的源代码总结查询接口的特征，其实还有很多的特征可以利用，比如查询接口在页面中的视觉布局信息、所在页面的频繁词汇信息等。

3.2 查询接口模式的抽取

查询接口的模式是一组领域相关的属性集合，通过对其中若干属性的赋值形成一个对该查询接口所代表 Web 数据库的查询。对查询接口模式的抽取可以获得一个查询接口的查询能力。查询接口的模式可以被看作是建立在对应 Web 数据库上的一个视图。

对查询接口模式的抽取是指对查询接口属性的获取与分析。对查询接口模式的抽取主要目的是为了下一步的 Web 数据库分类和查询接口集成，其关键是把查询接口所包含的各个属性准确得抽取出来。

文献[4]提出了以文法分析的方式来完成对查询接口模式的抽取。对于整个页面结构的分析已经有了较为细致的工作，如文献[5-7]，但针对查询接口结构的分析该工作属于开创。这种方法首先通过观察与统计提出了这样一个假设：所有查询接口都是由隐藏的文法构建而成。为了能够准确地从一个具体的查询接口中将表示属性的各个元素组合方式识别出来，该工作通过构建解析树对整个查询接口进行解释，确定它们的语义角色，并利用优先次序解决分组方式之间存在着冲突的可能性，这样就把查询接口中的属性尽可能的发现出来了。其 precision 为 80%，recall 为 89%，显然还不能完全达到实际应用的程度。

完成属性抽取后，需要把查询接口形式化地表现出来以便于为下一步的工作提出模型化的解决方案。查询接口形式化的表达方式与应用目的相关，如果是为了对Web数据库的分类，关注的是查询接口整体的信息，即可以查询哪个域的信息，如果是为了查询接口集成，则是关注查询接口内各个属性的细节信息，即找到不同查询接口之间属性的最佳的匹配关系。最直观的方法是将查询接口看作是一个属性的集合。文献[8]提出了较为完备一种形式化的表达方式，首先整个查询接口表示一个三元组，包括查询接口所在网站的相关信息、属性的集合，由属性形成查询条件之间的关系，比如连接、非连接、排斥等。属性集合是对每个属性信息的描述，每个属性表示为一个七元组，包括属性的名称，属性在查询接口中的布局位置，属性的域类型，属性的缺省值，属性的值的类型，属性值的单位。可以看出，包含了查询接口所有有关的细节信息，可以对下一步Web数据库的分类和查询接口的集成提供足够的信息。

3.3 Web 数据库的分类

根据文献[2]在 2004 年的估计，整个 Web 中大约有 450000 个可访问的 Web 数据库，而且数目还在快速地增长。为了有效的利用这些 Web 数据库中的信息，需要将其按领域进行分类。如果手工地来完成对所有 Web 数据库的分类是个庞大而费时的工程，因此需要以尽可能自动的方式来完成对 Web 数据库的分类。由于对 Web 数据库按领域进行分类才有实际的应用意义，因此目前所提出的分类方法也都基于领域的。在查询接口上提交查询是获取 Web 数据库信息的主要途径，对 Web 数据库的分类实质上是对查询接口的分类。

分类方法共分为两类：指导方式和非指导方式。文献[9]针对应用意义最广泛的 e-commerce 的 Web 数据库提出了一种有效的分类方法。这种方法是一种非指导的方式，主要利用了 e-commerce 的 Web 数据库的查询接口所在的页面上可用的特征信息，包括接口中出现的频繁词和商品的价格特征。其实验结果表明，按这种分类方式进行分类，precision 和 recall 都在 90%左右。文献[10]完全利用查询接口的模式信息提出了一种更一般的 Web 数据库分类解决方案，属于指导方式。他们统计认为查询接口的模式信息可以作为对 Web 数据库分类的依据。基于这样的统计结论，他们提出通过建立概率模型来表示所有可能出现的属性在每个领域中出现的可能性。对于一个给定的查询接口，考察其属性集合在这个模型上计算出这个查询接口与每个领域的相似性。前面两种方法都是基于查询接口的特征信息实现对 Web 数据库的分类，另外还提出了两种利用提交样本查询来实现分类的方法，文献[11]从返回查询结果数量来分析一个 Web 数据库属于哪个领域，文献[12]则从分析返回文本的内容

来确定一个 Web 数据库的领域。这两个工作针对的不是结构化信息，而是文本信息，但其通过查询进行分类的思想可以为 Web 数据库的分类所借鉴。

3.4 查询接口的集成

查询接口的集成是为了给用户提供一个对属于同一个领域的 Web 数据库统一的访问途径，而对 Web 数据库的访问方式主要是通过查询接口，因此对 Web 数据库集成重要的一步就是查询接口的集成。集成的查询接口合并了同一领域的查询接口集合中表示同一语义的属性，保留了一些查询接口中特定的属性，并尽可能的保持该领域查询接口的结构特征和属性的顺序性。如果把各个被集成的查询接口看作 Web 数据库的一个本地视图的话，那么集成的查询接口就是建立在这些本地视图之上的全局视图。

通过属性分析是查询接口集成最主要的途径，至今已经有了许多的工作，如文献[11,13-16]。这种方式主要发掘给定查询接口的模式信息和语义信息，利用这些语义信息来识别不同查询接口上属性之间的匹配关系，在这些具体的查询接口之上获得一个集成的查询接口，达到同时访问多个Web数据库的目的。模式匹配与集成[17-21]是实现这一方式以及后面进行数据合并的一个关键技术，但主要是对已有技术的应用，所以不作过多叙述。目前对查询接口的集成主要是手工的方式，这样虽然可以达到比较高的准确性，但是在大规模集成查询接口情况下效率很难得到保证，因此需要以自动的方式来完成这个集成的过程。过去对查询接口自动集成的实现方式上可分为两大类，一类属于局部方式，是基于给定的要进行集成的查询接口集合，分析属性的隐含信息，特别是语义信息，在它们之间作属性的匹配，得到一个新的全局接口，另一类属于整体方式，是基于某个确定的主题通过对这个主题范围内大量接口的处理，发现这个主题上一般的查询接口。

局部集成方式：Wise-Integrator[8]是对e-commerce进行数据集成的一个系统，接口的集成是该系统其中的一个重要组成部分。它是一个综合的解决方案，首先对每个查询接口进行分析，获取其中的属性信息。在语义分析的过程中用到了一个很重要的工具Wordnet[3]。然后就是属性匹配，在完成对所有查询接口的属性匹配后，要为匹配的属性在集成的查询接口上确定它的全局名称和它的类型和取值范围，这样就得到了一个集成的查询接口。在实验中从正确性和完整性两个方面来衡量集成的质量，这两项的实验结果分别为95.25%和 97.91%。该工作总的来说实现了接口的集成，但也存在着不足：首先把查询接口看作是一个平的结构，实际上查询接口具有很丰富的结构信息；其次是只考虑了查询接口之间属性1:1的映射情况，但现实中的查询接口存在着大量的复杂映射。针对这些不足，文献[13]对查询接口的集成提出了较大的扩展与改进：首先，把查询接口的模式看作有层次的树状结构；其次，通过“搭桥”的方式对查询接口的属性实现更准确地匹配聚类；由于复杂的1:m映射频繁的出现，针对这种情况，对复杂映射划分为aggregate和is-a两种类型；让用户参与到集成过程中来，对集成过程加以指导。实验将完全自动和用户参与两种方式进行了对比，完全自动的方式平均precision和recall分别为88.2%和91.1%，而通过用户者的参与两项指标分别提高了7.8%和2.9%。

整体集成方式：与上面通过属性分析来发现两个查询接口之间属性对匹配方式不同，文献[22]提出了利用统计模式匹配的方案，这种方式认为 Deep Web 中同一域的数据源隐藏着一个共同的模式模型，这个模型可以刻画该域的所有查询接口共同的特征。基于这个共同的模式模型以整体的方式匹配同一个域的所有模式。基于这个思想，一个一般的模式匹配框架 MGS 被提出，包括假猜模型化、假猜生成和假猜选择。假猜模型化这一步是定义一个模型，这个模型是针对特定问题的。假猜生成是对所定义模型的参数的设定。由于会产生多个可能

[3] http://www.cogsci.princeton.edu

的模型，假猜选择则是从这些可能的模型中选择出合适的模型。文献[23,24]提出了查询接口属性相关性的观点：所有属性可分为正相关、负相关和相互独立三类。对于如何判断两个属性是正相关、负相关还是相互独立的问题，提出了自己判断标准 H-measure。在确定了不同查询接口之间属性对的关系之后要从中选择最合适的匹配。其实验结果表明对查询接口中 1:1 的匹配可以全部准确的发现，而对 m:n 的匹配可以发现全部的负相关匹配，正相关匹配在测试数据集中仅错误的一例。

3.5 小结

查询接口是访问 Web 数据库的主要途径，目前对其开展的研究工作主要集中在分类和集成两个方面，在实现方法上又可分为属性分析的方式和基于模型的方式两种。但无论怎么划分，对查询接口的分析主要包含两个方面：模式信息和属性语义。对于特定主题的查询接口，目前的工作可以得到比较理想的结果，但对于简单的查询接口（比如只有一个关键字查询的接口）和综合主题的查询接口（比如电子商务网站）就没有合适的解决办法。在未来的工作中我们可以尝试对其查询结果的分析来确定其主题和查询能力，这就需要设计一个与主题相关的样本数据库并可以自动填写查询接口和提交查询。

另外值得我们关注的是，随着 Web 服务不断发展，越来越多的大型商务网站为访问者提供了 Web Service，使得信息的获取和处理变得容易且稳定，以这种方式对信息的访问将是未来的发展趋势，但目前提供 Web Service 的网站仍然只占较小的比例，而且基于页面大规模集成的方式仍将是一个必不可少的手段。

4、Deep Web 数据查询的处理

当用户在集成查询接口上填写并提交查询时，是为了同时得到从多个 Web 数据库中获取符合该查询的结果，并把这些异构的数据以统一的模式存储或展现，这就是对 Deep Web 数据查询的处理。为了能达到这个目的，需要完成若干步骤。首先能够为用户选择合适的 Web 数据库，其次把查询近似等价地转化到在这些具体 Web 数据库查询接口上的查询，然后是从返回的结果页面中抽取查询结果并添加语义注释，最后将这些结果合并在一起。下面对这些方面做逐一介绍。

4.1 Web 数据库的选择

当完成对一个领域的查询接口的集成后，Web 数据库集成系统的用户在集成的查询接口上提交查询，这样从属于这个领域的 Web 数据库中获得所需的信息。在对 Web 数据库按领域进行分类后，对于每一个领域中 Web 数据库的数量仍然十分巨大。以商业和教育这两个领域为例，根据 CompletePlanet 的统计，都存在上千个 Web 数据库，由于 CompletePlanet 只是发现了整个 Deep Web 中大约 7%的 Web 数据库，所以在现实中还要远远大于 1000 这个数字。

当在集成的查询接口上对某一个领域进行查询时，如果只是简单地把集成接口上的查询转换到对该领域每个 Web 数据库的查询，并不是一个可行的方案，原因有三个：首先是由于一个领域中存在大量可访问 Web 数据库，虽然在理论上来说可以获得足够丰富的查询结果，但伴随而来的是，因访问大量的 Web 数据库在 Internet 上花费的代价也是难以承受的；其次并不是每一个 Web 数据库都能够满足一个特定的查询，显然任何一个领域的 Web 数据库不可能包含这个领域中所有的信息，因此也不可能满足这个领域的任意查询；第三是一个

领域中大部分的 Web 数据库之间存在着冗余的信息，因此对一个查询而言，访问的 Web 数据库越多，返回信息的冗余度也会越大，使得冗余信息的处理难度大大增加。因此，在 Web 数据库的选择这一步要达到的目标是如何从一个领域中大量的 Web 数据库选择出合适的部分，使得在满足一个特定查询的前提下尽可能的减少所访问的 Web 数据库的数量和使得查询结果中冗余度足够小。

对一个特定的查询，为了能够知道每个可访问的 Web 数据库对这个查询的满足程度，即每个可访问的 Web 数据库中符合该查询的信息量，要事先获取 Web 数据库的有用特征。由于 Web 数据库分为结构化和非结构化两类，结构化的 Web 数据库的特征是指其模式中各个属性上值的分布特征，而非结构化的 Web 数据库主要是指文本数据库，对文本数据库的查询主要使用了信息检索的技术，因此其特征是指所存储的文档集合与域相关的关键词的相似性关系。而对于搜索引擎的选择已有了许多较为成熟的工作，如文献[25-27]，其中一些技术思想可以借鉴到对结构化的 Web 数据库选择的实现中。目前对 Web 数据库特征的获取唯一途径是通过对提交查询而得到的查询结果进行分析，非结构化的 Web 数据库主要关注一个特定查询返回结果的数量，而结构化的 Web 数据库除了返回结果的数量外更主要是关注各个属性上值的分布特征。

由于结构化的 Web 数据库中存储的是由若干属性组成的现实世界的实体，在对结构化的 Web 数据库选择除了根据其大小是根据各个属性上特征表现，现在主要是在数字属性（价格、日期等）上利用直方图的方法进行特征概括。为了获得某个属性上值分布的特征，显然获取的该属性值越多越能够得到与实际相一致的特征。因此要以尽可能少的查询来获得尽可能多的结果并使得查询结果能够均匀的分布在整个 Web 数据库中，这就需要设计具有代表性的查询，既要与 Web 数据库的领域紧密相关，又要能够近似反映出当前数据库中的信息在各个属性之上的分布。另外，对 Web 数据库提交一次特定的查询往往会返回较多的查询结果，而大部分的用户并不是关注查询的全部结果，只需要前 N 位的结果就可以满足他们的查询需求了。因此，在集成各个 Web 数据库的查询结果的同时，能快速的得到最符合查询的 N 个结果是非常有应用意义的。数据分散在各个 Web 数据库之中，我们需要的前 N 个结果可能只是在某几个 Web 数据库的结果中。如果可以只向这一小部分 Web 数据库提交查询，就可以降低计算代价。文献[28]提出了一种基于直方图的 Top-N 的选择方法。该方法分为两步。第一步是判断数据库与特定查询之间的相关性。第二步是确定最适合提交查询的数据库和从返回的结果中选择最合适的记录。算法实验表明，作者这种计算 Top-N 查询的方法是非常有效的。

4.2 查询结果的抽取

Web 数据库返回的查询结果主要是通过 Html 语言编写的页面来展现的，而 Html 语言的特点是在 Web 上发布的、内容多样、形式各异，使得 Web 上的数据半结构甚至是无结构，给 Web 数据库集成系统的建立造成了极大的困难。从页面中将查询结果抽取出来的过程是指将 Web 页面上半结构和无结构的数据通过各种技术手段进行抽取出来，保存为可以自动处理的 XML 文档或关系模式，作为下一步处理的基础。

目前普遍的 Web 数据抽取方式是编写特定的抽取程序，主要具备两个功能：搜寻、发现并抽取特定的数据；以适当的格式保存数据供进一步处理，比如 XML 和关系模式。其中最大的挑战是如何从页面上大量的数据中完整准确地发现查询结果。当把 Web 数据库中的信息以 Html 页面的表现形式展现时，数据库相关模式结构信息就完全丢失了。对页面抽取的一个主要目的就是通过把信息以结构化的格式存储来反转这个过程。

目前这个研究领域已经开展了大量的研究工作，有了很多Web数据抽取的工具，按使用

的技术大致可以分为几类，下面分别作简要介绍。

页面抽取语言：是指开发一种特定设计的语言帮助使用者实现抽取过程，因此抽取是用手工的方法编写程序来实现的。抽取过程是基于过程化的程序，但是，抽取结果依赖于文档的结构。这方面主要的工作有Minerva[29]、TSIMMIS[30]、Web-OQL[31]。Minerva是Araneus[32]系统的一个重要组成部分，它结合了基于语法的声明方式和典型的过程化语言。Minerva使用的语法以EBNF定义：对每个文档，定义生成式的集合；每个生成式根据终结符和其它非终结符和定义一个语法的非终结符的结构。TSIMMIS可以通过用户写的规范文件来配置。规范文件由一系列定义抽取步骤的命令组成，通过规范文件解析Html页面，发现感兴趣的数据并进行抽取。Web-OQL其最初的目的是在Web上能够执行象SQL那样的查询。Web-OQL是一种陈述性的查询语言，能够定位在HTML页面上所选择的数据快。为了达到这种目的，包装器将页面解析抽象的语法树hypertree来表示页面。通过这种语言，可以写查询在语法树上定位感兴趣的数据并以已合适的格式输出这些数据。

基于DOM树的工具：依赖于Html页面的内在的结构特征，在抽取之前将页面转化成DOM树，可以反映页面标签的层次结构，然后自动或半自动的抽取规则在此树上应用。主要的工作有XWRAP[33]、RoadRunner[34,35]、lixto[36,37]、MDR[38]和MDRII[39]。XWRAP有一个组件库提供抽取规则生成的基本模块，这个工具引导用户通过一系列的步骤，选择每一步中正确的组件。最后，XWRAP输出特定源上的一个抽取规则。在对象抽取这步中，为Html页面预定义了六个启发式，用户可以使用其中的启发式定位感兴趣的数据对象。用户也可以为了使抽取结果更符合自己的要求限制或放宽每个对象的组件数目或指定数据类型。RoadRunner其方法是进一步发掘Html文档内在的特征来自动产生抽取规则。通过比较样本页面得到一个结果模式，从这个模式可以推测出一个能够识别出样本页面中的实例。为了准确的捕获在样本页面所有可能的结构变量，必须提供多于两个的样本页面。所有的抽取过程都基于这样一个算法，比较样本页面的标签结构产生规则的表达式来处理结构之间不匹配的情况。过程完全自动化是RoadRunner独一无二的特性。它可以说是第一个完全自动的抽取工具，具有里程碑的意义。但它对模式的推导时间复杂性是指数量级，因此在大量样本页面的情况下代价过高。MDR和MDRII这两种抽取方法都是由美国Illinois大学同一研究小组提出，其独特的地方在于能够十分准确地在DOM tree中完成对多记录页面的抽取。它们的实现关键在于利用页面的嵌套结构和表现特征把查询结果从整个页面中分离出来，并将结果中的多个记录从中彼此精确的划分，其意义是把每个记录作为现实世界的实体对待，首先从这个角度完成第一步抽取，第二步把每一条记录从属性的角度进行分解。MDR把标签树中节点的路径看作一个字符串，并使用了比较字符串编辑距离的思想从数据区中发现代表数据记录的结点，而MDRII则是以树的结构信息代替标签字符串，从而达到对数据记录更准确的识别结果。对于结果页面中记录的界定在文献[40]中早已提出，随着对页面结构和布局的不断认识，这种方式被重新加以发展深化。

抽取规则推导工具：从给定的训练样本中产生基于分隔符的抽取规则，更适合HTML文档，但需要大量的样本页面。主要的工作有WIEN[41]、STALKER[42]。WIEN是归纳工具类中的先驱，它将已经标好感兴趣数据的页面作为样本输入，与每个样本一致的抽取规则作为输出。这些页面有预定义的结构和特定启发式用来产生特定的抽取规则，但不能处理嵌套结构和典型半结构化数据的变量。STALKER能处理层次数据的抽取，输入是：(1)以一系列包含被抽取数据的符号的形式的训练样本；(2)页面结构的描述，叫做ECT。STALKER产生一个抽取规则尽量覆盖给定的样本。如果存在未被覆盖的样本，它产生一个新的分离的规则。当所有正例被覆盖后，STALKER返回一个规则的集合。使用ECT，STALKER能处理嵌套层次的对象。

基于模式的工具：为感兴趣的对象给定一个目标结构，尽量使页面上的数据部分符合这

个结构，通过图形界面与用户交互，由用户指出页面上感兴趣的区域。由于需要和用户交互，从自动化程度上来讲属于半自动抽取工具。主要的工作是NoDoSE[43,44]、DEByE[45]和SG-WRAP[46]。NoDoSE是一个半自动化交互的工具，使用图形化的用户接口，用户层次的分解文档，划出感兴趣的区域并描述它们的语义。DEByE是一种交互工具，把简单页面的样本对象集合作为输入，产生能够从其它类似页面抽取新对象的抽取模式。SG-WRAP这种方法是一种预定义模式引导的数据抽取方式，通过图形化的界面把在样本页面中要抽取的数据与预定义的模式进行连接匹配，通过这种操作产生抽取规则，完成对同类页面的有效抽取。

其它方法：抽取过程的实现还有很多方法。有的是针对页面中特定的能够结构化表现数据的标签，如文献[47,48]，显然这种方法有着很大的局限性，应用范围窄，所以这里不做过多的介绍。值得注意的是，页面中的视觉信息越来越受到研究者们的注意，目前已经有了相当的工作利用视觉信息对页面进行分析[49,50]，这里有一个重要的原因：网页被设计出来的目的是为了方便人们浏览从中获取有用的信息，而不是被计算机自动处理，因而获取页面的视觉信息可以从某种程度上模拟人类的行为对页面信息的识别。文献[51,52]在利用视觉信息对页面分块的基础上进行了Web搜索和链接方面的研究，而利用视觉信息在Web数据库查询结果抽取方面目前是作为一种有用的辅助手段。文献[39]在由页面形成的DOM树中为元素添加了在浏览器中的位置信息，并认为每个节点在视觉上占据了一个矩形的区域，而且父节点所占据的矩形区域包含子节点占据的区域，通过节点的位置和大小信息可以准确地发现在DOM树中不连续的数据记录，而这种情况对以往只利用页面的源码作抽取的Wrapper来说是无法解决的。文献[53]是针对搜索引擎的查询结果而提出的工作，它把视觉信息和DOM树结构结合起来发现和分离查询结果。

从前面可以看到，到目前为止已经有了如此多的抽取工具，并按照实现技术进行了分类，如何评价抽取工具的性能，可以从下面几个角度来看待。准确性，这是最为重要的标准，可以借用信息检索的两个主要概念准确率 (Precision)和召回率 (Recall)来衡量：准确率在这里指抽取到的正确结果与抽取到的全部结果的比；召回率在这里指抽取到的正确结果与要抽取页面的全部结果的比。自动化的程度，这是另一个比较重要的标准，关系到在抽取的过程中使用者参与的程度。这也是对Web抽取工具的另一个分类方式，即手工、半自动和完全自动。目前完全自动的抽取方法已经完全取代了手工和半自动的方式成为主要的趋势。弹性和适应性，由于Web页面的内容和结构经常发生变化，抽取工具要有自适应的能力，即当页面结构发生较小的变化时也能继续正常工作，这成为弹性。一个抽取工具为某个特定领域的页面而生成，如果它也能为这个领域另一个数据源的页面工作，这称为适应性。这对于高度动态的Web而言尤为重要。使用的方便程度，提供图形化界面使抽取规则的生成更加容易。这主要是针对半自动的方式而言。另外大部分抽取工具都或多或少的需要调整参数，参数过多或过于复杂也会使其可用性降低。

Web 数据抽取是 Web 数据库集成系统中发展最为成熟的部分，我们对 Web 数据抽取工具进行了分类和总结，分类的方法主要根据技术实现的角度，可以看出涉及了各种各样的方法，而且随着 Web 的发展，新的方法会不断地出现。作为 Web 数据集成的重要一环，在这个领域还远没有达到令人满意的程度，尤其是在准确性上。语义 Web 的提出就是为了使计算机能够对页面中的数据进行自动处理，不过在目前看来要做到全面替代传统的 Html 页面还有很长的路要走。

4.3 小结

至今在对 Web 数据库查询处理这一模块中，各个研究问题发展很不平衡。从页面中抽取数据已经比较成熟，各种技术方法被提出从理论和应用中解决这个问题。相对来说，其它

的子问题，比如 Web 数据库的选择、数据的语义添加、数据合并等，还处于空白阶段或刚刚开始被研究者们关注，但作为 Deep Web 数据集成系统不可缺少的组成部分，需要研究者们在这些研究问题上给予更多关注与努力。

5、未来工作的展望

随着 Web 数据库在 Web 中不断大量的涌现，对 Web 数据库进行大规模集成的研究成为一个非常迫切的问题。至今，人们在 Deep Web 领域已经作了大量的研究，所提出的 Deep Web 数据集成系统有文献[8,54]，但它们只是属于研究性的原型系统，因此确切地说至今还没有一个真正可以作为实际应用的 Deep Web 数据集成系统。前面对这些工作按照 Deep Web 数据集成系统的框架进行了分类和概括总结，然而大部分工作仍然处于探索性的阶段，只有查询接口的集成和查询结果的抽取这两个方面的工作相对成熟，有些方面的工作到目前可以说是刚刚开始甚至仍然是空白。因此要实现一个真正可用的集成系统仍然有许多的问题有待更深入的研究。下面就 Deep Web 数据集成系统框架中仍然需要开展的工作做初步的展望。

Web 数据库的发现：利用成熟的传统搜索引擎完成对 Web 数据库的搜索是一种行之有效的办法。由于查询接口存在于静态的页面中，因此可以被传统的搜索引擎爬取到。如果能够借助搜索引擎强大的搜索能力，那么就大大降低了搜索代价。这种方法虽然是可行的，但也包含了挑战性的工作。搜索引擎的作用是搜索 Web 中的页面，获取页面唯一途径是提交关键词查询，而包含 Web 数据库查询接口的页面只占全部页面很小的比例，如果提交的关键词不合理，会导致搜索到的页面结果集中所包含的查询接口比例太小，使得不仅每次获得的 Web 数据库数量少，而且也会使筛选的代价过高。因此设计合理的关键词查询是利用搜索引擎获取 Web 数据库的关键问题。由于 Web 数据库的查询接口在页面中以 Form 表单形式表现，但 Form 表单还可以有很多种用途，搜索引擎和元搜索引擎的查询接口在页面中的表现形式与 Web 数据库的查询接口更加相似，如何把 Web 数据库查询接口从中准确的发现出来至今仍未达得到很好的解决。为了能够准确地判断一个页面中的 Form 表单元素是否是一个真正的 Web 数据库的查询接口，有两个十分有用的方法还未加以利用：首先，页面中一般包含有比较丰富的语义信息，通过这些语义信息可以用来帮助我们判断一个 Form 表单元素的用途；其次，通过提交试探性查询，根据返回结果的数量来判断，比如判断一个 Form 表单是不是一个图书信息的查询接口，可以提交“Thinking in Java”，如果有包含该书的查询结果信息，则说明此 Form 表单极有可能是一个图书信息的查询接口，甚至可以进一步判断为一个计算机图书信息的查询接口。

Web 数据库的分类：已有的工作总的说仍未把 Web 数据库的分类问题彻底解决，其根本原因是只是利用了查询接口自身及所在页面所提供的信息，当属性信息非常类似时就会无法区分。另外有些领域的 Web 数据库为了方便用户的查询，提供了极其简单的查询接口，如音乐和图书领域，经常只需填写关键字，使得仅依赖查询接口的模式信息很难判断出这个接口属于哪个领域。为了解决这些情况，可以从两个方面考虑。首先根据领域之间的不同特征要能够实时调整相似性判断函数里的判断标准，并可以多阶段的执行分类过程。其次通过在查询接口上提交与领域相关的查询，根据返回结果进行分类，这是直接判断一个 Web 数据库属于哪个领域的最有效方法。以汽车、音乐和图书三个领域的分类为例，如果提交“Thinking in Java”的查询，前两个领域的 Web 数据库将不会返回任何结果，而图书领域的 Web 数据库则可能返回若干结果。提交样本查询也是 Web 数据库发现的一种有效方法，进一步说，如果能够设计一个合适的领域相关的样本查询集合，就可以把 Web 数据库发现和分类两个步骤合并在一起，叫做基于领域的 Web 数据库的发现，这样不仅保证了更高的准确性和效率，而且更具有实际应用意义。

Web 数据库的选择：由于 Web 数据库数量的不断增长使得 Web 数据库的选择成为一个急待解决的问题。为了能够降低对 Web 数据库的访问代价和获得高质量的数据，需要在同一个领域中选取合适的 Web 数据库进行查询，不可避免要对这些 Web 数据库进行特征概括，通过这些特征概括来判断一个 Web 数据库是否与给定的查询相关程度。目前已有的工作主要是针对搜索引擎和 Web 数据库中非结构化的文本数据库提出，而对于比例最大的结构化 Web 数据库而言，现有的工作是在数字属性(如价格、日期等)和离散属性(如有限种选择的属性)上进行特征概括，虽然对 Web 数据库的选择起到了一定的作用，但还未从根本上解决问题。因此下一步的研究工作要能够对非数字的不可穷举属性进行有效的特征概括，这就要提出不同的方法来处理这类属性。随着本体和语义领域理论的不断成熟，可以借助于建立一个特定域的本体来对一个 Web 数据库进行特征概括，建立一个概念的层次树结构，最低层节点是属于父节点概念的实例集合，这样通过实例查询可以估计每层的每个分类在一个 Web 数据库中所拥有的信息比例，从而能够更好的刻画 Web 数据库在这个属性上的特征总结。另外如果把 Web 数据库中所有的数据全部通过查询的方式获取出来，虽然可以对其做最好的特征概括，但考虑到对 Web 数据库的访问代价和 Web 数据库内容频繁的更新，就失去了其应有的意义。因此需要设计有效的查询获得 Web 数据库中的部分数据，使得这些部分的数据能够具有对全部数据的代表性，通过这种“以偏概全”的方式来降低对 Web 数据库特征概括的代价。通过在查询接口上提交试探查询的方法不仅对 Web 数据库的选择，对 Web 数据库的发现和分类也有极其重要的意义，因此如何对一个特定的 Web 数据库构建高质量的查询自动生成器是将来迫切需要解决的问题。

对查询结果的语义注释：为了使从页面中抽取到的数据具有使用价值，必须要为其添加语义注释，而目前在这方面的工作还在初步阶段，都是以启发式规则的方式对抽取到的数据进行语义注释[55,56]，不仅准确性还未达到实际应用的标准，而且更重要的是不能对抽取到的全部数据添加语义注释。为了能够把从各个Web数据库的数据有效的集成起来并加以利用，要对这两个方面做较大的改进。对于抽取数据的自动语义添加，如果是针对一个特定的Web数据库，可以通过机器学习的方式预先在一组样本页面上训练形成一个自动添加语义的程序，学习出数据与对应语义之间的关系，从而能够处理新的页面。考虑是在Web数据库集成系统的环境下，有的Web数据库的查询接口或结果中能够得到某个属性数据的语义，而有的Web数据库对这个属性的数据则没有语义注释，如果能够对各个Web数据库的模式之间建立匹配关系，利用预先建立的模式匹配关系就可以以互补的方式达到对数据语义的添加，但要保证语义的正确性前提是要保证这种模式匹配关系的正确性，由于页面结构化程度很差，目前还很难保证在页面中模式匹配较高的正确性。

Web数据的合并：在Web数据库集成系统中，最终要把从各个Web数据库获得的数据合并到一个统一的模式下。在实际中，各个Web数据库的数据经常存在大量的重复，因此在合并过程中必须要解决的一个重要问题是数据的去重。由于这些数据描述的是现实世界的实体，如果能从实体的角度把重复的数据加以识别将会有效地达到去重的目的。实体识别的问题其实普遍存在于对多个数据源集成的领域中，以前的关于对实体的识别还是主要采用手工的办法根据特定的应用领域和使用者的要求来定制特定的规则，为了达到较高的准确性，不可避免的要求大量人工的参与，而且只能适用于特定的领域。目前实体识别只是在关系模式和半结构化的XML模式上开展[57,58]。在实体识别的问题中有两个关键的子问题：建立实体之间属性的映射关系和属性之间值的比较。实体之间属性的映射即模式匹配，由于Web页面结构化程度很差，传统的模式匹配方法难以直接应用，对于大规模的集成手工的方式更加不可取，因此需要提出新的方法来解决Web环境下的属性匹配问题；属性之间值的比较则首先选取能够代表实体的属性，然后在这些代表性的属性上值的比较，由于各个Web数据库的异质性，要从语义角度来判断属性值对之间是否表达统一语义。在Web环境下至今还没有真正

意义上实体识别的工作存在，但这又是Web数据库集成系统中不可缺少的一个关键环节，这将是未来最为迫切的问题之一。

Web 数据的增量维护：Web 数据库的数据经常处于频繁更新的状态，而用户总是希望能够得到当前 Web 数据库中最新的内容。在多数据源集成的研究领域，对集成数据的增量维护是一个无法避免的问题， 同样对 Web 集成数据增量维护问题的重要性将随着 Web 数据库集成系统的不断成熟显得日益突出。由于 Web 数据处于快速动态更新的状态，而且 Web 页面模板也频繁的发生变化，使得增量维护变得更加复杂，需要提出新的方法来自动检测增量的 Web 数据并适应 Web 页面模板的变化。

6、结束语

随着 Web 数据库数量和其蕴含数据量飞速的增长，对 Deep Web 数据的集成越来越成为研究领域关注的问题，目前人们已经在这个方面做了大量的工作，本文对最近几年来国际上在该领域的主要研究成果进行了回顾与总结，综述了 Deep Web 数据集成系统中若干主要问题的研究现状，包括 Web 数据库的发现、Web 数据库模式的抽取、Web 数据库的分类、查询接口的集成、Web 数据库的选择、结果数据的抽取等等，并在综述的同时指出仍然存在的问题和将来可能的解决办法。但总的来说对 Deep Web 数据集成的研究仍然处于刚刚起步的阶段，离应用阶段还有很长的路要走，仍然有大量关键的问题还需要做深入细致的研究。

参考文献

[1] Fetterly D., Manasse M., Najork M., Wiener J. L.. A large-scale study of the evolution of web pages. In: Proceedings of the 12th International World Wide Web Conference, Budapest, 2003, 669-678

[2] Chang K. C., He B., Li C., Patel M., Zhang Z.. Structured databases on the web: Observations and Implications. SIGMOD Record, 33, 3, 61-70

[3] Cope J., Craswell N., Hawking D.. Automated discovery of search interfaces on the Web. In: Proceedings of the 14th Australasian Database Conference(ADC 2003), Adelaide, 2003, 181-189

[4] Zhang Z., He B., Chang K. C.. Understanding Web query interfaces: best-effort parsing with hidden syntax. In: Proceedings of the 23th ACM SIGMOD International Conference on Management of Data, Paris, 2004, 107-118

[5] Arasu A., Garcia-Molina H.. Extracting structured data from Web pages. In: Proceedings of the 22th ACM SIGMOD International Conference on Management of Data, San Diego, 2003, 337-348

[6] Crescenzi V., Mecca G., Merialdo P.. RoadRunner: towards automatic data extraction from large web sites. In: Proceedings of the 27th International Conference on Very Large Data Bases, Italy, 2001, 109-118

[7] Wittenburg K. Weitzman L.. Visual Grammars and Incremental Parsing for Interface Languages. In: Proceedings of the IEEE Symposium on Visual Languages (VL), Skokie, 1990, 111-118

[8] He H., Meng W., Yu C. T., Wu Z.: WISE-Integrator: an automatic integrator of Web search interfaces for e-commerce. In: Proceedings of the 29th International Conference on Very Large Data Bases, Berlin, 2003, 357-368

[9] Peng Q., Meng W., He H., Yu C. T.: WISE-cluster: clustering e-commerce search engines automatically. In: Proceedings of the 6th ACM International Workshop on Web Information and Data Management, Washington, 2004, 104-111

[10] He B., Tao T., Chang K C.. Clustering structured Web sources: a schema-based, model-differentiation Approach. In: Proceedings of the 9th International Conference on Extending Database Technology, Heraklion, Crete, 2004, 536-546

[11] Ipeirotis P. G.., Gravano L., Sahami M.. Probe, count, and classify: categorizing hidden Web databases. In: Proceedings of the 19th ACM SIGMOD International Conference on Management of Data, Santa Barbara, 2001, 67-78

[12] Meng W., Wang W., Sun H., Yu C.. Concept hierarchy based text database categorization. Knowl. Inf. Syst., 2002, 4, 2: 132-150

[13] Wu W., Yu C. T., Doan A., Meng W.. An interactive clustering-based approach to integrating source query interfaces on the Deep Web. In: Proceedings of the 23th ACM SIGMOD International Conference on Management of Data, Paris, 2004, 95-106

[14] He H.i, Meng W., Yu C. T., Wu Z.. Constructing interface schemas for search interfaces of Web databases. In: Proceedings of the 6th International Conference on Web Information Systems Engineering, New York, 2005, 29-42.

[15] He H., Meng W., Yu C. T., Wu Z.: Automatic integration of Web search interfaces with WISE-Integrator. VLDB Journal, 2004, 13, 3: 256-273

[16] Wu Z., Raghavan V., Du C., Sai K. C., Meng W., He H., Yu C. T.. SE-LEGO: creating metasearch engines on demand. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, 2003, 464

[17] Li W., Clifton C.. Semantic integration in heterogeneous databases using neural networks. In: Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, 1994, 1-12

[18] Miller R J., Ioannidis E Y., Raghu R.. Schema equivalence in heterogeneous systems: bridging theory and practice. Inf. Syst., 1994, 19, 1: 3-31

[19] Milo T., Zohar S.. Using schema matching to simplify heterogeneous data translation. In: Proceedings of the 24th International Conference on Very Large Data Bases, New York, 1998, 122-133

[20] Gio Wiederhold: Meditation to Deal with Heterogeneous Data Sources. In: Proceedings of the 2th International Conference on Interoperating Geographic Information Systems, Zurich, 1999, 1-16

[21] Doan A., Domingos P., Levy A. Y.. Learning source description for data integration. In: Proceedings of the 3th International Workshop on the Web and Databases, Dallas, 2000, 81-86

[22] He B., Chang K. C.: Statistical schema matching across Web query interfaces. In: Proceedings of the 22th ACM SIGMOD International Conference on Management of Data, San Diego, 2003, 217-228

[23] He B., Chang K. C., Han J.. Discovering complex matchings across web query interfaces: a correlation mining approach. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, 2004, 148-157

[24] He B., Chang K. C., Han J.. Mining complex matchings across Web query interfaces. In: Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Paris, 2004, 3-10

[25] Leake D. B., Scherle R.. Towards context-based search engine selection. In: Proceedings of the 5th International Conference on Intelligent User Interfaces, Santa Fe, 2001, 109-112

[26] Meng W., Yu C. T., Liu K.. Building efficient and effective metasearch engines. ACM Comput. Surv., 2002, 34, 1: 48-89

[27] Yu C., Liu K., Meng W., Wu Z., Rishe N.. A methodology to retrieve text documents from multiple databases. IEEE Trans. Knowl. Data Eng., 2002, 14, 6: 1347-1361.

[28] Yu C. T., Philip G., Meng W.. Distributed top-N query processing with possibly uncooperative local systems. In: Proceedings of the 29th International Conference on Very Large Data Bases, Berlin, 2003, 117-128

[29] Crescenzi V., Mecca G.. Grammars have exceptions. Inf. Syst., 1998, 23, 8: 539-565

[30] Hammer J., Hector G., Nestorov S., Yerneni R., Breunig M. M., Vassalos V.. Template-based wrappers in the TSIMMIS system. In: Proceedings of the 16th ACM SIGMOD International Conference on Management of Data,

Tucson, 1997, 532-535
[31] Arocena G. O., Mendelzon A. O.: WebOQL: restructuring documents, databases, and Webs. In: Proceedings of the 14th International Conference on Data Engineering, Orlando, 1998, 24-33
[32] Mecca G., Atzeni P., Masci A., Merialdo P., Sindoni G.. The Araneus Web-base management system. In: Proceedings of the 17th ACM SIGMOD International Conference on Management of Data, Tucson, 1998, 544-546
[33] Liu L., Pu C., Han W.. XWRAP: An XML-enabled wrapper construction system for Web information sources. In: Proceedings of the 16th International Conference on Data Engineering, San Diego, 2000, 611-621
[34] Crescenzi V., Mecca G., Merialdo P.. RoadRunner: towards automatic data extraction from large Web sites. In: Proceedings of the 27th International Conference on Very Large Data Bases, Roma, 2001, 109-118
[35] Crescenzi V., Mecca G., Merialdo P.. RoadRunner: automatic data extraction from data-intensive web sites. In: Proceedings of the 21th ACM SIGMOD International Conference on Management of Data, Madison, 2002, 624
[36] Baumgartner R., Ceresna M., Gottlob G., Herzog M., Zigo V.. Web information acquisition with lixto suite. In: Proceedings of the 19th International Conference on Data Engineering, Bangalore, 2003, 747-749
[37] Baumgartner R., Flesca S., Gottlob G.. Visual web information extraction with lixto. In: Proceedings of 27th International Conference on Very Large Data Bases, Roma, 2001, 119-128
[38] Liu B., Grossman R. L., Zhai Y.. Mining data records in Web pages. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, 2003, 601-606
[39] Zhai Y., Liu B.. Web data extraction based on partial tree alignment. In: Proceedings of the 14th International World Wide Web Conference, Chiba, 2005, 76-85
[40] Embley D. W., Jiang Y. S., Ng Y.. Record-boundary discovery in Web documents. In: Proceedings of the 18th ACM SIGMOD International Conference on Management of Data, Philadelphia, 1999, 467-478
[41] Kushmerick N.. Wrapper induction: efficiency and expressiveness. Artif. Intell., 2000,118, 1-2: 15-68
[42] Muslea I., Minton S., Knoblock C. A.. Hierarchical wrapper induction for semistructured information sources. Autonomous Agents and Multi-Agent Systems, 2001, 4, 1/2: 93-114
[43] Adelberg B., Denny M.. Nodose version 2.0. In: Proceedings of the 18th ACM SIGMOD International Conference on Management of Data, Philadelphia, 1999, 559-561
[44] Adelberg B.. NoDoSE - a tool for semi-automatically extracting semi-structured data from text documents. In: Proceedings of the 17th ACM SIGMOD International Conference on Management of Data, 1998, 283-294
[45] Laender A. H. F., Berthier A. R., Altigran S.. DEByE - data extraction by example. Data Knowl. Eng., 2002, 40, 2: 121-154
[46] Meng X., Lu H., Wang H., Gu M.. SG-WRAP: a schema-guided wrapper generator. In: Proceedings of the 18th International Conference on Data Engineering, San Jose, 2002, 331-332
[47] Cohen W. W., Hurst M., Jensen L. S.. A flexible learning system for wrapping tables and lists in HTML documents. In: Proceedings of the 11th International World Wide Web Conference, Budapest, 2002, 232-241
[48] Pinto D., McCallum A., Wei X., Croft W. B.. Table extraction using conditional random fields. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, 2003, 235-242
[49] Cai D., Yu S., Wen J., Ma W.. Extracting content structure for Web pages based on visual representation. In: Proceedings of the 5th Asian-Pacific Web Conference, Xian, 2003, 406-417
[50] Song R., Liu H., Wen J., Ma W.. Learning important models for web page blocks based on layout and content analysis. SIGKDD Explorations, 2004, 6, 2: 14-23
[51] Cai D., Yu S., Wen J., Ma W.. Block-based web search. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, 2004, 456-463
[52] Cai D., Yu S., Wen J., Ma W.. Block-level link analysis. In: Proceedings of the 27th Annual International

ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, 2004, 440-447
[53] Zhao H., Meng W., Wu Z., Raghavan V., Yu C. T.. Fully automatic wrapper generation for search engines. In: Proceedings of the 14th International World Wide Web Conference, Chiba, 2005, 66-75
[54] Chang K. C., He B., Zhang Z.. Toward large scale integration: building a MetaQuerier over databases on the Web. In: Proceedings of the 2th Biennial Conference on Innovative Data Systems Research, Asilomar, 2005, 44-55
[55] Arlotta L., Crescenzi V., Mecca G., Merialdo P.. Automatic annotation of data extracted from large Web sites. In: Proceedings of the 6th International Workshop on Web and Databases, San Diego, 2003, 7-12
[56] Wang J., Lochovsky F. H.. Data extraction and label assignment for web databases. In: Proceedings of the 12th International World Wide Web Conference, Budapest, 2003, 187-196
[57] Lim E., Srivastava J., Prabhakar S., Richardson J.. Entity identification in database integration. Inf. Sci. 1996, 89, 1: 1-38
[58] Wei W., Liu M., Li S.. Merging of XML documents. Conceptual Modeling - ER 2004, 23th International Conference on Conceptual Modeling, Shanghai, 2004, 273-285

可信数据库系统研究

肖珍 尹少宜(mobile 组) 谢敏(xml 组)

1. 引言

长期以来，邮件，财务记录，医疗图像，质量保证文档，订单记录一直是有巨大价值的资产，它们记录的事件是商业运作等关键性事务的决策基础。随着信息技术的发展，这些资料越来越多地以电子记录的方式被保存在数据库系统里，使得用户能快速的读写这些数据。但另一方面，为了谋取私利的攻击者也可能对这些数据进行修改而不留下痕迹，从而影响商业决策，损害国家和公众的利益。

1.1. 社会背景

2001 年 12 月，美国最大的能源公司------安然公司，突然申请破产保护，此后，公司丑闻不断，规模也“屡创新高”，特别是 2002 年 6 月的世界通信会计丑闻事件，“彻底打击了（美国）投资者对（美国）资本市场的信心”（Congress report, 2002）。在安然事件中，公司 CEO，CFO 等通过发布虚假消息（虚假交易事件）和编造虚假的公司财务报告（虚报利润等）来误导公众，哄抬股价。另一方面，负责对安然公司财务进行审计的独立审计公司安达信的审计人员每年都会收到安然的巨额审计费，所以违背了审计人员的道德准则，对安然存在的问题视而不见，没有及时的向公众披露。在此案的审理过程中，美国检察官发现安然的公司的财务报告等运营数据完全是不可靠的，不真实的，从而导致了最后公司破产、股价暴跌、雇员失业，股民财产损失巨大，而公司的 CEO，CFO 等却通过抛售获得了巨大的个人收益。

在这一案件中，正是由于电子数据的管理不善，所以带来了国家和公众的巨大损失。随后，针对美国国内大规模上市公司的财务丑闻和审计标准屡屡拉响警报。在这些丑闻中，许多公司主管声称他们不应当对虚假财务报表负责，甚或他们根本不知道这些是虚假报表。为了转变这一信用危机局面，挽回公众对政府的信心，投资者对资本市场的信心。美国国会和政府加速通过了一系列法律法规，来规定企业的电子数据如何保留和储存的问题，从而准确地从上市公司的电子文档记录中掌握有关上市公司内部运营的信息。

其中包括：约束证券经纪商的美国证券交易委员会规范 SEC （Securities Exchange Commission），全美证券交易商协会行为规定（NASD 3110），约束医疗保健业的美国健康保险便利和责任法案（Health Insurance Portability and Accounting Act，简称 HIPAA），规定生命科学的联邦条例 21CFR 第 11 部分，规定美国国防部电子记录管理应用（RMA）软件的设计标准的 DOD 5015.2-STD 标准，约束上市公司财务管理等行为的萨班斯-奥克斯莱法案（Sarbanes-Oxley）法案，金融服务业的数据安全性规范 Gramm-Leach-Bliley 法案 (Gramm-Leach-Bliley Act) 等众多法案。这些法案对电子记录在完整性、保密性、可存取性、可靠性等各方面都有明确规定。

1.2. 法律背景

1.2.1. 企业财务：萨班斯法案

在安然事件等一系列华尔街财务丑闻中，许多公司主管声称，他们不应当对虚假财务报表负责，也根本不知道这些报表是虚假的。美国政府由此制定了影响极为深远的萨班斯法案，该法案的另一个名称是“公众公司会计改革与投资者保护法案”。法案的第一句话就是“遵守证券法律以提高公司披露的准确性和可靠性，从而保护投资者及其他目的”。该法案管制对象仅限于在资本市场运作资金超过 7500 万，并且每季度必须向证券交易委员会提交报表的股份公司的美国上市公司和美国企业的海外分支机构及子公司。该法案规定了强化信息披露、监管责任、内部控制和外部审计等制度，要求公开上市公司需定时地公布准确详细的财务报告，强制设置审计委员会，并规定该委员会由独立董事组成，该独立董事没有担任公司管理层级职务，也不从公司领取薪金。其主要目的是确保审计人员的独立性，授予其审核公司财务记录的权限与自主性，并监督经理人的工作绩效。该法案对各公司信息保存有如下规定（http://www.kahnconsultinginc.com）：

- 公司的首席执行官和首席财务官必须亲自证明，他们提交的财务报表是真实的，否则将承担刑事责任。
- 对信息保护的能力。要求公司运用“细致入微”的存取控制和保护手段，防止非授权或因疏忽而更改、毁坏或破坏业务记录和财务信息。
- 对信息准确跟踪的能力。要求公司能够提供审计人员与保存关键记录和信息的系统之间所有交互行动的“审计轨迹”。信息和记录以及文档管理软件、硬件和安全存储环境形成了关键的“内部控制”，它确保各公司其财务和业务信息是准确和可靠的。
- 对信息长期保存的能力。要求公司确保用于保留要求记录的存档和存储系统和介质将支持长期可靠存取。对某些特定信息要求保存长达七年的时间。
- 从该材料提交的财政年度末开始计算，若不可提供 5 年内工作材料供审计或审查，将处以最高 5 年的监禁，并可被课以罚款，或单独课以罚款。
- 为了阻碍美国联邦调查，故意更改、销毁、损坏、隐匿、包庇、伪造任何记录、文件或者有形物体，导致记录的完整性受损或者在任何记录、文件或者有形物体中制造虚假条目导致记录的可靠性受损的人都将承担刑事责任，使其作为官方处理证据的价值受影响，可以被判以最高 20 年监禁，并可被课以数额不等罚款，或单独课以数额不等的罚款。

1.2.2. 证券交易：SEC 17a-4

如今的证券业的信息交流基本上是通过电子方式，包括电子邮件、即时信息传输以及各种电子表格（票据、文件、批准书等）。美国证券交易委员会要求券商将所有客户通信的电子资料和其他经纪记录都存储在不可删除、不可改写的介质上。除此之外，美国证券交易委员会还要求企业对各种频繁和广泛的信息需求做出迅速反应。美国证券交易委员会 SEC（Securities & Exchange Commission) 的第 17a-4 条法案（简称：SEC 17a-4）对交易记录的保存有如下规定：

（http://www.law.uc.edu/CCL/34ActRls/rule17a-4html）

- 数据必须存储在“不可改写、不可删除”的介质上，也就是说只能以“不可覆盖，不可擦除”的方式保存记录；
- 自动验证存储介质记录过程的质量和准确性；
- 将原存储介质及其副本单元（如果合适）以及此电子介质上存储信息的保存期的日期和时间序列化；
- 有能力响应交易委员会或自律机构的要求随时将电子存储介质上保存的索引和记录方便地下载到任何可接受的介质上。

对电子记录所做出的以不可改写、不可擦除的格式保存的要求是要确保信息的完整性。验证信息的质量和准确性实际上也是对信息完整性的要求。在 SEC 17a-4 所规定的“能够及时下载保存在电子存储介质上的索引和记录”，是对电子记录的可存取性的具体规定。当然，保密性对所有经纪行的运营来说也是一个重要考虑事项。

1.2.3. 医疗健康：HIPAA 法案

1996 年，美国国会通过了健康保险便携性和责任法案（HIPAA），其中有关个人的医疗健康信息隐私权的条款于 2003 年 4 月 14 日生效。该法案打破了传统的由医疗提供者拥有患者个人医疗记录的观念，转化为以客户为主，由个人自己来决定自己的医疗信息将如何被使用的观念。政府的立场是，通过在线访问患者资料的方式可以大大提高医疗服务的质量，但同时也应当保护患者的隐私，防止对秘密数据的不正当使用。尽管该法案本身并没有对数据的存储方式做出要求，但要求医疗机构和其他相关机构（医院、保险公司和卫生维护组织）使用安全的系统和介质来对所有患者的记录进行电子化管理，保存 HIPAA 安全标准要求的相关文档，并接受对这些资料和相关过程的定期复查。

该法案对多种医疗健康产业都具有规范作用，包括交易规则、医疗服务机构的识别、从业人员的识别、医疗信息安全、医疗隐私、健康计划识别、第一伤病报告、病人识别等。该法案的主要目标如下：

- 保证劳动者在转换工作时，其健康保险可以随之转移；
- 保护病人的病例记录等个人隐私；
- 促进国家在医疗健康信息安全方面电子传输的统一标准。

HIPAA 条例定义：

- 受保护的医疗信息（Protected Health Information，简称 PHI）包含以任何形式或者媒体传播的所有的医疗信息，不管是口头的还是有记录的。
- PHI 主要是由以下对象所创建或者接收到的：医院、健康计划部门、保健服务商、相关票据交换所、医疗信息系统提供商、医科大学、甚至只有一个内科医生的办公室，雇主，保险公司，学校等。
- PHI 是和以下信息相关的：某个个人过去、当前或者未来的身体或者精神健康状况；向患者提供的医疗服务；过去、当前或者未来对医疗服务的支付费用
- PHI 能够直接或者间接用于确认患者个人身份。

HIPAA 条例有以下规定：

- 对任何形式的 PHI 的存储、维护和传输都必须遵循 HIPAA 的安全条例规定，并且大部分组织必须在两年内达到要求。
- 对于违反 HIPAA 安全条例的行为，可以处以最高为 25 万美元的罚款和最长为 10 年的监禁。
- 保密性：对数据访问的保护和监控，保护数据免受非法访问，如病人的病例属于个

人隐私，应予以保密

- 一致性：保护数据免受非法修改和删除。
- 可用性：系统和数据处于可访问和运行阶段的时间长度。

1.2.4. 生命科学和制药行业：FDA 法案

美国食品与药物管理局 FDA (Food and Drug Administration) 的 21 CFR (Code of Federal Regulations) Part 11 (Electronic Records and Signatures)法案，意在简化药品从开发到投入市场的过程，并使之更有效。该条例的目标是，编制并管理有关药品开发、药品测试和批量制造的信息流，从而加快这一过程，并使之更加安全。无论是确保药品在获得审批之前经过彻底测试，还是保证药品经过充分的实验和调查，严格的记录都是必不可少的。由于大多数制药公司掌握的药品临床测试数据都与患者有关，它们必须同时按照 HIPAA 法案的要求，保证数据的机密。具体的对电子记录和签名有如下规定：（http://www.21cfrpart11.com）

- 确保电子记录的真实性、完整性。
- 验证系统以确保具有准确性、可靠性，一致的、所希望的性能，和识别无效或被篡改报告的能力。
- 保护记录以使它们在整个保留期内都准确而且可以随时检索。
- 将系统访问权限制到有权访问的人。
- 使用安全、由计算机生成而且加盖时间戳的审核跟踪，“其保留期至少应与主电子记录保留期一样长”
- 对开放系统的控制。“文档加密等附加措施”

通常,制药厂在将一种新药品推向市场的过程中，将会产生大量的文字资料。FDA 发布这一管理法规，提出确认系统以确保精确性，及保护记录以支持记录的“精确性和及时检索”标准。其基本要求就是提供系统验证和保留管理能力，以确保数据的完整性。

从以上这些美国法案对于电子记录存储的规定举例，我们可以看到，为了达到这些要求，需要有一个可信的数据库来存储和管理电子记录，使得里面存储的电子记录是可信的。那么可信的数据库包括哪些方面的技术，有什么要求呢？下面将详细的分析。

2. 可信的数据存储

2.1. 理论和案例分析

记录管理实际上就是记下那些对于企业和组织的业务非常重要的事情作为历史备案，便于审计及今后的所有调查、取证和分析等工作。可信数据库存储的电子记录需要满足以下要求，提供以下特性：

- 信息的完整性：所有相关记录都保存。
- 信息的准确性：数据记录描述准确，精确。
- 信息的可靠性：数据是真实可靠的，没有虚假数据。
- 信息的安全性：相应权限控制；数据不能被非法使用；
- 信息的可获取性：能够在适当的时间以适当的格式访问任意的数据。
- 信息的不可更改性：历史数据记录不能被更改（被覆盖，被擦除）。
- 信息的时效性:所有信息都是在各项相关政策和规定所要求的时限上满足以上要求

的，数据有自己的生命期。

- 日志：能够安全地记录所有数据操作：创建、改动和删除等日志信息。

以公司的交易记录为例：

- 完整性要求对某一项交易相关的数据都要有记录，以免出现数据的不一致性，比如收支不平衡等；
- 准确性要求相关数据一定要与事实相符合，交易的价格数量等不能有超过容忍的偏差，利润必须与事实吻合，不得虚报以哄抬股价，也不得漏报以偷税漏税；
- 由于数据反映了已经发生的事件的情况，是一种“证据”，所以必须是可靠的，不能有虚假的交易记录并不存在的交易事件；
- 安全性要求逻辑上由相应的访问权限控制等，比如只有交易员有权限提交和修改交易记录，物理上要求对存储介质有备份措施等；公司必须明确应当如何移动和存储数据，授权的人员应当如何以及在何时访问并修改数据，以及是否应当在一定的时期之后销毁数据。公司策略还必须确保未授权的人员无法对数据进行不正当的访问、更改或者删除数据。
- 可获取性要求任何对数据的合理的访问都应该能够快速响应，以满足业务的效率要求和相关部门的审计要求。比如审计师能够在可以容忍的响应时间内查询到需要审计的交易记录；对电子信件、数据、文件的察看有响应时间上的要求，是要避免被审计机构如公司利用时间拖延而伪造相关记录，时间拖延越长，造假的可疑性就越高。
- 由于电子记录非常容易修改，删除并且不留下痕迹，所以必须要求历史记录以不可覆盖，不可擦除的方式存储，对记录的修改只能以添加新记录的方式进行，以完全的反映该记录从创建开始的所有变化情况，只有当该记录过期之后，才可以删除或者备份。
- 时效性要求文档的保存都有规定的时间，而不同产业的电子数据也各有不同年限要求。比如萨班斯法案，要求会计等相关资料要留存 4 年以上；比如医疗记录必须保存 21 年，HIPAA 要求保存 30 年；甚至美国证券交易委员会的第 17a-4 条法案（SEC 17a-4）更要求资料要保存到该业者结束营运为止。

在美国推出的众多关于电子记录管理的法案中，上述要求或多或少的贯穿于全部管理法规要求，是电子记录管理的趋势所在。这些法案的颁布也促使公司采取相应的技术措施来保证“法规遵从性”，以便在突如其来的配合调查中能够从容因应，否则将会受到法律的处罚。

2002 年 11 月左右，美国政府对华尔街（Wall Street）5 家知名的金融公司予以罚款，包括高盛证券（GS）、摩根史坦利（MWD）、花旗集团（C）的投资银行部门：U.S. Bancorp.（USB）所属的 U.S. Bancorp Piper Jaffray、以及德意志银行（DB）的证券部门，总计罚款达 830 万美元，受罚原因就是：未依联邦主管部门的要求将电子信件进行留存。这就是违反了可信数据库的电子记录存储的完整性。美国对证券业者业务相关的电子文件保存规范要求 2 年内的数据要能立即被察看，而即便 2 年过后也依然要再保存 1 年，但这 1 年的保存就不再硬性规定立即调阅，可以用其它方式储存。诸如此类的要求，多是为了日后配合法令调查，包括金融业者的客户可能涉及洗钱，或金融机构本身可能违约交易等，届时持续留存的电子数据、文件就成为记录的证据。

为了提供可信的电子数据，可信数据库采用的存储介质要求考虑安全性、数据完整性、总拥有成本、性能、访问能力以及搜索功能。一个设计良好的系统也将具有灾难恢复功能，除非发生重大的事故，否则记录将不能被更改，不会发生任何安全问题，重要数据也不会丢失。

为了满足上述的要求和特性，同时也要减轻企业的负担。可信数据库将采取什么样的技

术呢？下面将分析一些已有的技术。

2.2. 技术实现

可信的数据存储要求：数据从创建到使用过程中一直保持可信。对于存储在一个数据库系统内的数据，如果是可信的，那么给定任何一组特定的输入，其输出一定是期望的结果。这要求数据库系统在任何时刻都具有正确的计算和推理能力，以及可靠的存储能力（存储的任何数据都可以被检索到，并且，检索到的任何数据都是被合法存储的）。也就是说，系统本身不存在任何错误的和欺诈的行为，同时也能够阻止任何恶意的入侵。（这些“错误的”、“欺诈的”、“恶意的”等的定义是针对具体的应用而言的）

为了实现这一要求，我们规定对于电子数据的存储满足以下要求：

- 为所有已经发生的事件创建正确的数据记录
- 没有虚假数据记录并不发生的事件
- 已存储的记录是规定的保留期内不可修改的（只能添加新的记录，不能对历史记录进行修改和覆盖），过期之后可以删除
- 有获取控制管理
- 在数据创建过程中，周期性的审计创建操作

为了满足上述要求，提供可信的数据，当前最普遍使用的存储技术是 WORM（write once read many）。如图 1 所示：它构建在普通的光盘、磁带、磁盘存储介质上，通过硬件设备的控制实现了数据只能一次写入，只能添加新的数据，不能修改历史数据的语义限制，从而为电子数据提供了最安全的保证。它提供了类似文件系统的接口，并被扩展以支持对文件和块的添加操作。我们做了一些合理的假定 1)任何用户不能从物理上来破坏 WORM 存储设备 2）记录的创建是正确的，WORM 存储设备的运转也是正常的 3）查询是正确的 4）任何用户不能干涉查询的过程，例如修改查询接口等。

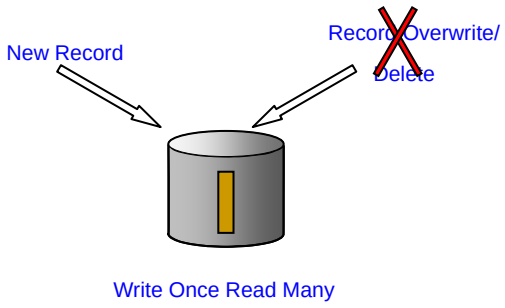


图 1：WORM 存储技术

在数据被创建到被使用的过程中，将面临怎样的威胁，我们将采用什么样的技术来保护数据不被攻击者破坏呢？我们将用一个示例来说明：

如图 1 所示：Alice 是一个合法的用户，她创建了一个记录 R，并提交到基于 WORM 存储的数据库里，在未来的某个时间，将有审计用户 Bob 提交查询并得到 R 作为结果。在 R 被创建到被查询的期间，有一个恶意的用户（Adversary）Mala，她不想让 Bob 得到 R 这个记录，所以会采取一些恶意的措施来掩盖 R 的存在。

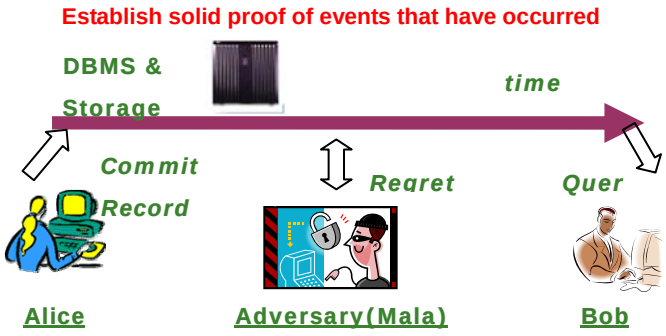


图 2：威胁模型 1------攻击者具有普通权限

如果 Mala 是一个普通的用户，那么她不能获得修改记录的权利，普通的获取控制机制和 WORM 存储特性就可以保证数据的安全。

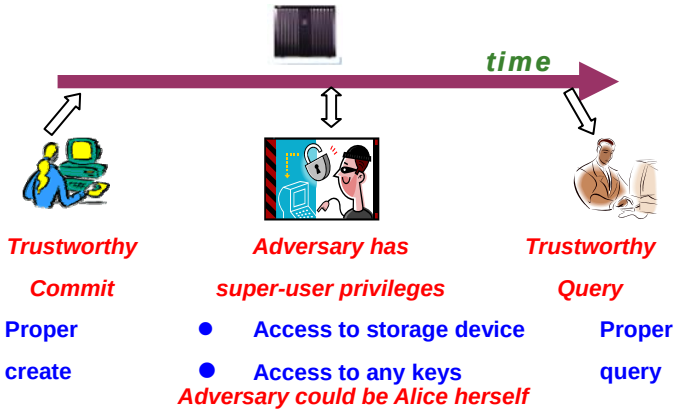


图 3：威胁模型 2------攻击者具有超级权限

如果 Mala 是一个高级用户，情况就有所不同了，这在实际生活中也是常见的。公司的高层（CEO，CFO）或者高级技术人员等等往往具有超级用户的权限，能够进行所有普通合法用户的操作，例如往数据库里面写任何数据，读任何数据。如图 3 所示：在这样的情况下，普通数据库系统的获取控制机制就不起作用了，因为超级用户的权限是最大的。那么，简单的 WORM 技术还能保证数据的可信吗？不能，因为随着数据量的不断增大，数据库的查询主要是通过索引来执行的，所以虽然记录的提交和记录的查询都是可信的，但具有高级权限的用户（比如公司高层管理人员，技术专家）可以通过恶意篡改索引来从逻辑上隐藏某些数据，从而导致查询结果失真。如图 4 所示：

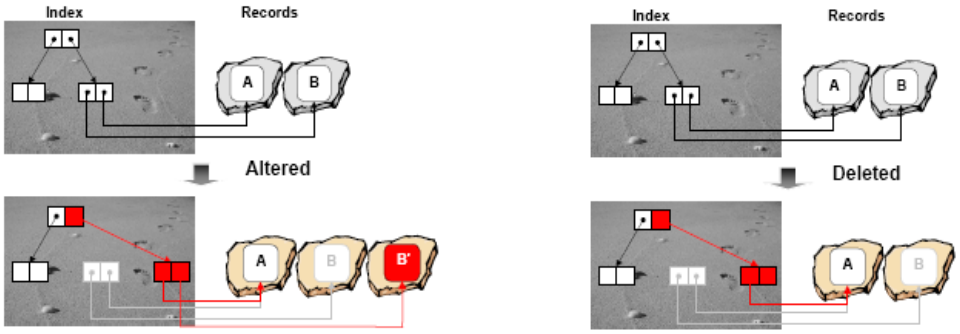


图 4：允许逻辑上修改数据的不可信索引

在沙地上记录的是不可信索引，表示索引很容易被修改，在石头上记录的是真实的记录，表示不能被修改。虽然记录 A，B 都存储在 WORM 存储设备里，但我们在查询的时候是通过索引来访问真正的数据。该索引是不可信的，它允许逻辑上的修改。因此通过修改索引指针，可以轻易的将 B 替换为 B’或者将 B 隐藏，这样就达到了从逻辑上来隐藏记录 B 的作用。在这种情况下，我们面临的威胁模型变为如图 5 所示：

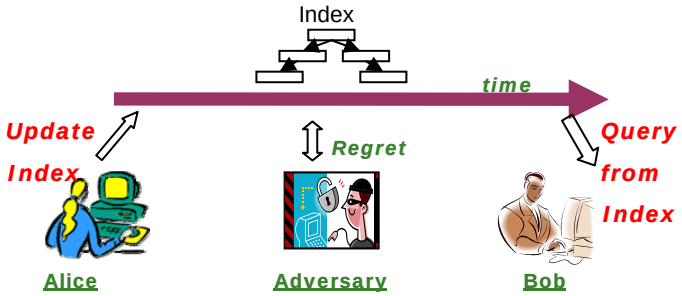


图 5：威胁模型 3------攻击者修改索引

记录的创建者提交记录的同时也更新了索引，记录的访问者通过索引来查询记录。攻击者由于是超级用户，所以可以从逻辑上来直接修改索引。该不可信的索引可能是存储在 WORM 里，也可能是存储在普通的存储介质里，但攻击者均可以通过添加新的路径来掩盖记录。

为了防止上述的攻击，我们提出了对可信索引的要求：

- 正确性：一旦记录创建，对该记录的索引项和路径都是不可更改的。更新索引的代码不会导致历史记录的隐藏和修改。这意味着一旦记录被创建，则相应的索引更新也被提交到 WORM，除非 WORM 发生物理上的问题，否则该记录一定是通过创建时的索引来访问的。也就是说，记录的获取只依靠 WORM 存储设备的不可重写性。
- 持久耐用性：索引能支持记录的快速增长，对新记录即时更新，不需要周期性的删除和重建索引。如果没有即时更新，把记录放在缓存中，会面临该记录被修改的威胁。
- 索引必须支持高效的查询。
- 索引引起的空间增长必须是可接受的。
- 索引是不可分解的，已经过期的删除的数据不能通过索引被推测出来。

同时，对查询也要求是可信任的：即是正确的查询

3. 可信的数据隐私保护

3.1. 背景介绍

随着信息技术的发展，网络的连通性和磁盘存储空间的增大，整个社会正在经历一场数字化的革命，各种各样的电子数据不断产生，人们纷纷热衷于用数据来表达自己的意愿，来描述复杂的事物。对这场数字革命的到来欢欣鼓舞的人们，只沉醉于享受新事物的愉悦，毫无戒心的甚至争先恐后的将自己的个人信息公之于众，不管是填写各种注册信息还是使用各种诱人的服务，完全没有意识到自己已经毫无隐私可言。只有当清晨开机时那烦人的垃圾短信，以及邮箱里新增加的莫名其妙的垃圾邮件，才会让你好好反省是否应该保护自己的隐私！在使用各种服务时，一个不可避免的事实是只有提供更多的个人信息，才能获得更好的服务。

例如随着个人手持设备（PDA，Smart Phone 等）的普及，人们越来越多的使用基于位

置的服务（Location Based Service：LBS），包括紧急救援服务，基于位置的游戏，移动黄页服务等。虽然服务提供商不要求人们在请求服务的同时发送自己的唯一标志例如姓名，网络地址等，但要求用户发送自己的当前位置，只有个人位置信息越精确，获得的服务才越满意。在这种情况下，用户的位置就成为了个人隐私信息。服务商（攻击者）可以通过把用户位置和地图进行匹配以及某些经验观察来发现用户的真实身份，进而对用户的服务请求进行分析，发现用户的个人爱好等隐私。

另一方面，政府机构以及公共服务机构越来越多的发布包含个人信息的数据，比如医疗数据，选民数据等等，这些数据甚至可以作价出售。如果没有可信的隐私保护，那么攻击者将利用多个数据之间的联系来获得个人隐私信息。如图 6 所示，左图的医疗信息是从专门为政府雇员购买医疗保险的机构购买的，选民信息是从负责选举的机构处购买的。该医疗信息可以认为是匿名的，因为没有病人的姓名等唯一标志信息。但如右图所示，当攻击者把医疗信息和选民信息结合之后，通过出生日期，邮编，性别的匹配，就可以把选民姓名和疾病联系起来，从而获得了非常隐私的个人信息。

Hospital Patient Data

Birthdate	Sex	Zipcode	Disease
1/21/76	Male	53715	Flu
4/13/86	Female	53715	Hepatitis
2/28/76	Male	53703	Brochitis
1/21/76	Male	53703	Broken Arm
4/13/86	Female	53706	Sprained Ankle
2/28/76	Female	53706	Hang Nail

Voter Registration Data

Name	Birthdate	Sex	Zipcode
Andre	1/21/76	Male	53715
Beth	1/10/81	Female	55410
Carol	10/1/44	Female	90210
Dan	2/21/84	Male	02174
Ellen	4/19/72	Female	02237

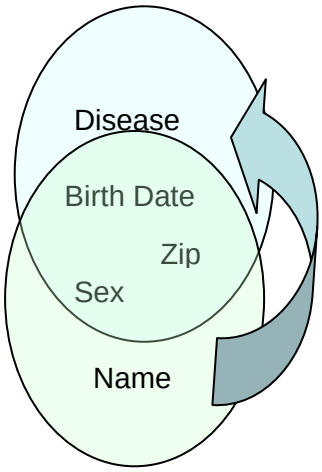


图 6：通过数据之间的匹配来识别隐私

3.2. 技术实现

3.2.1. 关系数据库环境

在不同的环境下，可信的数据隐私保护有不同的要求。公共发布数据环境下，包括公共医疗调查报告，选民数据发布等，要求首先必须隐藏能够惟一标志用户的个人信息，比如显式的名字。另外，还需要使某一个特定个人的信息不能从所有数据中被攻击者识别出来，比如生日，性别，邮编，电话号码等属性很容易被匹配到个人。当前普遍采用的一个方法是 k 匿名模型：一个关系满足 k 匿名，如果其中每一个元组所代表的个人信息都至少和关系中其他的 k-1 个元组不能区别。如图 7 所示：该关系中 Problem（疾病）是个人的隐私。在 Race, Birth, Gender, ZIP 属性上，每一个元组都至少包含了一个并发（相同的属性），所以攻击者不能识别出某一个特定个人的疾病信息。

Race	Birth	Gender	ZIP	Problem
Black	1965	m	0214*	short breath
Black	1965	m	0214*	chest pain
Black	1965	f	0213*	hypertension
Black	1965	f	0213*	hypertension
Black	1964	f	0213*	obesity
Black	1964	f	0213*	chest pain
White	1964	m	0213*	chest pain
White	1964	m	0213*	obesity
White	1964	m	0213*	short breath
White	1967	m	0213*	chest pain
White	1967	m	0213*	chest pain

图 7：满足 k 匿名的关系

3.2.2. LBS 环境

在 LBS 环境下，用户发送的位置越精确，获得的服务越好，但精确的位置更容易泄漏用户的隐私。为了保护用户的位置隐私，通常采用的方法是对用户的真实位置采用 cloaking 技术，使得用户的真实位置点（location point）被扩大为一个区域（cloaking region），服务提供商只能接收到该位置区域而不能识别用户的真实位置。使用广泛的是两种服务模型。

第一是采用匿名代理的三方模型，如图 8 所示：

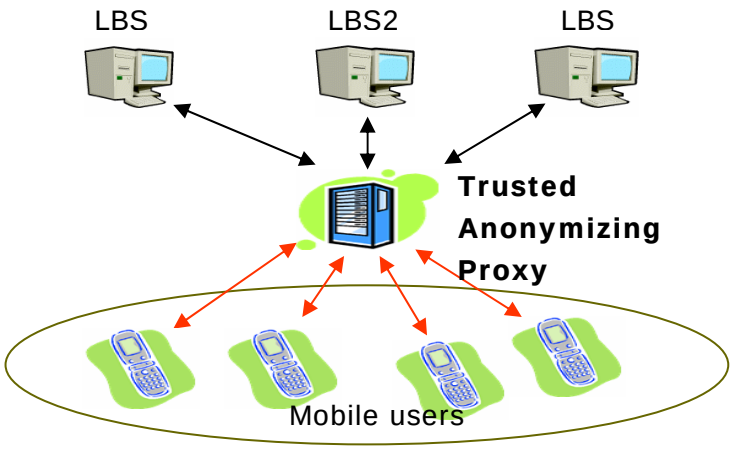


图 8：可信隐私保护的 LBS 模型

该模型由三类对象组成：移动对象，可信的匿名代理，服务提供商。移动对象提出基于位置的服务请求，匿名代理对用户的位置进行匿名处理以保护用户的隐私，服务提供商提供各种不同的服务。在模型中信息的交互过程是：1）移动对象向可信的第三方匿名代理提出服务请求，其中包含自己的服务内容和真实的位置信息（location point），还有一些关于隐私的要求。该请求通过安全的传输（签名机制等）到达匿名代理。2）可信的匿名代理使用 cloaking 技术对用户的位置进行扩大，再把扩大后的位置区域和服务请求发送给服务提供商。3）服务提供商响应服务请求，并把查询结果返回给匿名代理。4）匿名代理对结果进行求精，选出最适合用户真实位置的结果返回给用户。

在该处理过程中，涉及到的关键技术是：如果进行位置的匿名处理；如何响应服务请求，

处理位置相关的查询；如何对结果进行求精。

匿名处理通常采用的方法也是 k 匿名模型。LBS 环境下的 k 匿名模型主要是针对位置信息的。如果某个用户的位置不能和其他 k-1 个用户的位置相区别，则该用户的位置满足 k 匿名。匿名代理在对用户的位置进行 cloaking 时，主要的问题就是怎么样找到一个合适的 cloaking region 使得它能够同时覆盖住 k 个用户的真实位置，并且还要满足用户的隐私要求（比如响应时间，匿名质量（cloaking region 的最小最大范围限制）等）。通常有两种找到 cloaking region 的方法。第一是静态的基于空间的划分方法。整个用户空间被预先划分为等大小的基本单元，用户的 cloaking region 为从用户所在的基本单元开始扩展而找到的最小的能够覆盖用户的并且满足隐私要求的区域。代表方法有基于 Quad-Tree 和基于 Grid。第二是动态的基于用户位置的划分方法。用户的 cloaking region 是从用户位置开始扩展，找到离自己最近的并且满足隐私要求的 k-1 个邻居之后所组成的区域。代表方法有基于 Graph 的。

处理位置相关的查询主要是处理 cloaking region 的查询，比如范围查询（range query），最近邻查询（knn query），采用方法主要是基于移动对象环境下的查询方法，代表方法有。。。

第二是用户-服务器结构的两方模型，如图 9 所示：

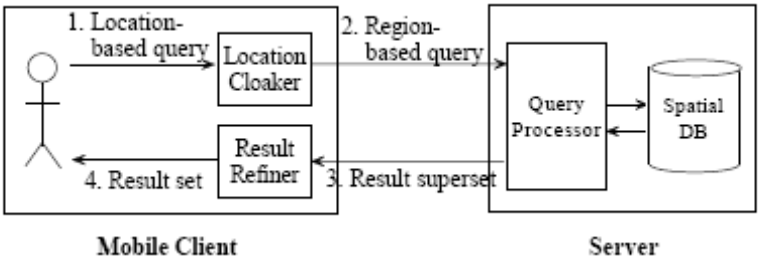


图 9：用户-服务器结构的 LBS 模型

该模型由两类对象组成：移动对象和服务器。移动对象能够使用 GPS 等设备来获取自己的位置并具有匿名处理及结果处理的能力，它和服务器直接进行通讯。在模型中信息的交互过程是：1）location cloaker 接受移动对象的服务请求，其中包含服务内容和真实的位置信息（location point），还有一些关于隐私的要求。2）location cloaker 使用 cloaking 技术对用户的位置进行扩大，再把扩大后的位置区域和服务请求发送给服务器。3）服务器中的 query processor 响应服务请求，并把查询结果返回给用户的 result refiner。4）result refiner 对结果进行求精，并选出最适合用户真实位置的结果。

在该处理过程中的主要问题和上面的模型相似。

其他很多环境下，也需要考虑可信的隐私保护。比如 sensor 环境下，RFID 环境下等。

4. 可信的数据存储和可信的数据隐私保护的结合

可信的数据存储和可信的隐私保护是两种不同的契约规定，它们没有层次高低的区别，也没有相互依赖的关系，只是不同的应用要求，可以同时存在，也可以单独实现。只是不同的情况下对数据管理系统的实现提出不同的要求。类似安然案件这样的情况，公司的自身的运营情况记录等需要完全真实的提供给国家，审计者，投资者，调查者和所有公民个人以利于决策的，我们要求数据的存储是可信的，这里不要求隐私保护。

类似医院的病例记录，保险公司受理的保险记录等这些涉及到公众的私人信息的，同时也要完全真实的提供给国家的，我们要求数据的存储是可信赖的，但是对外发布的时候还要求可信的隐私保护（也就是发布公共信息的隐私问题）

类似用户请求服务时，用户的真实信息必须有一个第三方来存储，但这里的存储不一定要求可信，或者可信的要求更弱。由第三方再向外发布的时候要求可信的隐私保护。

5. 可信数据操作：外包数据库 —— 新的展望

对于简单的应用，也许仅仅有可靠的数据存储，有可靠的数据隐私保护，就能满足我们的需要。对于复杂的应用场景，我们往往还需要服务商能够提供针对这些数据的访问服务。一个提供数据库外包的服务商需要能够提供相关的查询，更新，访问控制等的操作。

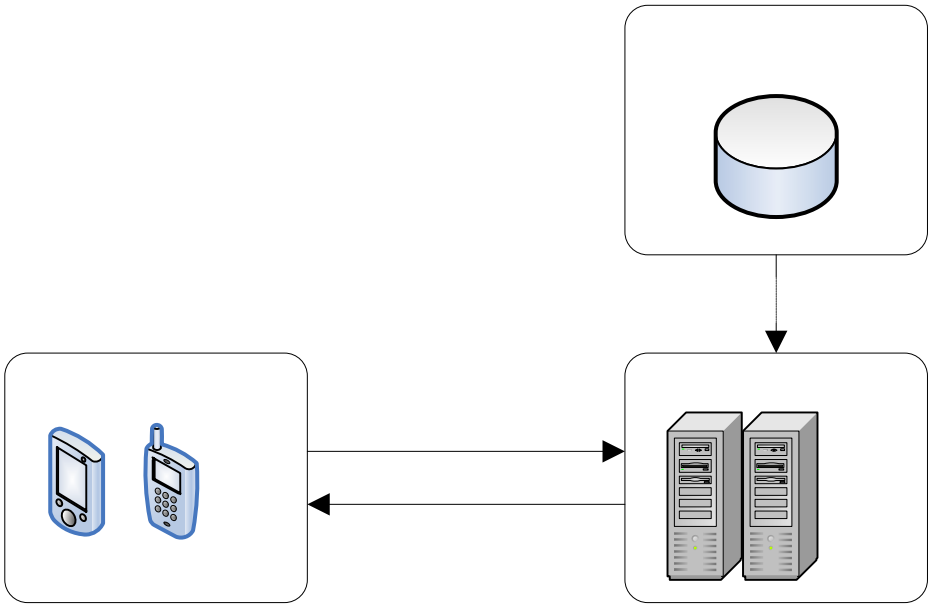
数据库外包会跟很多小的企业应用带来很多好处：可以避免维护数据库软件，硬件；可以节省人员 DBA 的开销等等。但是，数据库外包也会带来很多的挑战，例如：数据从远程访问，网络访问延时的影响；数据的安全性、隐私的考虑；数据操作的安全性考虑。

现今网络技术正在以惊人的速度发展，千兆网，万兆网的出现都是提供远程数据库服务成为了可能。对于数据的安全性，我们也可以通过加密数据来使得我们的数据不会被一个恶意的服务提供商获得。

但是还有一个非常重要的，非常有挑战性的问题在于我们如何保证在这些加密后数据上操作的可信性，换句话说，我们如何来保证我们的查询、更新操作真正得到了正确的、完全的执行？

我们说这个问题需要一分为二来看：首先，由于数据经过加密，在这些加密后的数据上的操作必然会受到限制，这是挑战之一；其次，假设我们在加密后的数据上能够执行查询，我们怎么去验证一个数据库服务提供商正确的执行了我们的查询、更新操作？我们将在下面的讨论中考虑上面的问题。

在讨论具体的问题之前，我们先来看看一个比较通用的数据库外包场景：



可以看到，在这个典型的数据外包场景中，一个数据拥有者（Database Owner）将所有加密后安全的数据存放到一个数据库外包服务提供商（Service Provider），所有的用户（User Device）通过一个加密后的查询到服务提供商查询需要的数据。

在这样一个典型的数据外包场景中，一个值得注意的问题是我们的用户设备往往是一些存储和计算能力都有限的小型的手持设备，那么我们如何在这种受限的情况如何去验证一个不可信的服务提供商是否正确完整地执行我们的查询、更新操作，是否可信就是一个具有相当挑战性的任务。

下面我们将分两个部分来讨论，首先是如何在一个加密后的数据源上执行查询，然后讨论如何验证一个数据库外包提供商是否可信。

5.1. 在加密后的数据上执行查询

对于一些比较敏感的数据，为了避免一个恶意的服务提供商去利用数据去获取利益，我们往往需要将其加密，但是在加密后的数据上往往难于执行灵活的查询，因为，对于一个非等值查询来说，由于传统的加密算法不能保证加密后的数据和加密前的数据由相同的顺序，所以只能支持等值查询。

一部分研究工作中指出，我们可以通过把每一个准确值抽象成一个区间，然后通过在服务提供商存储这个抽象的区间而不存放原来的每个精确值来实现数据的加密，同时也可以保证我们能够执行所有的查询。但是这种方法由于丢失到数据的准确值返回的结果往往是一个结果的超集，需要用户对返回的结果进行过滤，这样会带来额外的代价。同时这种方法也面临着一个不可调和的问题就是，为了减小用户的代价，我们需要抽象的区间尽可能精确，但是抽象的区间尽可能精确又会造成安全性的降低，所以这种方法有其本身的缺陷。

另一部分最新的工作通过将原来的数据映射到一组新的数据上来达到加密的目的，这种方法维护数据项在加密后加密前数据中的顺序性来保证可查询性，这种方法没有之前方法存在的种种问题，而且查询返回的是一组准确的结果，所有较之之前的方法有较大的优势。

5.2. 查询的正确性，完全性检查

在数据库外包中，判断一个服务提供商是否可信，两个最为重要的条件就是：用户提交的查询返回的结果是否正确？用户提交的查询返回的结果是否完全？

正确性：

查询返回的结果是否正确是指，服务提供商返回的所有元组是否是原来数据库中的元组，服务提供商有没有修改我们的元组或者有没有返回一个恶意生成的元组。

如果一个服务提供商不能够满足查询的正确性，我们说这个服务提供商一定不是可信的。

因此，在之前的工作中，有很多的工作对这个问题进行了讨论，提供了各种加密，签名的方法来验证一个服务提供商是否满足查询正确性。

完全性：

一个服务提供商如果能够满足查询正确性，我们依然不能判定这个服务提供商是可信的，因为一个恶意的服务提供商还有可能删除部分数据，或者只返回原来结果的一部分。所以除了验证一个服务提供商的查询正确性之外，我们还需要知道一个服务提供商的查询完全性。

所谓的查询完全性是指，对于给定的一个查询，服务提供商能够给我们返回所有的查询结果，没有遗失任何一个正确的查询结果。

查询完全性的验证是一个非常重要的，非常具有挑战性的课题。之前的所有的工作往往都只能支持非常有限的查询类型，而且对数据由很强的假设，需要建立各种索引数据结构来完成验证的过程，所以非常具有局限性。

所以，能够提出一种开销小的，能够支持各种查询的完全性验证方法非常的有意义。

6. 研究点归纳

（1）继续分析“可信的数据库”在不同行业的应用需求及不同的处理方式，并区别它和传统概念如安全性、保密性。

（2）根据可信的本质特征及（法规）要求，提出相应的实现技术或解决方案。可信的数据存储从技术角度来看，既可以通过数据存储在数据库之后的内部实现机制（如存储方式、索引结构等）来从根本上对法律规范进行支持和遵从，又可以借助外部监测技术来监督和验证数据库系统的可信性。此外，还可以提出与技术相结合的一整套解决方案，如使用哪些（已成熟）的技术经过何种步骤来最终保证数据库中数据的可信性。

（3）继续研究可信的数据隐私保护和可信的数据外包服务。

Flash-based DBMS

尹少宜 （Mobile 组）

1. 引言

1956 年，第一块硬盘诞生。从此，人类存储数据的方式被改写。那些被堆积成山的企业报表所淹没的人们看到了希望，但是，当他们兴高采烈地把报表“数字化”之后突然又眉头紧锁——如何才能迅速地从数字报表中查找记录？如何在报表之间建立关联？如何让计算机利用已有数据生成新的报表？是最初的数据库人帮他们解答了这些问题，一方面将企业数据抽象为关系模型，一方面又利用磁盘的性质给出关系数据库的实现方案。企业管理人员终于如愿以偿地使用了硬盘来管理自己的数据，这也便成就了 Oracle、DB2、SQL Server 等今日的辉煌。

1969 年，互联网诞生。从此，人类传输和共享数据的方式发生了根本性变化。广大拥有 PC 机的人们欢欣鼓舞，然而，当他们放肆地把自己的数据共享又从网上获取到大量信息之后开始迷茫——在浩瀚的网页数据中，如何才能在瞬间得到自己最想要的信息？又如何才能让自己发布的信息在任何时候都能被需要的人看到？聪明的数据库工作者知道传统的关系数据库不能解决这个问题，人们需要的只是一个面向网页的搜索引擎，于是他们根据搜索目的和网页特征，用新的方法把数据存储在磁盘上。人们开始从漫无目的的网上冲浪转变为一针见血的网络搜索，Google 的神话广为传唱。

从数据库的辉煌到 Google 的神话，在瞬间颠覆了人们的思考方式，而未曾改变的是，磁盘一直都被用作大量数据的存储介质。然而，人类的脚步没有停止，人们采集和使用数据的方式在日复一日地发生着变化：人们不满足于在固定的位置使用 PC 机，于是发明了便携式的计算设备；人们不愿意使用陈旧的人工方式来控制工业机器，于是发明了嵌入式控制器……即便如此，人们仍不满足于只有这样的计算和控制，而是希望随时随地将收集的数据在瞬间保存、将需要的数据在瞬间获取，于是便发明了便携式的存储设备和嵌入式的存储器。这时，磁盘再也经不起考验了——它的机械特性暴露无遗：体积太大、读写周期太长、工作温度范围太小、噪声太大、抗震性太差、耗电量太高等等，都使其无法胜任移动和高温等环境下的数据存储任务。于是，1988 年，一种叫做 Flash 的永久性存储介质诞生了，它读写速度快、抗震能力强、体积小、存储密度大、噪声小、耐高温，因此其应用开始广泛而迅速地渗透到磁盘所不能及的各个领域。

Flash 的种类有很多，而目前最适合用作大量数据存储介质的当属 NAND 型 Flash。喜欢随时随地管理数据的人们当然也会喜欢大容量的移动存储介质，然而他们到底要管理什么样的数据？对数据要进行什么样的操作？关系数据库是否仍能满足其需求？而这些数据应该如何在

Flash 上存储？基于磁盘的数据库实现技术在 Flash 上是否还能行得通？问题又回到了最初——什么数据，怎样存储，如何查询？好奇的数据库人又开始寻找新的答案。

2. NAND Flash 介绍

和磁盘类似，NAND Flash 读写数据的基本粒度为页（page），然而读、写数据所需的时间却不一样，一般写入时间是读出的 3 到 10 倍。此外，Flash 不允许数据的直接覆盖写，必须首先擦除旧有数据才能写入新的，而擦除的粒度为块（block），通常一个块包含有 64 个页。擦除的速度是很慢的，而且每一块的擦除次数也有限，平均为 100 万次。下面将给出一个例子来具体说明 NAND Flash 的特点。图 1 是三星 K9K8G08U0M 芯片的 Flash 组织结构图。

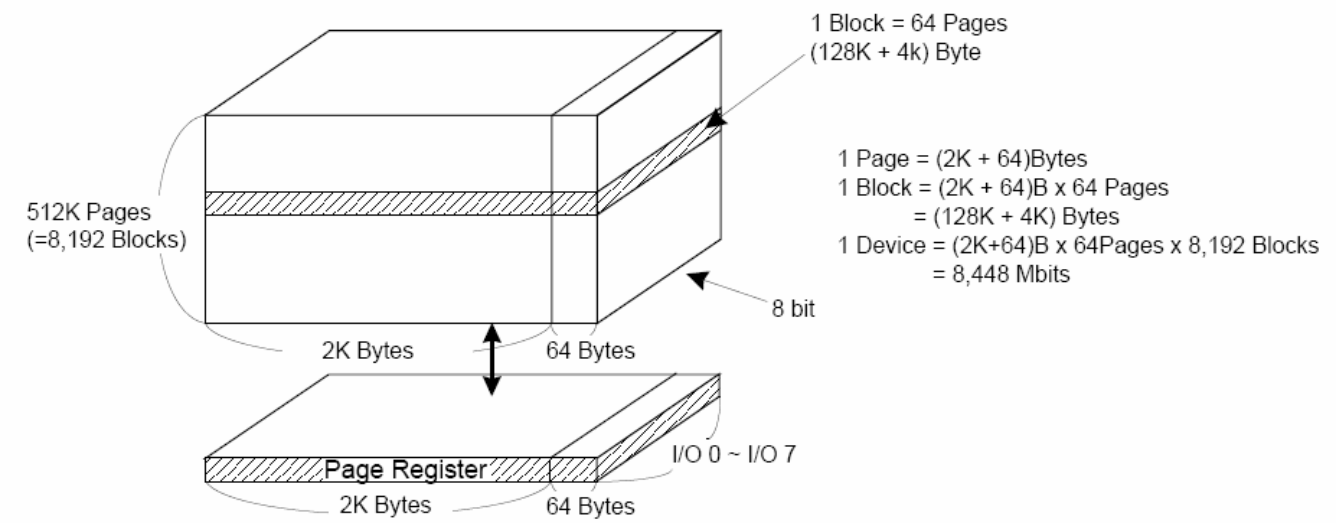


图 1：NAND Flash 组织结构图

从图中可以看出，整个 Flash 的容量为 8448Mbit，其中含有 8192 个物理块，每块包含 64 个页，每页除了 2K 的数据区之外，还有 64 字节的额外空间用于存放错误校验码等信息。除了 Flash 的主体，我们还可以看到一个一页大小的数据寄存器。事实上，每一次 Flash 的读写操作都需要使用这个寄存器来完成。例如，读操作实际上分成两个阶段，首先是把指定地址中的整页数据载入到寄存器中（时间为 20 微秒），然后再从寄存器输出数据，可以连续输出，也可以根据指定偏移量随机输出，且随机输出的次数不限（每个字节的输出周期为 20 纳秒）；而写操作也分为两个阶段，第一阶段是把数据从外界输入到寄存器，可以整页输入，也可以随机输入，且随机输入的次数不限，每字节的输入周期也是 20 纳秒，第二阶段便是把寄存器中的数据固化到 Flash 上，时间为 200 微秒，每一页原则上只能有一次固化操作，然而大部分 NAND Flash（如本例）允许在同一页中的几个片断按照先后顺序分几次写入，本例中为 4 次，这种情况叫做 partial page programming。

除了 partial page programming，NAND Flash 还有另外一种有意思的操作，也和数据寄存器相关，叫做 copy-back。就是说，当某个数据页需要被复制到新的位置时，只需将其载入到数据寄存器中，然后根据接收到的目标地址将该页数据固化到 Flash 上新的位置，这样就省去了整页数据在寄存器和 RAM 之间的输入输出过程。另外，在数据被载入到寄存器之后，Flash 允许程序使用随机输入数据的方式改写该页的部分内容，然后再固化到目标位置。

3. 基于 NAND Flash 的数据库系统研究

在引言中，我们曾提到 Flash 的应用环境广泛，而不同应用环境下的数据可能有不同的特征，因此对数据库功能的要求也各不相同。事实上，普适计算环境下的数据建模本身也是一个值得研究的问题，而这里，我们将重点针对某一种模型下的数据库来研究其在 Flash 上的实现问题。关系数据库模型不可能在任何环境下都适用，然而能够使用关系数据库的例子却比比皆是，例如公司职员将重要的客户和产品资料保存在自己的 PDA 上，医生和病人将病历资料存放在带有智能卡的 USB key 上等等。因此，我们不妨把关系数据库在 NAND Flash 上的实现作为研究的第一步。而这其中，值得研究的问题又有很多，主要包括：

3.1 Flash-based DBMS 不同于磁盘数据库的评价指标

由于 I/O 瓶颈是用磁盘管理数据的主要难题，I/O 次数越少，数据库的性能就越好，因此传统数据库实现的目标也十分明确，就是尽可能减少 I/O 代价。存储、索引、查询等的设计和实现目标也都是最小化 I/O 次数。然而，flash 与磁盘具有完全不同的性质，例如读、写速度不等，但都快于磁盘；重写前要擦除；擦除次数有限等。针对这些性质，用单纯的 I/O 次数来评价系统是否仍然合理？至少，既然输入（I）和输出（O）的代价不等，在新的指标评测中就应该考虑区别对待了。此外，Flash 大多用于资源受限的环境中，那么内存使用情况、电源消耗情况等是否也应该作为更加重要的指标来考虑，从而影响整个系统的设计？我们的研究就是要将这些指标分析得更加透彻，归纳出哪些指标是由 Flash 本身决定的，具有广泛约束力，而哪些指标是由特殊硬件环境或者软件应用决定的，分析这些指标间的相互关系，并最终给出在不同指标限制下的不同实现方案。需要解决的问题可概括如下：

- 系统整体性能与 Flash 读、写、擦等的次数之间有怎样的定量关系？
- 系统电源消耗与读、写、擦次数之间有何定量关系？
- 通常可以通过增加内存来提高系统性能，或者为了节省内存而降低系统性能，那么内存占用和性能优化二者之间如何权衡？是否可以将系统目标设定为：在任何给定大小的内存限制下，系统都能够最大化利用资源从而使得性能最优？也就是说，系统实现不依赖于内存大小，同时又能够最大化地利用已有内存。

3.2 Flash-based DBMS 的存储和索引模型（以及相应的垃圾回收方案）

从对研究现状的分析中可以看出，使用传统的以“块”为单位的存储方式会带来 Flash 的大量浪费和系统性能的下降。而使用日志式追加的存储方式又会使得数据记录变得无序，降低查找效率。怎样有效地组织 Flash 上的数据使得 Flash 资源在数据库应用环境下得到最合理的利用是一个十分有意义的研究问题。当然，这个问题本身也包括了索引结构的设计。经过深入分析，我们发现传统的索引结构直接用于 Flash 之上存在非常严重的问题，并将这些问题进行了归纳总结，发现了 Flash 上索引结构设计的瓶颈所在，然后正在试图寻找新的方案来解决这些问题。由于 Flash 在重写前需要擦除这一特性，无论使用何种存储和索引方案，都必须考虑垃圾回收和损耗平衡的问题，而如何解决这些问题也将对系统最终的性能产生巨大的影响。也就是说，使用不同的垃圾回收策略可能导致系统性能的巨大差异。上述问题可总结为：

- 在 Flash 上如何组织记录和索引？具体来说，就是数据库一个表中原始记录和索引条目分别以何种方式存储在 Flash 上？不同的表之间的数据存储位置又有什么关系？同一个表的不同索引结构之间的存储有何关联？
- 上述结构如何维护？也就是在有数据库插入、删除和更新操作之后，怎样最小化对 Flash 的更新？我们现在的方案是将更新分解为插入和删除，因此需要建立一个删除列表，那么这个删除列表又应该如何维护？
- 怎样设计缓冲策略，在尽量少使用内存的情况下最小化索引维护的代价？
- 无论如何存储和维护，垃圾回收都是不可避免的。那么采用什么策略才能顺应系统的总体目标，同时又不破坏损耗平衡？

3.3 Flash-based DBMS 的查询处理和优化

在任何一个数据库管理系统中，查询处理和优化都是系统实现中的关键。考虑到 Flash 的特殊应用环境，查询处理所能使用的资源可能受到巨大限制；考虑到 Flash 的读写特性，查询优化的目标和内容也会有所变化；考虑新的存储和索引方案，查询处理和优化的实现策略必然要以此为基础。查询处理和优化的算法应满足以下约束：

- 不应依赖于 RAM，但又能充分利用可用的 RAM。也就是说，当 RAM 资源紧张时，系统能够正常运行，但是 RAM 资源充足时，系统又可以利用这些资源来提高效率；
- 合理利用已有的存储和索引结构。最大化发挥这些数据结构的优势，以充分利用 Flash 的特性（如读写速度不等）。

3.4 Flash-based DBMS 的事务和恢复

在数据库管理系统中，要保证数据的一致性和持久性，事务处理是必不可少的。故障之后的恢复通常是利用日志记录来重做或撤销事务。然而，在 Flash 上以何种方式来管理日志记录也是值得考虑的问题。可研究的问题包括：

- 首先，由于 Flash 数据不可覆盖的特性，数据本身往往就包含了故障恢复所需信息，日志的内容可以适当缩减；
- 其次，检查点的设置也不应该是随意的，而是根据存储和索引模型针对 Flash 空间利用情况作适当优化。

3.5 其它（安全、并发等）

在上述研究基础上，加入数据库的安全访问控制和数据加密算法，并考虑支持并发操作。尤其是在并发操作中，各种锁的实现需要充分考虑数据的存储和索引结构，利用 Flash 特性来最大化并发度。

四、结束语

当数据库邂逅 Flash，不变的是什么，改变的又是什么？在人类需求的加速膨胀中，基于 Flash 的数据库到底会扮演怎样的角色？对人们的生活和工作方式将会产生怎样的影响？

我们有太多的资料要搜集，我们有太多的问题要思考，我们有太多的推断要证明。

我们，一直在努力。

参考文献：

[1] M-Systems white paper. Two Technologies Compared: NAND vs. NOR. July, 2003.

[2] Samsung data sheet. 1G x 8 Bit / 2G x 8 Bit / 4G x 8 Bit NAND Flash Memory. Nov, 2005

[3] E. Gal and S. Toledo. Algorithms and data structures for flash memories. ACM Computing Surveys, 37(2):138–163, 2005.

[4] H. Garcia-Molina, J. Ullman, J. Widom. Database System Implementation. New Jersey: Prentice Hall, 2000.

PIM：一个新的研究焦点

李玉坤　　（Web 组）

摘要：科学技术的发展使个人信息量成倍增长，并成为影响个人生活秩序和生活质量的重要因素，于是产生了一个新的研究领域：PIM，即个人信息管理（Personal Information Management），本文对 PIM 历史发展、基本概念进行了系统介绍，分析了 PIM 的研究内容和关键的技术问题，特别对于个人信息的获取、存储、输出技术以及目前的研究情况进行了分析，对比分析了目前国内外的研究情况，对未来 PIM 研究进行了展望。

1．引言

科学技术的发展为我们提供了巨大的信息量，报刊、手机、电视、电脑、互联网、人与人的交流，甚至个人的思考，都使我们时时地接触信息，每个人都处在信息的包围之中，信息在人们生活中的作用越来越重要，同时也使个人信息处理面临越来越多的问题。相信很多人都经历过这样的场景：

偶然的机会遇到自己非常感兴趣，或对于自己非常有价值的信息，如在上网的时候偶然发现了一篇对自己非常有用的文章；在旅途中发现了一个市场需求信息；与人交流的时候，突然有了一个非常好的想法；等等，但是无法用有效的手段及时将其记忆下来；

自己积累的信息，往往随机记录在纸上，或保存在自己的计算机内，因为版本问题、存放位置问题、信息组织方法问题，往往不能随时随地访问自己要用到的信息，如某张照片，某个电话号码等；

记忆中确实将某张照片或某个电话号码保存到了自己的计算机或其他存储介质中，但是需要用到这个信息的时候，始终查询不到，或者查询成本非常高；

由于硬件损坏或丢失，如手机、笔记本电脑、硬盘等，造成自己重要信息的不可恢复性的丢失，给工作、生活带来严重影响；

忘记重要的日程安排，带来难以挽回的损失。

邮件系统受到容量限制、系统性能等各种问题的困扰，信息交流出现障碍，有时查询一封邮件往往成为很困难的事。

人们类似的经历还非常多，可以说，在这样一个充满信息的世界中，人们生活状态的好坏、工作效率的高低很大程度上依赖于信息处理的效率和及时性。特别是计算机技术、网络技术、web技术等的发展，为每个人提供了一个巨大的、共享的Web信息空间。使信息管理问题更加突出。据“网器”公司监测统计，2006年10月网站数量突破1亿，其中4700万或4800万家网站更是频繁更新网上信息[1]。Web网页每周增长率是8%,每年新内容的增长率是50%[2]。

除Web信息外，数据流、传感器、数字影像、数字电器、移动通信等技术的发展和应用，使我们每天所面临的信息更加丰富多样。如何将遇到的信息及时分析、保存；如何在需要的时候快速找到所需要的信息；如何在自己忘记的时候及时得到提醒；如何在信息管理中保护自己的隐私等等，这些问题变得越来越重要，处理的好坏直接影响到我们的生活质量和工作效率。如何解决这些问题，就引发产生了一个新的研究分支：个人信息管理（PIM）。

2． PIM 基本知识介绍

目前大家认同的最早提出PIM这一思想的是Vannevar Bush，他在 1945 年发表的文章“As

we may think”[3]中第一次提出了个人信息管理-Memex的概念，他这样描述：Memex是一种能够记录所有书籍、唱片、交流信息的设备，它能够快速、自动、灵活的帮助人们找到所需要的信息。Bush只是为我们描述了一种远景，随着信息科学技术的发展，人们试图从不同视角对PIM给出一个准确的定义，在 2005 举办的第一届PIM Workshop的报告中，对这一概念进行了总结和阐述[4]：

PIM 是我们日常对于信息的处理、分类、访问----Lansdale (1988) 。

- 为个人创建的供其在一个工作环境中使用的系统，其中包含人们获取信息的规则与方法；对信息进行组织与存储的机制，以及维持系统运行的一些规则与过程，以及对信息进行访问、处理、产生输出的方法机制。 ----Barreau (1995)
- 存储信息以使能够在以后被访问。 ----Boardman (2004)

由以上定义可以看出，PIM 的定义与信息技术的发展有密切关系，Lansdale 只是对 PIM 给出了一个宏观的描述；Barreau 指出 PIM 中应包含获取信息的规则、方法，以及存储信息的策略、机制；到 2004 年，Web 技术的成熟和存储技术的发展，使海量信息数据的存储成为可能， Boardman 认为 PIM 的核心是数据的存储(store)和再访问（finding/refinding）。这些关于 PIM 的描述，成为进一步研究、定义 PIM 的基础。

2.1 PIM 基本概念

表面看来，对 PIM 定义是非常简单的，因为我们每天都接触它、使用它，其实 PIM 是很难定义的，以至于一直是一个挑战性的问题。首先，对 PIM 研究领域的界定比较困难，必须合理界定 PIM 与其他研究领域的关系。其次，从字面来看，PIM 是一个包含主体、信息、工具的人机交互系统，涉及的概念很多。在 PIM 2005 workshop 上，与会专家对 PIM 的概念进行了讨论，对一些概念进行了阐述：

个人信息（Personal Information）：在 PIM 的研究中，我们聚焦于研究信息世界的一个信息子集，其中每个信息元素对于主体都有一定的影响能力。这样，就把 PIM 中的信息和我们平常的信息区分开来，这就是信息的相关性。或者称为信息的有用性，即 PIM 所研究的信息对于主体是有用的，这种有用性可能是现实的，也可以是潜在的。例如，我们到某地旅游，选择旅馆，关于旅馆的信息很多，如位置、价格、经理、员工数目、营业状况等，如果对主体做出选择产生影响的因素只有位置和价格，那么在 PIM 中关于旅馆的信息可以只包含旅馆的位置、价格。

因为主题需求是动态变化的，因此 PIM 的信息集合也是变化的，但具有相对稳定性。

在PIM研究中，**个人信息（PI）**包括以下三层涵义：

（1）个人保存并为自己所用的信息。

（2）和一个人有关但被其它实体控制的的信息，例如，医疗保险机构掌握着我们的健康信息。

（3）一个人经历过的但不为自己所控制的信息，例如我们访问过的网页。

信息项（Information Item）：信息项是与主体相关的信息集合的一个单元，也可以叫做信息包。在传统的以纸为介质的 PIM 中，一篇文章，一封信都可以看作信息项；现在的信息中包含大量的数字信息，一个信息项可以是一封电子邮件，保存下来的一个网页、一个文件等。每个信息项有一个信息框（information form），信息框与具体的应用和工具有关，这些应用和工具用来命名、移动、修改、复制、组织信息项，也可以为信息项赋予一些属性。

个人信息空间（PSI）：个人信息空间(Personal Space of Information)是指其所能够控制，或名义上能够控制的所有数据项的全体组成的集合（并不是指物理上对数据专属，例如邮件系统），一个PSI往往包括一个人的书籍、Paper文档，Email地址信息，Email文档、或其它存储在不同计算机上的与主体有关的文件，也包括网页链接。关于PSI的几点说明：

（1）一个PSI是个人信息项组成的集合。PSI的大小是动态变化的。

（2）PSI包含的是主体记忆过的个人信息项。PSI不能包括我们访问过的，但是没有记录的信息（如尚在缓存中的网页）。

（3）每个主体只有一个PSI。

（4）PSI是可供我们通过多种方法利用的潜在的数据源。对PSI中信息的有效重用，可以提高我们的工作效率。同时PSI的动态变动也引来了信任、安全问题。

个人信息管理（PIM）：鉴于PIM的最终目的就是通过对数据的存储，以达到信息的重用，从这种数据存储的角度， PIM本质上是一系列操作行为的集合，其行为目的是建立、使用、保持信息及需求之间的映射。对PSI中有关的行为按照input-storage-output分类，可以归为三种：输入、存储和输出。在此基础上提出了一个如图1所示的PIM概念框架[4]。

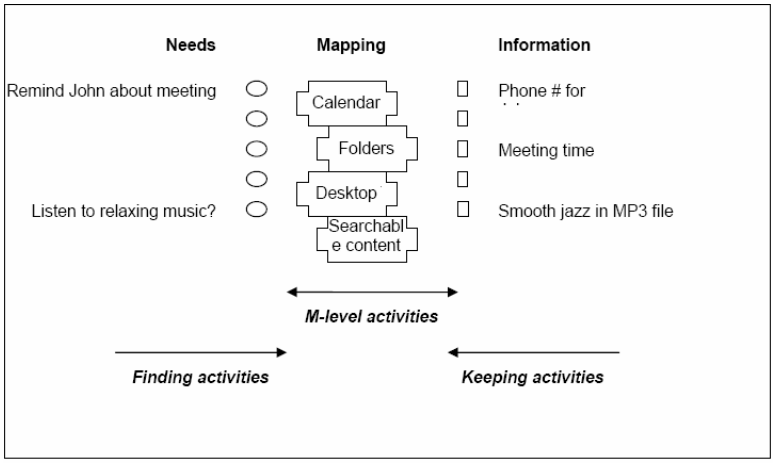


图1　　PIM概念框架

由图1所示的PIM概念框架可以看出，PIM涉及的行为可以分为三类：

• **信息保持行为：**影响到PSI中数据输入的一系列行为。具体来说，是指完成从信息到需求所进行的行为，例如，当我们遇到信息的时候，如某人电话号码、会议时间等，我们要将这些信息保存下来已备将来之用，这类行为包括，信息的分析、分类、记忆、增强、记录等。

• **信息发现行为：**影响到PIM中信息输出的一系列行为。具体来说，是指完成从需求到信息所进行的行为。例如，当我们需要用到某项信息（如某人电话、一封邮件、一张图片等）的时候，将个人需求提交，并从PSI中得到该信息。这类行为包括查询语言、人机界面、搜索技术、信息分析、自动提醒等，需要区分的是：这里所说的信息发现，和通常的信息搜索不同，这里指的是从个人信息空间中发现自己曾经记忆过的信息，而不是在公共数据空间中搜索某项信息。

• **“M-level activities” ：**影响PSI中数据映射的一系列行为。要高效地完成上面的两种映射，需要解决数据的存储、索引、安全性、一致性等一系列问题，这类行为就主要针对解决这些问题，其中核心的问题是个人数据空间的管理。

根据PIM的定义，作为一个研究分支，PIM聚焦于研究个人信息管理中的一系列行为，以提高各种行为的执行效率，最终提高个人信息管理的水平。

2.2 PIM 成为新的研究焦点

科学技术的发展使个人信息管理问题变得更突出，同时也为应对这一问题创造了条件，人们逐步关注到这一问题并开始进行这方面的研究，使 PIM 成为新的研究热点。

（1）PIM Workshop 的举办

PIM 研究引起了广泛的关注，其标志是 2005 年第一届 PIM 2005 Workshop 的举办，这是第一次专门针对 PIM 研究的专题研讨会，参加会议的有数据库领域的研究人员，也有微软、IBM 等公司的专家，从 PIM 基本概念、研究内容、研究目标等方面进行了讨论，取得了很大成果，我个人认为，最重要的是提出了很多重要的研究课题。

在 2006 年 PIM Workshop 上，收到论文 32 篇，其中有些针对 2005 Workshop 上提出的一些问题进行了深入的研究，有的论文针对个人信息管理中具体的问题进行了研究，这些论文有以下特点：

涉及面广。涉及到 PIM 研究的众多领域，包括基本理论、信息保存、信息分类、信息存储、隐私保护、邮件系统、行为分析、信息提醒等众多研究课题。即包括对基本概念、基本理论知识的研究，也包括针对特定应用需求的研究。

总体来说，处于起步阶段，这些论文很多针对 PIM 的某个具体课题，提出了自己的观点，并初步进行了论证，但对于具体的算法、数据的模型、系统的框架还没有作深入的研究和量化的实验分析，而且对于有些基本概念问题，也没有达成一致的看法。这些都为研究者提供了很好的机遇。

（2）国际会议中开始出现有关 PIM 的论文

近两年关于PIM课题的研究论文也开始出现，如 2006 VLDB关于个人数据空间研究的会议论文[7]。在CHI 2006 发表的会议论文中，有一些是关于PIM人机接口设计方面的问题的研究。

（3）PIM 为我们提出了许多新的具有创新意义的研究课题

跨学科研究成为 PIM 研究新的特点，PIM 与数据库技术、人机接口技术（CHI\HCI）、认知科学、人工智能的结合，为我们提出了许多跨学科的研究课题，如 PIM 中的主体行为分析、个人信息挖掘、个人数据空间管理、特殊环境下的人机接口设计、数字记忆与信息自动提醒等。

同时，PIM 研究也面临重大挑战，一方面是由于信息的多样性，信息类型多样，位置各异，成为一个个“信息孤岛”；此外，主体在 PIM 中的关键作用使得信息的获取、组织、输出都充满个性化，PIM 的评测也受主体因素影响而变得复杂；但是，正是这些挑战的存在，给我们提供了进行创新性研究的课题和动力。

3．PIM研究技术分析

由上节的PIM概念模型可以看出，对PIM的研究将重点放在对三类行为的研究方面：信息保持、信息存储、信息重用，每个环节都涉及很多技术问题，也有许多研究成果，本节从这三个方面对于PIM研究内容、关键技术问题做一分析。

3.1 PIM中的信息保持技术

PIM中的信息保持是指将所需要的信息项从公共信息空间复制到个人信息空间的一系列行为，以使主体能够重复访问、使用该信息项；同时也包括将无用的信息从个人信息空间中清除的行为，如图2所示。在Barreau’s definition的框架模型中，将这一阶段定义为信息保持（keeping），这是为了区分传统的信息输入的概念，信息保持不仅仅包括主体通过手工输入的个人信息。 也包括其他偶然遇到并复制到个人信息空间中的信息，如浏览到的某个网页、收到的某个邮件、临时记录的某个电话号码等。

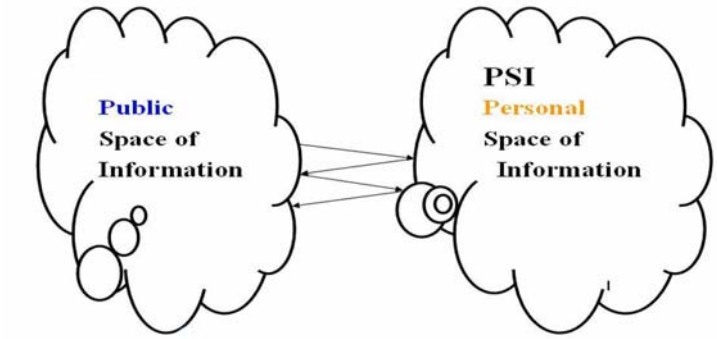


图2 PIM中的信息保持

由于个人信息形式的多样性，以及缺乏相应的工具，使得个人信息保持（输入）异常重要，也成一个技术难点。原因主要有：

（1）信息的隐蔽性，很多数据以隐式数据的形式存在于公共数据空间中，在公共数据空间遇到、发现这些数据本身就是一个很难的过程；

（2）当我们积极（寻找）或消极（偶然）遇到某些信息的时候，由于缺乏相应的技术和工具，也无法有效的将信息保持在我们的个人信息空间中；

（3）信息的异构性，数据存在于众多的“信息孤岛”当中，如何将这些异构数据转化为PSI中的数据元素，也是一个困难的过程；

（4）主体本身的因素。个人信息管理不仅包括数据等客体因素，主体本身也在这一系统中起重要作用，由于主体个性差异巨大，也为信息保持带来了很大难度，例如不同个体获取信息的能力是不同的；遇到信息后对其价值的判断也是一个难点问题。

信息保持主要通过以下几种行为实现：

（1） 手工输入方式

这种方式是指用户将自己所需要用到的信息项手工输入到个人信息空间中，如写邮件、输入自己的电话号码本、输入自己的个性化信息、写文章等，元数据输入需要采用这种方式。

（2） 信息集成

信息集成是指将现有的与个体有关的信息项输入到自己的信息空间中。信息集成的基础是信息感知（遭遇），只有遇到我们需要的信息后才能进行集成。信息感知又分为主动和被动两种：

被动信息感知是指我们无意偶遇到与我们有关的信息；

主动信息感知是指我们有意识的借助一些外部工具去寻找我们所需要的信息；主动信息感知往往要借助外部工具，如Web搜索工具，数据分析工具等等。从研究内容区分，感知到信息是PIM信息管理的起点。

与信息保持相关的技术问题：

信息记忆技术：由于信息遭遇的随机性，可能在任何时间、任何地点遭遇到有价值的信息，如何在这种情况下将信息快速的记忆下来是一个难点问题。

信息分类技术：遇到信息以后，保持信息的目的是为了以后信息的再利用，因此，“如何对信息进行分类、组织、存储，以保证用户需要该信息时能够快速找到”也是一个问题。

异构数据的处理：由于信息类型多样，需要研究不同的处理技术。

信息可用性的判定：遇到信息以后，要评估其价值以确定是否进行保存。由于主体个性差异巨大，如何判定信息的价值也是一个难点问题。

信息保持的人机接口技术：信息保持需要高效完成，主体的多样性也决定了接口设计的

难度。例如，针对不同的用户人群研究其适用的信息保持界面；针对用户所处于的不同的物理位置环境研究其适用的用户界面；或者针对局部的具体PIM应用研究其界面。高效的工具和人机界面才能使PIM高效的为个人信息管理服务。

个人信息挖掘技术。信息挖掘也不是一个新名词，但是，不同领域的数据有其自身特点，对知识信息的需求也不同，PIM也是如此，PIM数据挖掘是指根据PSI中现有个人信息，挖掘尚未被主体认识到的信息的过程，而不是指从公共数据空间挖掘数据信息。首先个人数据信息有自身特点，包括数据分布、数据量大小等方面；另外，受主体差异的影响，个性需求差异也很大，例如，不同职业、不同行业的个体关注的信息是不同的，因此个人信息挖掘技术也是一个重要的研究方向。一方面是针对有共性需求的数据挖掘技术研究；一方面是针对个性需求的数据挖掘技术研究。

3.2 PIM中的数据存储技术

PIM中的数据存储技术研究，实际上就是个人数据空间管理系统（PDSMS）的研究。目前，因特网的发展，使人们对于数据资源的存储、访问等出现了新的特点，从而传统的数据库技术不能完全满足新的数据管理的需要，google、百度等网站对数据处理方式的改变也说明了这一特点，于是人们提出了一个新的概念：数据空间（DataSpace）. 由于个人数据信息的特点，Dataspace将是存储个人信息的理想方式，在PIM中称为PDS（Personal Data Space）。目前在这一研究领域。Jens-Peter Dittrich Marcos, Antonio Vaz Salles等人作了大量工作，他们在04-05年共发表了2篇关于个人数据空间管理的论文[5]-[6]，在2006 VLDB论文中，系统阐述了个人数据空间管理的概念，提出了一个新的数据模型：*iMEMEX*, 提出了数据源视图的概念，并基于此实现了一个PDS原型。他们的工作为个人数据空间管理的研究建立了一个框架模型，但是需要研究的问题还很多。

首先是对PSI中数据存储策略的研究，面对如此巨大的个人数据信息，在一台机器上进行数据的存储是不可能的，同时从安全性、可访问性等方面进行考虑，也不是很好的。因此，对于Dataspace存储策略的研究成为一个基础性的问题。

因为Web数据变动的特点，有的数据会随时消失，这样就要求我们对数据的安全性策略进行研究，对数据安全性进行评估，以确定数据的处理策略。

数据模型的研究也非常重要，传统的数据库管理系统大都是基于关系模型的，在PDS中，传统的关系模型是否适用，随着PIM的使用，PSI中的数据量会越来越大，这样就会造成用户访问数据代价增高，因此，如何提高用户访问效率就会成为一个大问题。和传统数据库不同，在数据空间中，影响访问效率的关键因素可能不再是磁盘的I/O，那么这种情况下应改采用什么样的索引策略，在Dataspace中索引的涵义是什么。这些都是需要研究的问题，也是下一代数据库系统所必须解决的问题。

目前国外在这方面的研究还不多，进行这些理论问题的研究，对于下一代的数据库研究意义重大。

3.3 PIM中的数据输出技术

PIM的数据输出技术是指从客户提出需求到获取信息结果的技术，即信息查询技术。即研究客户如何快速、高效的从PDS中获取自己想要的信息。这与传统DBMS中的查询类似。因此也涉及到查询语言、查询优化等一系列技术，又由于查询语言、查询优化都依赖于数据存储模型，因此不同的数据模型也会影响到数据输出技术的研究。

PIM中的数据输出，不仅包括传统的基于用户查询的方式，也包括自动提醒，类似于传

统数据库的触发器，提醒（reminding）技术也是重要研究内容之一，由于PIM系统的特殊性，提醒机制也要考虑到主体的各种情况，包括所处的环境，提醒的方式等，这与传统的触发器概念又有质的不同。信息提醒的研究是数据输出中的重要课题，也是难点问题，数据提醒的前提是数据的分析，因此人工智能（AI）技术和推出信息(Push)技术的应用会成为数据提醒技术的研究重点。

人机界面设计也是这一部分的重要内容，由于 PIM 的研究目的是使人们能够高效快捷的享受信息带来的巨大便利。客户个体的差别，所处环境的差别，要求系统能够将信息以用户最方便的形式展示出来。

4．PIM 研究现状

PIM研究是跨学科的研究，它涉及信息搜索、人机接口、认知科学、数据库技术、人工智能等众多研究领域。目前PIM的研究还处于起始阶段，国外对PIM的一些基本问题进行了研究，取得了一些成果。在PIM Workshop 2006 的 32 篇会议论文中，其中多是阐明的关于PIM的一些观点,有些论文着眼于PIM中的某个具体问题，如个人邮件信息的处理[7]、移动环境下的PIM管理[8]、个性化的信息查询[9]。也有一些文章涉及到PIM的一些理论问题[10][11]，如数据模型，PIM研究的一些前提等，这些文章提出了一些非常有价值的观点和概念。这些工作表明，PIM的研究已经起步，并将引起研究者的广泛关注。

与国外相比，目前国内专门针对PIM、PSI的研究还比较少，软件学报的两篇综述文章“数据库技术发展趋势”[12]和“个性化服务技术综述”[13]，对于数据集成技术、用户界面技术、个性化服务技术进行了分析总结，此外，也有一些针对信息搜索、基于用户行为进行信息分析的研究成果，但是，总的来说，对一些基础性的、核心性的技术研究不多，如数据空间中的数据存储技术、数据模型、索引技术、优化技术等，相对国外的研究还有差距。由于对PIM的研究会涉及新一代数据库技术方面的一些问题,因此,应当引起国内该领域研究者的关注，不断跟踪新技术动向，争取在PIM研究方面取得一些高水平的成果。

5．PIM 研究展望

PIM 为我们提供了新的机遇和挑战。目前来看，近期 PIM 研究将主要围绕以下几个方面：

（1）数据空间技术的研究，PIM 中将侧重于研究个人数据空间技术，具体包括：数据模型、数据存储、数据独立性、索引技术、查询优化等。

（2）数据的保持技术。

（3）数据的发现/再发现技术。数据的再发现是 PIM 研究的目的，特别是信息提醒技术的研究，需要应用人工智能的相关技术成果。

（4）对于 PIM 技术、工具的评价方法学、框架、基准的研究。因为 PIM 实际上是一个包括主体、客体的系统，对其评价是非常复杂的，但这又是 PIM 研究的基础工作，因此会有很多研究围绕这方面展开。

（5）PIM人机接口技术的研究，

综上所述， PIM 研究具有重要的理论意义和现实应用意义。PIM 将研究 Web 环境中的个人数据管理问题，这些问题是下一代数据库技术必须解决的问题，同时为新的数据库技术提供了应用环境。PIM 是面向应用的，在研究过程中会不断开发出面向不同用户、不同领域的 PIM 工具软件，从而提高人们的信息管理水平，使个人从信息的枷锁中解放出来，产生巨大的社会效益。

参考文献

[1] http://news.xinhuanet.com/world/2006-11/04/content_5290138.htm

[2] Alexandros Ntoulas, Junghoo Cho, Christopher Olston: What's new on the web?: the evolution of the web from a search engine perspective. WWW 2004: 1-12

[3] Vannevar Bush：As we may think. The Atlantic Monthly , July 1945

[4] William Jones , Harry Bruce . Report on the NSF PIM Workshop, January 27-29, 2005, Seattle **A** Report1 on the NSF-Sponsored Workshop on Personal Information Management, Seattle, WA, 2005

[5] Dittrich, Jens, M. Salles, S. Karaksashian. *iMeMex: A Platform for Personal Dataspace Management.*, A SIGIR 2006 PIM Workshop Position paper.

[6] VLDB 2006 regular paper. *iDM: A Unified and Versatile Data Model for Personal Dataspace Management,* JensPeter Dittrich, Marcos Antonio Vaz Salles

[7] Yu, Xiaoyan, Mohammad Alkandari, Pengbo Liu, & Manuel A. Perez-Quinones, *Visualizing a Personal Social Network of Email Archives for Re-Finding*. A SIGIR 2006 PIM Workshop Position paper.

[8] Singh, Gurminder, *PIM for Mobility*. A SIGIR 2006 PIM Workshop Position paper.

[9] Cutrell, Edward, Susan Dumais, & Raman Sarin, *New directions in personal search UI.* , Personal Information Management.A SIGIR 2006 PIM Workshop Position paper.

[10] Kirsh, David, *Personal information objects & Burden of multiple personal spaces*. A SIGIR 2006 PIM Workshop Position paper.

[11]Spurgin, Kristina, *A Sense-Making Approach to Personal Information anagement*. A SIGIR 2006 PIM Workshop Position paper.

[12] 孟小峰, 周龙骧, 王珊. 数据库技术发展趋势， 2004,Vol.15, No.12

[13] 曾春, 邢春晓, 周立柱.个性化服务技术综述， 2004,Vol.13, No.10

Mashups——一种新型 Web 应用程序

凌妍妍 （Web 组）

1. 引言

一种新型的基于 Web 的数据集成应用程序正在 Internet 上逐渐兴起。通常用术语 mashup 表示。根据 Wikipedia 的解释，Mashup 是将多个不同的支持 web api 的应用进行堆叠而形成的新型 web 服务。这种新型的基于 Web 的数据集成应用程序正在 Internet 上逐渐兴起。它利用了从外部数据源检索到的内容来创建全新的创新服务，将来自不止一个数据源的内容进行组合，创造出更加增值的服务。Mashup 所能利用的外部数据源格式多种多样，表现出惊人的兼容性，它涵盖 public APIs, XML/RSS/Atom feeds, web services, HTML 等。人们普遍认为 Mashup 具有 Web 2.0 的特点。Web 2.0 的主要思路是在互联网上建立起大众的贡献的共享的信息平台，协作和共享是这种思想的精髓。Mashup 技术也是建立在各种 Web 应用程序贡献出自己的服务和内容，同时共享其他人和其他组织提供的信息和服务的基础上的，自此基础上进行组合、增值从而构造出更多更具吸引力的新的 Web 应用程序。随着越来越多的 Web 站点公开了自己的 api，许多人已经和正在用 eBay, Amazon, Google and Yahoos APIs 构建新的 Mashups，使得这种新型的 Web 应用模式成为了现实。这篇文章对 Mashup 的分类和架构进行了深入的研究和探索，另外您还将看到对 Mashup 在企业应用中引出的研究问题的一些分析。

2. Mashup 分类

本节将简要介绍[2]中对出名的 Mashup 类型进行调查的一些成果。现今涌现的 Mashup 应用大致由以下几类构成：

视频图像 Mashup——Mashup 设计者利用与图像相关的元数据（例如谁拍的照片，照片的内容是什么，何时何地拍摄的等）对视频和图像资源进行关联。CelebrityV 就是这样的一个应用—它将来自 Flicker 的明星照片和来自 youtube 对应的明星视频剪辑进行了匹配和组合。

搜索购物 Mashup——搜索和购物 mashup 在 mashup 这个术语出现之前就已经存在很长时间了。在 Web API 出现之前，有相当多的购物工具，例如 BizRate、PriceGrabber、MySimon 和 Froogle，都使用了 B2B 技术或屏幕抓取（screen scraping）的方式来累计相关的价格数据并进行比较。之后为了促进 mashup 和其他有趣的 Web 应用程序的发展，诸如 eBay 和 Amazon 之类的消费网站已经为通过编程访问自己的内容而发布了自己的 API。

新闻 Mashup——新闻源（纽约时报、BBC 或路透社）已从 2002 年起使用 RSS 和 Atom 之类的联合技术来发布各个主题的新闻提要。以联合技术为基础的 mashup 可以聚集一名用户的提要，创建个性化的报纸，从而满足读者独特的兴趣。Diggdot.us 正是这样一个应用。

地图 Mashup——地图 Mashup 蓬勃发展的一种主要动力就是 Google 公开了自己的 Google Maps API。现阶段，几乎所有包含位置数据的数据集均可利用地图通过令人惊奇的图形化方式呈现出来。Microsoft（Virtual Earth）、Yahoo（Yahoo Maps）和 AOL（MapQuest）也不甘示弱，很快相继公开了自己的 API。

3. Mashup 架构

此架构[3]非常简单。来自客户机浏览器的请求传向 Mashup 站点所在的 Apache Web 服

务器。请求的页面包括 HTML 和 JavaScript。JavaScript 调用一个或多个 API 内容提供者提供的服务后，按照该 Mashup 的逻辑进行内容组合。举一个 Mashup 的例子，用户向 Mashup 站点提交房屋的所在地点和房价查询是否高于当地平均水平，一个请求就传向一个与后台房屋信息数据库连接的 Web 服务器，同时调用 Google Maps API 提供的服务，执行 Mashup 逻辑并将组合的内容在客户机端浏览器中显示。一般来说，Mashup 服务主要涉及 3 方面的内容：Mashup 站点，API 内容提供者以及客户机的 Web 浏览器。

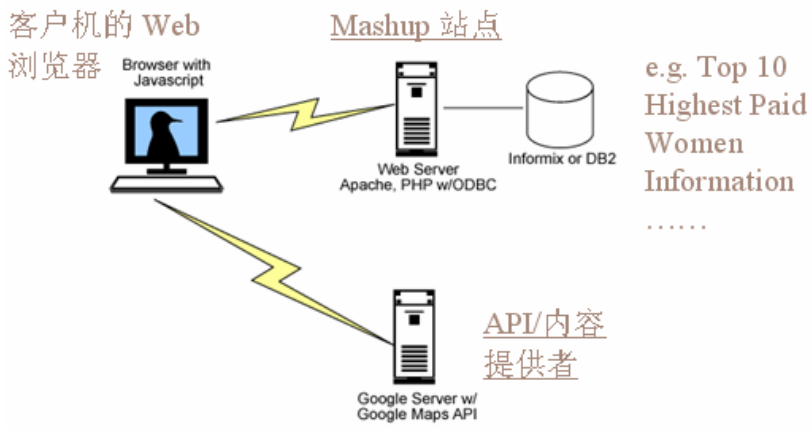


图 1：Mashup 架构

3.1 Mashup 站点

一个 Mashup 应用可能需要将来自多个数据源的信息进行合并组合，构造新的服务，因而这里所说的 Mashup 站点也就是 Mashup 逻辑所在的地方。尽管 Mashup 这类新型的 Web 应用程序可以采用以往的 Web 服务器技术（Java servlets、CGI、PHP 或 ASP）构造传统 Web 应用程序，我们却发现越来越多的 API 开始设计成通过浏览器端的 JavaScript 进行访问。于是对于 Mashup 的执行来说，合并内容也可以直接在客户机的浏览器中通过客户机端脚本（即 JavaScript）或 applet 生成。我们将 Mashup 使用的这种方法称为胖 Internet 应用程序（简称 RIA）。

在客户机端进行 Mashup 应用中的内容合并，其优点可以概括如下 ：首先，从 Mashup 服务器的角度来说，对服务器的所产生的负载较轻（数据可以直接从内容提供者那里传送过来）；其次，从用户的角度来说，具有更好无缝用户体验（页面可以请求对内容的一部分进行更新，而不用刷新整个页面），当然这样的功能得益于 Ajax 这样的 Web 应用模型的诞生。

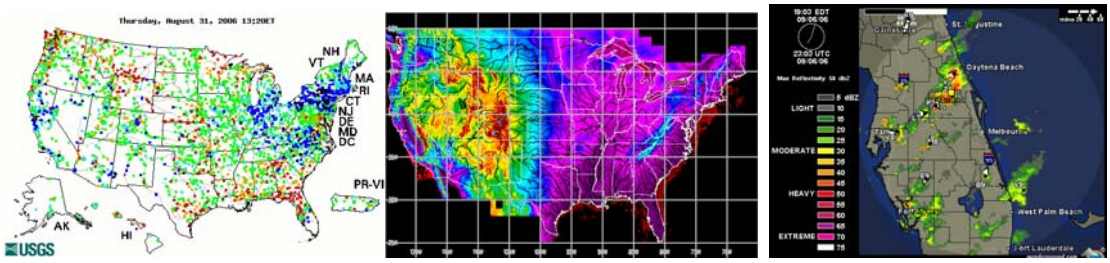
3.2 API 内容提供者

他们提供的内容为 Mashup 应用程序所用。为了方便外界获取和使用，他们将自己的内容通过 Web 协议对外提供（例如 REST、Web 服务和 RSS/Atom）。按照其通常的功能和使用，Web 协议可以分为两组。第一组处理消息传递、接口描述、寻址和交付的问题。最有名的是消息传递协议，称为简单对象访问协议（Simple Object Access Protocol，SOAP）。此协议对消息进行了编码，这样就可以通过传输协议（如 HTTP、IIOP、SMTP 或其他协议）在网络上传递它们。下一组协议和规范定义了服务如何公开它们自己以及如何在网络上相互发现。对于要相互查找的服务，统一描述、发现和集成（Universal Description, Discovery and Integration，UDDI）为查找和访问服务定义了注册中心和相关的协议。当然，Web 上还存在着很多有趣的潜在数据源可能并没有方便地对外提供 API，Mashup 应用如果想用到这些信息，可以通过前面我们提到的屏幕抓取的技术实现，通过对特定页面进行分析，提取 Mashup 感兴趣的信息。

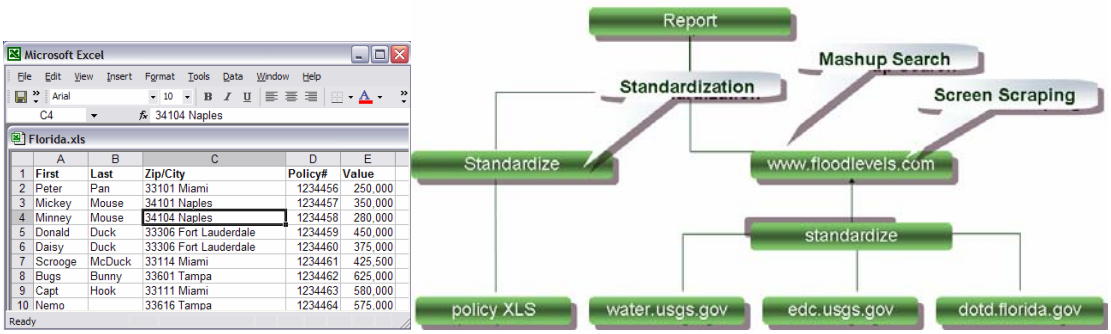
4. 企业级 Mashup 及引出的研究问题

Mashup 技术并非只会提供消费者网站使用的、加了注释的地图，这项技术具有真正的企业应用前景。Mashup 这样轻量级的集成在企业中有着很多的先例：从历史悠久的股票报价系统，到将 UPS 或 FedEx 等快递公司的跟踪数据与订单记录组合起来，提供订单状态单一视图的电子商务网站等等。IBM 等门户服务器厂商提供了诸如 StrikeIron 之类的图形化操作工具，帮助用户集成来自不同地方的数据源，实现简单、个性化的 Web 应用。

一方面，企业必须将许多原本并不能很好彼此共存的管理系统和应用程序拼凑到一起。DBMS、内容管理系统、数据挖掘包和工作流系统都可以购买，但该公司必须自行开发集成软件以集成它们。每当增加了新的数据源或信息必须流转到新的目标时，就必须扩展客户自制的解决方案。因此，企业需要一个提供所有这些服务的统一视图的健壮平台，这样的平台应该突破存在于 DBMS、内容管理系统、中间层高速缓存和数据仓库之间的界限。另一方面，[4]中指出即时应用的出现使得利用企业信息架构之外的信息成为新的需求 (电子邮件，报告和文档，网页，电子表格，决策支持数据等等)。这样的即时应用对效率和易用性都提出了新的要求，因此在企业级 Mashup 应用中出现了用“assembly”来代替“programming”的思想。



(HUC = Hydrological Unit Code)　(Geocode = Latitude/Longitude)　(Geocode = Latitude/Longitude)



(Zipcode)

图 2：构造 Mashup 的过程

首先来看一个企业即时应用的实例（例 1）——一佛罗里达州的一名保险经纪看见了一则美国风暴灾害的新闻报道，公司要求他递交一份最新灾害损失分析报告，对这场风暴给公司带来的损失进行评估。如图 2 所示，该名保险经纪为了完成这份受灾情况分析报告，他首先需要将电子表格里客户的 ZipCode 通过网站 Water.usgs.gov 转化为水文学上的一种编码，然后利用网站 edc.usgs.gov 将水文学编码转化为经纬度的表示，最终从网站 dotd.florida.gov 上通过经纬度定位当地的受灾情况。当然，如果 Web 上已经存在了这样的一个 Mashup 应用 www.floodlevels.com 合并了来自上述三个网站的内容和服务，即直接提供了从 ZipCode 查询受灾情况的功能。那么如此现存可利用的资源更应该得到有效发掘和运用。

下面我们将从构造企业级 Mashup 应用的四个方面分别讨论存在的研究问题。

- **资源发掘**

构造企业级 Mashup 应用并不一定是从零做起，Web 上存在着许多对处理企业即时应用有效的资源和服务。比如，例 1 中提到的网站 www.floodlevels.com 就是一个现存的可满足从 ZipCode 查询受灾情况的 Mashup 站点，它隐藏在 Web 信息海洋中。可想而知，利用现存可利用的 Mashup 作为构建企业解决方案的基础，在效率和有效性上都会带来质的飞跃，也大大省去了 Mashup 应用重复开发带来的资源浪费。

于是一种新形式的搜索引擎正在酝酿。这种搜索引擎试图通过使用者键入简单的查询，来发掘 Web 中所有与该查询相关的 Mashup 资源。如使用者期望通过查询“Flood Levels”来找到这样的一个 Mashup 资源 www.floodlevels.com 为我所有。Mashup 资源发掘的问题和 Deep Web 研究领域中特定类别数据源的发现问题有异曲同工之处。相关资源的发掘需要解决两方面的问题，一是对资源特性的描述，而是资源与用户需求的匹配程度。Deep Web 数据源的特点相对来说比较明确，我们可以从接口的复杂性，结果的结构化等来对 Deep Web 数据源进行定义并分析该数据源的类别和接口形式。但是对于 Mashup 数据源来说，最大的困难是如何让搜索引擎理解 Mashup 的逻辑，从而推断与用户需求的匹配程度。

- **需求表达**

构造企业级 Mashup 应用的第二阶段是用户的需求表达，这主要包括两方面的含义：首先，如果在资源发掘阶段能够找到现存的可直接利用的 Mashup 很好地解决用户的需求，那么用户可以在此基础上进行上层应用的构建。比如一旦 Mashup 资源 www.floodlevels.com 得以发掘，那么剩下的只需要在客户资料（电子表格数据源）和该 Mashup 之上构建异质数据源集成的解决方案。另一方面，如果在资源发掘阶段无法找到或者实际并不存在可直接利用的 Mashup，那么用户只有考虑从现有的资源出发，着手从最底层开始构思多个异质数据源之间的 Mashup 逻辑，表达自己的需求。

- **应用构建**

合理的 Mashup 逻辑从构思到实现，一个最基本的问题就是如何在一个用户友好的环境下，实现各个逻辑模块的组装（代替传统的编程）。于是，良好应用平台的构建是当务之急。首先必须考虑到对于 Mashup 企业级应用来说，它的使用者是最普通的用户。这样的用户既不是一个 Java script 专家，也不懂 PHP/Java/Ruby 这些编程语言。其次，我们还必须考虑到即使用户在资源发掘阶段能够找到了一个现存可利用的 Mashup 应用，但是如果该站点对外不提供 API，那么这种情况甚至还要求用户自己构建屏幕抓取程序来处理该站点，这是不现实的。也就是说，广大最普通的需求者亟需一个良好的平台去构建自己的应用。

[4]中对如何方便构建 Mashup 进行了一些基本分析。如图 3 所示，构建 Mashup 最直接的方法是用编程语言自己动手编写一些过程性的代码（Procedure Code），这种方法对使用者技术要求很高，哪怕一个细小的读取操作都要由使用者自己来指定实现细节。第二种方法是采用类似于 XQuery 的声明性查询语言（Declarative Queries），这种方法实现起来更为简单，至少使用者只需要遵守查询语言的格式，指明操作步骤，而不用关心每一个细节的具体实现；然而对于广大最普通的需求者来说，我们不能做出他们熟悉计算机语言的假设。实际生活中，往往图形化的开发平台才是最能虏获人心的。

由此看出，构建Mashup应用所需要的平台应该至少应具有如下两个特点：容易使用，表达力强。实际中的Mashup应用可能涉及到的数据源种类繁多，形态各异，涵盖public APIs, XML/RSS/Atom feeds, web services, HTML等等，因此如何搭建一个能很好处理多种格式异

质数据源的平台是极其具有挑战性的。

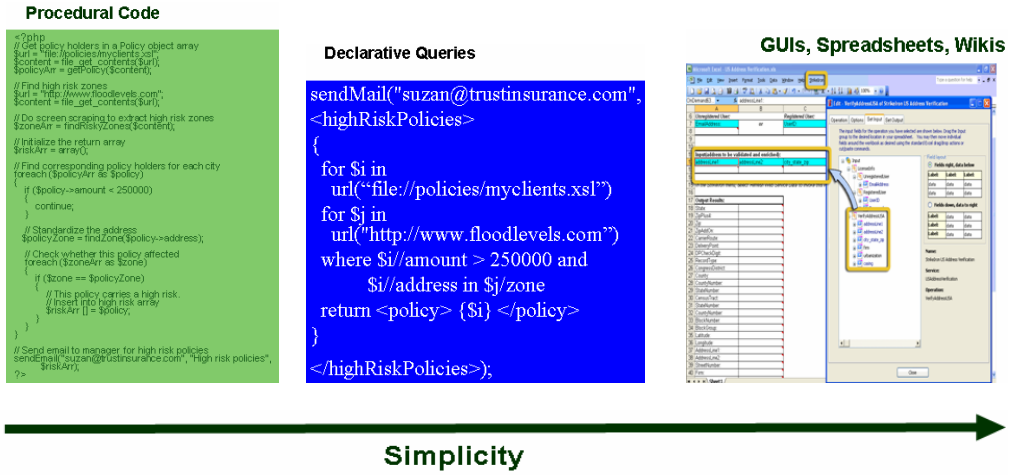


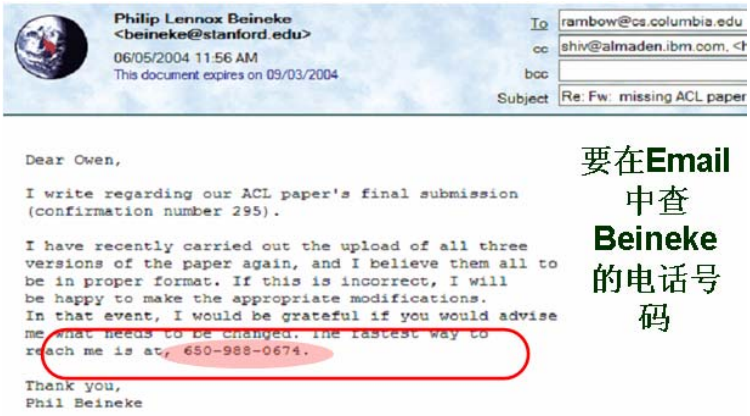
图 3：构建自己的 Mashup

有的读者可能会问，帮助广大最普通的用户实现这种企业的即时应用是否还有更简单的方式呢？最理想的方式是，当用户在类似搜索引擎的环境下以自然语言的方式表达自己的整个需求，整个 Mashup 能够自动构建，就像搜索引擎一样直接将结果返回给用户，无须用户参与中间过程的构建。比如，例 1 中保险经纪通过搜索“flood levels for zipcodes 33101, 34106, etc.”而直接获得答案。这种智能化的人机交互方式首先需要自动发现和用户查询内容相关的，能帮助回答查询的若干资源（如图 2 中的若干网站）；并且要在理解这些资源的逻辑的基础上，在资源间进行类似 join 的 Mashup 操作。[1]在处理 Deep Web 数据集成的过程中，自动选择合适的相关数据源进行查询，并通过数据源查询能力的组合来回答用户提出的查询。例如，在左图所示的 5 个数据源上，用户想查找 1992 年之后生产的运动车型的价格和评论。那么首先可以排除 Source 3 和 Source 4，因为这两个数据源不提供与用户查询相关的内容；其次我们需要组合 Source 1，Source 2，Source 5 的内容。Source 1 或者 Source 2 的输出与 Source 5 的输入在 model 和 year 字段上是可以做连接的，从而通过不同 Deep Web 数据源的横向连接，形成能帮助用户回答查询的统一视图。在这篇文章中，处理的对象相对来说比较单一，即 Deep Web 网站，而且假定对每个网站的查询能力都事先进行了分析，定义了明确的输入输出接口。然而，在 Mashup 应用程序中，我们面临的任务更加艰巨，如何理解各个类型数据源的处理能力，接口以及提供服务的范围成为了新的研究课题。

Source 1: Used cars for sale. Accepts as input a category or model of car, and optionally a price range and a year range. For each car that satisfies the conditions, gives model, year, price, and seller contact information.
Source 2: Luxury cars for sale. All cars in this database are priced above $20,000 Accepts as input a category of car and an optional price range. For each car that satisfies the conditions, gives model, year, price, and seller contact information.
Source 3: Vintage cars for sale (cars manufactured before 1950). Accepts as input a model and an optional year range. Gives model, year, price, and seller contact information for qualifying cars.
Source 4: Motorcycles for sale. Accepts as input a model and an optional price range. Gives model, year, price, and seller contact information.
Source 5: Car reviews database. Contains reviews for cars manufactured after 1990. Accepts as input a model and a year. Output is a car review for that model and year.

● 语义和非结构化

在解决 Mashup 数据集成的问题里，语义（Semantic）和对非结构化数据（Unstructured data）的处理是两大难题。语义方面举一个最简单的例子，在自己储存的一份客户资料里，客户的名字可能使用自己熟知的一些昵称或简称代替的，当某个 Mashup 应用需要处理到这



份资料并与另外一些信息在该字段上做连接的时候，机器是无法识别这些昵称或简称的。在关于语义的领域里，存在大量的研究工作。同时我们很欣喜地看到随着信息的高速增长，可用的标准化服务越来越多（将<male, female>，<M,F>,<1,2>等表示同一语义的不同方式识别并统一起来就是标准化的一个简单例子），Microformats 也正在越来越多地被重视和使用，企业中也逐渐将元数据的管理提上了日程。种种这些努力都在为逐步跨越语义这道鸿沟推波助澜。另一方面，人们总在试图在信息领域所有非结构化的数据上做工作，解析成为结构化的数据，这样机器就可以直接处理了。可是如何从非结构化的数据中抽取出结构化的信息是一个非常头疼的问题，尤其是 Mashup 企业级应用面临了不同于以往我们所考虑的数据抽取问题。举一个例子，当一个 Mashup 应用需要在一份客户资料和一封电子邮件之间在姓名和电话号码字段上进行连接操作，那么这时候的数据抽取问题就是全新的——如何从非结构化的电子邮件文本中抽取出<姓名，电话号码>这样的二元组。如图 4（1）所示，落款为 Beineke 的邮件中包含了一个电话号码，那么这种情况下，我们能相对容易识别并抽取这样的姓名和电话号码二元组。但是如果出现如图 4（2）所示的情况，文中出现的电话号码并不是 Beineke 的，而是其伙伴的，那么对这样混淆情况的识别对新形势下的数据抽取提出了更高的要求。于是，在对非结构化数据进行解析的过程中，文本语义的理解和对领域知识的利用也变得越来越重要。从这个例子中，我们对搜索引擎未来的发展方向也可窥见一斑。用户或许可以只需要向搜索引擎提交简单的查询“Beineke phone”，搜索引擎就能将解析之后得到的<姓名，电话号码>二元组返回给用户，并且要保证返回结果的正确性，避免出现如图 4（2）所示的混淆情况。

（1）

（2）

图 4：从 Email 中抽取结构化信息

5. 结束语

Mashup 是一种令人兴奋的交互式 Web 应用程序，它利用了从外部数据源检索到的内容来创建全新的创新服务。第一个关键词是“交互式”。实际生活中的 Mashup 应该是面向广大普通使用者的，简单易用表达力强的新型 Web 应用。用户应该能在一个交互式的用户友好的平台上进行各种异质数据源的“组装”，而不是以繁琐编程的方式构造 Mashup 应用程序。当然这样的平台必须是足够智能化的，它不仅需要帮助用户寻找现成可利用的资源，

还需要使得具体实现细节透明化，如对语义的理解和对非结构化数据的处理等。第二个关键词是“创新”。Mashup 应用将已有的来自多个数据源（public APIs, XML/RSS/Atom feeds, web services, HTML）的信息进行加工，融合和进一步利用，从而使之产生更大的价值。这有点

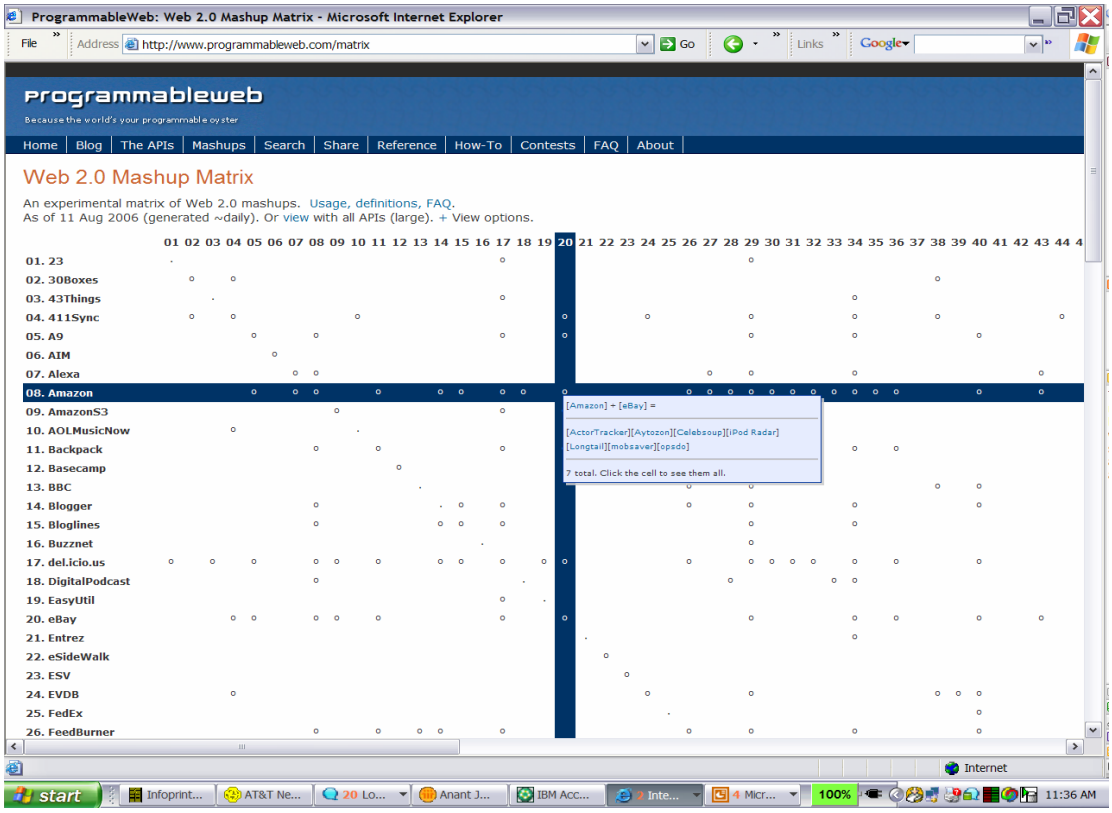


图 5：Mashup 矩阵

像做菜：把各种原材料按照某种方式烹饪在一起（当然不是简单的混合），形成一道美味。又比如把磨锋利的石头绑在木棍上，做成一个斧子，却不是石头和木棍两样东西功能的简单累加，也就是实现了“1+1>2”的突破。ProgrammableWeb.com 对目前 internet 上涌现的 Mashup 应用进行了概要性的统计和描述。如图 5 中 Mashup 矩阵所示，矩阵中每一行列交叉点都代表着已经存在这样的 Mashup，它在该行和该列所代表的信息源的基础上进行了组合和创新。随着越来越多的信息源开始对外公开了自己的 API，越来越多的有趣的 Mashup 应用也如雨后春笋般涌现，这个 Mashup 矩阵将不再稀疏。

参考文献

[1] Alon Y. Levy, Anand Rajaraman, Joann J. Ordille: Querying Heterogeneous Information Sources Using Source Descriptions. VLDB 1996: 251-262

[2] http://www-128.ibm.com/developerworks/cn/xml/x-mashups.html Mashups：Web 应用程序新成员（Duane Merrill，2006）

[3] http://www-128.ibm.com/developerworks/cn/db2/library/techarticles/dm-0602lurie/ DB2 和开放源代码：在 Linux 上使用 Google Maps API、DB2/Informix 和 PHP 创建地图（Marty Lurie 和 Aron Y. Lurie，developerWorks，2006）

[4] http://aitrc.kaist.ac.kr/~vldb06/slides/K-1.ppt Anant Jhingran: Enterprise Information Mashups: Integrating Information, Simply. VLDB 2006.

Essential Google

——由 Google File System 所想到的

王仲远（web 组）

引言：十年前，微软依靠 Windows 操作系统，成为 IT 业界的的神话；而十年后的今天，Google 以其强大的互联网搜索能力征服了全世界，成为当今 IT 界最耀眼的公司之一。本文以 Google 的文件系统为切入点，介绍了 Google File System 的工作原理，论述了作者对 Google File System 与 Datebase 的一些思索和比较，试图探讨出一个全新的属于数据库人的研究方向。

一、李开复与 Google 中国

这是一个造星的时代，当一切成功都被神化以后，它外面所笼罩的美丽光环，使我们常常不能清醒地认识一个公司、一个人，他所起的作用，他所获得的真正成就。Google 就是这样一个公司，李开复就是这样一个人。

去年，李开复来我们学校青年大讲堂作报告，他信誓旦旦地说“微软将会是我度过余生的一个地方”。但一年之后，言犹在耳，可是物是人非：他由微软全球副总裁的身份变为了 Goolge 中国区总裁。顿时，李开复成为国人关注的焦点，Goolge 也成为了国人关注的焦点。

Google 中国李开复(左)和周韶宁(右,已离职)

李开复的加盟，使得 Google 成为大学校园里一大批学生向往的地方。Google 对我们来说，也似乎不再是那个遥不可及的在纳斯达克一上市就创造奇迹的地方，而是一个触手可及的承载着太多荣耀和梦想的地方。Goolge 中国风暴席卷大学校园！

就如几年前的微软神话一样，Google 中国也是大家心中的神话：宽松的工作环境，随手可取的零食，带着宠物上班……进入了 Google 就意味着站在了 IT 时代潮流的浪尖。但是，Google 真的就是这么的完美无瑕吗？

二、Google 人才观

正当大家对 Google 充满无限遐想的时候，Google 举行了一个大型的实习生招聘活动。所有人都跃跃欲试，期待着 Google 再给大家一个惊喜。但是，当笔试卷子发下来时，所有人都惊讶了，笔试题目是如此的平凡无奇，全部是最基本的算法题和计算机知识。于是，许多人又叫嚣着“Google 神话破灭了！”。

但真的是神话破灭了吗，或者是 Google 并非是一个神化般的公司呢？

我们依然清晰地记着，去年 10 月，周韶宁踌躇满志的加盟 Google，与李开复成为 Google 大中华区联合总裁。但是，周韶宁向 Google 总部提出的一系列本地化策略并没有得到 Google 总部尤其是两位创始人谢尔盖·布林和拉里·佩吉的认同。Google 总部并不认可其一系列“激

进”的措施，这也成为周韶宁离职的重要原因之一。

从一系列事情中，我们可以看出一些端倪：Google 确实是在不断的创新之中，但是它所需要的人才或许并非是有着非常强的创新能力的人，而是那些有着非常扎实基础的计算机人才。

为什么呢？

这就得从 Google File System 说起……

三、Google File System

当我们使用 Google 进行关键字搜索，享受 Google 强大搜索所带来的便捷的时候；当我们赞叹 Google 地图搜索是如此之精确以至于能够清楚地看到我们所居住的房屋的时候；当我们已习惯使用 Google 个性化主页，让 Google 按照我们的想法随心所欲地提供我们想要的资讯的时候……是否有人静下心来思考这样一个问题：在这样强大的搜索背后，究竟是什么技术在支持呢？是什么系统在管理这样一个已超出我们所能想象的巨大的数据资源呢？

“世界上最出名的搜索引擎公司 Google 所使用的竟然不是数据库！”当听到这个消息的时候，对于我这个以为数据库已经是无所不能的即将进入数据库领域的人来说，确实是一个惊人的消息！数据不用数据库存储，而是用已经被我们所淘汰的文件系统来存储，这是一个让人费解的事情，这是一个让人动摇信念的事情。

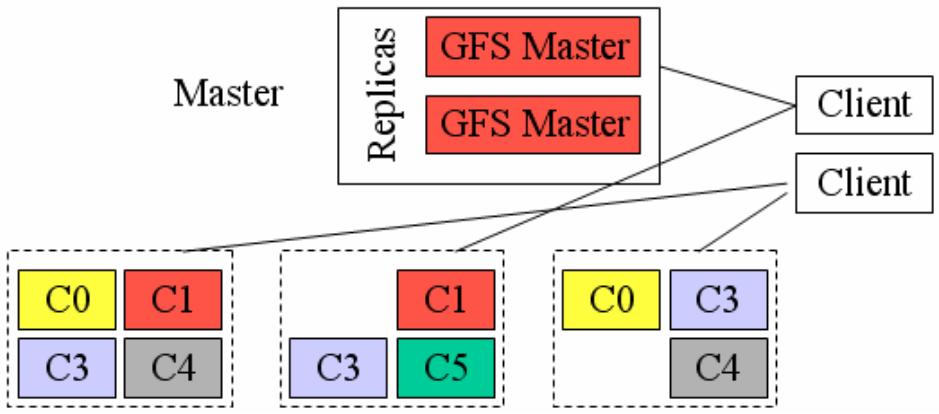
那 Google 为什么不用数据库呢？

Google 说“现有的数据库没法满足我们海量数据存储的需求，即使有，存储及查找代价也会让人无法忍受”。Google 每天所面对的，是成千上万台服务器，是上千 TB 的数据，是每秒数百万的读/写。而且，在这样的情况下，还要实现高效的查询。因此，Google 理直气壮地说：“数据库，No！”

于是 Google 利用极其便宜的 PC 机，来代替昂贵的高性能服务器，并且重新拾起被数据库人即将遗忘的文件系统，成为他们内部的数据管理系统，他们兴奋地说：“Google File System 能够达到我们全部的目标，能够实现高效的存储，具有非常强的容错性！”这些话，不禁让人遥想起 50 年前，当文件系统代替人工管理时，或许也是这般的兴奋。但仅仅十年，数据库就取代了让人激动不已的文件系统，成为数据管理的主要工具。

诚然，在当今网络盛起的时代，面对着 Internet 上数十亿的网页、上百 TB 的卫星图片，传统的关系数据库显得有些吃力甚至都无法管理这样的海量数据。因此，RDBMS 已经无法适应网络时代的需求，需要有新的突破。在 Google 内部，这种突破就是 Google 引以为豪的 Google File System！

那么，Google File System 与以往的文件系统有什么区别吗？



Google文件系统[1]

Google的文件系统是一个大规模的分布式文件系统，它能够处理大规模的分布式数据。它包括控制服务器（Master）和块服务器（Chunkservers），两者之间的信息传输通过GFS的客户端（Client）实现。控制服务器负责管理元数据，它主要存储文件和块的名空间、文件到块之间的映射关系以及每一个块副本的存储位置；块服务器存储块数据，一个典型的块大小为 64M，它通过懒惰算法（Lazy space allocation）来管理存储在它上面的块。控制服务器通过文件系统客户端向块服务器发送数据请求，而块服务器则会将取得的数据直接返回给文件系统客户端[2]。在块服务器中，一个块可能有多个备份，这样做的目的是为了保障数据的安全性，当然，也能够实现负载均衡。

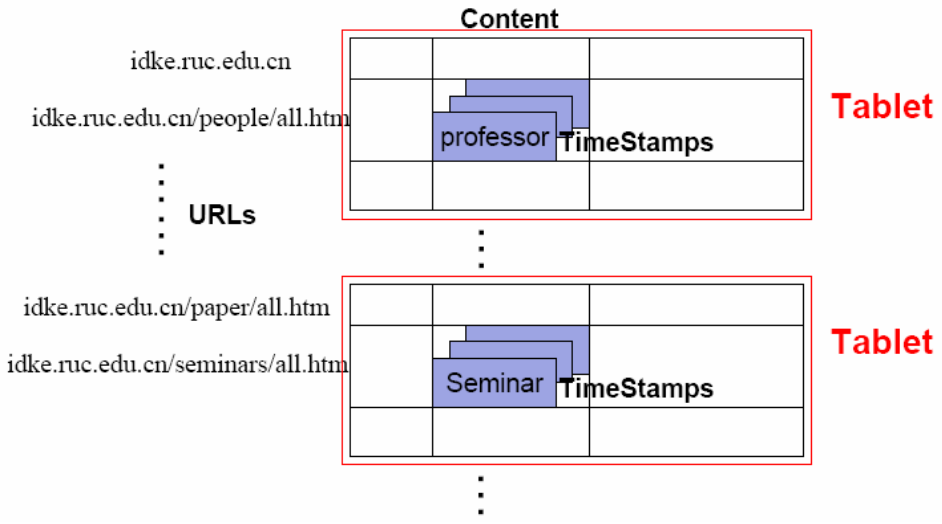
这就是 Google 文件系统的体系结构。然而，光有一个良好的体系结构是不够的，只有具体实现才是硬道理，因此我们需要好好看看建立在 Google File System 上的 BigTable 是如何工作的。

四、BigTable

BigTable，顾名思义，就是一张“大表”，是一张稀疏多维图。Google 于 2004 年初开始研发，到现在它已经运行了两年多，基本上能够满足 Google 的需求：处理海量数据，实现高速存储与查找。

BigTable由行和列组成，每个单元（BigTable Cell）是一个三元组，由行、列、时间戳组成。在一个典型的单元格中，行可以是URLs，列可以是属性/规则，时间戳则是用来标识版本的，它可以在多个备份中，将最新的信息提供给用户[1]。

BigTable 就如它的名字一样，是一张“大表”，以至于为了便于管理，需要将 Bigtable 按照行拆分成 Tablets。如果说 BigTable 是一块布，Tablets 就好像是从这块布上扯下的布条。即使这样，Tablets 也是不小的。每个 Tablets 大约有 100~200M，而每台机器大约会存储 100 个左右的 Tablets。由于 Google 采用的是廉价的 PC 机，而不是使用高端的服务器，因此采用 Tablets，就将 BitTable 化整为零，分布式地存放在各台 PC 机上。



Google’s BigTable

同时，由于采用了 Tablets，也非常容易地实现了负载均衡和快速恢复。如果某台机器的某个 Tablets 经常被访问，则它可以将原来存储在它上面的其它 Tablets 转移到别的机器上，然后专门负责这个 Tablets，而这个 Tablets 也可以完全载入内存，提高访问速度。如果某台机器坏了，不难想象，这台机器上的 Tablets 只需要由其它 100 台机器，每台机器恢复一个

Tablets，系统就重建起来了，因而机器损坏的影响也会降到最低。这点其实是很重要的，因为 Google 采用的是最普通的机器。“如果你买了一台机器，也许用三年也不会有什么太大的问题；但如果你拥有上千台机器，你要做好每天 down 掉一台的准备”。Google 拥有许许多多的普通 PC，因此，每天都会有机器不断损坏，也会又机器不断补充进来，在这种情况下，具有非常好的容错性是很重要的一点。

为了实现对数据的管理和恢复，日志是必不可少的。不难想象，如果每个 Tablets 就有一个日志，那对于这些日志本身的管理就将是一个巨大的工程，所以 Google 选择了同一台机器上的所有 Tablets 共享一个日志的方式。但这种方式虽然减少了日志的的个数，却带来另一个问题：一个日志块将很快被写满，于是系统将非常频繁地开始一个新的日志块。看来，鱼和熊掌确实是不可兼得啊!

同时，由于Tablets采用的是不可修改的（immutable）的SSTables存储方式，因此系统将产生大量的冗余数据，面对这些冗余数据，Google主要采取两种压缩方式进行数据压缩：BMDiff[3]和Zippy。这两种压缩方式，与Gzip和LZW等压缩方式相比，在压缩率上并没有什么优势，但它们都有一个很大的特点，这就是压缩速率和解压速率都非常快，而这正是Google所需要的。能够快速地压缩以节约空间，又能快速地解压获得数据，这对Google来说，远比其它特性要重要得多。

至此，一个建立在 Google File System 上的 BigTable 系统就已呈现在我们面前，从这个系统中，我们可以看到 Google 的核心理念：**低价的机器，高速的处理，大量的冗余，极强的容错**。

采用了 BigTable 后，Google 完全实现了这些理念。

Google File System 虽然很好，但数据库原有的与文件系统相比的优势难道就荡然无存了吗？

五、Google File System 和 Database

从 Google File System 可以看出，虽然 Google File System 有一些自己的特色，有一个不同的应用背景——网络环境，但它仍具有普通文件系统最重要的特点：冗余与单一应用。

当我们数据库人指责文件系统缺陷时，总是忍不住指责其冗余性所带来的坏处：占用存储空间、造成数据不一致性。但这些问题却恰恰在 Google 中都不存在。Google 采用的是便宜的 PC 机，因此，他可以买很多很多机器来解决存储容量问题。至于不一致性，对于 Google 来说并不是一个大问题，他能够通过时间戳给用户提供最新的信息，而且，由于 Google 应用的特殊性，他并不存在修改问题，他的数据一旦写入，就是不可修改的。此外，由于 Google 使用的是廉价 PC，面临着机器随时损坏的可能，使用冗余能够实现系统迅速的恢复。

至于文件系统常见原子性问题、完整性约束、同步性问题等，则由于 Google 目前的主要应用——关键字搜索，使得这些问题对 Google 来说已经不成问题了。

但，难道 Google File System 真的是无懈可击的吗？

不。

我觉得Google现在之所以使用文件系统使用得得心应手与Google现在所从事的应用有关——即 Google 虽然提供了众多服务，但其核心仍然是对大量数据的搜索。这也正是问题的关键所在。

我们知道，数据结构化是文件系统与数据库系统最本质的区别[4]。也就是，数据库中的数据都是结构化的，它能够针对不同的应用；而文件系统则只常常只是针对某一特定应用，但应用发生改变时，需要建立另一个系统。

因此，我们不难猜想到，Google 的成功是有其特殊背景的：在 Web 广泛应用后，出现了大量 html 等不规整的数据，而面对这些数据，又有查询、处理的需求，因此面对这一特定应用，传统数据库已经不再适用，需要有新的系统来适应这种环境。而 Google 选择了专门为这种应用开发了一个系统，这就是 Google File System！

可以预见，在将来应用需求越来越多之后，例如面临的是语义网络或其它更多的扩展，Google File System 很可能就会力不从心。而红极一时的 Google File System，就会像 50 年前的文件系统一样，被其它能够应对更多应用的系统所取代。至于取代它的是不是 Web 上的 Database，还有待观察和实践。

六、Google's Database: Google Base

当然，Google 也会意识到他所存在的危机，也会寻求他自身的突破。

2005 年 10 月，号称是Google的数据库的Google Base在网民的关注下上线了。从首日就有 200 万相关网页的诞生，可以看出Google的影响力。面对Google Base，Google是这样说的："Google Base是Google的数据库，用户可以往里面添加自己的数据，然后Google会让这些数据出现在搜索范围之内"[5]。

Rankings of Top 20 Google Domains Week Ending 5/13/06		
Rank	Name	Market Share
1	Google	79.98%
2	Google Image Search	9.54%
3	Google Mail	5.51%
4	Google News	1.49%
5	Google Maps	0.82%
6	Froogle	0.46%
7	Google Video Search	0.45%
8	Google Groups	0.43%
9	Google Scholar	0.27%
10	Google Book Search	0.25%
11	Google Earth	0.22%
12	Google Desktop Search	0.18%
13	Google Directory	0.10%
14	Google Answers	0.09%
15	Google AdWords	0.07%
16	Google - Local	0.05%
17	Google Finance	0.03%
18	Google Calendar	0.01%
19	Google Talk	0.01%
20	Google Labs	0.01%

在Google今年年中公布的旗下服务访问量中，Google Base 并没有进入前 20 名，看来 Google Base 任重而道远啊

也就是说，Google 在试图利用自己的影响力来收集元数据。这其中涉及到一个满有趣的问题，也就是"先有鸡还是先有蛋"的问题。即到底是先有规整的数据，再有处理这些数据的工具呢；还是先有工具，再把原本不规整的数据转变为规整的数据呢？很显然，Google Base 采取的是前者。因此，虽然有人在说 Google Base 的出现，是对 Ebay、Amazon 等电子商务网站的挑战，是对网络社区的挑战，但我却觉得 Google 的野心远不止如此。我相信，Google Base 是在试图建立一个局部的语义网络。这是一个尝试，是一个对下一代网络的尝试，是一个对建立在 Semantic Web 上的搜索的尝试！它代表了 Google 对未来的摸索与试探，代表了 Google 希望在网络上保持霸主地位的野心与决心！

但这种试探的实际现状如何，我们无法得知，或许只有天知、地知、Goolge 内部的人知；这种试探的结局会如何，我们也无法得知，恐怕只有天知、地知、举头的三尺神明知。

我们只能拭目以待……

七、结语

但无论如何，Google File System 对于我们数据库人来说，是一个机遇，也是一个挑战。因为我们身处在一个瞬息万变的时代，我们面临着即将到来的又一次科技上的巨大变革，我们是要做未来潮流的引导者，还是要做错过机遇的叹息人？

答案，就在我们每一个人的手中。

参考文献：

[1] Jeff Dean：BigTable A System for Distributed Structured Storage
[2] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung：The Google File System 2003
[3] Bentley, Mcllroy: Data Compression Using Long Common Strings. DCC’99
[4] 萨师宣,王珊: 数据库系统概论(第三版). 北京：高等教育出版社, 2005: 9
[5] http://googlebase.blogspot.com/

RFID Data Management

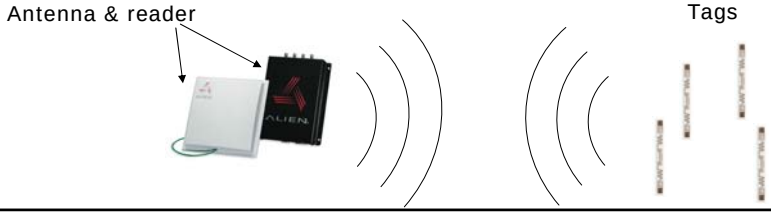
Xiao Pan

Outline

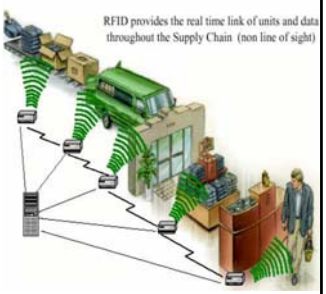
- Introduction to RFID technology
- Characteristics of RFID Data
- Research of RFID data management
  - Beginning of the research
  - Fruits of RFID data management
    - Storage and model of RFID
    - Warehousing and Mining Massive RFID Data Sets
    - Data Cleaning
    - Demo
- Conclusion

What is RFID?

- What is it?
  - RFID(Radio Frequency IDentification) is a technology that allows a reader to detect, from a distance, and without line of sight, a unique electronic product code(EPC) that is transmitted by a tag.
- Tag
  - Attached to items
  - Store item EPC
- Reader
  - Periodical tag scans
  - Records (EPC, time)

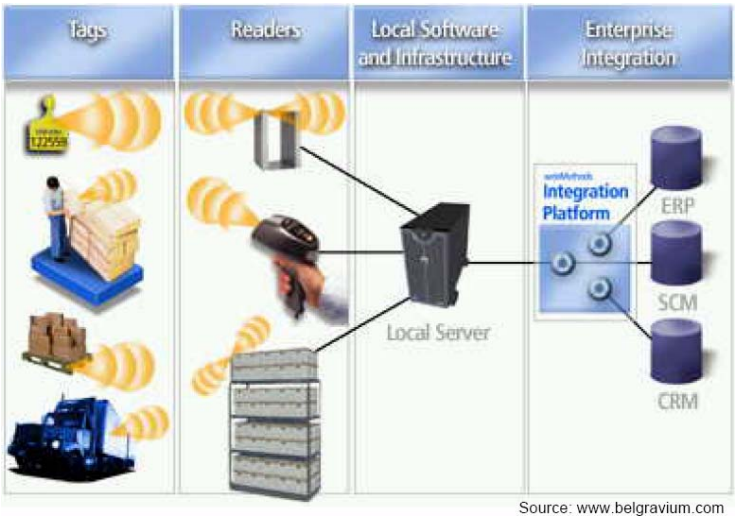


Applications



- Supply chain management
  - for example in retail store
- Healthcare
- Airline luggage management
  - (British airways) Implemented to reduce lost/misplaced luggage
- Library
- Something interesting applications
  - CocaCola
  - Fetch money by mobile phone in ATM machine
- …

RFID System



Source: www.belgravium.com

Outline

- Introduction to RFID technology
- Characteristics of RFID Data
- Research of RFID data management
  - Beginning of the research
  - Fruits of RFID data management
    - Storage and model of RFID
    - Warehousing and Mining Massive RFID Data Sets
    - Data Cleaning
    - Demo
- Conclusion

Characteristics of RFID Data

- Large volume
  - A retail with 3000 stores sells 10,000 items a day per store
    (EPC, location, time)
    Each item 10 traces before leaving store
    How manly tuples it will generate each day?
    10,000×10×3,000=300,000,000 (without redundancy)
  - Walmart is expected to generate 7 terabytes of RFID data per day

-> model and storage of RFID data

- Inaccurate data
  - Noisy data and duplicate readings

-> Data cleaning of RFID data

- Implicit semantics
  - Observations imply location changes, aggregations, and business processes

-> Query and data mining of RFID data

- Temporal oriented

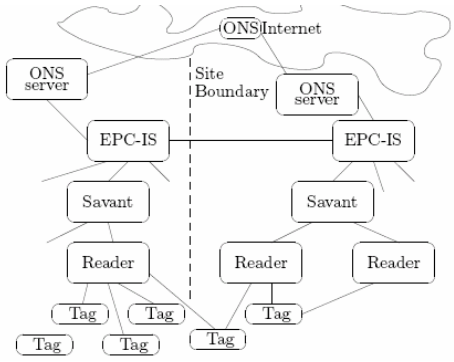
Outline

- Introduction to RFID technology
- Characteristics of RFID Data
- Research of RFID data management
  - Beginning of the research
  - Fruits of RFID data management
    - Storage and model of RFID
    - Warehousing and Mining Massive RFID Data Sets
    - Data Cleaning
    - Demo
- Conclusion

Managing RFID Data_VLDB2004(invited)

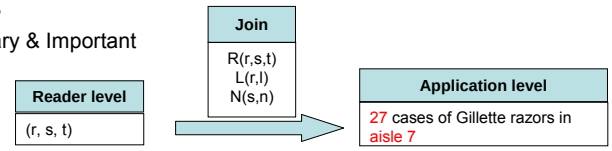
Sudarshan S. Chawathe, Venkat Krishnamurthy etc.

- A layered architecture
  - RFID tags
  - Tag readers
  - Savant/Midderware
    - Mapping the low-level data stream form readers to a more manageable from that is suitable for application-level interactions
  - EPC-IS
    - Most interesting and challenging tasks: Combing business logic with the stream of data emerging from the sensing framework below them
  - ONS
    - Essentially a global lookup service

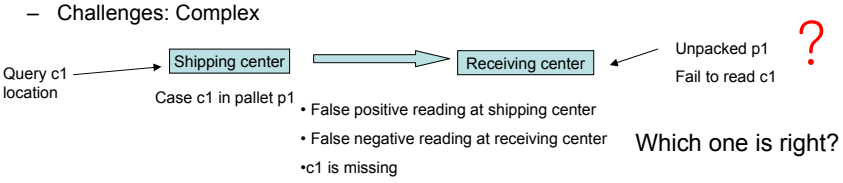


Managing RFID Data_VLDB2004(Contd.)

- Inferences
  - Necessary & Important



  - Challenges: Complex

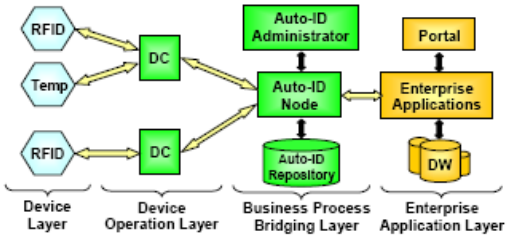


- RFID data management vs. warehousing
  - Analogous task: collecting data, data cleaning etc
  - Differences: Currency of data
    Station-local activities
- Configuration Design
  - Determining number, type, and placement of readers, and the manner connected to other sensors
  - Design choice affects the amount and nature of data that must be stored at other layers

Integrating Automatic Data Acquisition with Business Processes Experiences with SAP's Auto-ID Infrastructure_VLDB2004(invited)

Christof Bornhovd,Tao Lin,Stephan Haller,Joachim Schaper

- Auto-ID infrastructure



- Open Issues
  - Different Qualities of Service
  - Distributed Smart Items Infrastructure
  - Seamless Integration of Environmental Sensors
  - Privacy

Outline

- Introduction to RFID technology
- Characteristics of RFID Data
- Research of RFID data management
  - Beginning of the research
  - Fruits of RFID data management
    - Storage and model of RFID data
    - Warehousing and Mining Massive RFID Data Sets
    - Data Cleaning
    - Demo
- Conclusion

Supporting RFID-based Item Tracking Applications in Oracle DBMS Using a Bitmap Datatype_VLDB2005

Ying Hu,Seema Sundara,Timothy Chorma,Jagannathan Srinivasan

- Observation
  - Groups of items in the same proximity - e.g. on a shelf, on a shipment
  - Groups of items with same property - e.g. Same product
- epc_bitmap_segment Datatype
  - A new type to represent a collection of EPCs with a common prefix

Header 2-bits | EPC_Manager 21-bits | Object_Class 17-bits | Serial_Number 24-bits

0x4AA890001F62C160
.....
0x4AA890001FA0B38E

Len	Suff_len	Prefix	Suff_start	Suff_end	bitmap
64	24	0x4AA890001F	0x62C160	0xA0B38E	101001…00010

With EPC Collections

Store_id	Prod_id	Time	Item_collection
s1	p1	t1	epc11, epc12, epc13, …
s1	p2	t2	epc21, epc22, epc23, …
…	…	…	…

With epc_bitmaps

Store_id	Prod_id	Time	Item_bmap
s1	p1	t1	bmap1
s1	p2	t2	bmap2
…	…	…	…

- Bitmap datatype: multiset of epc_bitmap_segment type

```
CREATE TYPE epc_bitmap IS
TABLE of epc_bitmap_segment;
```

Supporting RFID-based Item Tracking Applications in Oracle DBMS Using a Bitmap Datatype (Contd.)

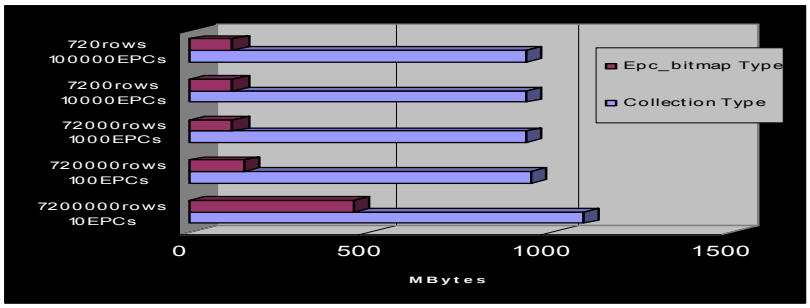
- epc_bitmap Operations
  - Conversion Operations
    epc2Bmap, bmap2Epc, and bmap2Count
  - Pairwise Logical Operations
    bmapAnd, bmapOr, bmapMinus, and bmapXor
  - Maintenance Operations
    bmapInsert and bmapDelete
  - Membership Testing Operation
    bmapExists
  - Comparison Operation
    bmapEqual
- Use of these operations in SQL
  - Query: Determine the items added to a shelf between time t1 and t2

Table Shelf_Inventory

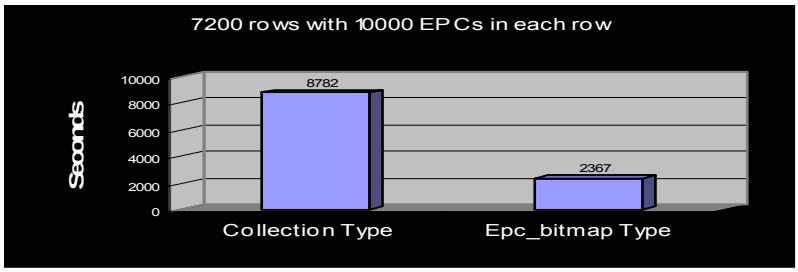
Shelf_id	Time	Item_bmap
sid1	t1	bmp1
sid1	t2	bmp2
…	…	…

```
SELECT bmap2Epc(bmapMinus(s2.item_bmap,s1.item_bmap))
FROM Shelf_Inventory s1, Shelf_Inventory s2
 WHERE s1.shelf_id = <sid1> AND
        s1.shelf_id = s2.shelf_id AND
        s1.time=<t1> AND s2.time=<t2>;
```

Storage Comparison



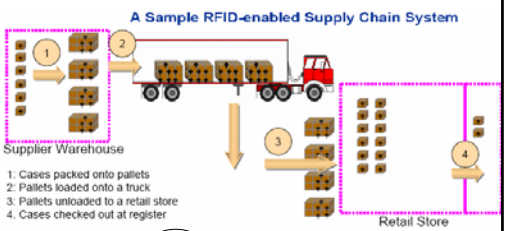
Bulk Load Performance

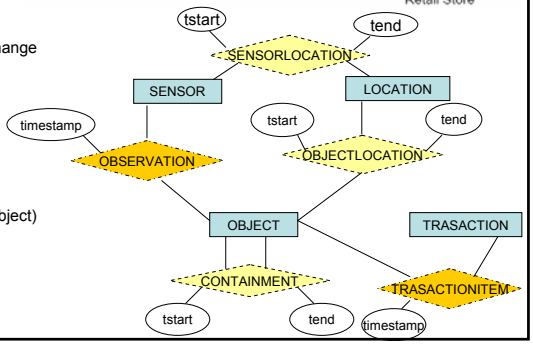


Temporal Management of RFID Data_VLDB2005

Fusheng Wang, Peiya Liu

- Dynamic Relationship ER Model (DRER)
  - Fundamental Entities in RFID Systems
    - Objects
    - Sensors/readers
    - Locations
    - Transactions
  - Two dynamic relationships added
    - State-based dynamic relationship
      - Object location change
      - Object containment relationship change
      - Reader location change
      - two timestamp attributes : Represents the lifespan of a state
    - Event-based dynamic relationship
      - New event
        Observations (reader + object)
        Transacted items (transaction+ object)
      - A timestamp attribute:
        Represents event occurring time





Temporal Management of RFID Data_VLDB2005(Contd.)

- Rules-based RFID Data Transformation

– Rules for location transformation

OBSERVATION("R2", e, t) ->
UPDATE:OBJECTLOCATION(e,"L002", t, "UC")

– Rules for data aggregation

seq(s,"r2",Tseq);OBSERVATION("r2", e, t) ->
INSERT:CONTAINMENT(seq(s,"r2",Tseq),e,t,"UC")

– Rules for data filtering

OBSERVATION(Rx, e, Tx), OBSERVATION(Ry, e, Ty),
Rx <> Ry, within(Tx, Ty, T) ->
DROP:OBSERVATION(Rx, e, Tx)

- Fusheng Wang, Shaorong Liu, Peiya Liu, Bridging Physical and Virtual World: Complex Event Processing for RFID Data Streams, EDBT2006

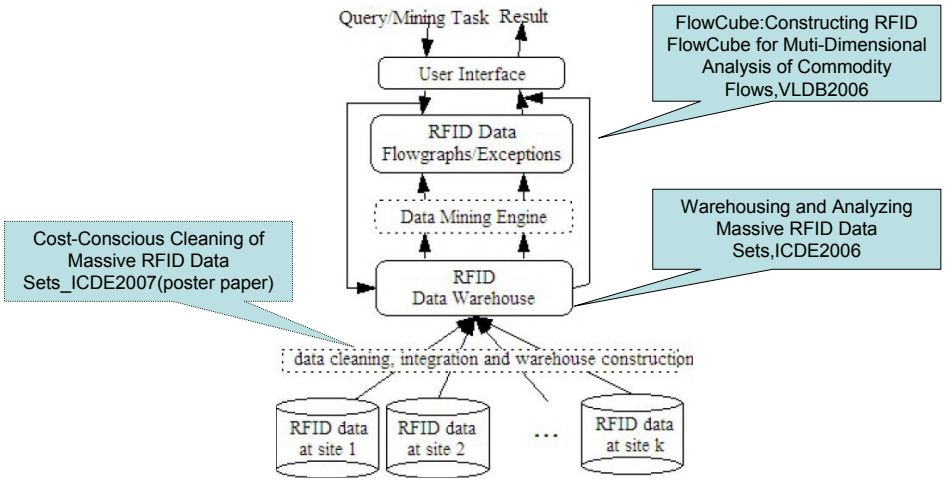
Outline

- Introduction to RFID technology
- Characteristics of RFID Data
- Research of RFID data management
  - Beginning of the research
  - Fruits of RFID data management
    - Storage and model of RFID data
    - Warehousing and Mining Massive RFID Data Sets
    - Data Cleaning
    - Demo
- Conclusion

Warehousing and Mining Massive RFID Data Sets_Keynote for ADMA2006

Jiawei Han

- RFID Warehouse Architecture



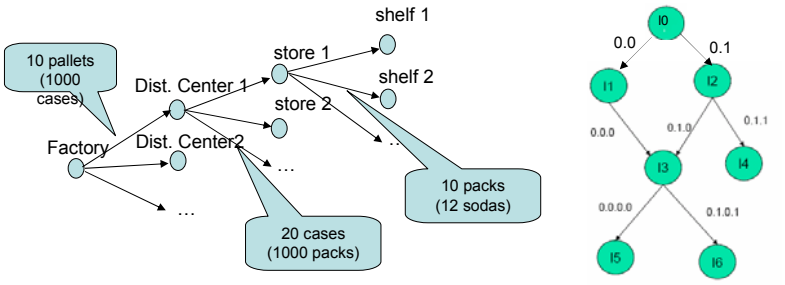
Warehousing and Analyzing Massive RFID Data Sets_ICDE2006

Hector Gonzalez, Jiawei Han, Xiaolei Li, Diego Kabjia

- Why traditional data cube fails?
  - View the cleansed RFID data: fact table - (object epc, location, time in, time out : measure).
  - measure: count - Number of items that stayed at a given location for a given period.
  - Does not consider links within records.
  - Example
    - Get the number of items of product *P* that traveled from the distribution center *L* to stores *U* ?
    - We have the count of product *P* for each location but we do not know how many of those items went from the first location to the second.
    - Hard to get this information.
    - We need a more powerful model capable of aggregating data while preserving its path-like structure.

Warehousing and Analyzing Massive RFID Data Sets_ICDE2006

- Compression Idea: Bulky object movements
  - Objects often move and stay together through the supply chain.
  - If 1000 packs of product P stay together at the distribution center : register a single record for all of them.
  - (GID, location, time_in, time_out:measures).
  - GID is a generalized identifier that represents the 1000 packs that stayed together at the distribution center



Warehousing and Analyzing Massive RFID Data Sets_ICDE2006（Contd.）

- RFID-Cuboid Architecutre
  - **Stay Table:** (GIDs, location, time_in, time_out: measures)
    - Records information on items that stay together at a given location
  - **Map Table:** (GID, <GID1,..,GIDn>)
    - Links together stages that belong to the same path. High level GID points to lower level GIDs
    - If saving complete EPC Lists: high costs of IO to retrieve long lists, costly query processing
  - **Information Table**: (EPC list, attribute1,...,attribute n)
    - Records path-independent attributes of the items, e.g., color, manufacturer, price
- Query: Get the number of items of product *P* that traveled from the distribution center *L* to stores *U* ?
  - GIDs for L <0.0.0>,<0.1.0>
  - GIDs for U <0.0.0.0>, <0.1.0.1>
  - Prefix pairs: p1: (<0.0.0>,<0.0.0.0>)
    p2: (<0.1.0>,<0.1.0.1>)
  - Retrieve stay records for each pair (including intermediate steps) and compute measure

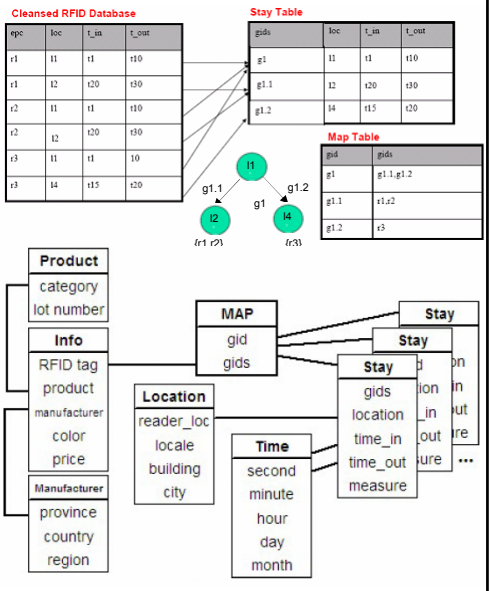
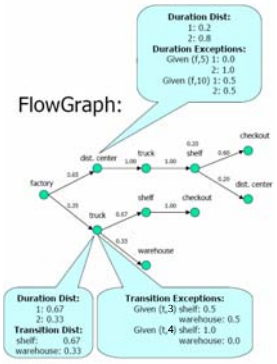


Fig. RFID-Cube Architecture

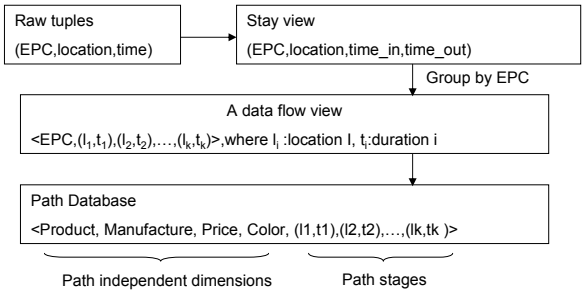
FlowCube:Constructing RFID FlowCubes for Muti-Dimensional Analysis of Commodity Flows_VLDB2006

Hector Gonzalez, Jiawei Han, Xiaolei Li

- Data Flow Analysis:FlowGraph
  - Tree shaped workflow that summarizes the flow patterns for an item or group of items
    - Nodes Locations
    - Edges: Transitions
  - Each node is annotated with:
    - Distribution of durations at the node
    - Distribution of transition probabilities
    - Exceptions to duration and transition probabilities



- RFID data: A Path Database View



Path Database:

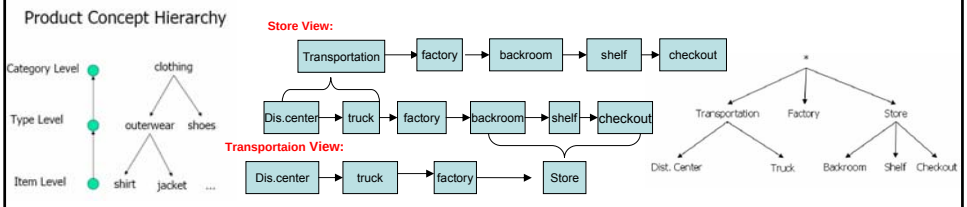
id	product	brand	path
1	tennis	nike	(f,10)(d,2)(t,1)(s,5)(c,0)
2	tennis	nike	(f,5)(d,2)(t,1)(s,10)(c,0)
3	sandals	nike	(f,10)(d,1)(t,2)(s,5)(c,0)
4	shirt	nike	(f,10)(t,1)(s,5)(c,0)
5	jacket	nike	(f,10)(t,2)(s,5)(c,1)
6	jacket	nike	(f,10)(t,1)(w,5)
7	tennis	adidas	(f,5)(d,2)(t,2)(s,20)
8	tennis	adidas	(f,5)(d,2)(t,3)(s,10)(d,5)

FlowCube:Constructing RFID FlowCubes for Muti-Dimensional Analysis of Commodity Flows_VLDB2006(Contd.)

- **FlowCube**
  - Each cuboid has an associated level in the item and path abstraction lattices.
    - Level in the item lattice.
      (product category, country, price)
    - Level in the path lattice.
      (<transportation, factory, backroom, shelf, checkout>, hour)

Path Database:

id	product	brand	path
1	tennis	nike	(f,10)(d,2)(t,1)(s,5)(c,0)
2	tennis	nike	(f,5)(d,2)(t,1)(s,10)(c,0)
3	sandals	nike	(f,10)(d,1)(t,2)(s,5)(c,0)
4	shirt	nike	(f,10)(t,1)(s,5)(c,0)
5	jacket	nike	(f,10)(t,2)(s,5)(c,1)
6	jacket	nike	(f,10)(t,1)(w,5)
7	tennis	adidas	(f,5)(d,2)(t,2)(s,20)
8	tennis	adidas	(f,5)(d,2)(t,3)(s,10)(d,5)

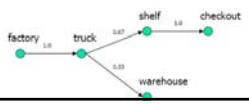


- The measure for each cell in the FlowCube is a FlowGraph computed on the paths aggregated in the cell.

Cuboid for <product type, brand>

cell id	product	brand	path ids
1	shoes	nike	1,2,3
2	shoes	adidas	7,8
3	outerwear	nike	4,5,6

FlowGraph for cell 3



Outline

- Introduction to RFID technology
- Characteristics of RFID Data
- Research of RFID data management
  - Beginning of the research
  - Fruits of RFID data management
    - Storage and model of RFID data
    - Warehousing and Mining Massive RFID Data Sets
    - Data Cleaning
    - Demo
- Conclusion

Issues in Data Cleaning

- False negative reading
  - In this case, RFID tags might not be read by the reader at all while present to a reader
  - Caused by
    - RFID readers capture only 60-70% of all tags that are in the vicinity
    - RF collisions
    - Water or metal shielding
- False positive reading
  - In this case, besides RFID tags to be read, additional unexpected reading are generated
  - Caused by
    - RFID tags outside the normal reading scope of a reader are captured by the reader
    - RFID tags has moved away its vicinity, but reader fails to capture it
    - Unknown reasons from the reader or environment, one of our readers periodically sends wrong IDs
- Duplicate Readings
  - Caused by
    - Tags in the scope of a reader for a long time are read by the reader multiple times
    - Multiple readers are installed to cover larger area or distance, and tags in the overlapped areas read by multiple readers
    - To enhance reading accuracy, multiple tags with same EPCs are attached to the same object, thus generate duplicate readings
- Logical anomalies: tend to be application dependent
  - For example: cycle anomalies

    (e1, t1, r1, back room)
    (e1, t1+2, r2, sales floor)
    (e1, t1+5, r1, back room)
    (e1, t1+9, r2, sales floor)

A Pipelined Framework for Online Cleaning of Sensor Data Streams, ICDE 2006(short paper)

Shawn R. Jeffery, Gustavo Alonso, Michael J. Franklin, Wei Hong, Jennifer Widom

- Extensible Sensor stream Processing (ESP):

  A declarative query-based framework
  - Point

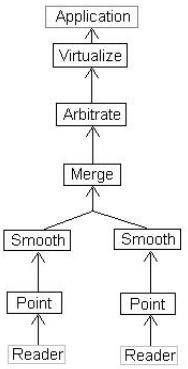
    Operates over a single reading in a receptor stream, filtered individual readings(e.g., obvious outliers)
  - Smooth

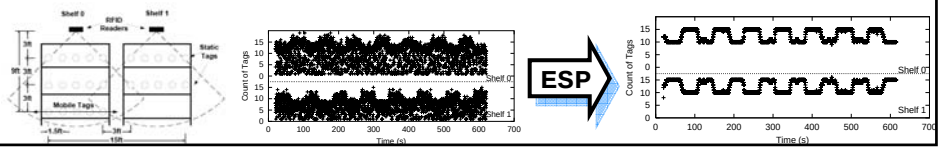
    Granularity defined by applications to correct for missed readings temporally (over one input only); uses aggregate function over the input.
  - Merge

    Granularity specified by the application to correct for missed readings spatially
  - Arbitrate

    Deals with conflicts between different spatial granules
  - Virtualize

    Combine readings from different types of devices and data sources

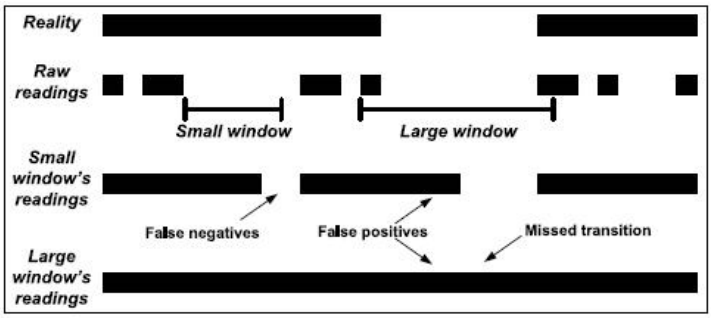




Adaptive Cleaning for RFID Data Streams_VLDB2006

ShawnR. Jeffery, Minos Garofalakis, Michael J.Franklin

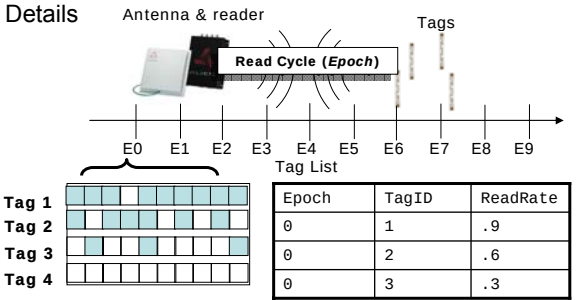
- Window Size for RFID Smoothing



- Solution
  - SMUF(**Statistical Smoothing for Unreliable RFID Data)**
  - Adapt the window size in response to data

Adaptive Cleaning for RFID Data Streams_VLDB2006

- Key Insigh:A Statistical Sampling Perspective
  - RFID data ≈ random sample of present tags,
  - Map RFID smoothing to a sampling experiment
- RFID's Gory Details



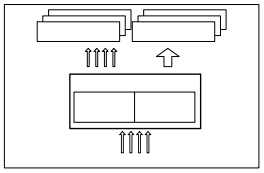
Epoch	TagID	ReadRate
0	1	.9
0	2	.6
0	3	.3

- RFID Smoothing to Sampling

RFID	Sampling
Read cycle (*epoch*)	Sample trial
Reading	Single sample
Smoothing window	Repeated trials
Read rate	Probability of inclusion (p_i)

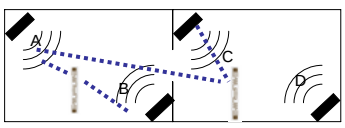
Adaptive Cleaning for RFID Data Streams_VLDB2006

- Per-tag cleaning
  - Completeness $w_i = \left(\frac{1}{p_i^{avg}}\right) * \ln\left(\frac{1}{\delta}\right)$
  - Transitions $|\underbrace{|S_i|}_{\text{\# observed readings}} - \underbrace{w_i * p_i^{avg}}_{\text{\# expected readings}}| > 2\sqrt{\underbrace{w_i * p_i^{avg} * (1 - p_i^{avg})}_{\text{Is the difference "statistically significant"?}}}$
- Mechanisms for:
  - Per-tag and multi-tag cleaning



- Not fit for

Two rooms, two readers per room



Efficiently Filtering RFID Data Streams_VLDB-CleanDB2006

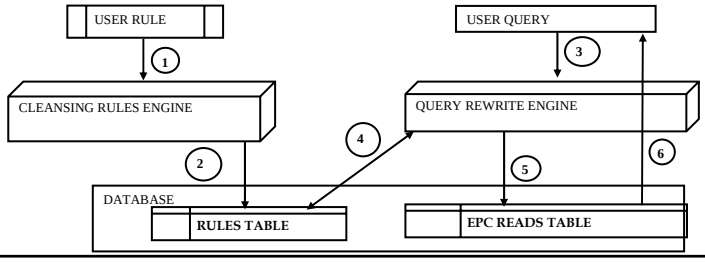
Yijian Bai, Fusheng Wang, Peiya Liu

- Main ideas
  - False positive readings
    - The noise readings are readings with count of distinct tag EPC values below a noise threshold
    - Essentially performs the following operations: within any time window with size of window_size ,
      - if the count of the readings with same tag EPC values appears equal to above threshold, then the observed EPC value is not noise and needs to be forwarded for further processing;
      - otherwise the reading is discarded.
  - Duplicate readings
    - If a reading is within max_distance in time from the previous reading with the same key, then this reading is considered a duplicate.
    - Otherwise, it is considered a new reading and is output
- Good points: preserve the original order
- Bad points
  - Not give the cleaning method for false negative reading
  - Don't mention how to confirm the threshold

A Deferred Cleansing Method for RFID Data Analytics_VLDB2006

Jun Rao, Sangeeta Doraiswamy, Hetal Thakkar, Latha S.Colby

- Motivation
  - Conventional approach to cleansing is eager
    - Before loading into a warehouse (ETL)
    - Clean once, reuse at query time
    - Typically reducing data size
    - Best strategy if applicable
  - Sometimes eager cleansing is not applicable
    - Don't know how to clean until analyzing the data
    - More than one cleaned version (app-dependant anomalies)
    - Law enforcement (pharmaceutical e-pedigree tracking)
  - Propose deferred cleansing
    - Load everything
    - Clean at query time
    - Has runtime overhead
    - Complementary to eager cleansing
- Overview of This Approach



Cost-Conscious Cleaning of Massive RFID Data Sets_ICDE2007(poster paper)

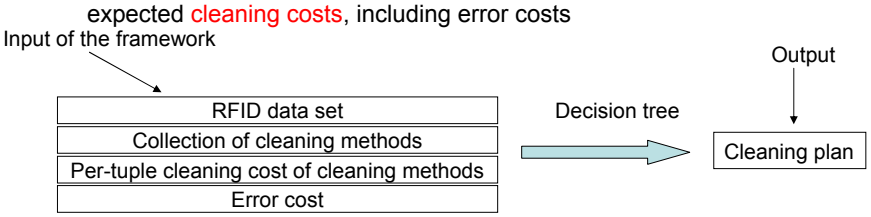
Hector Gonzalez, Jiawei Han, Xuehua Shen

- Motivation

  Existing cleaning techniques have focused on the accurate methods, but have disregarded the very high cost of cleaning in a real application

- Contribution: propose a cleaning framework

  Identify the conditions under which a specific cleaning method or a sequence of cleaning methods should be applied in order to minimize the expected cleaning costs, including error costs



Outline

- Introduction to RFID technology
- Characteristics of RFID Data
- Research of RFID data management
  - Beginning of the research
  - Fruits of RFID data management
    - Storage and model of RFID data
    - Warehousing and Mining Massive RFID Data Sets
    - Data Cleaning
    - Demo
- Conclusion

DEMO(1)

- HiFi(High Fan-in System)
  - Database Research Group of UC(University of Californ) Berkeley
  - HiFi is a distributed stream processing system designed to support large-scale sensor-based network
  - Architecture
    - Reader/Sensor
    - Mid-tier devices
    - Host Computer
  - HiFi: A Unified Architecture for High Fan in System, VLDB2004 demo

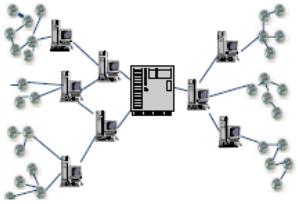
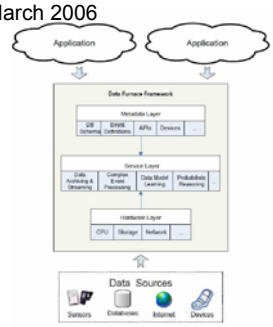


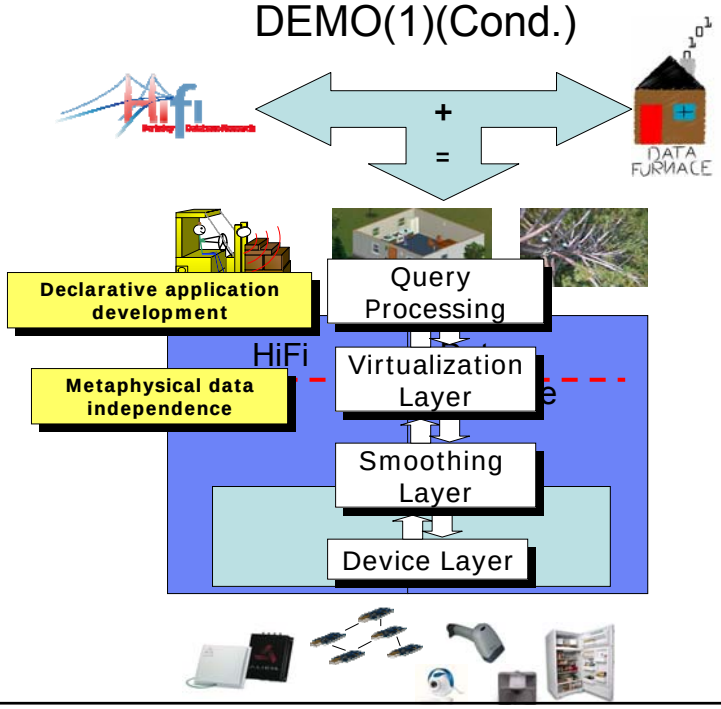
Figure 1 - A High Fan-in Architecture

- Data Furnace
  - Intel Research & UC Berkeley
  - Data management for pervasive applications
  - Probabilistic data a first-class citizen
  - Event language-based
  - Probabilistic Data Management for Pervasive Computing: The Data Furnace Project. IEEE Data Engineering Bulletin, Vol. 29, No. 1, March 2006



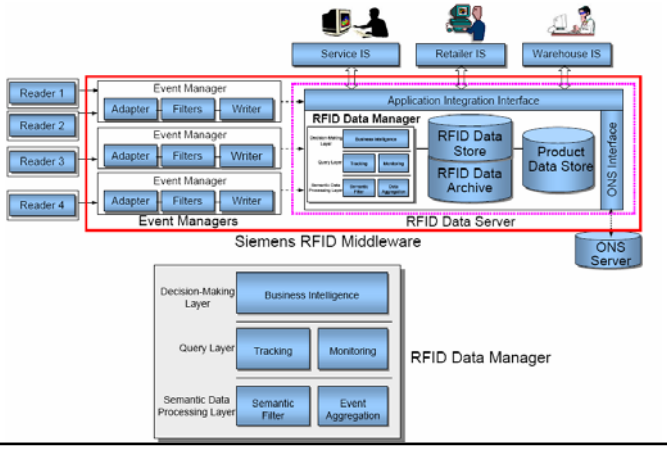
The Data Furnace Architecture

DEMO(1)(Cond.)



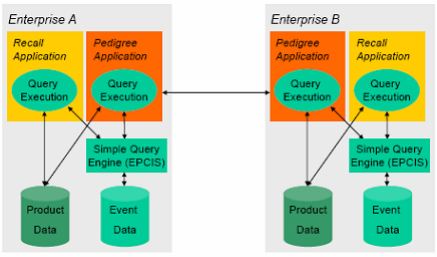
DEMO(2)

- Simens RFID Middleware
  - Develop and demonstrate at Siemens Corporate Research
  - Applied in health care to increase healthcare safety and workflow efficiency
  - Temporal Management of RFID Data，VLDB2005



DEMO(3)

- Theseos
  - IBM Almaden Research Center
  - A query engine on top of sovereign, distributed RFID databases, to facilitate traceability query processing
  - Theseos:A Query Engine for Traceability across Sovereign, Distributed RFID Databases,ICDE2007 demo



EPCglobal Approach to Traceability

Outline

- Introduction to RFID technology
- Characteristics of RFID Data
- Research of RFID data management
  - Beginning of the research
  - Fruits of RFID data management
    - Storage and model of RFID
    - Warehousing and Mining Massive RFID Data Sets
    - Data Cleaning
    - Demo
- Conclusion

Conclusion

- What is RFID?
- Some fruits of research on RFID data management
  - Storage and model of RFID data
  - Warehousing and Mining Massive RFID Data Sets
  - Data cleaning of RFID data
  - Existing demo
    - HiFi and Data Furnace
    - Simens RFID Middleware
    - Theseos
- What can we do??

发表论文精选

Deep Web 数据集成研究

(Deep Web Data Integration)

Web Database Integration

Wei Liu, Xiaofeng Meng

In Proceedings of the Ph.D Workshop in conjunction with VLDB 06 (VLDB-PhD2006), Seoul, Korea, September 11, 2006

DEEP WEB 数据集成中的实体识别方法

凌妍妍，刘伟，王仲远，艾静，孟小峰

计算机研究与发展, 卷 43(增刊):46-53,2006．(第 23 届中国数据库学术会议,广州.)

In Proceedings of the Ph.D Workshop in conjunction with VLDB 06 (VLDB-PhD2006),
Seoul, Korea, September 11, 2006

Web Database Integration

Wei Liu
School of Information
Renmin University of China
Beijing, 100872, China
gue2@ruc.edu.cn

Xiaofeng Meng
School of Information
Renmin University of China
Beijing, 100872, China
xfmeng@ruc.edu.cn

ABSTRACT

More and more accessible databases are available in the Web. In order to provide people a unified access to these Web databases and achieve information from them automatically, a comprehensive solution for Web database integration is proposed in this paper. After summarizing the research status in this area, the works which are the focus of my PhD thesis are presented.

1. INTRODUCTION

With the rapid development of Web, more and more accessible databases are available in the Web. Such databases are usually called Web database (or WDB in short) by researchers. From this angle, the Web can be divided into two parts: Surface Web and Deep Web. The Surface Web refers to the static Web pages which can be crawled and indexed by popular search engines, while the Deep Web refers to the contents stored in Web databases and published by dynamic Web pages.

The abundant information stored in Web databases is "hided" behind the query interfaces in Web pages. This means that the main approach people access Web databases is through their query interfaces. Figure 1 gives the query interface provided by Amazon which is a very popular e-commerce Web site.

According to the survey[1] released by UIUC in 2004, there are more than 300,000 Web databases and 450,000 query interfaces available at that time, and the two figures are still increasing quickly. Besides the scale of Web databases, the contents in Web databases are spanning well across all topics. Some Deep Web portal services provide Deep Web directories which classify Web databases in some taxonomies. For example, CompletePlanet[2], the biggest Deep Web directory, has collected more than 7,000 Web databases and classified them into 42 topics. Combing the above two aspects, we can conclude that theses Web databases are just like a huge repository and provide people a great opportunity to get their desired information.

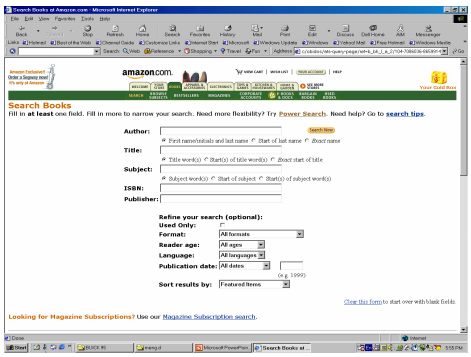


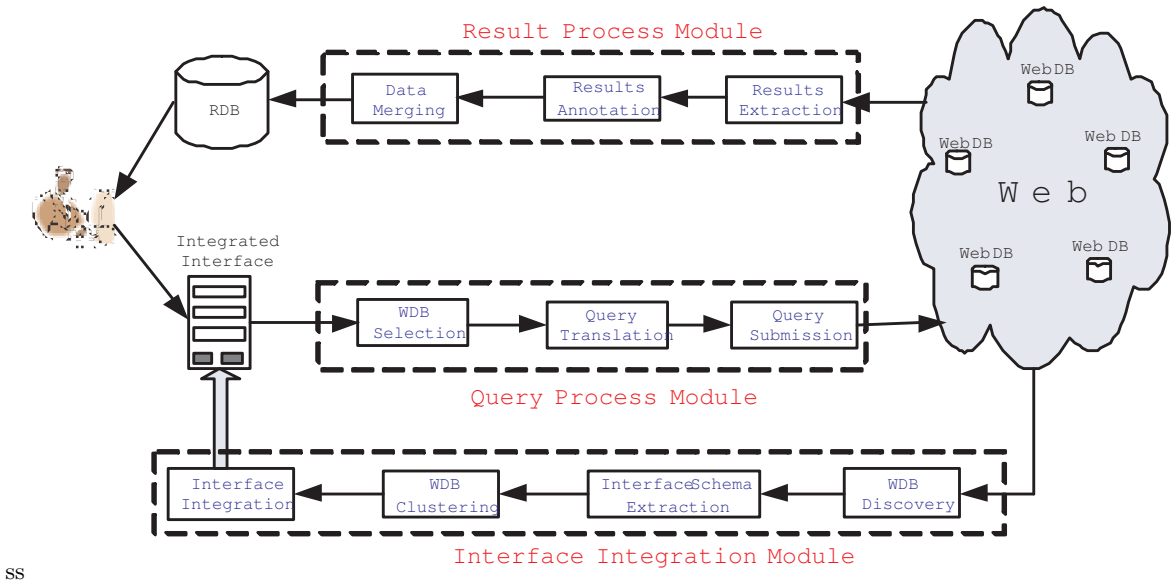
Figure 1: The query interface of Amazon

With proliferation of Web databases, it is not only an opportunity but also a challenge for people. At present, people access to Web databases mainly by manual approach, and his will bring an overhead problem.

Here is an example to explain the problem. Suppose Jane wants buy a book on Java. There are several tasks she has to complete. First, she must find the Web sites which sell books. If she wants save money, more Web sites are needed to compare. Second, she fills the query interfaces with an appropriate query (for example, fill book title with "think in java") and submits them. Third, when the Web pages contain query results returned (these Web pages are called response pages generally), she browses them in turn and chooses the best book. The whole process is time-consuming. Maybe Jane will spend half a day for this. Therefore, the challenge of manual approach is people often have difficulties in first finding the right sources and then querying over them.

It is impending and compulsory to integrate Web databases and to provide people a unified access to them and achieve information automatically. Web databases integration can be considered as the heterogeneous data source integration in Web context. The traditional heterogeneous data source integration generally focuses on the heterogeneity and autonomy of data sources. According to my investigation, Web databases also have four distinct characteristics which are different to other heterogeneous data sources:

**Proceedings of the VLDB2006 Ph.D. Workshop
Seoul, Rep of Korea, 2006**



ss

Figure 2: A comprehensive solution for Web database integration

- ***Scale:*** There are myriads of Web databases in Web, and even under a special topic the quantity of Web databases is still striking.
- ***Dynamic:*** First, Web databases are very sparsely distributed in Web, and they appear and disappear endlessly. So searching for appropriate Web databases in Web is really like looking for a few needles in a haystack. Second, the contents in Web databases are usually updated frequently. Especially in some topics, such as airline and job, everyday a batch of new contents will be added to Web databases and the outdated part will be removed. So the information in Web databases is "ever" not "forever" to you.
- ***Access through query interfaces:*** Due to the peculiar access approach, the schema of a Web database can not be captured directly. We can only infer the schema from their query interfaces and response pages.
- ***Heterogeneity:*** The query interfaces and response pages are designed by different persons and there are no design standards to follow. Even in the same topic, the query interfaces and response pages are often very dissimilar.

In a word, the research on Web database integration aims to help people make use of the abundant information in Web databases effectively and efficiently. But due to the distinct characteristics of Web databases, there are many challenging research issues in this area.

My PhD thesis is focusing on building a Web database integration system and addressing several challenging issues in this area. In this paper a comprehensive solution for Web database integration is presented and my current and future research works in this area is indicated .

There is a fact which should not be neglected. Some Web sites provide Web Services for their Web databases, and people can use a customized program to access Web databases. But this approach has two limitations: first, only a small portion of Web sites provide Web Services for their Web databases; second, this approach must depend on a customized program, and this is not an easy thing for common users. So in this paper we focus on the popular approach of accessing Web databases through the query interfaces in Web pages.

The rest of this paper is organized as follows. Section 2 gives the solution for Web database integration; Section 3 summarizes the research status in this area; Section 4 presents the works we are focusing now and will focus in the future; Section 5 is the conclusion.

2. A SOLUTION FOR WDB INTEGRATION

In this section, a comprehensive solution for Web database integration is proposed, which is the pursuit in my PhD track. Figure 2 is the architecture of the solution. This solution includes three primary modules: integrated interface generation module, query processing module and results processing module.

Integrated interface generation module: Produce an integrated interface over the query interfaces of the Web databases to be integrated. There are four components in this module. The functions of them are described as following:

- *Web database discovery:* Search Web sites which have Web databases behind, and identify the query interfaces among the Web pages in these Web sites.
- *Query interface schema extraction:* Extract the attributes in query interfaces (such as "Title" and "Author" in Figure 1), and the meta-information about each attribute (such as value type, default value, etc).
- *Web database clustering by topic:* Cluster all discov-

ered Web databases into different groups. The Web databases in each group belong to the same topic.

- *Interface integration:* Given the Web databases in the same topic, merge the same semantic attributes in different query interfaces into a global attribute, and finally form an integrated interface.

Query processing module: Process a user's query filled in integrated interface, and submit the query to each Web databases. There are three components in this module. The functions of them are described as following:

- *Web database selection:* Select appropriate Web databases for a user's query in order to get the satisfying results at minimal cost.
- *Query translation:* Try to translate the query on integrated interface equivalently into a set of local queries on the query interfaces of Web databases.
- *Query submission:* Analyze the submission approaches of local query interfaces, and submit each local query automatically.

Result processing module: Extract the query results achieved from Web databases, and merge the results together under a global schema. There are three components in this module. The functions of them are described as following:

- *Result extraction:* Identify and extract the pure results from the response pages returned by Web databases.
- *Result Annotation:* Append the proper semantics for the extracted results.
- *Result merging:* Merge the results extracted from different Web databases together under a global schema.

These components work together and make up of a comprehensive solution for Web database integration. It's not difficult to found that there are dependency relationships between them. Figure 2 has disclosed such dependency relationship. For example, query processing module depends on integrated interface generation module (high level), interface integration depends on Web database clustering (low level). So the quality of the implementation of a component will affect the next component greatly.

In fact, each component can be considered as a research issue itself. In order to build a practical Web database integration system, these issues must be solved well in theory first. In Section 3, the research status in this area will be discussed.

3. RESEARCH STATUS IN THIS AREA

Until now, large numbers of efforts are devoted to this area. Due to the space limit, the related works can not be discussed comprehensively and in detail. We only discuss them summarily according to the issues they address, and we also give the representative works.

Unfortunately, the development of research in this area is uneven very much though the great efforts have been done. Several issues have been already addressed well and are mature enough we can resort to (developed issues), some issues is developing and need be researched deeply (developing issues), and some issues have not been touched yet (undeveloped issues). We summarize the research status according to the development of these issues.

3.1 Developed Issues

Interface integration It has received enough attention, and several effective approaches[3][4][5][6] are proposed solve this problem. These approaches match attributes of query interfaces by exploiting the semantic similarity between labels as well as that between data instances.

Query interface schema extraction In order to understand query capabilities a query interface supports, [7] transforms query interfaces into a visual language, and develops a 2P grammar and a best-effort parser to realize a parsing mechanism.

3.2 Developing Issues

Besides introducing the current approach for developing issues, the shortcomings of them are pointed out at the same time.

Web database discovery [9] proposed a strategy does that by focusing the crawl on a given topic and choosing links to follow within a topic that are more likely to lead to pages that contain query interfaces. It can not assure the quantity of discovered Web databases. [10] use automatic feature generation to describe candidates and C4.5 decision trees to detect query interfaces. It can not differentiate the query interfaces of search engines from that of Web databases.

Web database clustering [11] performs the clustering based on the features available on the interface page. [12] proposed an objective function, model-differentiation, to compute the probability which topic a query interface belongs to. Their accuracy depends on the schema information of query interfaces, so they are not good at dealing with the query interfaces with simple schema.

Result extraction There are lots of approaches proposed to address this issue. Most of them[13][14][15] first transform the response page into a HTML tag tree, then identify and extract data records or data items by analyzing tree structure and tag information. They can only deal with the Web pages designed by HTML language, so it is a latent shortcoming with the development of Web.

Result annotation This problem is often solved during the process of Result extraction. [17] find the proper the annotation of an extracted data item in the response page by some heuristic rules. They are very effective if a data item really has its annotation in the response page. But they can not ensure all data items get their annotations.

Entity identification Entity identification is one of the key components of data merging. Several approaches have been proposed to solve this problem. For example, [16] applies a set of domain-independent string transformations to compare the entities' shared attributes in order to identify matching entities. All current approaches assume that they have

achieved the well-build schema match between Web databases, but schema match in Web context have not been solved yet.

3.3 Undeveloped Issues

The undeveloped issues include *Web database selection*, *Query translation*, and *Data merging*. These issues have been well studied in some contexts(such as data warehouse), but there have not been approaches proposed to address these issues in the context of Web database integration, and they are compulsory in Web database integration.

Among these developing and undeveloped issues, *Entity identification*, *Result extraction* and *Web database selection* are in my PhD track at present and in the future, which are discussed in Section 4.

4. SEVERAL RESEARCH WORKS

In this section, several research works are proposed for discussion, which are being done at present and will be done in future.

4.1 Entity Identification among Web Databases

Entity identification is a key operation in integrating data from multiple sources. This issue has been well studied for years. As discussed in Subsection 3.2, though several solutions have already been proposed for Web databases, all of they are based on such assumption that the schema match between Web databases has been built well. As well known, due to the poor structure of Web pages, schema match in Web context is a very hard work, and there is still not automatic solution for it.

So we are trying to find a way to implement entity identification between Web databases without the help of schema match. Our basic consideration is described as following. We do not try to analyze the structure (or schema) of data records in response pages. Instead, given two Web databases A and B, each data record from A or B is considered as a text document. We judge whether data record a (from A) and data record b (from B) by comparing the text similarity of them. Obviously, it is very naive to compute the text similarity of two data records directly, and the accuracy is also not satisfying in our test. The reason is that, the importance of every part in a data record is different, and there is much noise information in a data record (for example, the words "author" and "price" often appear in the book data records). In order to make the similarity of a and b more reasonable (ideally, if a and b refer to a same entity, and a and c do not, then the similarity of a and b must be bigger than that of a and c), our approach is implemented as following:

1. filter the noise information from a and b as possible;
2. segment a into several blocks, and each block of a is formulated into a query for b;
3. compute the similarity of each block and b;
4. assign an appropriate weight for the similarity of each block and b, and sum up them;
5. judge whether a and b refer to a same entity according to the whole similarity.

At present, we are engaging to find an effective algorithm to train the weights and threshold of the whole similarity by a small set of sample data records pairs. A data record pair is two data records from different Web databases, and they refer to a same entity. The algorithm is now being detailed. The primary experiment result is very satisfying under the book topic. Further, the experiments under other topics (car, estate, etc.) will be done.

4.2 Vision Based Result Extraction

Most current approaches extract the results from response pages based on HTML language. But they have several inextirpable limitations. First, besides HTML, some other languages, such as XML and XHTML, have been introduced design Web pages. Second, HTML is still evolving. New versions of HTML will be proposed in the future, and new tags may appear and applied continuously. Third, as more and more web pages use more complex JavaScript and CSS to influence the structure of web pages, the applicability of the existing solutions will become lower. Fourth, if HTML is replaced by a new language in the future, then previous solutions will have to be revised greatly or even abandoned, and other approaches must be proposed to accommodate the new language.

Based on such motivations, it is important to find an approach which is vision based and language independent. In current phrase, we only aim at the response pages with multiple data records. Our basic idea is that, though the data records in a response page are different on the contents, they are similar on the appearance. The following is the implementation we are engaging in:

1. achieve the vision information (such as the font of a text, the size of an image, and their location in the Web page) by accessing the program interface of Web browser;
2. build a vision based block tree by VIPs[18] algorithm. A data record is composed by one or more blocks in the vision based block tree. So result extraction here is to find these blocks and judge which blocks compose a data record.
3. locate the data region (the region contains all data records in a response page) in the vision based block tree.
4. find the boundaries of all data records by computing the vision similarity of blocks in the vision based block tree.

The primary experiment has indicated that this approach is not only HTML language independent, but also very suit for extracting information-rich data records.

4.3 Web Database Selection

There are myriads of Web databases in the Web. So maybe a lot of Web databases are integrated under a topic. If a user submits a query on the integrated interface and the query is dispatched to all the Web databases integrated, it will be time-consuming and overhead to process all the returned results, especially data cleaning and deduplication. In most

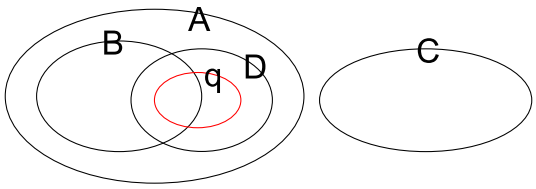


Figure 3: An example for Web Database Selection

cases, we only need select several ones among them to get the satisfying results. So *Web Database Selection* aims to select appropriate Web databases for a given user's query on integrated interface, which can help users get their desired results at the lowest cost.

In order to judge whether a Web database should be selected to answer a given query, there are two aspects must be considered. One is the pertinency of the Web database and the given query; the other is the query capability of the query interface of the Web database. The following gives some our considerations about the two aspects.

The prerequisite of selecting a Web database is it is pertinent to the given query. Extremely, it is meaningless to query a Web database if it does not has any useful information for the query. Figure 3 gives an example to illustrate this. Suppose A, B, C, and D are four Web databases, and q is a query to them. Where the size of A, B, C and D is the quantity of data records in them, the size of q is the quantity of data records satisfies q. Instinctively, C does not satisfy q at all, B satisfies q partly, A and D can satisfy q completely, but at last D is the best selection compared with A. So we need achieve the features of Web databases in advance. The features of a Web database include the size, the update ratio, the distribution on each attribute, etc. Because we can only access a Web database through its query interface, it is impossible to understand a Web database directly. The challenge is how to obtain the features by the query interface only. In the future, we want to design a sample records retriever to address this problem. Sample records retriever is a tool that can obtain a small set of data records which are distributed evenly in the Web database. We can profile the Web database by analyzing the obtained data records. Sample records retriever should have two components: query interface analyzer and query generator. Query interface analyzer is to obtain the necessary information of each attribute; query generator produces a set of smart queries according to the information obtained by query interface analyzer.

The query interfaces are often different about the query capability among Web databases, and this will influence the accuracy of a query. For example, in the book topic, a query on the integrated interface is "title=java and price<20$". If the query interface of a Web database contains both the two attributes , it can answer the query accurately. But if it only contains the attribute "title" or "price", then the results returned from the Web database will contain quite many data records which do not satisfy the query. So the challenge tasks are how to how to make the returned results be satisfying(for example, the minimal superset or maximal subset of the query).

5. CONCLUSIONS

With the rapid increasing of Web databases, it is impending to integrate these Web databases and provide people a unified access to them and achieve information automatically. In this paper, a comprehensive solution for Web database integration is proposed. There are a number of components in the solution, and each of them is also a research issue in this area. After summarizing the research statuses of the issues in this area, we introduce the issues which are being focused on now and will be addressed in the future. In conclusion, the focuses of my PhD thesis are building a Web database integration system and addressing several issues in this area.

6. REFERENCES

[1] K. C. Chang, B. He, C. Li, M. Patel, Z. Zhang. Structured Databases on the Web: Observations and Implications. SIGMOD Record 33(3): 61-70 (2004).

[2] http://www.completeplanet.com/.

[3] B. He, K. C. Chang. Statistical Schema Matching across Web Query Interfaces. SIGMOD Conference 2003: 217-228.

[4] H. He, W. Meng, C. T. Yu, Z. Wu. WISE-Integrator: An Automatic Integrator of Web Search Interfaces for E-Commerce. VLDB Conference 2003: 117-128.

[5] W. Wu, A. Doan, C. T. Yu. WebIQ: Learning from the Web to Match Deep-Web Query Interfaces. ICDE Conference 2006.

[6] E. Dragut, W. Wu, A. P. Sistla, C. T. Yu, W. Meng. Merging Source Query Interfaces on Web Databases. ICDE Conference 2006.

[7] Z. Zhang, B. He, K. C. Chang. Understanding Web Query Interfaces: Best-Effort Parsing with Hidden Syntax. SIGMOD Conference 2004: 107-118.

[8] H. He, W. Meng, C. T. Yu, Z. Wu. Automatic extraction of web search interfaces for interface schema integration. WWW Conference 2004: 414-415.

[9] L. Barbosa, J. Freire. Searching for Hidden-Web Databases. WebDB 2005: 1-6.

[10] J. Cope, N. Craswell, D. Hawking. Automated Discovery of Search Interfaces on the Web. ADC Conference 2003: 181-189.

[11] Q. Peng, W. Meng, H. He, C. T. Yu. WISE-cluster: clustering e-commerce search engines automatically. WIDM 2004: 104-111.

[12] B. He, T. Tao, K. C. Chang. Clustering Structured Web Sources: A Schema-Based, Model-Differentiation Approach. EDBT 2004: 536-546.

[13] B. Liu, R. L. Grossman, Y. Zhai. Mining data records in Web pages. KDD Conference 2003: 601-606.

[14] Y. Zhai, B. Liu. Web data extraction based on partial tree alignment. WWW Conference 2005: 76-85.

[15] H. Zhao, W. Meng, Z. Wu, V. Raghavan, C. T. Yu. Fully automatic wrapper generation for search engines. WWW Conference 2005: 66-75.

[16] S. Tejada, C. A. Knoblock, S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. KDD Conference 2002: 350-359.

[17] J. Wang, F. H. Lochovsky. Data extraction and label assignment for web databases. WWW Conference 2003: 187-196.

[18] D. Cai, S. Yu, J. Wen, W. Ma. Extracting Content Structure for Web Pages Based on Visual Representation. APWeb Conference 2003: 406-417.

计算机研究与发展, 卷43(增刊):46-53,2006 .（第23届中国数据库学术会议,广州.）

DEEP WEB 数据集成中的实体识别方法

凌妍妍 刘伟 王仲远 艾静 孟小峰

（中国人民大学信息学院 北京 100872）

（lingyy@ruc.edu.cn）

摘 要 互联网上存在着大量可访问的Web数据库，不同Web数据库之间存在着内容上的重叠。来自不同Web数据库的记录虽然在网页上的表现形式不同，但是可能描述的是同一实体。因此实体识别是 Deep Web 数据集成中数据合并过程里一个必不可少的环节，而且是一个很具有挑战性的工作。本文对这个问题进行了深入的探讨，提出了一种新颖的方法自动完成实体识别，该方法克服了传统的实体识别工作以模式匹配为前提的弊端，并且与领域无关。实验表明，这种方法在 Deep Web 环境下可以达到相当高的准确性。

关键词 Deep Web; Web数据库; 实体识别; 数据合并

中图法分类号 TP391

Entity Identification for Deep Web Data Integration

Ling Yan-Yan, Liu Wei, Wang Zhong-Yuan, Ai Jing and Meng Xiao-Feng

(School of Information, Renmin University of China, Beijing, 100872)

Abstract Nowadays, growing number of Web Databases emerge from the web with their contents duplicated. Two or more instances from different sources may refer to a single entity in the real world, though they are presented variously on WebPages. Therefore, entity identification is a crucial step in Web Databases integration but it's also a challenging task. In this paper, we have probed into this issue and proposed a novel automatic approach which is domain independent. Unlike traditional approaches, our approach is implemented without schema matching. The intensive experiments on real web sites show that the proposed approach can achieve high accuracies.

Key words Deep Web; Web databases; Entity identification; Data merge

1. 引言

随着 Web 飞速发展，其所蕴含的信息量也在急剧增长。整个 Web 按照信息深度的不同，可分为 Surface Web 和 Deep Web 两大类。根据 Brightplanet 在 2000 年发布的调查[1]，Deep Web 中包含的信息量超过 Surface Web 上千倍，而且这个比例仍在持续地上升。UIUC 大学在 2004 年的调查[2]对整个 Deep Web 的规模作了一次估计，结果表明目前 Deep Web 中可访问的 Web 数据库的数量超过了 45 万个。

为了能够有效利用 Deep Web 中丰富的信息，建立 Deep Web 数据集成系统成为了当前最迫切的需求。由于 Web 数据库的异质性和自主性，对从各个 Web 数据库中抽取结果的合并是一项十分具有挑战性的工作。为了对抽取结果进行清洗和去重，实体识别则是数据合并过程中的一个必不可少的环节。

作为 Deep Web 数据集成的一个应用，商品价格比较系统，要求把来自不同购物网站表示现实世界同一商品的信息识别出来并进行价格比较。以购书为例，如果我们希望从出售图书的电子商务网站（比如 Amazon 和 Bookpool）购买到最便宜的关于数据挖掘方面的图书，那么需要将来自这两个不同售书网站的查询结果中表示同一本书的记录识别匹配在一起。

传统的实体识别的工作中所提出的方法都是以模式匹配为前提的，即通过对实体之间在同一属性上的值进行比较来判断两个记录是否为同一实体。Web 中信息主要是以 Html 页面的形式发布，由于 Html 页面主要的作用是信息的表现，因此结构化程度很差，在这个前提下难以完成准确的模式匹配。本文在这个基础上，提出了一种在 Deep Web 数据集成环境下进行实体识别的方法。该方法不同于传统的实体识别方法，没有试图在结构化程度很差的 Html 文档上进行模式匹配，而是把每个结果记录看作一个文本文档，通过比较结果记录之间在文本上的相似性将表示现实世界中相同实体的结果记录对识别出来。此外，由于电子商务网站在 Deep Web 占有很大的比例，而在电子商务网站在结果记录中都包含有价格信息，因此我们针对这个特点在前一步的基础上进而把价格因素考虑进来，进一步提高了实体识别的准确性。通过实验表明，我们提出的这种方法在 Web 的环境下可以达到相当高的准确性。

本文其余部分组织如下：第 2 节阐述问题描述；第 3 节提出了 Deep Web 数据集成环境下自动实体识别的方法；第 4 节是相关工作的比较；第 5 节给出实验结果及分析。第 6 节是对全文的总结。

2. 问题描述

在关系数据库领域存在大量成熟的实体识别、数据清洗的工作，但是这些成果在 Web 环境下并不直接适用。因为不同于关系数据库中结构化的数据，Web 中的数据主要是通过网页发布的，因此是一种特殊的半结构化的数据。如图 1 中出现的记录，现存的工作很难以较高的准确性完成记录内部各数据项的分割，以及在各个数据项上的语义添加（Annotation）。所以以往基于结构化数据的实体识别方法并不能直接应用于 Web 数据集成这个新环境。

以购书为例（图 1）假设我们要在 Amazon 返回的 M 条记录和 Bookpool 返回的 N 条记录上进行匹配，单从某一项(比如书名)是不能保证一定能够判断出是不是同一本书的。因为在购书领域，即使存在同样的书名，也有可能是源自不同作者的不同的书。所以在一个领域，仅凭单独的某一个数据项并不能提供给我们足够的信息去判定实体间的匹配关系，而且在半结构化的 Web 数据上，现有的工作还无法将某个数据项（如书名）准确地提取出来。

在购书这个领域，书名和作者一起构成了唯一确定一个实体的主码，如果能在主码字段上进行类似 group by 的操作，似乎就能将表示同一实体的记录匹配在一起，但是我们不能忽略实体识别问题所处的 Web 大环境。首先，对于不同的领域，想要准确确定每一个领域（如航空领域，餐饮领域等）的主码，并非易事，而且，我们致力于产生一种一般性的领域无关的方法，试图能够适应 Web 上的不同领域。其次，即使我们能确定某个领域中唯一标识一个实体的主码，主码字段也不一定都在 Web 页面上返回的记录中出现（比如“作者”可能不出现在记录中，也许在 detail 页面中才可能找到，或者根本就不出现）。况且即使出现了的字段也还无法通过现有的工作准确地提取出来。于是，在我们的方法中，我们保留了半结构化数据本身的特点，在对来自两个网站的结果集中的数据进行匹配的过程中，综合考虑了出现在记录范围内的所有字段的内容，并采用了比较文本相似性的方法，在不需要进行传统模式匹配（Schema Match）和语义添加（Annotation）的基础上，有效地实现了 Deep Web 数据集成中的实体识别，而且是领域无关的。

3. Deep Web 数据集成中的自动实体识别

假设我们要在两个任意的网站 A 和 B 上建立实体之间的匹配关系，对于网站 A 中的每一个实体 A_i，我们都试图在另一个网站 B 中找到与之匹配的实体。于是在我们的方法中，我们计算 A_i 与 B 中所有可能匹配实体之间的相似度值，并考虑最大相似度值与阀值之间的关系，从而判断 B 中与 A_i 相似度值最大的实体是否就是真正与 A_i 匹配的实体。

在上述基础之上，我们通过以下 3 个步骤来解决 Deep Web 数据集成中实体识别的问题：

a) 记录块划分
b) 相似度计算
c) 迭代的训练

3.1 记录块划分

通过观察来自任意一个网站的查询结果列表，可以发现在每一条记录中，不仅包含被描述实体本身的信息，也包含了一些元数据（Metadata）信息。以图 1（A）为例，其中“ourprice:”“published”“you save”等都是显示记录时用到的一些描述性模版信息，不是在 Web 数据库中存储的实体本身的信息，因此这些元数据信息对于实体本身的识别并不起到积极作用，反而由于它们在每个记录中同样出现，会干扰实体间相似度的判别。所以我们要事先将这些元数据信息当成类似于 IR 中的阻止词（Stop Words）进行去除。

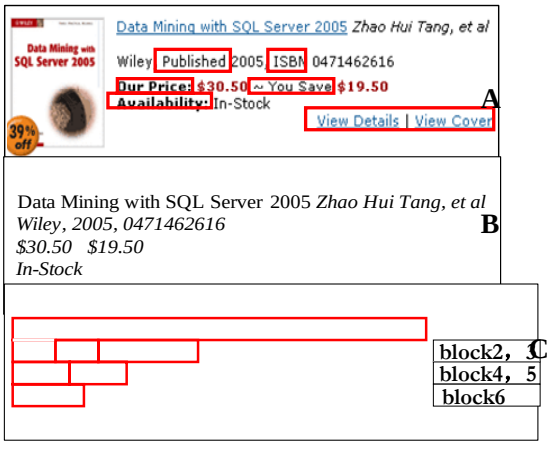


图 1 记录块划分示例

现在图 1（B）所示记录中只保留了进行实体识别所必需的实体本身的信息。如果我们将整个记录看作是描述实体的一段文本，然后在文本间进行相似度的比较，从而确定实体间的相似度，这样的方法准确度低，错误率高，因为它忽略了一个很重要

的因素，那就是记录文本中各个部分在决定实体是否匹配的过程中，发挥着不同的重要性。直观地看，决定两个记录是否表示同一本书，“书名”起到的作用似乎比“图书所属分类”起到的作用要大。如果采用传统的在记录内部划分语义块的方法，其复杂度高，将会成为整个系统效率的瓶颈。

我们采用了一个比较巧妙的办法来解决这个问题，由于 W3C HTML 规范定义了 93 个标签，其中有些标签(如 TABLE、P、DIV、SPAN 等)是用来将网页进行布局、划分为语义上的结构的。我们利用这类布局标签来对记录进行语义块划分不仅效率高，而且能使整体的准确性达到很高的水平。图 1（C）给出了基于布局标签的划分结果——图 1（A）中的记录被划分为 6 块。

3.2 相似度计算

Deep Web 数据集成中，对任两个网站返回的结果进行实体识别，我们需要一种合理的方法衡量来自不同网站的两条记录所代表的实体之间的匹配程度。3.1 节中已经介绍了基于标签对记录块进行划分，每个划分之后的语义块在决定实体之间匹配程度的重要性上存在着一定的区别，于是我们采用一组权值来量化地表示这种区别。对于图 1(c)中划分得到的 6 个语义块，我们分别赋予权值 W_1，W_2…W_6。在计算此记录与来自另一网站的记录之间的相似度值时，问题就转化为计算此记录中各语义块与另一记录之间相似度值的加权相加之和。

定义 1（实体相似度） 实体 A 和实体 B 的相似度值等于实体 A 内各语义块（设为 m 块：$A\text{-}block_1$，$A\text{-}block_2$……$A\text{-}block_m$）与实体 B 之间相似度值的加权相加之和。其中 W_i 代表语义块 $A\text{-}block_i$ 在决定实体匹配程度上的重要性。

公式 1： $$S(A,B)=\sum_{k=1}^{m} W_k \times S(A\_block_k, B)$$

具体到实体 A 内的各语义块，我们并不试图在与之比较的实体 B 内部进行划分，从而找到相匹配的块进行对应块之间相似度计算，这是因为：(1) 在结构性很差的 Html 文档之间进行模式匹配的工作难度大，而且一旦在我们的方法中引入模式匹配出现匹配错误时，会严重影响实体识别工作的准确性。(2) 我们的方法以简单高效的基于标签划分语义块为基础，不同于严格意义上划分记录的各个属性，加剧了模式匹配工作的难度。因此，为了提高实体识别效率，避免由于引入模式匹配造成的错误匹配，我们采用了基于文本比较的标准 TFIDF 余弦-相似度计算的方法[5]计算语义块与另一个记录的相似度值。

在 Deep Web 数据集成的应用中，电子商务的网站占到了很大比例。在处理这类网站的实体识别问题时，价格可以看作是一个很有价值的线索，因为不同电子商务网站出售的同一种商品在价格上一般是非常接近的。对于价格这种特殊的数据类型，标准的计算文本相似度的方法是不适用的，我们需要一种新的衡量标准来计算价格之间的相似度。

首先，在记录块内部我们要将价格正确地识别出来。不难发现，价格在页面上的出现必然伴随着一些特殊的前缀或后缀信息，如“￥”,“＄”,“Price”，“价格”,“元”等。通过识别出这些有限的前、后缀信息，就能很容易地将出现在记录块内部的价格识别出来。其次，考察两个价格之间的匹配程度，绝对的数值差异是不恰当的，我们需要考察数值的相对差值。如$28 与$30 绝对差值是$2，而$2800 与$3000 绝对差值是$200，价格的匹配程度不能由绝对差值来衡量。差异系数可以很好的解决这一问题。

定义 2（价格的相似度） 价格 A 与价格 B 的相似度值 Sp 等于价格 A 与价格 B 的差异系数的补值：$S(P_1,P_2)=1-DC=1-\left(\frac{\sqrt{(P_1-\overline{P})^2+(P_2-\overline{P})^2}}{\overline{P}}\right)$ **公式 2**

其中，DC(Differential Coefficient)指的就是两个价格 P_1 和 P_2 的差异系数，式中的 $\overline{P}$ 指的是两个价格 P_1 和 P_2 的平均值。至此，利用上述实体相似度的定义（定义 1），来自两个网站的记录两两之间的相似度值可由各语义块相似度值加权相加得出。于是，如何量化地衡量各语义块不同的权值，即它们在决定实体匹配性中不同的重要性，就成为了一个关键问题。下面将详细介绍在我们的方法中，是如何通过积极有效的迭代训练得到这组关键权值的。

3.3 迭代的训练

在 Deep Web 数据集成的过程中，对任意两个网站进行实体识别的工作。在已知不同记录之间的相似度值计算方法的前提下，有两组未知数亟待确定：

[1]. 用一组量化的权值来衡量各语义块在实体匹配过程中不同的重要性。其中，每个语义块对应一个权值；

[2]. 用一组阀值来衡量实体匹配的最终结果。依据这组阀值，我们可以将经过相似度计算的两个实体划分到不同的类别中去——依据匹配程度由高到低，分为三类：匹配/疑似匹配/不匹配。

3.3.1 训练样本

我们试图在两个给定的网站上（网站 A 和网站 B）建立实体之间的匹配关系，首先我们在这两个网站上手动选取出 N 对匹配的记录（彼此描述同一实体）作为训练样本。假设 A_1，A_2……A_n 是来自网站 A 的 N 个样本记录，同样 B_1，B_2……B_n 是来自网站 B 的 N 个样本记录，相应地，A_i 与 B_i 是匹配的记录对，于是 A_i 与 B_j（$j \neq i$）都是不匹配的记录对。总的来说，在我们的训练样本中，来自网站 A 的每一个记录 A_i，在 B 中都能唯一地找到一个与之匹配的记录 B_i，而与 B 中剩余的 $N-1$ 个记录都是不匹配的。

3.3.2 权值的确定

在对分别来自网站 A 和 B 的记录 A_x 和 B_y 进行相似度值计算的时候，先将 A_x 进行语义块划分，并给每一个划分得到的语义块赋予不同的权值，这里假设得到 M 个块分别对应 M 个权值（W_1，W_2……W_m）。在我们的训练样本中，对于每一个 A_i，在网站 B 中都能找到一个与之匹配的 B_i，以及 $N-1$ 个与之不匹配的 B_j（$j \neq i$）。可以肯定的是，A_i 与匹配记录 B_i 之间的相似度值一定大于 Ai 与不匹配记录 Bj（$j \neq i$）之间的相似度值。于是对于来自网站 A 的每一个 A_i，都能得到如下的一组不等式：

公式 3

$$\begin{cases} S(A_i, B_i) \geq S(A_i, B_1) \\ \cdots\cdots \\ S(A_i, B_i) \geq S(A_i, B_{i-1}) \\ S(A_i, B_i) \geq S(A_i, B_{i+1}) \\ \cdots\cdots \\ S(A_i, B_i) \geq S(A_i, B_n) \end{cases}$$

同时，记录 A_i 与 B_j 之间的相似度值又可以由 A_i 中各语义块与 B_j 的相似度值加权相加得到，于是我们利用相似度计算公式（公式 1）对公式 3 中的每一个不等式进行展开。对于训练样本中网站 A 的每一个记录 A_i，都能得到以 W_1，W_2…W_m（M 个语义块对应的 M 个权值）为未知数的一组 $N-1$ 个不等式。而网站 A 中共有 N 个记录，于是一共能得到 $N*(N-1)$ 个不等式。实际情况中，不等式的个数如果像这样按照 N 的指数倍增长，那将是不可忍受的。于是我们观察由每一个 A_i 得到的 $N-1$ 个不等式，只需保证 A_i 与匹配记录 B_i 之间的相似度值大于 Ai 与所有不匹配记录 Bj（$j \neq i$）之间的最大相似度值，就可以保证 A_i 与匹配记录 B_i 之间的相似度值一定大于 Ai 与每一个不匹配记录 Bj（$j \neq i$）之间的相似度值。于是，对于每一个 Ai，如下的一个不等式足以保证上述 $N-1$ 个不等式成立：

公式 4 $$S(Ai,Bi) \geq MAX\{S(Ai,Bj)\} \quad (i,j=1,2...,n; j \neq i)$$

相应地，对于网站 A 的 N 个记录，不等式数量由原来的 $N*(N-1)$ 个迅速降为 N 个。

我们知道，对于一个不等式组，它的解集对应的是空间中的一个凸多面体。每一个未知数（在这里就是 W_1，W_2…W_m）对应解空间里的一维。我们取其取值范围的平均值作为相应权值的解，最终可以保证这个 M 维的向量必然是解空间中的一个解。初始计算出来的每个权值往往都落在一个很大的取值范围内，因此并不能精确地量化反映出每个权值对应的语义块在决定实体是否匹配重要性上的权重。之后，我们通过迭代训练的方法来不断地将每个权值都缩小在一个更精确的范围内。

3.3.3 阀值的确定

根据上述由训练样本得出的初始不等式组，我们得到一组初始的权值。由公式 1 中实体相似度的定义，实体 A_x（来自网站 A）和实体 B_y（来自网站 B）之间的匹配程度，可以根据这组权值，对实体 A_x 内各语义块与实体 B_y 的相似度值进行加权相加。于是，对于训练样本中任两条来自不同网站的记录都可以计算出相似度的值，共有 $N \times N = N^2$ 个匹配值。其中，每一个匹配值对应一个可能的组合。

我们知道在训练样本中 N^2 个可能的组合里，有 N 个组合被认为是匹配的，另外的 $N \times (N-1)$ 个组合是不匹配的，因为对来自网站 A 的每一个记录 A_i，在 B 中能且只能找到一个与之匹配的记录 B_i。图 2 中对于每一个 Ai 我们在 0-1 的数轴上标示出它与 B 中每一个实体的相似度的值（介于 0 到 1 之间）。圆圈表示 Ai 同与之匹配的实体 Bi 之间的相似度，方块表示 Ai 同不匹配实体 Bj（$j \neq i$）之间的相似度，代表匹配实体的圆圈必然在每个数轴上最接近 1。N 个匹配的组合对应于图中的 N 个圆圈，我们将这 N 个匹配值中的最小值作为阀值 ***T1***（虚线），同样 $N \times (N-1)$ 个不匹配的组合对应于图中的 $N \times (N-1)$ 个方块，再将这 $N \times (N-1)$ 个匹配值中的最大值作为另一个阀值 ***T2***（实线）。如图，训练样本中的 N^2 个组合就被这两个阀值成功地划分为两类：匹配/不匹配。对于样本以外等待判断的两个实体，相似度值超过阀值 ***T1*** 被认为是匹配的；相似度值小于阀值 ***T2*** 被认为是不匹配的；而相似度值介于两个阀值之间的就被认为是疑似匹配的，还需要用户手动来判断。

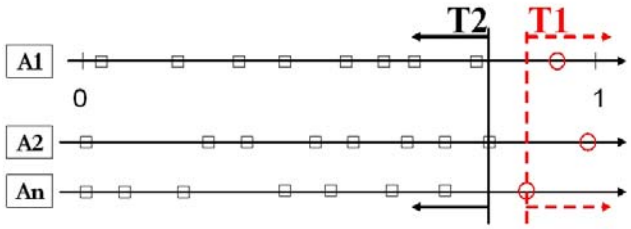


图 2 不交叉情况

图 3 交叉情况

我们考虑图 3 中的情况。图 3 在图 2 的基础上对代表相似度值的数轴上的点进行了一些调整，在每个 *Ai* 对应的数轴上仍然满足代表匹配实体的圆圈比代表不匹配实体的方块更接近 1。但是综合来看所有的实体对应的数轴，我们并没有保证所有 *N* 对匹配样本的相似度（圆圈表示）都大于所有 $N\times(N-1)$ 对不匹配样本的相似度（方块表示），即出现图中所示不匹配记录对 A_2 和 $B_y(y\neq 2)$之间的相似度 $S(A_2,B_y)$ 反而大于匹配记录对 A_1 与 B_1 之间的相似度 $S(A_1,B_1)$ 的情况。在这种情况下，训练样本中的 N^2 个组合就无法被这两个阀值 ***T1***（虚线）和 ***T2***（实线）成功地划分为两类：匹配/不匹配。相似度值落在交叉区域内的两个实体在是否匹配上具有二义性。

因此，一旦原先的不等式组得到的一组权值导致样本记录之间的相似度值在数轴上出现了交叉情况，我们就要重新对不等式组做出限制，使得调整之后的权值不会导致二义性价差区域的出现。图 3 中我们找到造成交叉情况出现的两个点——最小匹配样本匹配值 $S(A_1,B_1)$和最大不匹配样本匹配值 $S(A_2,B_y)$——我们将 $S(A_1,B_1)\geqslant S(A_2,B_y)$作为一个新的限制加入到原不等式组中去。这样就保证了新解出来的一组权值必然使得这个新不等式成立，于是造成交叉的原因得到了破除。

3.3.4 迭代训练器

至此，我们成功地得到了一组阀值去衡量实体匹配的最终结果。依据最小匹配样本相似度值 ***T1*** 和最大不匹配样本相似度值 ***T2***，我们可以判断经过相似度计算的两个实体是否匹配。但是，我们对权值的训练还没有停止。通过以权值作为未知数的不等式组可以得到空间中的一个凸多面体作为解空间，每个权值对应解空间中的一维。我们希望通过迭代训练的方法尽量使解空间缩小到一个更精确的范围内，并最终趋于稳定。

在原不等式组中，我们观察每一个不等式的含义。不等式左边是匹配的记录，右边是不匹配的记录，整个式子表示匹配记录对的相似度值大于不匹配记录对的相似度值。这个条件只是表示了大小关系，没有量化地衡量它们之间匹配值的差距。观察图 2 可以发现，当保证了匹配样本序列（所有的圆圈）与不匹配样本序列（所有的方块）不交叉时，匹配记录对和不匹配记录对之间的匹配值就至少相差了 ***T1-T2***（两个阀值的间距）。我们将这个新的限制加入到原不等式组（公式 3）中，得：

公式 5

$$\begin{cases} S(A_i,B_i)\geq S(A_i,B_1)+(T_1-T_2) \\ \cdots\cdots \\ S(A_i,B_i)\geq S(A_i,B_{i-1})+(T_1-T_2) \\ S(A_i,B_i)\geq S(A_i,B_{i+1})+(T_1-T_2) \\ \cdots\cdots \\ S(A_i,B_i)\geq S(A_i,B_n)+(T_1-T_2) \end{cases}$$

由于每一个不等式中都加入了匹配值差距至少为 ***T1-T2*** 的约束，产生出来的一组新的权值 W_1，$W_2\ldots W_m$构成的解空间逐渐被控制在一个比较精确的范围内。通过这组更精确反映各语义块重要性的权值，我们又可以重新计算样本中匹配样本序列和不匹配样本序列的匹配值，投影到数轴上获得一组新的阀值——***T1*** 和 ***T2***。重复上述过程，将新的限制 ***T1-T2*** 作为新的约束重新规范不等式组，产生新的一组权值，将解空间控制在更精确的范围内。这样循环反复的过程就是迭代训练的方法，直到两阀值 ***T1*** 和 ***T2*** 趋于稳定，这时得到的权值是对各语义块在决定实体是否匹配中重要性的真实反映。

4. 相关工作

在传统数据库领域，实体识别工作也被称为数据清洗和去重。[6，7]就是在同一个表内寻找等价的元组，在表的模式信息已知的前提下，比较两个元组在对应属性上文本的相似程度。这些工作都是在具体的领域上开展的，扩展性差而且代价很高。在 Web 环境下同样也存在一些工作试图将不同数据源提供的数据匹配起来。[8]是较为完善的指导在多个异质数据源上如何进行实体识别工作的方法，它提出了多种减少匹配代价提高匹配效率的策略，但是在结构化很差的 Web 数据上很难直接应用；[9]提出了一

系列被赋予权值的字符串转换规则，并将其应用在共享属性上从而进行实体是否匹配的判别，这是以模式匹配和发掘共享属性为前提的。在[10]中提出了一种 PROM 的方法，利用专家制定的或实际中发掘出来的各属性间的限制来协助实体识别工作，但是针对 Web 的各个领域，完整地制订出不同属性间依赖关系的规则库对开发者来说是一个很大的负担。而我们的工作一个很突出的优点就是能自动地学习并获取针对不同数据源的不同的权值，尽可能减少用户参与，而且准确性很高。具体到 Deep Web 数据集成环境下，鉴于不同数据源的异质性和自主性，模式匹配是很难进行的工作。我们的方法则是在模式匹配关系未知条件下进行实体识别的首次尝试。

5. 实验

为了验证本文所提出实体匹配方法的正确性，我们利用 Lucene(基于 Java 的全文索引引擎工具包)实现了一个研究原型，并在本节给出了实验结果。

5.1 数据集

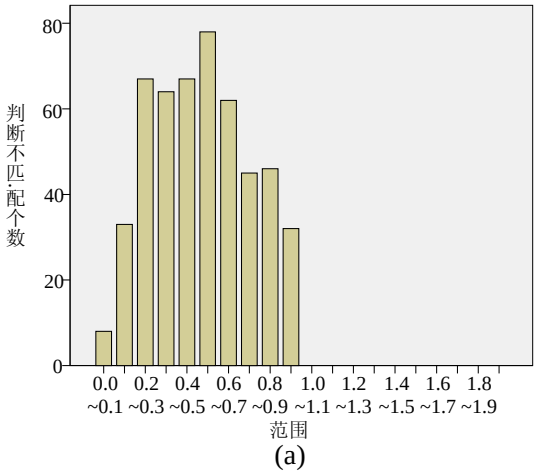
本实验的数据集在图书这个主题下从大家熟悉的 Amazon 和 Bookpool 两个购书网站获取。我们通过对它们的查询接口提交 8 个特定查询的方式获取数据记录。对每个查询，从两个网站的查询结果中的起始处各选取大致相同的记录数目。数据集共分为两类：训练集和测试集。训练集用来选取样本训练权值和阀值，包括来自 3 个查询的记录，每个查询在两个网站各选取 40 个记录。测试集用来验证本文所提出方法的正确性，包括来自 5 个查询的记录，

该数据集具有以下几个特点：第一，同一个查询下两个网站的查询结果具有较高重复比例；第二，足够规模，整个数据集超过 2000 个记录；第三，查询之间相互独立，多个互不相同的查询保证彼此查询结果基本没有交叉。基于这三个特点，该数据集可以保证实验结果的客观性。

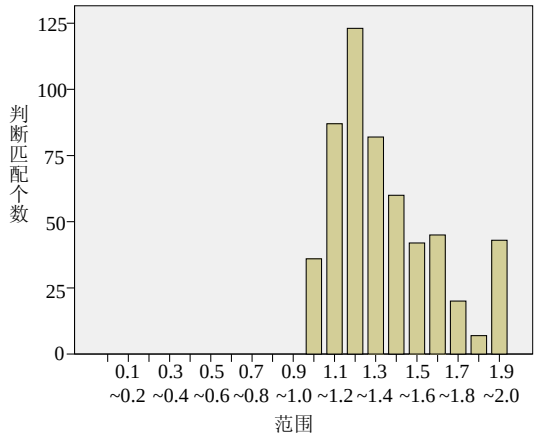
5.2 评价标准

- 匹配准确率：判断为匹配的所有记录对中判断正确的比例；
- 不匹配准确率：判断为不匹配的所有记录对中判断正确的比例；
- 总体准确率：所有进行判断的记录对中判断正确的记录对所占的比例，包括匹配与不匹配的记录对；
- 无法判断率：所有进行判断的记录对中无法判断的记录对所占的比例。

5.3 实验分析



(a)



(b)

图 4 相似度值的分布

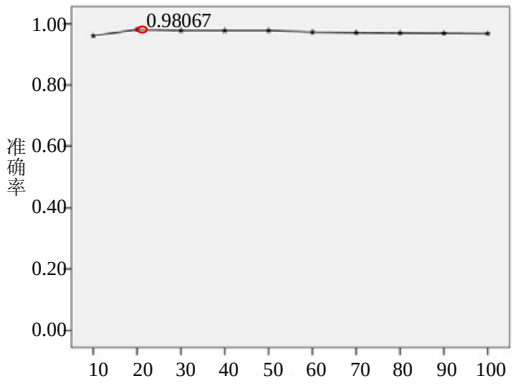


图 5 样本数量与准确率之间的关系

我们首先在训练集上进行训练得到记录中每个块的权值，然后计算得到匹配阀值与不匹配阀值。根据匹配阀值与不匹配阀值，所有记录对相似值集合可分为三个部分：不匹配记录对(小于不匹配阀值)、匹配记录对(大于匹配阀值)和无法判断记录对(介于不匹配阀值和匹配阀值之间)。将每个实体与另一个网站的所有实体的相似度值从大到小排序，取其最大值，将其投影到数轴上进行观察。从图中可以观察

到记录对的相似度值分布的两个明显的特征：第一，匹配记录对的相似度值主要聚集在 1.0-1.5 之间，而不匹配记录对则主要集中在 0.2-0.7 之间；第二，只有很小比例的记录对被判断为无法判断。

表 1 是实验结果的详细数据。从表中可以得出三个结论：第一，准确性高，在四个标准上都达到了比较理想的效果，匹配准确率、不匹配准确率和总体准确率都在 98%以上；第二，人工干预量小，对于无法判断的记录对需要进行人工判断，而无法判断率仅有不到 2%；第三，稳定性好，不同的数据集在匹配准确率、无法判断率和总体准确率相差很小，在不匹配准确率上只有 xml 与其它相差略大。总的来说，我们所提出的自动的判断方法能够达到令人满意的效果。

	匹配正确数量	匹配准确率	不匹配正确数量	不匹配准确率	无法判断数量	无法判断率	总体正确数量	总体准确率
Database	95	0.9895833	91	1	4	0.0209424	190	0.9947644
Linux	139	0.972028	58	0.983050	2	0.0098039	199	0. 97549
OS	121	0. 968	73	0.986486	5	0.0245098	199	0. 97549
Software	88	0.9777778	111	0.991071	2	0.0098039	201	0.9852941
xml	98	0. 989899	73	0.948051	4	0.0222222	175	0. 972222
总计	541	0. 9783	406	0.983051	17	0.017294	964	0. 980671

表 1 具体实验数据

另外在实验中我们发现了一个有趣的问题：(整体)准确率与训练集的数量并非正相关的。图 5 给出了二者之间的关系，从图中可以发现训练集数量在 20 附近准确率达到峰值，然后会缓慢下降。这说明训练样本数过多过少都不好，关键在于样本选取得当。训练样本数过少，无法真正体现各 block 的重要性；训练样本数过多，可能出现一些干扰不等式，影响权值及阀值的确定。

综上所述，我们所提出的自动的实体识别方法能够在较少训练集的情况下达到非常高的准确性和极低的人工干预量。

6. 总结

传统实体识别工作都是在良好的模式匹配前提下实现的，而在 Deep Web 环境下模式匹配仍未得到很好的解决。本文提出了一种在 Deep Web 环境下进行实体识别的方法。该方法无需以模式匹配为前提，通过对表示实体的记录进行划分和比较记录对之间文本的相似性来达到实体识别的目的。实验结果表明，这种方法可以达到非常高的准确性。

7. 参考文献

[1] http://www.brightplanet.com/technology/DeepWeb.asp

[2] Kevin Chen-Chuan Chang, Bin He, Chengkai Li, Mitesh Patel, Zhen Zhang: Structured Databases on the Web: Observations and Implications. SIGMOD Record 33(3): 61-70 (2004)

[3] W.Frakes and R. Baeza-Yates. Information retrieval: Data structures and algorithms, Prentice Hall 1992.

[4] William W. Cohen: Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity. SIGMOD Conference 1998

[5] Sunita Sarawagi, Anuradha Bhamidipaty: Interactive deduplication using active learning. KDD 2002

[6] E.Winkler, W.: The state of record linkage and current research problems. In: Proceedings of of the Survey Methods Section. (1999)

[7] Sheila Tejada, Craig A. Knoblock, Steven Minton: Learning domain-independent string transformation weights for high accuracy object identification. KDD 2002

[8] Doan, A., Lu, Y., Lee, Y., Han, J.: Object Matching for Information Integration: A Profiler-Based Approach. IIWeb 2003

凌妍妍，女，1985 年生，硕士研究生，研究方向：Deep Web 数据管理。

刘伟，男，1976 年生，博士研究生，研究领域：Deep Web 数据管理。

王仲远，男，1985 年生，本科三年级。

艾静，女，1985 年生，本科三年级。

孟小峰，男，1964 年生，教授（博导），研究领域：Web数据管理，XML数据库，移动数据管理。

Web 数据抽取研究

(Web Data Extraction)

Vision-based Web Data Record Extraction

Wei Liu, Xiaofeng Meng, Weiyi Meng

In Proceedings of the 9th SIGMOD International Workshop on Web and Databases (SIGMOD-WebDB2006), Chicago, Illinois, June 30, 2006

Hybrid Method for Automated News Content Extraction from the Web

Yu LI, Xiaofeng Meng, Qing Li, Liping Wang

In proceeding of 7th International Conference on Web Information Systems Engineering(WISE2006),pages 327-338,Wuhan,China,October 2006

RecipeCrawler: Collecting Recipe Data from WWW Incrementally

Yu Li, Xiaofeng Meng, Liping Wang, and Qing Li

In Proceedings of the Seventh International Conference on Web-Age Information Management(WAIM2006), pages 263-274, Hong Kong, China, 17-19 June, 2006. Lecture Notes in Computer Science 4016, Springer 2006

In Proceedings of the 9th SIGMOD International Workshop on Web and Databases (SIGMOD-WebDB2006), Chicago, Illinois, June 30, 2006

Vision-based Web Data Records Extraction

Wei Liu, Xiaofeng Meng
School of Information
Renmin University of China
Beijing, 100872, China
{gue2, xfmeng}@ruc.edu.cn

Weiyi Meng
Dept. of Computer Science
SUNY at Binghamton
Binghamton, NY 13902
meng@cs.binghamton.edu

ABSTRACT

This paper studies the problem of extracting data records on the response pages returned from web databases or search engines. Existing solutions to this problem are based primarily on analyzing the HTML DOM trees and tags of the response pages. While these solutions can achieve good results, they are too heavily dependent on the specifics of HTML and they may have to be changed should the response pages are written in a totally different markup language. In this paper, we propose a novel and language independent technique to solve the data extraction problem. Our proposed solution performs the extraction using only the visual information of the response pages when they are rendered on web browsers. We analyze several types of visual features in this paper. We also propose a new measure ***revision*** to evaluate the extraction performance. This measure reflects perfect extraction ratio among all response pages. Our experimental results indicate that this vision-based approach can achieve very high extraction accuracy.

Keywords

Web DB, response page, data record

1. INTRODUCTION

The World Wide Web has close to one million searchable information sources according to a recent survey[1]. These searchable information sources include both search engines and Web databases. By posting queries to the search interfaces of these information sources, useful information from them can be retrieved. Often the retrieved information (query results) is wrapped on response pages returned by these systems in the form of data records, each of which corresponds to an entity such as a document or a book. Data records are usually displayed visually neatly on Web browsers to ease the consumption of human users. In Figure 1, a number of book records are listed on a response page from Amazon.com.

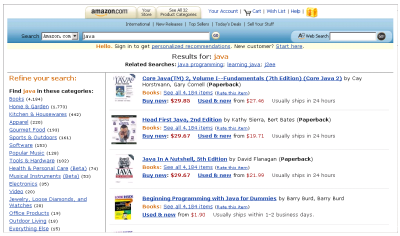


Figure 1: A response page from Amazon

However, to make the retrieved data records machine processable, which is needed in many applications such as deep web crawling and metasearching, they need to be extracted from the response pages. In this paper, we study the problem of automatically extracting the data records from the response pages of web-based search systems.

The problem of web data extraction has received a lot of attention in recent years[2][5][6][7][8]. The existing solutions are mainly based on analyzing the HTML source files of the response pages. Although they can achieve reasonably high accuracies in the reported experimental results, the current studies of this problem have several limitations. First, HTML-based approaches suffer from the following problems: (1) HTML itself is still evolving and when new versions or new tags appear, the previous solutions will have to be amended repeatedly to adapt to new specifications and new tags. (2) Most previous solutions only considered the HTML files that do not include scripts such as JavaScript and CSS. As more and more web pages use more complex JavaScript and CSS to influence the structure of web pages, the applicability of the existing solutions will become lower. (3) If HTML is replaced by a new language in the future, then previous solutions will have to be revised greatly or even abandoned, and other approaches must be proposed to accommodate the new language. Second, traditional performance measures, precision and recall, do not fully reflect the quality of the extraction. Third, most performance studies used small data sets, which is inadequate in assuring the impartiality of the experimental results.

There are already some works [9][12] that analyze the layout structure of web pages. They try to effectively represent and understand the presentation structure of web pages, which are physical structure independent. But the research on vision-based web data extraction is still at its infancy. It is well known that web pages are used to publish information for humans to browse, and not designed for computers to extract information automatically. Based on such consideration, in this paper we propose a novel approach to

extract data records automatically based on the visual representation of web pages. Like [8][7], our approach also aims at the response pages that have multiple data records. Our approach employs a three-step strategy to achieve this objective. First, given a response page, transform it into a Visual Block tree based on its visual representation; second, discover the region (data region) which contains all the data records in the Visual Block tree; third, extract data records from the data region.

This paper has the following contributions:

1. We believe this is the first work that utilizes only the visual content features on the response page as displayed on a browser to extract data records automatically.

2. A new performance measure, ***revision***, is proposed to evaluate the approaches for web data extraction. The measure ***revision*** is the percentage of the web sites whose records cannot be perfectly extracted (i.e., at least one of the precision and recall is not 100%). For these sites, manual revision of the extraction rules is needed.

3. A data set of 1,000 web databases and search engines is used in our experiment study. This is by far the largest data set used in similar studies (previous works seldom used 200 sites). Our experimental results indicate that our approach is very effective.

2. RELATED WORKS

Until now, many approaches have been reported in the literature for extracting information from Web pages. Recently, many automatic approaches [5][6][7][8] have been proposed instead of manual approaches [2] and semi-automatic approaches [3] [4]. For example, [6] find patterns or grammars from multiple pages in HTML DOM trees containing similar data records, and they require an initial set of pages containing similar data records. In [5], a string matching method is proposed, which is based on the observation that all the data records are placed in a specific region and this is reflected in the tag tree by the fact that they share the same path in DOM tree. The method DEPTA[7] used tree alignment instead of tag strings, which exploits nested tree structures to perform more accurate data extraction, so it can be considered as an improvement of MDR[8]. The only works that we are aware of that utilize some visual information to extract data records are [13][14]. However, in these approaches, tag structures are still the primary information utilized while visual information plays a small role. For example, in [13], when the visual information is not used, the recall and precision reduce by only 5%. In contrast, in this paper, our approach performs data record extraction completely based on visual information.

Although the works discussed above applied different techniques and theories, they have a common characteristic: they are all implemented based on HTML DOM trees and tags by parsing the HTML documents. In Section 1, we discussed the latent and inevitable limitations of them.

Since web pages are used to publish information for humans to browse and read, the desired information we want extracted must be visible, so the visual features of web pages can be very helpful for web information extraction. Currently, some works are proposed to process web pages based on their visual representation. For example, a web page segmentation algorithm VIPs is proposed in [9] which simulates how a user understands web layout structure based on his/her visual perception. Our approach is implemented based on VIPs. [10] are proposed to implement link analysis based on the layout and visual information of web pages. Until now, the layout and visual information is not effectively utilized to extract structural web information, and it is only considered as a heuristic accessorial means.

3. INTERESTING VISUAL OBSERVATIONS FOR RESPONSE PAGES

Web pages are used to publish information on the Web. To make the information on web pages easier to understand, web page designers often associate different types of information with distinct visual characteristics (such as font, color, layout, etc.). As a result, visual features are important for identifying special information on Web pages.

Response pages are special web pages that contain data records retrieved from Web information sources, and the data records contained in them also have some interesting distinct visual features according to our observation. Below we describe the main visual features our approach uses.

Position Features (PF): These features indicate the location of the data region on a response page.

- *PF1:* Data regions are always centered horizontally.
- *PF2:* The size of the data region is usually large relative to the area size of the whole page.

Though web pages are designed by different people, these designers all have the common consideration in placing the data region: the data records are the contents in focus on response pages, and they are always centered and conspicuous on web pages to catch the user's attention. By investigating a large number of response pages, we found two interesting facts. First, data regions are always located in the middle section horizontally on response pages. Second, the size of a data region is usually large when there are enough data records in the data region. The actual size of a data region may change greatly for different systems because it is not only influenced by the number of data records retrieved but also by what information is included in each data record, which is application dependent. Therefore, our approach does not use the actual size, instead it uses the ratio of the size of the data region to the size of whole response page.

Layout Features (LF): These features indicate how the data records in the data region are typically arranged.

- *LF1:* The data records are usually aligned flush left in the data region.
- *LF2:* All data records are adjoining.
- *LF3:* Adjoining data records do not overlap, and the space between any two adjoining records is the same.

The designers of web pages always arrange the data records in some format in order to make them visually regular. The regularity can be presented by one of the two layout models.

In Model 1, The data records are arrayed in a single column evenly, though they may be different in width and height. LF1 implies that the data records have the same distance to the left boundary of the data region. In Model 2, data records are arranged in multiple columns, and the data records in the same column have the same distance to the left boundary of the data region. In addition, data records do not overlap, which means that the regions of different data records can be separated. Based on our observation, the response pages of all search engines follow Model 1 while the response pages of web databases may follow either

Table 1: Relevant visual information about the four data records in Fig. 1

	Images (pixel)	plain texts		link texts	
		Total font number	Shared font number	Total font number	Shared font number
Data record 1	944	5	4	3	3
Data record 2	1056	5	4	3	3
Data record 3	871	5	4	3	3
Data record 4	912	4	4	3	3

of the two models. Model 2 is a little bit more complicated than Model 1 in layout, and it can be processed with some extension to the techniques used to process Model 1. In this paper, we focus on dealing with Model 1 due to the limitation of paper length.

We should note that feature LF1 is not always true as some data records on certain response pages of some sites (noticeably Google) may be indented. But the indented data records and the un-indented ones have very similar visual features. In this case, all data records that satisfy Model 1 are identified first, and then the indented data records are extracted utilizing the knowledge obtained from un-indented data records that have already been identified.

Appearance Features (AF): These features capture the visual features within data records.

- *AF1:* Data records are very similar in their appearances, and the similarity includes the sizes of the images they contain and the fonts they use.
- *AF2:* Data contents of the same type in different data records have similar presentations in three aspects: size of image, font of plain text and font of link text (The font of text is determined by font-size, font-color, font-weight and font-style).

Data records usually contain three types of data contents, i.e., images, plain texts (the texts without hyperlinks) and link texts (the texts with hyperlinks). Table 1 shows the information on the three aspects of data records in Figure 1, and we can find that the four data records are very close on the three aspects.

Our data record extraction solution is developed mainly based on the above three types of visual features. Feature PF is used to locate the region containing all the data records on a response page; feature LF and feature AF are combined together to extract the data records accurately.

Content Feature (CF): These features hint the regularity of the contents in data records.

- *CF1:* All data records have mandatory contents and some may have optional contents.
- *CF2:* The presentation of contents in a data record follows a fixed order.

The data records are the entities in real world, and they consist of data units with different semantic concepts. The data units can be classified into two kinds: mandatory and optional. Mandatory units are those that must appear in each data record. For example, if every book data record must have a title, then titles are mandatory data units. In contrast, optional units may be missing in some data records. For example, discounted price for products in e-commerce web sites is likely an optional unit because some products may not have discount price.

4. WEB DATA RECORD EXTRACTION

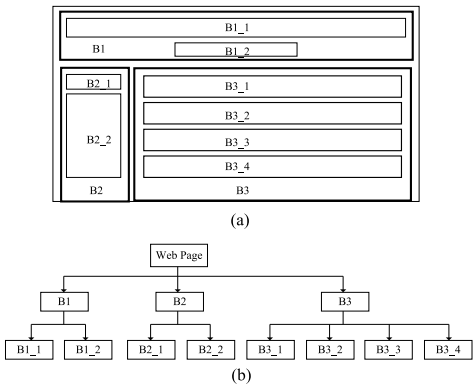


Figure 2: The content structure (a) and its Visual Block tree (b)

Based on the visual features introduced in the previous section, we propose a vision-based approach to extract data records from response pages. Our approach consists of three main steps. First, use the VIPs [9] algorithm to construct the Visual Block tree for each response page. Second, locate the data region in the Visual Block tree based on the PF features. Third, extract the data records from the data region based on the LF and AF features.

4.1 Building Visual Block tree

The Vision-based Page Segmentation (VIPs) algorithm aims to extract the content structure of a web page based on its visual presentation. Such content structure is a tree structure, and each node in the tree corresponds to a rectangular region on a web page. The leaf blocks are the blocks that cannot be segmented further, and they represent the minimum semantic units, such as continuous texts or images. There is a containment relationship between a parent node and a child node, i.e., the rectangle corresponding to a child node is contained in the rectangle corresponding to the parent node. We call this tree structure Visual Block tree in this paper. In our implementation we adopt the VIPS algorithm to build a Visual Block tree for each response page. Figure 2(a) shows the content structure of the response page shown in Figure 1 and Figure 2(b) gives its corresponding Visual Block tree. Actually, Visual Block tree is more complicated than what Figure 2 shows (there are often hundreds even thousands of blocks in a Visual Block tree).

For each block in the Visual Block tree, its position (the position on response page) and its size (width and height) are logged. The leaf blocks can be classified into three kinds: image block, plain text block and link text block, which represent three kinds of information in data records respectively. If a leaf block is a plain text block or a link text block, the font information is attached to it.

4.2 Data region discovery

PF1 and PF2 indicate that the data records are the primary content on the response pages and the data region is centrally located on these pages. The data region corresponds to a block in the Visual Block tree (in this paper we only consider response pages that have only a single data region). We locate the data region by finding the block that satisfies the two PF features. Each feature can be considered a rule or a requirement. The first rule can be applied directly, while the second rule can be represented

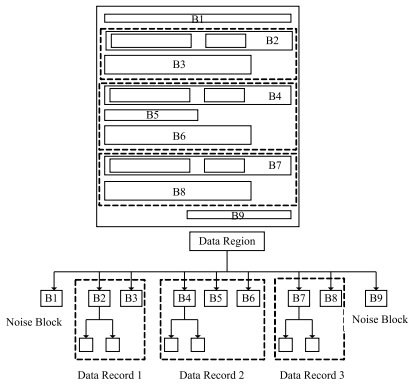


Figure 3: A general case of data region

by $(area_b/area_{responsepage}) \geq T_{dataregion}$, where $area_b$ is the area of block b, $area_{responsepage}$ is the area of the response page, and $T_{dataregion}$ is the threshold used to judge whether b is sufficiently large relative to $area_{responsepage}$. The threshold is trained from sample response pages collected from different real web sites. For the blocks that satisfy both rules, we select the block at the lowest level in the Visual Block tree.

4.3 Data records extraction from data region

In order to extract data records from the data region accurately, two facts must be considered. First, there may be blocks that do not belong to any data record, such as the statistical information (about 2,038 matching results for java) and annotation about data records (1 2 3 4 5 [Next]). These blocks are called noise blocks in this paper. According to LF2, noise blocks cannot appear between data records and they can only appear at the top or the bottom of the data region. Second, one data record may correspond to one or more blocks in the Visual Block tree, and the total number of blocks one data record contains is not fixed. For example, in Figure 1, "Buy new" price exists in all four data records, while "Used & new" price only exists in the first three data records. Figure 3 shows an example of a data region that has the above problems: Block B1 (statistical information) and B9 (annotation) are noise blocks; there are three data records (B2 and B3 form data record 1; B4, B5 and B6 form data record 2; B7 and B8 form data record 3), and the dashed boxes are the boundaries of data records.

This step is to discover the boundary of data records based on the LF and AF features. That is, we attempt to determine which blocks belong to the same data record. We achieve this with the following three sub-steps: *Sub-step1:* Filter out some noise blocks; *Sub-step2:* Cluster the remaining blocks by computing their appearance similarity; *Sub-step3:* Discover data record boundary by regrouping blocks.

4.3.1 Noise blocks filtering

Because noise blocks are always at the top or bottom, we check the blocks located at the two positions according to LF1. If a block is not aligned flush left, it will be removed from the data region as a noise block. In this sub-step, we cannot assure all noise blocks are removed. For example, in Figure 3, block B9 can be removed in this sub-step, while block B1 cannot be removed.

4.3.2 Blocks clustering

The remaining blocks in the data region are clustered based on their appearance similarity. Since there are three kinds of information in data records, i.e., images, plain text and link text, the appearance similarity of blocks is computed from the three aspects. For images, we care about the size; for plain text and link text, we care about the shared fonts. Intuitively, if two blocks are more similar on image size, font, they should be more similar in appearance. The appearance similarity formula between two blocks B1 and B2 is given below:

$$sim(B_1, B_2) = w_i \times simIMG(B_1, B_2)$$
$$+w_{pt} \times simPT(B_1, B_2) + w_{lt} \times simLT(B_1, B_2)$$

where simIMG(B_1, B_2) is the similarity based on image size, simPT(B_1, B_2) is the similarity on plain text font, and simLT(B_1, B_2) is the similarity on link text font. And w_i, w_{pt} and w_{lt} are the weights of these similarities, respectively. Table 2 gives the formulas to compute the component similarities and the weights in different cases. The weight of one type of contents is proportional to their total size relative to the total size of the two blocks.

A simple one-pass clustering algorithm is applied. The basic idea of this algorithm is as follows. First, starting from an arbitrary order of all the input blocks, take the first block from the list and use it to form a cluster. Next, for each of the remaining blocks, say B, compute its similarity with each existing cluster. Let C be the cluster that has the maximum similarity with A. If sim(B, C)$> T_{as}$ for some threshold T_{as}, which is to be trained by sample pages (generally, T_{as} is set to 0.8), then add B to C; otherwise, form a new cluster based on B. Function sim(B, C) is defined to be the average of the similarities between B and all blocks in C computed using the Formula above.

As an example, by applying this method to the blocks in Figure 1, the blocks containing the titles of the data records are clustered together after clustering, so are the prices of data records and other contents.

4.3.3 Blocks regrouping

In 4.3.2, the blocks in the data region are grouped into several clusters. However, these clusters do not correspond to data records. On the contrary, the blocks in the same cluster likely all come from different data records. According to AF2, the blocks in the same cluster have the same type of contents of the data records.

The blocks in the data region are regrouped, and the blocks belonging to the same data record form a group. This regrouping process has the following three phases:

Phase 1. For each cluster C_i, obtain its minimum-bounding box R_i, which is the smallest rectangle on the response page that can enclose all the blocks in C_i. We get the same number of boxes as the clusters. Reorder the blocks in C_i from top to bottom according to their positions in web browser. Thus, $B_{i,j}$ is above $B_{i,j+1}$ on web browser.

Phase 2. Suppose C_{max} is the cluster with the maximum number of blocks. If there are multiple such clusters, select the one whose box is positioned higher than the others on the web browser (here "higher position" is based on the highest point in each block). Let the number of blocks in C_{max} be n. Each block in C_{max} forms an initial group. So there are n initial groups (G_1, G_2, , G_n) with each group G_k having only one block $B_{max,k}$.

Phase 3. For each cluster C_i, if R_i overlaps with R_{max} on

Table 2: Formulas and remarks

	Formula	Remarks
1	$\text{simIMG(B1, B2)}=\frac{\text{Min}\{sa_i(B1), sa_i(B2)\}}{\text{Max}\{sa_i(B1), sa_i(B2)\}}$	$sa_i(B)$ is the total area of images in block B.
2	$w_i=\frac{sa_i(B1)+sa_i(B2)}{sa_B(B1)+sa_B(B2)}$	$sa_B(B)$ is the total area of block B.
3	$\text{simPT(B1,B2)}=\frac{\text{Min}\{fn_{pt}(B1), fn_{pt}(B2)\}}{\text{Max}\{fn_{pt}(B1), fn_{pt}(B2)\}}$	$fn_{pt}(B)$ is the total number of the fonts of plain texts in block B.
4	$w_{pt}=\frac{sa_{pt}(B1)+sa_{pt}(B2)}{sa_B(B1)+sa_B(B2)}$	$sa_{pt}(B)$ is the total area of plain texts in block B.
5	$\text{simLT(B1,B2)}=\frac{\text{Min}\{fn_{lt}(B1), fn_{lt}(B2)\}}{\text{Max}\{fn_{lt}(B1), fn_{lt}(B2)\}}$	$fn_{lt}(B)$ is the total number of the fonts of link texts in block B.
6	$w_{lt}=\frac{sa_{lt}(B1)+sa_{lt}(B2)}{sa_B(B1)+sa_B(B2)}$	sa_{pt}(B) is the total area of link texts in block B.

the web browser, process all the blocks in C_i. If R_i is lower (higher) than R_{max}, then for each block $B_{i,j}$ in C_i, find the nearest block $B_{max,k}$ in C_{max} that is higher (lower) than $B_{i,j}$ and put $B_{i,j}$ into G_k. When all clusters are processed, each group is a data record.

The basic idea of the process is as follows. According to LF2 and LF3, no noise block can appear between data records, and its corresponding box will not overlap with others. So the boxes that overlap with others enclose all the blocks that belong to data records. In sub-step2 (section 4.3.2), the blocks containing the data contents of the same type will be in the same cluster (e.g., for book records, the blocks containing titles will be clustered together). According to CF1, if a cluster has the maximum number of blocks, then the blocks in this cluster are the mandatory contents in data records, and the number of blocks in it is the number of data records. If there is more than one such cluster, we select one as C_{max} (generally, the one whose box is higher than the others on the web browser is selected). We select the blocks in C_{max} as the seeds of the data records, and each block forms an initial group. In each initial group G_k, there is only one block $B_{max,k}$. Then we try to put the blocks in other clusters into the right groups. That means if a block $B_{i,j}$ (in C_i, C_i is not C_{max}) and a block $B_{max,k}$ (in C_{max}) are in the same data record, then $B_{i,j}$ should be put into the group $B_{max,k}$ belongs to. In another word, the blocks in the same data record are also in the same group. According to LF3, no two adjoining data records overlap. So for $B_{max,k}$ in C_{max}, the blocks that belong to the same data record with $B_{max,k}$ must be below $B_{max,k-1}$ and above $B_{max,k+1}$. For each C_i, if R_i is lower (higher) than R_{max}, then the block on top of R_i is lower (higher) than the block on top of R_{max}. According to CF2, this determines $B_{i,j}$ is lower (higher) than $B_{max,k}$ if they belong to the same data record. So we can conclude that, if $B_{max,k}$ is the nearest block higher (lower) than $B_{i,j}$, then $B_{i,j}$ is put into the group $B_{max,k}$ belongs to.

5. EXPERIMENTS

We have built an operational prototype system based on our method, and we evaluate it in this section. This prototype system is implemented with C# on a Pentium 4 2GH PC. For response pages with no more than 20 data records, the whole process takes no more than 3 seconds.

5.1 Data set

Most previous works on web data extraction conducted experimental evaluations on relatively small data sets, and as a result, the experimental results are often not very reliable. Sometimes, the same system/approach yields very different experimental results depending on the data sets used (e.g., see the experimental comparisons reported in [8][13] about three approaches). In general, there are two reasons that may lead to this situation: first, the size of the data set used is too small, and second, the data set used is not sufficiently representative of the general situation.

In this paper, we use a much larger data set than those used in other similar studies to avoid the problems mentioned above. Our data set is collected from the Completeplanet web site (www.completeplanet.com). Complete-planet is currently the largest depository for deep web, which has collected the search entries of more than 70,000 web databases and search engines. These search systems are organized under 43 topics covering all the main domains in real world. We select 1,000 web sites from these topics (the top 10 to 30 web sites in each topic). During our selection, duplicates under different topics are not used. In addition, web sites that are powered by well-known search engines such as Google are not used. This is to maximize the diversity among the selected web sites. For each web site selected, we get at least five response pages by submitting different queries to reduce randomness. Only response pages containing at least two data records are used. In summary, our data set is much larger and more diverse than any data set used in related works. We plan to make the data set publicly available in the near future.

5.2 Performance measures

Two measures, precision and recall, are widely used to measure the performance of data record extraction algorithms in published literatures. Precision is the percentage of correctly extracted records among all extracted records and recall is the percentage of correctly extracted records among all records that exist on response pages. In our experiments, a data record is correctly extracted only if anything in it is not missed and anything not in it is not included.

Besides precision and recall, there is an important measure neglected by other researchers. It is the number of web sites with perfect precision and recall, i.e., both precision and recall are 100% at the same time. This measure has a

Table3: Comparison of ViDRE and MDR

	ViDRE	MDR
DR_r	85,497	
DR_e	84,198	53,323
DR_c	83,103	45,485
Total Web sites	1,000	
Correct web sites	876	448
precision	98.7%	85.3%
recall	97.2%	53.2%
revision	12.4%	55.2%

great meaning for web data extraction in real applications. We give a simple example to explain this. Suppose there are three approaches (A1, A2 and A3) which can extract data records from response pages, and they use the same data set (5 web sites, 10 data records in each web site). A1 extracts 9 records for each site and they are all correct. So the average precision and recall of A1 are 100% and 90%, respectively. A2 extracts 11 records for each site and 10 are correct. So the average precision and recall of A2 are 90.9% and 100%, respectively. A3 extracts 10 records for 4 of the 5 sites and they are all correct. For the 5th site, A3 extracts no records. So the average precision and recall of A3 are both 80%. Based on average precision and recall, A1 and A2 are better than A3. But in real applications A3 may be the best choice. The reason is that in order to make precision and recall 100%, A1 and A2 have to be manually tuned/adjusted for each web site, while A3 only needs to be manually tuned for one web site. In other words, A3 needs the minimum manual intervention.

In this paper we propose a new measure called ***revision***. Its definition is given below.

$$revision = \frac{WS_t - WS_c}{WS_t}$$

where WS_c is the total number of web sites whose precision and recall are both 100%, and WS_t is total number of web sites processed. This measure represents the degree of manual intervention required.

5.3 Experimental results

We evaluate our prototype system ViDRE and compare it with MDR. We choose MDR based on two considerations: first, it can be downloaded from web site and can run locally; second, it is very similar to ViDRE (a single page at a time; data extracted at record level). MDR has a similarity threshold, which is set at default value (60%) in our test, based on the suggestion of the authors of MDR. Our ViDRE also has a similarity threshold, which is set at 0.8. We show the experimental results in Table 3.

From Table 3, we can draw two conclusions. First, the performance of ViDRE is very good. That means vision-based approach can also reach a high accuracy (precision and recall). Second, ViDRE is much better than MDR on revision. MDR has to be revised for nearly half of the web sites tested, while ViDRE only need to be revised for less than one eighth of these sites.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a fully automated technique to extract search result data records from response pages dynamically generated by search engines or Web DBs. Our technique utilizes only the visual content features on the response page, which is HTML language or any other language independent. This differentiates our technique from other competing techniques for similar applications. Our experimental results on a large data set indicate that our technique can achieve high extraction accuracy.

In the future, we plan to address several issues and improve our vision-based approach further. First, if there is only one data record on a response page, our approach will fail. We intend to tackle this problem by comparing multiple response pages from one web site. Second, while our data region discovery technique is fast, data record extraction is slow when the number of data records is large (say more than 50). We plan to look into the issue of improving the efficiency of our approach. Third, we plan to collect a set of response pages from real web sites which are not written in HTML, and show our vision-based approach is really language independent.

7. ACKNOWLEDGMENTS

This research was partially supported by the grants from the NSFC under grant number 60573091, 60273018, China National Basic Research and Development Program's Semantic Grid Project (No. 2003CB317000), the Key Project of Ministry of Education of China under Grant No.03044, Program for New Century Excellent Talents in University (NCET), and US NSF grants IIS-0414981 and CNS-0454298.

8. REFERENCES

[1] K. Chang, B. He, C. Li, M. Patel, and Z. Zhang. Structured Databases on the Web: Observations and Implications. In SIGMOD Record, 33(3), pages 61-70, 2004.

[2] G. O. Arocena, A. O. Mendelzon. WebOQL: Restructuring Documents, Databases, and Webs. In ICDE, pages 24-33, 1998.

[3] X. Meng, H. Lu, H. Wang. SG-WRAP: A Schema-Guided Wrapper Generation. In ICDE, pages 331-332, 2002.

[4] R. Baumgartner, S. Flesca, G. Gottlob. Visual Web Information Extraction with Lixto. In VLDB , pages 119-128, 2001.

[5] C. Chang, S. Lui. IEPAD: Information extraction based on pattern discovery. In WWW, pages 681-688, 2001.

[6] V. Crescenzi, G. Mecca, P. Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In VLDB, pages 109-118, 2001.

[7] Y. Zhai, B. Liu. Web data extraction based on partial tree alignment. In WWW, pages 76-85, 2005.

[8] B. Liu, R. L. Grossman, Yanhong Zhai. Mining data records in Web pages. In KDD, pages 601-606, 2003.

[9] D. Cai, S. Yu, J. Wen, W. Ma. Extracting Content Structure for Web Pages Based on Visual Representation. In APWeb, pages 406-417, 2003.

[10] D. Cai, X. He, J. Wen, W. Ma. Block-level link analysis. In SIGIR, pages 440-447, 2004.

[11] D. Cai, X. He, Z. Li, W. Ma, J. Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In ACM Multimedia, pages 952-959, 2004.

[12] Xiaodong Gu, Jinlin Chen, Wei-Ying Ma, Guoliang Chen. Visual Based Content Understanding towards Web Adaptation. In AH, pages 164-173, 2002.

[13] Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, Clement T. Yu. Fully automatic wrapper generation for search engines. In WWW, pages 66-75, 2005.

[14] Kai Simon, Georg Lausen. ViPER: Augmenting Automatic Information Extraction with Visual Perceptions. In CIKM, pages 381-388, 2005.

In Proceedings of the Seventh In proceeding of 7th International Conference on Web Information Systems Engineering(WISE2006), pages 327-338,Wuhan,China,October 2006

Hybrid Method for Automated News Content Extraction from the Web

Yu LI[1] , Xiaofeng Meng[1], Qing Li[2], and Liping Wang[2]

[1] School of Information, Renmin University of China, China
{liyu17, xfmeng}@ruc.edu.cn
[2] Dept. of Computer Science, City University of Hong Kong, HKSAR, China
{50095373@student, itqli@}.cityu.edu.hk

Abstract. Web news content extraction is vital to improve news indexing and searching in nowadays search engines, especially for the news searching service. In this paper we study the Web news content extraction problem and propose an automated extraction algorithm for it. Our method is a hybrid one taking the advantage of both sequence matching and tree matching techniques. We propose *TSReC*, a variant of tag sequence representation suitable for both sequence matching and tree matching, along with an associated algorithm for automated Web news content extraction. By implementing a prototype system for Web news content extraction, the empirical evaluation is conducted and the result shows that our method is highly effective and efficient.

1 Introduction

WWW has posed itself as the largest data repository ever available in the history of humankind, and Web search engines such as Google, Yahoo, and MSN Search have emerged as the most important Web services in recent years. Besides conventional keyword based and general purpose search, most search engines have recently launched a new searching service, named “Web news search”, which mainly focuses on Web pages of news.

Web news search causes some non-trivial problems to traditional information retrieval techniques. One of them is how to differentiate Web news content from others in Web pages. As we know, a Web news page usually contains not only the content (the title, date and context) of news, but also other facilities such as the navigation area of the whole Web site, links to related topics, links to supplemental materials, advertisements, and even ongoing events. Separating the news content from the others, and only indexing the content, are believed to be the way to further improve the searching quality.

In this paper we focus on automatically differentiating Web news content from others. In general, identifying the content of an article in conventional techniques requires the knowledge on semantics of phases, which is a hard problem in itself. However, in Web news scenario, alternative solutions are possible, because Web pages are semi-structured documents written in markup language (mostly in HTML), which contain tags and structures dividing the context into fragments with specific

semantics. By properly utilizing them, recently proposed techniques to reduce patterns or to match similar structures, reported can be applied to automatically extract specific parts of HTML documents. Therefore we are motivated to find a similar solution to automatically extract Web news content.

According to the Web page modeling method, existing Web extraction techniques generally fall into two main categories: tag sequence based v.s. tree based. Techniques of the former category view a Web page as a long sequence consisting of HTML tags and text fragments. And the basic idea is to apply traditional pattern reduction techniques in order to find a template. After continuous improvement, these techniques can be easily applied, with an acceptable time complexity; but they still rely on heuristics to certain extent for solving the semantic problem. On the other hand, tree based techniques can avoid the semantic problem with the help of tree structure, for Web page authors tend to cluster texts of different topics into different sub trees. However, general matching on trees is harder than on sequences, hence various restrictive assumptions were made in order to get an acceptable solution.

In this paper, we propose to combine the advantages of tag sequence based techniques and tree based techniques, so as to come up with a hybrid, effective and efficient solution. In particular, our contributions are as follows:

1. We propose an extended sequence representation of Web page, named as *TSReC*(Tag Sequence with Region Code), which can reserve necessary tree structure information. Building from one pass scanning of HTML and region code encoding, it is suitable for both tag sequence based and tree based extraction.
2. We propose an effective algorithm based on *TSReC* for automated Web news content extraction. It contains two procedures, namely, Sequence Matching and Tree Matching. The former one can detect and remove the identical parts of Web news pages, such as navigation bars, copyright notes. The latter one can match and remove the similar structures of Web news pages, such as advertisements and activities. Consequently, our algorithm can differentiate Web news content from others.
3. Empirical evaluation of our algorithm is performed on news pages crawled from real Web sites. The results show that our algorithm is both effective and efficient.

The rest of this paper is organized as follows. In section 2, we review some key Web extraction techniques from the literature, as well as the major work related to Web news extraction. In section 3 the problem of Web new content extraction is defined, and *TSReC* is discussed in section 4 along with an algorithm for building it. In section 5, we discuss how the Web news content extraction can be applied to a search engine, and actually provide the details of our algorithms. Finally, empirical evaluation is studied in section 6, and the conclusion is given in section 7.

2 Related Work

The great demand on Web data extraction has attracted many researchers [16] in recent years, and great efforts have been paid in finding automatic solutions to free people from manual work. Earlier research works were more on semi-automatic tools [10], whereas later ones on automated extraction techniques have become more popu-

lar. However, depending on how to model the HTML Web pages, these techniques mainly fall into two categories: tag sequence based v.s. tree based techniques.

Representational research works on tag sequence based are Stalker [6] and RoadRunner [2]. In Stalker (which also has a commercial version called FETCH), a wrapper program for extraction is learnt from the tag sequence of a Web page. For example, a template based on the fragment "513 Pico, Venice, Phone 1-800-555-1515" may be derived as "513 Pico, *, Phone 1-*-555-1515", which contains two fields for location and district number. By providing enough positive and negative examples, a simple fail-and-release strategy is proposed, which starts from the rigid template, and relaxes the restrictions whenever of the accuracy is found to be not good. As an advantage, utilizing tag sequence enables the authors to apply existing sequence matching techniques. In RoadRunner system, an algorithm is proposed to infer union-free regular expressions that represent page templates, based on tag sequence as well. Regular expression techniques are adopted thus a solid theory foundation exists. However, problems arise when iterating the sequence from one section into another (e.g., from "navigation" to "news body"), due to that there is no extra information in the tag sequence for differentiating them easily. The algorithm in RoadRunner has to try every possible template for the best result, which incurs an unacceptable huge searching space, and therefore leads to an exponential time complexity. In [3], Arvind and Hector proposed an improved version called ExLag with a polynomial time complexity by employing several heuristics. There are other works based on tag sequence-like data structures [7, 8], but still encounter similar problems. Overall, viewing a Web page in HTML as a tag sequence (single words and tags) makes the pattern detection to be easier, but causes handling nesting structures to be harder.

More recently, tree based extraction methods are proposed, such as [1] and [4]. Both of them limit the matching process to work only under sub trees, which helps to differentiate unrelated sections of a Web page. Although sub tree structure brings out some light, pattern reduction turns out to be the dark side. Based on the results of restricted tree edit distance computation process, the work in [1] attaches wildcards to tree nodes and employs heuristics when there is a need to generalize. In [4], an improved version, based on a novel technique named partial tree alignment, is proposed. It can align corresponding data fields in data records without doing wildcards generalizations. However, its assumption on no complex nesting structures limits its use in applications [5], and in special cases adaptation or new techniques have to be made.

Additionally, there are some other related works to ours, which try to utilize the visual cues of Web page to do extraction [9, 11, 12, 13, 14]. Visual cues of a Web page can be derived after a parsing and rendering process. According to visual features, such as layout, font size and color, extraction of specified pieces of Web page content is possible [14]. Further solutions for more complex extraction tasks can also be obtained [9, 11]. However, the feature selection nature of these techniques requires many thresholds or heuristics, which should be trained first and is usually domain specific. An example work in this category is [12], which attempts to automatically extract Web page titles in this category. Comparing to it, our method does not rely on as many heuristics or thresholds.

3 Web News Content Extraction – the Problem

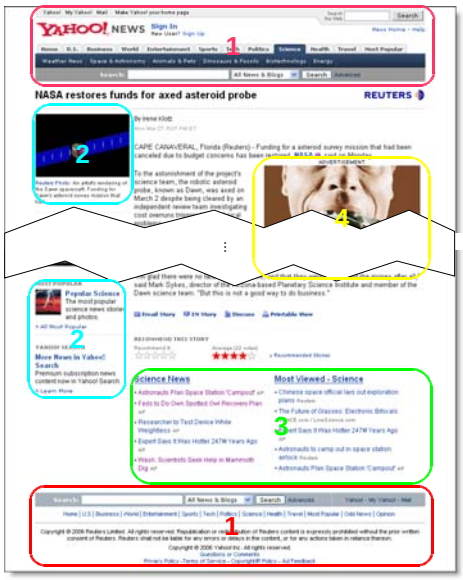


Fig. 1. Example Web news Page from Yahoo, with Navigation Area (Area 1), Events (Area 2), Relate Links (Area 3), and Advertisements (Area 4)

As an example, **Fig. 1** shows a Web news page crawled from Yahoo News, in which we can see that, besides news content, there are other components like navigation areas, events, related links, and advertisements. The objective of Web news content extraction is to find out just *the content*, which is the main part of the Web page after removing the components mentioned above.

One reasonable assumption is that related Web news pages coming from the same Web site share an almost similar page layout and structure, which is usually called the *template*. Actually these web pages are generated by filling a web page template with values queried from the backend database. Therefore most Web data extraction works rely on comparing or matching Web pages that follow the same template. Our method is also based on this idea.

Retrieving Web pages following the same template is possible and practical. Research on clustering or classifying Web pages [1, 15] enables us to do such retrieval in theory, and link analysis techniques in nowadays search engines make it practical. In the Web news domain, as we find out, even simply using the most similar links to the Web pages could achieve acceptable results.

Before we proceed to discussing the algorithms for identifying and extracting the content of Web news, we first give out the definition of *template*.

Definition (Template): A *template* is an incomplete Web page based on which a complete Web page can be generated by filling reserved fields with values. It usually consists of a *common part*, *regular part* and *content part:*

1. *Common part* refers to reserved texts which can not be changed.
2. *Regular part* refers to reserved rigid structure which contains unfilled fields for future values.
3. *Content part* is the reserved area which can be filled with arbitrary html fragments.

Referring back to **Fig. 1**, we have a feel for various parts in the template acccording to the above definition,. Navigation areas (marked by number 1) are the *common parts*, since they are fixed. Events (marked by number 2), relate links (marked by number 3), and advertisements (marked by number 4) are the *regular parts*, because they adhere to the same structure across different Web news pages even though the inner fields may change. The *content parts* are the rest of that page (not marked), which can be freely filled in during generation, and tend to have no fixed patterns.

With the understanding of the *template*, we can divide the Web news content extraction problem into two sub problems, namely, matching for the *common part* and matching for the *regular part*. As we have reviewed in section 2, the former one can be easily solved by sequence matching, whereas the latter one should be better done by tree matching. In order to combine them into the same framework, we have to first design a data structure that is suitable for both techniques.

4 *TSReC*: Tag Sequence with Region Code

In this section we introduce an extended version of tag sequence, namely, Tag Sequence with Region Code (*TSReC*), which is designed for applying both tag-sequence based and tree based extraction algorithms.

As we know, a tag sequence is suitable for extraction but it does not hold any structural information. This backward prevents the utilization of sub trees' information from solving cross trees' ambiguity. For example, by matching on the sequence, we are not able to differentiate the content and advertisement in HTML; but with the help of structural information, the boundaries of them are clear because they are usually resident under different sub trees.

Therefore the basic idea is to extend a tag sequence with extra structural information. In recent database research, the region code in XML processing [17] has proven to be an ideal way to attach structural information in element based storage. With extra storage of a few numbers (region code), all structural relationships can be reserved, such as parent-child, ascent-decedent, and sibling relationships. For our work, we adopt the idea behind the region code, and define *TSReC* as follows.

Definition (TSReC): *TSReC* is a sequence of elements, each of which is defined as

$$TS = <N, RC_b, RC_e, RC_p, RC_l, C>$$

where

1. N is the name of TS, which usually is the same as the HTML tag creating it.
2. RC_b, RC_e, RC_p, and RC_l are region codes, which correspond to begin, end, parent, and level, respectively.

3. *C* is the content of *TS*, which may contain inner HTML tags and text, or be empty.

In **Fig. 2**, there is an example illustration of *TSReC*. At the top part there is a fragment of HTML which shows some links of related articles in science, and there are two categories. In the bottom is its corresponding *TSReC* fragment. With respect to the definition of TSReC, we can see that each element represents a tag in HTML with a corresponding (identical) name. For each of them, four numbers, say RC_b, RC_e, RC_p, and RC_l , are stored in order to keep the structural information. As we will learn in the later algorithm for generating *TSReC*, RC_b and RC_e are begin-end region code identical to XML processing. We use the parent's begin code as the child's RC_p , and also calculate the child's RC_l according to the parent's level. Some TS in *TSReC* may have content consisting of simple HTML tags and texts, which is the same as the conventional tag sequence. As *TSReC* is an extended tag sequence, the sequence matching is supported naturally. On the other hand, the tree matching method can also be applied. Necessary operations, such as calculation of parent-child relationship (by utilizing the RC_p), calculation of ascent-decedent relationship (by utilizing the RC_b and RC_e), are supported. Taking the line 108, line 110 and line 117 as the example, with simple calculation, we know the line 108 and line 110 are under the same sub tree (for $RC_b(108) < RC_b(110) < RC_e(110) < RC_e(108)$), whereas the line 117 in another sub tree (for ($RC_b(117)$, $RC_e(117)$) is not in ($RC_b(108)$, $RC_e(108)$)).

Fig. 2. An fragment HTML and its *TSReC* representation

TSReC can be easily built up by one-pass scanning of a Web page. **Fig. 3** shows the algorithm for building it, which is modified from the conventional one for building tag trees. Basically the algorithm scans the Web page tag by tag (line 05, where text is also treat as a tag), and *TSReC* elements are created in lines 06-29 while computing the begin-end region code. According to the different types of tokens (open tag in lines 06-18, close tag in lines 26-29), different actions are taken. Note that a

new TS is created only when a tag comes to break the text flow (line 08, line 14), such as “P”, “DIV”, “TABLE” and so on. This heuristic helps us get rid of the effect of HTML decoration tags (such as “B”, “FONT”, “A”), which extraction algorithms usually do not care. Finally, after one pass scanning, a *TSReC* instance is returned as the result.

```
Algorithm buildTSReC(w) /* w as input is a web page */
01 TSReC tsrec /* variable for TSReC holder */
02 TS temp_TS /* temp variable of TS */
03 Stack S /* a stack facility used in the algorithm */
04 int count, level, parent
05 while t = readNextTerm(w) do
06    if t is open Tag then
07        ts = getTop(S)
08        if t breaks text flow then
09             if ts is null then
10                 count = 0, level = 0, parent = 0
11             else
12                 count++, level = ts.RC_b, parent = ts.RC_l+1
13             end if
14             temp_TS = createNewTS(t, count, level, parent)
15             push(S, temp_TS)
16        else // t does not break text flow
17            appendToContent(ts, t)
18        end if
19    elseif t is close Tag then
20        if t breaks text flow then
21            ts = pop(S)
22            count++
23            ts.ie = count
24            append(tsrec, ts)
25        end if
26    else if t is Text then
27        ts = getTop(S)
28        appendToContent(ts, t)
29    end if
30 end while
31 return tsrec
```

Fig. 3. buildTSReC - Algorithm for building TSReC from Web Page

5 Automated Web News Content Extraction

Having defined *TSReC*, in this section we discuss a hybrid method for automated Web news content extraction. We lable our algorithm as hybrid because it combines sequence matching techniques with tree matching techniques, in which the former is used for identifying and removing the *common part*, and the latter is for *regular part*. The overall flow chart of our method is as given in **Fig. 4**.

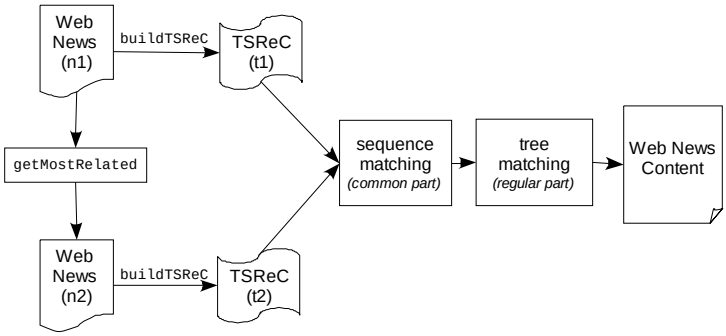


Fig. 4. The Processing Flow of Our Hybrid Method for Web News Content Extraction

The process starts with a Web news page (n1) as its input, and returns the content as its output. Firstly, a function called `getMostRelated` is invoked to get another Web news page probably sharing the same template (cf. section 3). In our evaluation, we simply use the Web page from the related links with its URL most similar to n1's. Then we build *TSReC*s for both Web news pages (i.e., (n1 and n2). After the *TSReC* structures are built up, sequence matching is performed which is followed by tree matching. Each of them will identify and classify specific parts by assigning class marks to elements of *TSReC*. Finally elements with no class marks are output, which are just the news content we need.

5.1 Sequence Matching for *Common Part*

Referring back to the definition in section 3, the target of sequence matching is to find out the *common part* which is supposed identical in two individual Web news pages (cf. Area 1 in **Fig. 1**). Intuitively, we may think that simply comparing two sequences (*TSReC*) can lead us to find out the *common part*. However, things are often a bit more complex, since there are usually more than one *common part*, between each two of which are variable *regular* and *content parts*. The matching process thus becomes not so easy as we have to skip some tags in the sequence to get the best matching result. Due to this reason, we are motivated to adopt a conventional string edit distance calculation algorithm to achieve the goal.

String edit distance calculation is to find out how similar two strings are. A solution based on conventional dynamic programming has a by-product showing the mappings. For example, after the calculation, besides the edit distance between S1("abc") and S2("ac"), we can also know that "*a* in S1 maps to *a* in S2", "*c* in S1 maps to *c* in S2", and "*b* in S1 has no mapping". Taking the advantage of that, we propose an algorithm in **Fig. 5**, based on the dynamic programming solution of string edit distance calculation for sequence matching.

```
Algorithm sequenceMatch(t1, t2)
01 int t1size = sizeof(t1)
02 int t2size = sizeof(t2)
03 int M[t1size+1][t2size+1]
04 M[i][0] = i, i=0,1,2,...,t1size+1
05 M[0][i] = i, i=0,1,2,...,t2size+1
06 for i=1 to t1size do
07    for j=1 to t2size do
08        ts1 = the ith ts of t1
09        ts2 = the jth ts of t2
10        int match = 1
11        if ts1 and ts2 have same tag name and content text then
12            match = 0
13        end if
14        M[i][j] = Min(M[i-1][j-1]+match, M[i-1][j], M[i][j-1])
15        if M[i][j] == M[i-1][j-1]+match then
16            mark matching of ts1 and ts2
17        end if
18    end for
19 end for
```

Fig. 5. Sequence Matching Algorithm on *TSReC* for Common Part

Our sequence matching algorithm takes two *TSReC*s (t1, t2) as the input, does matching for the *common parts* and marks them. Being the same as conventional calculation of string edit distance, our algorithm also uses dynamic programming techniques (lines 03-19). Different from comparing characters in string, our algorithm compares TSs in *TSReC* (line 11). If two TSs have the same tag name and content text, we regard them as they matched, in which case corresponding marking operations are performed (line 16). Otherwise, we move on to look for further matched tags. Note that, in our implementation, we always let t1 be the shorter sequence so as to optimize the algorithm further (ie., let the shorter sequence lead the other loop).

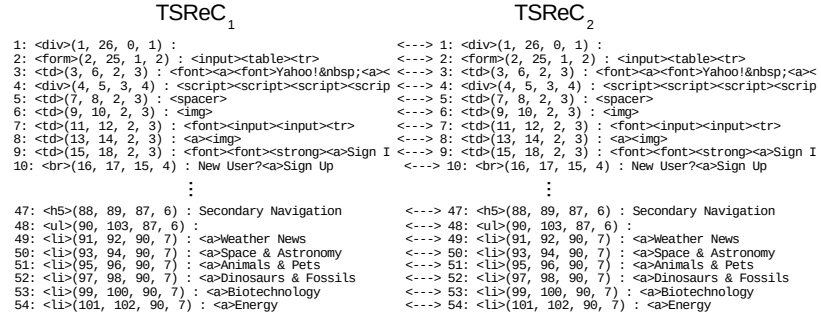


Fig. 6. Sequence Matching Result on *TSReC* – An Example

Fig. 6 shows an example result of sequence matching on two *TSReC*s, which are built up from two related Yahoo News pages. We can see that the *common parts* in each of them have been matched, therefore as the next step we only have to identify and remove the *regular parts*.

5.2 Tree Matching for *Regular Part*

However, differentiating the *regular part* from the *content part* of a Web news page is not as easy as finding *common parts*. Sometimes the mission is even impossible without utilizing semantics, if the *regular parts* in the template are too flexible to fill anything. So it is reasonable to assume that the *regular part* has rigid format structure while the *content part* has not. For example, the *regular part* is usually like "<UL>(<LI><A>(text)</A>)*</UL>", whereas the *content part* can be any free HTML fragment. Based on our observation, this assumption is common in real-life Web pages for news, and therefore our algorithm takes advantage of it in differentiating the regular part.

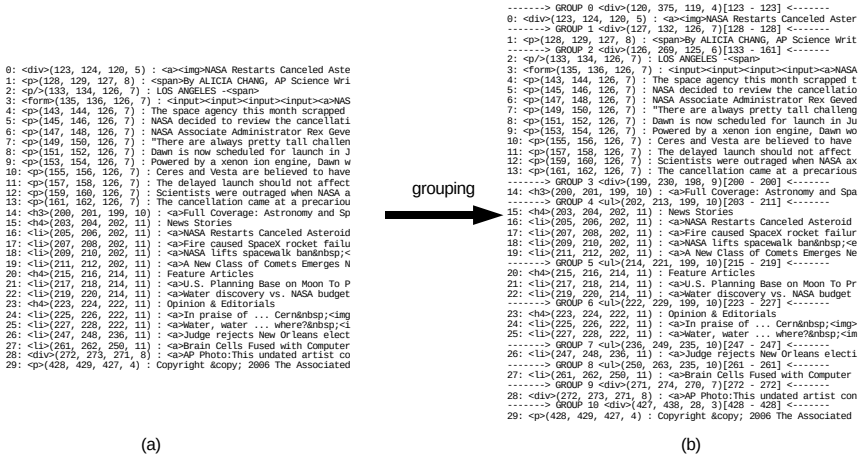


Fig. 7. (a) Different Parts after Sequence Matching. (b) Grouping Results on Different Parts

In our approach, before doing tree matching, we do grouping first. **Fig. 7** gives out an illustration on how grouping is performed. **Fig. 7**(a) is the various parts of a specific Web page derived after the sequence matching; as we can see, tags are organized one by one, being not aware of structures. The grouping process tries to find out which of them are under the same sub trees. As shown in **Fig. 7**(b), for example, line 13 and line 14 are in different groups, which were not known in **Fig. 7**(a). This grouping process is necessary in order to do tree matching subsequently; its algorithm is given in **Fig. 8**.

The grouping algorithm takes a list containing different parts (ie., the rest of the *TSReC* excluding *common parts*) as its input, and returns groups as the output. Each group is a TS (called group parent) with two numbers (called group region), namely, group beginning and group ending. Any TS whose region codes (beginning and ending) are in the group region belongs to that group. The grouping method is to simply check the parent and tree level (lines 06-07) of siblings. If siblings share the same parent and tree level, they are under the same sub tree so we add them into the same group (line 08) by simply extending the group region. Otherwise, a new group will be created (lines 10-11).

```
Algorithm subTreeGroup(dp)
01 List groups /* list for holding groups */
02 /* each group is a list of TS,
03    and initial a group using first element of dp */
04 List group = createGroup(dp[0], dp[0].level, dp[0].parent)
05 for i=1 to sizeof(dp) do
06    if dp[i].level == group.level &&
07       dp[i].parent = group.parent then
08        append(group, dp[i])
09    else
10        append(groups, group)
11        group = createGroup(dp[i], dp[i].level, dp[i].parent)
12    end if
13 end while
14 return groups
```

Fig. 8. Grouping Algorithm

After grouping, we get sub trees which correspond possibly to the *regular* or *content part*s. So we have to differentiate the *regular part* from the others. As we have discussed, the determination of whether a sub tree in a Web page is a *regular part*, is measured by weather there is a sub tree in the other Web page sharing the same rigid pattern. Accordingly, we have the tree matching algorithm as shown in **Fig. 9**.

```
Algorithm treeMatch(t1, t2, rsm)
01 List<TS> dp1 = getDifferentPart(t1, rsm)
02 List<TS> dp2 = getDifferentPart(t2, rsm)
03 List<TS> groups1 = getSubTreeGroup(dp1)
04 List<TS> groups2 = getSubTreeGroup(dp2)
05 int nextMatch = 0
06 for i=0 to sizeof(groups1) do
07    group1 = compactGroup(getGroup(groups1, i))
08    for j=nextMatch to sizeof(groups2) do
09        group2 = compactGroup(getGroup(groups2, j))
10        if groupMatch(group1, group2) then
11            markNotContent(group1)
12            nextMatch = j+1
13            break
14        end if
15    end for
16 end for
17 return groups1
```

Fig. 9. Tree Matching Algorithm on TSReC for Regular Part

Basically the tree matching algorithm tries to find for each group (sub tree) a matched group in the other Web page, which is done in a nest-loop (lines 06-16). The outer loop iterates on t1 (the Web page to be extracted), and the inner loop iterates on t2 (the Web page as reference). The parameter `rsm` is the sequence matching result, which is necessary in extracting different parts (lines 01-02). However, we can do a simple optimization according to the following observation: Usually *regular parts* share the same order even in different Web pages (e.g. “event” appearing before “advertisement” in one Web page seldom appears in reverse order in another Web

page). This optimization in our algorithm is as reflected in line 08 (ie, to start matching after the previous matching position).

A notable function in the tree matching algorithm is the one named `compactGroup` (line 07). It is designed to handle repeatable fields in the *regular part*. For example, referring back to the example of in **Fig. 2**, where "Science News" is a *regular part* which usually contains several lines for titles and links of news in identical topics. However, it is a repeatable field, which can have proper instances according to the underlying database. Since we target to find rigid patterns, multiple instances of a repeatable field should be reduced, and return in the form of "<UL>(<LI><A>(text)</A>)*</UL>". In the algorithm of `compactGroup`, we thus only check whether siblings share the same sequence patterns. **Fig. 10** shows the `compactGroup` algorithm which is of a fairly simple process.

After the tree matching process, we have identified both the *common part* (by sequence matching) and the *regular part*. Thus the rest of the Web page is unmarked and is the content part needed. By simply returning this part, we get the Web news content as desired.

```
Algorithm compactGroup(groups)
01 int i=1, j
02 while i<sizeof(groups) do
03    j = i+1
04    while j<sizeof(groups) do
05      if patternMatch(groups[i], groups[j]) then
06          j++
07        else break
08      end if
10    end while
11    i++
12 end while
```

Fig. 10. Algorithm of Compacting Group

6 Empirical Evaluation

As part of the proposed approach, we have built a prototype system for Web news content extraction, upon which empirical evaluation has been conducted. In this section, we describe the implementation of the prototype system, testing data bed used, evaluation method and evaluation results, and explanations of the results.

The Prototype System: All the proposed algorithms in this paper, as well as the *TSReC* data structure of our Web news content extraction system, are implemented in Java with the help of HTML Parser [18]. The prototype system accepts two Web news content pages (one to be extracted, one as the reference) sharing the same template, and returns the HTML fragment of Web news content. As we discussed in previous section, finding two Web news pages of the same template is not hard, and in fact, we just take the one having the most similar URL to the reference page's URL. We reserve the output in HTML form, which can be directly delivered to indexing facilities for parsing and indexing.

Testing Data Bed: The testing data bed used in our evaluation is manually crawled from real-life Web news sites. We have collected news pages from up to 50 Web news sites, covering news of politics, business, sports, entertainment, and life. For each Web news site, we randomly collect one page, then running a script to get from the related links the Web news page sharing probably the same template with the source. The 50 Web news sites, which are 25 in English and 25 in Simplified Chinese, are selected from the online categories of famous search engines (google.com and baidu.com). The names of the Web sites are given in **Table 1**.

Evaluation Method: As an empirical evaluation, our evaluation is conducted in the following steps. We first extract Web pages from each Web site by running our crawler. Then we manually check whether the result is the content of Web news. We take the content part as the content of Web news according to the definition in section 3, while the semantics of text is not considered.

Evaluation Results and Explanation: The result of evaluation is as shown in **Table 1**. For each Web site, we list out the URL[ψ], as well as two evaluation results (R1 and R2). R1 and R2 can be of the value S or F: S means that the extraction is successful and the content is correctly extracted (as judged by manual checking); F means that the extraction fails, which may be caused by various reasons.

In particular, F1 means that the extraction failed with no output. The reason is that the *content part* is incorrectly identified as the regular part, for its content may be simply a sequence of text sentence (no highlighted sub section titles, no hyperlinks for key words). So the `compactGroup` function takes a wrong action on it. This kind of situation is somehow common as there are 5 cases in our data set. However, it is easy to be handled by using a set of simple heuristic rules, such as by judging how long the text is (eg., when it exceeds specific threshold, we stop matching it). Actually, those simple heuristic rules are actually applied, as shown by the later results (R2).

F2 means that the extraction failed because the HTML source of the Web news page contains some invisible text, thus affecting the matching process. These texts are usually for Dynamic HTML effects, such as popup menu. By using a simple heuristic to remove it from the original source, the problem is easily fixed.

However, there are still error types F3 and F4 for which currently we do not have a solution. Specifically, F3 means that there are regular parts not in rigid forms, and F4 means that even the regular parts are highly dynamic. These errors do not follow the assumption we made before, therefore corrupt the extraction process. Fortunately, these errors are not common, and they are left as a future work.

Overall, as we can see from **Table 1**, the accuracy of our method is generally high (19/25=76% and 21/25=84% before applying the simple heuristic rules, and after is 23/25=92% and 24/25=96%). Given that our method does not rely on any a threshold nor machine learning knowledge, it is practically feasible and suitable to be applied in real life search engines.

[ψ] Due to the space limitation, however, the URLs are only indicative in Table 1. Readers are referred to [19] for the detailed URLs.

Table 1. The Result of Evaluation On Real-life Web News Sites

	Chinese Web News			English Web News		
	URL	R1	R2	URL	R1	R2
1	news.sina.com.cn...	S	S	news.yahoo.com...	S	S
2	beijing.qianlong...	F_3	F_3	news.yahoo.com...	S	S
3	news.qq.com...	S	S	www.cbc.ca...	S	S
4	politics.people....	F_2	S	www.cbc.ca...	F_3	F_3
5	news.tom.com...	S	S	www.cnn.com...	S	S
6	news.xinhuanet.c...	S	S	www.cnn.com...	S	S
7	www.gmw.cn...	S	S	www.msnbc.msn.co...	S	S
8	news.163.com...	S	S	www.msnbc.msn.co...	S	S
9	news.espnstar.co...	S	S	today.reuters.co...	S	S
10	www.southcn.com...	F_1	S	today.reuters.co...	F_1	S
11	news.21cn.com...	S	S	www.un.org...	S	S
12	heilongjiang.nor...	S	S	www.un.org...	S	S
13	www.cnhan.com...	F_1	S	www.cbsnews.com...	S	S
14	gb.chinabroadcas...	S	S	www.cbsnews.com...	S	S
15	www.yzdsb.com.cn...	S	S	articles.news.ao...	S	S
16	sports.sohu.com...	S	S	articles.news.ao...	S	S
17	news.sports.cn...	F_2	S	www.usatoday.com...	F_1	S
18	www.chinanews.co...	F_4	F_4	www.usatoday.com...	S	S
19	economy.enorth.c...	S	S	today.reuters.co...	S	S
20	news.17173.com...	S	S	today.reuters.co...	S	S
21	www.daynews.com....	S	S	www.latimes.com...	S	S
22	sports.nen.com.c...	S	S	www.latimes.com...	S	S
23	www.lanews.com.c...	S	S	www.heraldsun.ne...	F_1	S
24	news.huash.com...	S	S	www.heraldsun.ne...	S	S
25	www.jhnews.com.c...	S	S	smh.com.au...	S	S
A		19/25	23/25		21/25	24/25

7 Conclusion

In this paper we have studied the Web news content extraction problem and proposed an automated extraction algorithm for it. Our algorithm exhibits a hybrid method applying both sequence matching and tree matching techniques. Built on top of *TSReC* – a variant of tag sequence previously proposed by ourselves, the hybrid method benefits from combining the advantages of both sequence matching and tree matching. Empirical evaluation conducted shows that our method is highly effective and efficient. Because of the automatic nature of our method, it should be fairly straightforward for us to integrate it into a real life search engine, such as Yahoo! Or Google, as a preprocessing procedure to improve indexing quality. In addition, we plan to further improve the algorithm and enhance our prototype system to make it more robust, able to handle more types of errors as mentioned in section 6.

References

1. Reis, D. Golgher, P., Silva, A., Laender, A. Automatic Web news extraction using tree edit distance, WWW-04, 2004.
2. Crescenzi, V., Mecca, G. and Merialdo, P. RoadRunner: Towards automatic data extraction from large web sites. VLDB-01, 2001.
3. Arasu, A. and Garcia-Molina, H. Extracting Structured Data from Web Pages. SIGMOD-03, 2003.
4. Zhai, Y., and Liu, B. Web data extraction based on partial tree alignment. WWW-05, 2005.
5. Bing Liu and Yanhong Zhai. NET - A System for Extracting Web Data from Flat and Nested Data Records., WISE-05, 2005
6. Muslea, I., Minton, S. and Knoblock, C.. A hierarchical approach to wrapper induction.. Agents-99, 1999.
7. Wang, J., and Lochovsky, F. Data extraction and label assignment for Web databases. WWW-03, 2003.
8. Chang, C. and Lui, S-L. IEPAD: Information extraction based on pattern discovery. WWW-10, 2001.
9. Zhao, H., Meng, W., Wu, Z., Raghavan, V. and Yu, C. Fully automatic wrapper generation for search engines.. WWW-05, 2005.
10. Alberto H. F. Laender, Berthier A. Ribeiro-Neto, Altigran Soares da Silva, and Juliana S. Teixeira. A brief survey of web data extraction tools. SIGMOD Record, 31(2):84–93, 2002.
11. Bing Liu, Robert Grossman, Yanhong Zhai. Mining Data Records in Web Pages. KDD-2003, 2003
12. Yunhua H., Guomao X. , Ruihua S., Guoping H., Shuming S., Yunbo C., and Hang L., Title Extraction from Bodies of HTML Documents and its Application to Web Page Retrieval, SIGIR2005, 2005
13. D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. Vips: a vision-based page segmentation algorithm. Technical Report MSR-TR-2003-79, Microsoft, 2003.
14. Can Lin, Zhang Qian, Xiaofeng Meng, and Wenyin Liu. Postal address detection from web documents. In WIRI, pages 40–45, 2005.
15. Crescenzi, V., Mecca, G. and Merialdo, Wrapping-Oriented Classification of Web Pages. SAC2002, 2002
16. Bing Liu, WISE-2005 Tutorial: Web Content Mining. WISE2005, 2005
17. Q.Li, B.Moon. Indexing and Querying XML Data for Regular Path Expressions. VLDB 2001, 2001
18. Dhaval Udani, HTMLParser project, available at http://sourceforge.net/projects/htmlparser.
19. Yu Li, Evaluation of Hybrid Extraction Method, available at http://idke.ruc.edu.cn/hybrid.

In Proceedings of the Seventh International Conference on Web-Age Information Management(WAIM2006), pages 263-274, Hong Kong, China, 17-19 June, 2006. Lecture Notes in Computer Science 4016, Springer 2006

RecipeCrawler: Collecting Recipe Data from WWW Incrementally

Yu Li[1], Xiaofeng Meng[1], Liping Wang[2], and Qing Li[2]

[1] {liyu17, xfmeng}@ruc.edu.cn
School of Information, Renmin Univ. of China, China
[2] {50095373@student, itqli@}.cityu.edu.hk
Computer Science Dept., City Univ. of Hong Kong, HKSAR, China

Abstract. WWW has posed itself as the largest data repository ever available in the history of humankind. Utilizing the Internet as a data source seems to be natural and many efforts have been made. In this paper we focus on establishing a robust system to collect structured recipe data from the Web incrementally, which, as we believe, is a critical step towards practical, continuous, reliable web data extraction systems and therefore utilizing WWW as data sources for various database applications. The reasons for advocating such an incremental approach are two-fold: (1) it is unpractical to crawl all the recipe pages from relevant web sites as the Web is highly dynamic; (2) it is almost impossible to induce a general wrapper for future extraction from the initial batch of recipe web pages. In this paper, we describe such a system called *RecipeCrawler* which targets at incrementally collecting recipe data from WWW. General issues in establishing an incremental data extraction system are considered and techniques are applied to recipe data collection from the Web. Our *RecipeCrawler* is actually used as the backend of a fully-fledged multimedia recipe database system being developed jointly by City University of Hong Kong and Renmin University of China.

1 Introduction

WWW has posed itself as the largest data repository ever available in the history of humankind, which also is highly dynamic as there are web pages created and/or deleted on a daily basis. Utilizing WWW as a data source seems to be natural and many efforts have been made according to the literatures. However, devising generic methods for extracting Web data is a complex (if not impossible) task, because the Web is very heterogeneous as well as there are no rigid guidelines on how to build HTML pages and how to declare the implicit structure of the Web pages. Various systems, either prototypes or commercial products, try to solve the problem in two specific domains: (1) data intensive pages (such as the search results on Amazon) usually generated by online database search engines, and (2) data record pages (such as a single product page on Amazon) usually for product descriptions. The main difference between the two domains is that in the former case, there is more than one data record in each page whereas in the latter case, there is only one record in each page. Furthermore, data records of the first case share a common keyword since the web page is generated by a search engine, but for the second case the web pages usually share the same page template as they are formatted by a web page generator.

In this paper we focus on the latter case through building a robust system to collect structural data from WWW continuously. It is a part of a collaborative project between Renmin University of China and City University of Hong Kong, the goal of which is to build a fully-fledged multimedia recipe database by collecting as many recipe web pages as possible. We extract the data records from the collected recipe pages which will be later on used in a multimedia database application–*RecipeView* (Fig.2). Generally speaking, recipe web pages are very similar to online product web pages in that (a) one web page contains only one record, (b) they follow an underlying template, and (c) there are many optional attributes. Some examples of recipe pages are shown in Fig.1. Thus by applying existing techniques, which are roughly classified into two categories–wrapper induction and automatic extraction, our goal may be achieved. However, this turns out to be a non-trivial task because of the following reasons:

- It is unpractical to crawl all the recipe pages from a web site. In Fig.1(c), there is an example of a recipe category list. The webmaster will add/update some new recipe links (shown in red circle) while updating other links such as advertisements and activities. Naive crawling of all updated links will not only lead to an inefficient strategy but also impact the latter steps by introducing some noisy web pages. Thus we have to consider how to identify real recipe links while crawling pages incrementally.
- It is almost impossible to induce a general wrapper with initial batch of recipe web pages. Because of the continuous updating of recipe web sites, the changes of the underlying schema may cause the existing wrapper broken. For example, Fig.1(a) is a typical recipe web page when the web site was created. It only contains a name, a picture, a material list, a seasoning list and some cooking steps. As time elapses, the webmaster provides us with some new recipes, one of which is shown in Fig.1(b). Because some complex new optional attributes are added (e.g. two styles of sauce in Fig.1(b)) and the existing attributes are revised (e.g. seasoning turns to be repeatable), such of these variations not only cover simple representation changes, but also involve serious schema evolutions, which definitely makes conventional extraction techniques inapplicable.

Due to these observations, our approach is to build a system (called *RecipeCrawler*) that can automatically extract relevant content data, and be able to do so incrementally so that the new web pages containing new recipe records may be added dynamically. To this end it must support the following incremental features in extraction of newly crawled web pages from the recipe websites.

1. **Incrementally crawling specific web pages**. In our system, some web data sources, such as recipe web site's categories, recipe blog pages, or even recipe online forums, are monitored. Whenever the links are updated, crawler should not only grab the web page pointed by the link, but also justify whether it is the one we need. It is possible as we have some extracted recipe data records, which can give us the domain knowledge of recipes.
2. **Incrementally extracting web pages for data records**. Either wrapper based or automated method faces the problem of web site's schema evolutions. The extraction program should not only be able to adapt itself to meet the schema revision,

Fig. 1. Examples of Recipe and Category List Web Pages

but also be able to identify new attributes. This is important to help applications which rely on the extraction system to be of more concrete, useful, and valuable services. And it also enables the extraction system to be a reasonable and practical web data extraction system.

By putting all things together, we aim to build our system as a practical robust system which supports incremental automated data extraction. It is different from existing systems in that novel modifications are made upon the tradition architecture. In a nutshell, our contributions in this paper include: 1) a framework for building incremental web data extraction system, which is implemented in our prototype system for collecting recipe data from WWW incrementally; 2) solutions for adopting and adapting existing data extraction techniques under incremental scenarios.

In this paper we describe our *RecipeCrawler* system in detail. The rest of this paper is organized as follows. In section 2, we briefly review some existing techniques on web data extraction. Section 3 gives out an overview of *RecipeCrawler*. Section 4 and 5 discuss our main considerations in designing and implementing each component. Finally we give out a conclusion and future works in section 6.

2 Relate Work

One of the reasons why the Web has achieved its current huge volume of data is that a great and increasing number of data-rich web sites automatically generate web pages according to the data records stored in their databases. Taking advantage of this fact, several approaches have been proposed and systems have been built to extract these data in literature. Generally these systems fall into two categories: wrapper induction versus automatic extraction.

With wrapper induction techniques, some positive web pages are selected as positive examples and then wrappers are trained. Though using wrappers to do continuous extraction is possible, wrappers may expire in future [6]. Thus wrapper maintenance problems arose and efforts were paid in solving it. However, to our knowledge, it assumes that there are only few small changes in web pages' representation whereas in fact the underlying schema may change [8], such as:(1) attributes that have never appeared in previously extracted pages may subsequently be added; (2) attributes appeared in previously extracted web pages may later be removed. These can cause the templates induced from existing web pages to be invalid, thus intuitive extraction strategies can not be applied. Therefore wrapper induction is not practical towards long-time, continuous data extraction.

On the other hand, as automatic extraction techniques can automatically extract structural data without doing wrapper maintenance from web pages, it becomes more popular recently years. The first reported work on automated data extraction was done by Grumbach and Mecca [5], in which the existence of collections of data-rich pages bearing a similar structure (or schema) was assumed. In RoadRunner [3], an algorithm was proposed to infer union-free regular expressions that represent page templates. Unfortunately, this algorithm has an exponential time complexity hence it is impractical for real-life data extraction systems. Then Arvind and Hector [1] proposed an improved version with a polynomial time complexity by employing several heuristics. Both of these works view web pages in HTML as a sequence of tokens (single words and tags), so when it comes to infer a template from complex web pages with many nesting structures, their solutions are still inapplicable. Other researchers have tried to solve the automated data extraction problem by viewing web pages as a long string, through employing similar generalization mechanisms (e.g., [2] and [10]). Be aware of the tree structure of web pages, [9] and [11] presented techniques based on tree edit distance for this problem. Both of them utilize a restricted tree edit distance computation process to find mapping between two web pages and then do future data extraction. In [9], wildcards are attached to tree nodes and heuristics are employed when there is a need to generalize them. In [11], a more advanced technique named partial tree alignment was proposed, which can align corresponding data fields in data records without doing wildcards generalizations. In our system, we use a similar technique and make it applicable under incremental data extraction.

While some major works have been done on clustering or classifying web pages, few of them are on automated data extraction as far as we can see from the literature. In [4], several web page features were proposed for wrapper-oriented classification. In the news extraction system [9], a hierarchical clustering technique was proposed to cluster web pages according to their HTML tree structures. A basic distance measure-edit distance is calculated by comparing two HTML DOM trees, which can tell us how similar the two web pages are. When the web page contains more than one data record, there is almost no need to do the clustering. But new problems do arise. For example, how to identify data regions containing data records in such kind of web pages is a problem. In particular, several strategies have been proposed in [7] and [12].

Combining these existing automated data extraction techniques may lead us to a generic system that is able to crawl, cluster and extract structured data from a whole

web site once for all. For our recipe collection scenarios, we need to continuously collect recipe data from the web, hence modifications to such techniques or other novel techniques are needed. In the rest of this paper we show our approach to build an incremental data extraction system by adopting and adapting the existing web data extraction techniques.

3 *RecipeCrawler* - a Recipe Data Collection System

Starting from this section, we will discuss the general considerations on how to build a system to support incremental features in conventional architecture by introducing our recipe data collection system. As Fig.2 illustrates, general architecture of current existing extraction systems were applied. Besides adopting and adapting the classic components such as web crawler, web data extractor and annotator, a new component called "*Monitor*" is advocated to keep a close watch on recipe sources. Instead of digging into the details on how it is designed and implemented as well as how it supports incremental features, in this section we would like to give an overview on how recipe data are collected.

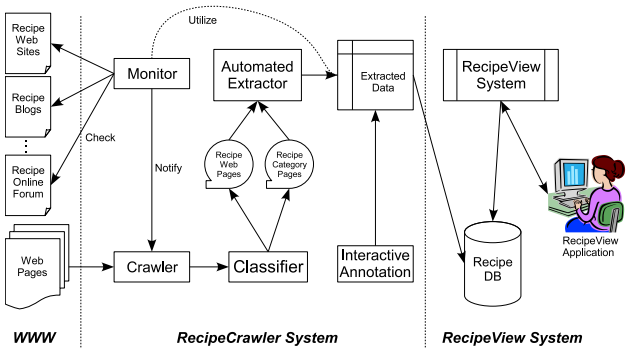


Fig. 2. Recipe Data Extraction System - An Overview

The mission of *RecipeCrawler* is to provide *RecipeView* with the recipe data records which are embedded in web pages. Here *RecipeView* is a user-centered multimedia view application built on top of the recipe database and means to provide user continuous, flexible user experience. It requires the extraction system (viz. *RecipeCrawler*) to be incremental because it needs recipe data updated every day on WWW.

Fig.2 shows an overall picture on how *RecipeCrawler* works. In particular, we incrementally grab recipe web pages by monitoring some data sources, which are as shown in the left part of Fig.2, including recipe web sites, recipe blogs and recipe online forums et. al. Considering that their indices are usually accessible (such as category lists in recipe web sites, taxonomy pages in recipe blogs and archive lists in recipe online forums), we establish a module called "*Monitor*" to find out the updated links from these sources. In order to identify whether the specific updated link is just the one we need,

extracted data has been used as domain knowledge to do data clarifications. And survivors, which are definitely the ones we need, are sent to "*Crawler*" which does basic crawling as well as validation and repairing on HTML pages.

Next the crawled web pages are delivered to the "*Classifier*" which puts pages into different categories. In this procedure, an algorithm proposed in [9] has been adopted and adapted to classify web pages according to their underlying structures (or underlying template). Two categories–"*Recipe Web Pages*" and "*Recipe Category Pages*" are derived through this step, where the former one usually contains the detailed information of each recipes and the latter one usually maintains taxonomy of recipes.

In the extraction procedure, web pages in each category are processed by an "*Automated Extractor*" and thus category information and recipe data are retrieved. Annotation was done by a module named "*Interactive Annotation*" which is operated by human, who tells the system what attribute is about what. As our system means to work in incremental way, being able to handle schema changes is critical so we proposed a method by adopting algorithms in [11]. We will further discuss it in section 5 as well as the mechanism of annotation process. So finally we get the desired data with corresponding annotations and thus can import them into DBMS for future applications, which is *RecipeView* in our case.

Before we go to the sections that discuss the details of each component, we want to emphasize the incremental nature of *RecipeCrawler* again. Incremental features in *RecipeCrawler* are the basic requirements and also the significant differences comparing with other systems. Though there is an initial web page set, which can be extracted before the *RecipeView* system is established, we can not guarantee that the wrapper induced or the schema learnt in them will always be valid for future cases, because we can not naively believe the webmaster will always update recipes activities, or assume the schema will not change. In other words, our *RecipeCrawler* should face the very dynamic perspective of WWW and the only choice is to make sure that each component of our system has the ability of doing incremental update.

4 Retrieving Recipe Web Pages

Monitoring, crawling and classifying procedures in *RecipeCrawler* are implemented to retrieve recipe web pages. In this section we mainly focus on the mechanism of monitoring and classifying procedures whereas crawling procedure is omitted because its implementation is fairly simple and straightforward.

4.1 Monitoring Recipe Data Sources

Recipe data sources on WWW usually have an index facility, such as category lists in recipe web sites, taxonomy pages in recipe blogs and archive lists in recipe online forums and so on. Monitoring them for updated recipe links generally should (1) find out whatever new/updated links, and (2) identify whether they are recipe-related links or not. The former step is easy by simply comparing current web page with history version whereas the latter one is complicated. The link discovery procedure of conventional crawler usually does simple identifications based on several rules, such as URL

domains, file types and so on. Few works are done on semantic link discovery because: (1) crawlers are usually of general use; (2) insufficient domain knowledge can be utilized to do it. However, in *RecipeCrawler*, we focus on recipe web pages, concerning not to introduce noisy web pages to subsequent procedures; we can even have domain knowledge by analyzing the extracted data of the initial set, which can always be selected out when first time we crawl the web site. With these characteristics in mind, we proceed to present a semantic link discovery method.

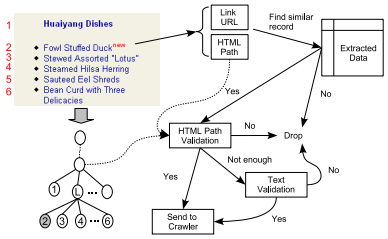


Fig. 3. Identifying Recipe Links Based on the Extracted Data - An Example

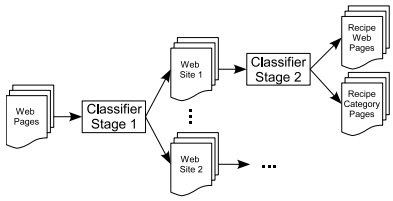


Fig. 4. Classifying Recipe Web Pages

As illustrated in Fig.3, our strategy of identifying recipe links on the basis of the extracted data works as follows. First, the current index of a web page is compared to the old one. In this way, the updated links, texts and HTML paths can be retrieved. For example, “Fowl Staffed Duck”(in short, FSD) with its link and HTML DOM path can be retrieved. Secondly we try to find records in extracted data which have similar links, as machine does not know whether it is a recipe link. Two links are similar if we can find a common pattern in them (In our system, we uses common URL prefix). Only considering URL pattern is sometimes not enough as there are still some links such as activities may survive. Therefore we utilize HTML paths and texts for further clarifications. After finding out similar records of a specified link, we first check how many records residing in the same subtree according to HTML paths. Referring back to the example, as we have FSD’s DOM path, we also know similar records’ DOM path (which are recorded in last time’s extraction), by finding common parent nodes, such as “L” node of the DOM tree in right bottom corner of Fig.3. Note that we only give a simple DOM tree here due to the space reason, in which number denotes the content. If we can not find any, this link is probably not a recipe link so we discard it. If we can only find few (in our system, we use 0.5 as the threshold, which means half out of total records), the text is used as the third judgment, which is simple keyword matching in our system, in the hope of finding common recipe keywords(such as “Beef”, “Pork” and so on). If most records reside in the same subtree, we let the link survive. Figure 3 illustrates the whole process we have just described, which, based on our practice, has been quite effective and efficient.

4.2 Classifying Recipe Web Pages

In the next step, we build a module "*Classifier*" to handle the web page classification. The classifier program in our system has two stages, as shown in Fig.4. In the first stage we organize the web pages according to URLs, thus obtaining categories of web sites. This stage is relatively easy. Next we further classify crawled web pages according to the tree structures. A clustering algorithm based on tree edit distance [9] has been adopted and adapted. As mentioned before, recipe web pages in our scenario may contain repeatable attributes, so we have modified the matching process to cover repeatable cases. It is called sibling matching which is also used in automated extraction procedure and the details will be given in section 5.1. After classification we will get two categories, namely recipe web pages and recipe category pages, for each web site. Subsequent extractions will be done in these categories.

The classification procedure is in nature incremental for cases where there are no big changes in page templates. When a template (or structure) changes a lot, a new initial data set needs to be generated so that a new classification process can proceed.

5 Retrieving Recipe Records

We now describe how *RecipeCrawler* retrieves recipe data from the crawled recipe web pages. There are two modules involved, namely "*Automated Extrator*" and "*Interactive Annotation*". Though they do different functions in retrieving recipe data, there is no rigid execution order. In *RecipeCrawler*, they are actually invoked asynchronously. Fig.5 gives an illustration on how these two modules cooperate with each other. The *Automated Extractor* continuously does extraction on web pages while the *Interactive Annotation* is notified each time new attributes are identified. *Automated Extrator* will generate two data tables, namely "*Recipe Data*" and "*Category Data*", from recipe web pages and recipe category pages, respectively. Each table may contain some new attributes during the incremental extraction. Thus an execution of annotation procedure is needed. Then we select data fields that have been annotated from these two tables, and join them according to URLs. Finally data is extracted and ready to be imported into DBMS.

5.1 Automated Extraction

In this module, we adopt techniques proposed in [11] for automated extraction. As reported in [11], an algorithm named *partial tree alignment* based on the simple tree matching was used to extract data records in data intensive web pages, such as result pages returned by online retailer web sites. The recipe category web pages in our system are also data intensive web pages, so data records can be directly extracted by applying this algorithm. But we need to modify it in order to extract new/updated records in it for supporting incremental features. This can be done by comparing currently extracted results to the former ones, so the details are omitted here.

On the other hand, extracting data from recipe web pages is not so easy. It is a non-trivial problem because: (1) attributes that have never appeared in previously extracted

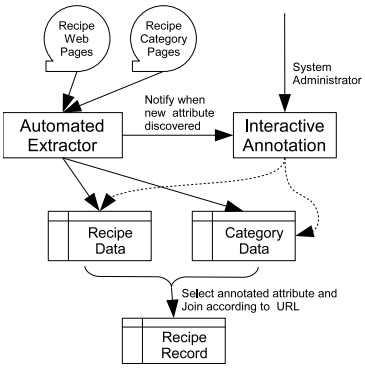


Fig. 5. Retrieving Recipe Data from Web Pages

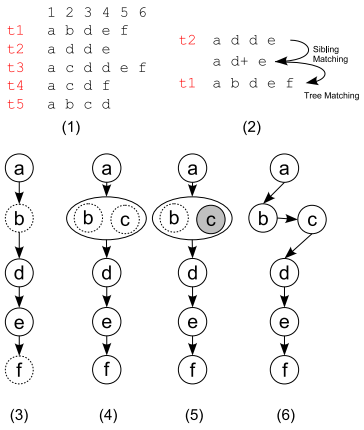


Fig. 6. Illustration of How Automated Recipe Data Extraction Works

pages may subsequently be added; (2) attributes that appeared in previously extracted web pages may later be removed; (3) attributes that appeared as singleton in previously extracted web pages may be modified to be repeatable. For example, referring back to Fig.1, the "sauce" attribute appearing in Fig.1(b) is an example of added attributes, and the "seasoning" attribute appearing both in Fig.1(a) and Fig.1(b) is an example of revised attributes, which later can be repeatable. There is no example of removed attributes in Fig.1, but it is easy to give out: any optional attribute can be it when we start from web pages containing it to web pages without it. Though the technique proposed in [11] can roughly handle these situations by selecting and starting from the maximal web page in the hope of that it contains as many optional nodes as possible, it is unfortunately inapplicable in our incremental crawling scenario. So we have adapted it to fulfill the incremental requirements.

Instead of explaining the detailed algorithm used by *RecipeCrawler*, we give an illustrative example in Fig.6 to show how it works. We suppose there are 5 recipe web pages, and to be simple, we present them in simple characters sequence, in which each character denoting a subtree directly contains text values, such as "<LI>Materials:
Beef 150g</LI>" . We can get the sequence by specific traversal of HTML tree [11], and we use pre-order traversal here. According to [11], partial tree alignment first selects the biggest web page as the seed and then do multiple tree alignment. In our example, the biggest one is t3. But in an incremental situation, t3 may not be in the initial set because it is not created by any webmaster at all. In our example, we assume that the initial set has t1 t2, whereas t3, t4 and t5 are added subsequently.

For the initial set we apply the partial tree alignment technique. First we do a sibling matching (as shown in Fig.6(2)), which is used to handle repeatable attributes ("d" in t2). The sibling matching scans each tree and tries to match siblings in it. If two sibling nodes match, they will be replaced by a single example node (we simply take the first one). We do not consider non-sibling nodes because usually a list of repeatable

attributes will not be interrupted by other attributes. (For example, the webmaster will not insert some cooking steps in the middle of listing materials.) And the sibling matching performs whenever we match a web page to another (as well as the template, see below). After doing that we make the tree matching based on the edit distance computation to find mappings. By taking the biggest one t2 as the template, we can align t1 to it and by applying partial tree alignment techniques [11] we can also align optional nodes. The basic idea of partial template alignment is trying to find the unique insertion location for each unmatched nodes. In our example, “b” of t1 is unmatched, but we can find a unique insertion location in t2 for it, because “a” and “d” are matched and there is nothing between them in t2. So “b” should be inserted between “a” and “d” in t2 to form a template. After inserting all optional nodes as proven in [11], recipe data is extracted and a template (shown in Figure 6(3)) is induced. Then an annotation process may be invoked, but at this time we are not sure that the nodes “b” and “f” are the data attributes we need (they can be useless values such as “copyright by” et. al.). Another reason is that they may be disjunctions as we have only few instances. So, in our example, we simply suppose no annotation in it, so actually we only extract “a”, “d” and “e”.

Now we come to the part of incremental extraction. Supposing that t3, t4 and t5 will be updated and crawled one by one, Fig.6(4,5,6) shows how the extraction is done. The basic idea is to match new crawled web pages with the existing template and insert the unmatched nodes into the template. When there is no unique insertion location for the specific node, we insert it by merging it as a possible value into a possible node. In our example, when t3 comes, we find that “c” does not have a unique insertion location as there is already an unannotated “b” between “a” and “d”, so we merge “c” as a possible value into “d”, thus the template can cover t3 (as shown in Fig.6(4)). At this time t3 can be partially extracted with some part left in the induced template, which may be further matched or annotated (extraction process will give annotation process a notification at this time). Another node, say “f”, matches with the one in the template, thus we have enough instances to identify “f” as an attribute and both “f” nodes in t2 and t3 will be extracted.

After processing t3, t4 comes in subsequently. This time we match it with the template too. The difference is that when matching with node “b c”, we need to match two times to find the best one. We can see that “c” will be matched thus attribute “c” will be identified. But we can not take it out from the “b c” node for there is still no unique insertion location. The template after matching and extracting t4 is as be seen in Fig.6(5). After t5 comes, matching with t5 will identify the attribute “b” too. And the order of attributes “b” and “c” can be identified since we have t5 as the instance (there is a “b” “c” sequence in t5). Thus all attributes are identified and can be extracted. The induced template is shown in Fig.6(6). Next time when new web pages come in, the same processing techniques can be used.

Note that currently we do not consider disjunctions in our strategy due to two reasons. Firstly, disjunctions are actually not that serious when we are doing incremental extraction. By using following web pages as examples (Fig.6(6)), identifying whether there are disjunctions is easy. Secondly, the chances of disjunctions making our strategy broken are fairly few. For example, considering a web page t6(“a c b d e”), our strategy would break while handling it. But this is rare because t6 means that web master

changes the order of attributes (such as giving “cooking steps” before “materials”). It is almost unlikely and we did not find any cases in our practice, so we leave this problem to be a possible future work.

5.2 Interactive Annotation

Currently in *RecipeCrawler* the annotation procedure is designed as an interactive program. It can be asynchronously invoked by a system operator while the system does automated extraction. The template induced by automated extraction will be presented to the operator for annotation instead of requiring him to do annotation on each record. When a new attribute is identified, a notification will be given. Then the system operator can check the revised template and examples to decide what kind of attribute it is. Having annotations made to the extracted recipes and category data, they will be selected out and joinned based on URLs to generate the final extraction results. Unannotated data will be reserved in the extracted data storage for future annotation. This mechanism ensures us to be able to incrementally extract meaningful recipe data for *RecipeView* as soon as newly crawled web pages come in. In our practice, we perform the interactive annotation when the initial set was extracted and when enough (e.g., 10) new web pages are extracted. The current practice of *RecipeCrawler* shows that such an approach is quite reasonable and effective.

5.3 Importing Recipe Data Records

As shown in Fig.2, the extracted recipe data records by *RecipeCrawler* are to be utilized by a front-end application system called *RecipeView*. Since the retrieved recipe data records come from various sources, they should go through an importation procedure before they can be fully utilized. This procedure is called “Preprocess” in *RecipeView*, which involves *Filtering* and *Standardization*. The *Filtering* module makes sure that all the recipe records are qualified for the system requirements (e.g. by checking whether the data fields of each record are correctly identified). In the *Standardization* module, all the recipe records have to conform to a standard presentation by fusing different data formats together. For instance, the display sequence of the data fields in each record must be the same. Thus they become uniform and consistent. After the “Preprocess” procedure, the recipe data records are imported into an underlying DBMS for possible user manipulations within the *RecipeView* system.

6 Conclusion and Future Work

As we believe, building incremental data extraction is a critical step towards practical, continuous, reliable web data extraction systems that utilize WWW as the data source for various database applications. In this paper, we have described such a system (viz., *RecipeCrawler*) which targets at incrementally collecting recipe data from WWW. General issues in establishing an incremental data extraction system are considered and techniques applied to recipe data collection from the Web. Our *RecipeCrawler* has served as the backend of a multimedia database application system (called *RecipeView*)

and offers good experimental results. Various techniques proposed in literature for data extraction from WWW are adopted and adapted to do the automated recipe data extraction as well as to support incremental features. As for future research, besides evaluating and improving our system, we also plan to address other important issues, including better crawling strategies and automated annotation algorithms.

7 Acknowledgments

This research was partially supported by the grants from the Natural Science Foundation of China under grant number 60573091, 60273018; the National 973 Basic Research Program of China under Grant No.2003CB317000 and No.2003CB317006; the Key Project of Ministry of Education of China under Grant No.03044 ; Program for New Century Excellent Talents in University(NCET).

References

1. A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In *Proceedings of the* 22^{th} *ACM SIGMOD International Conference on Management of Data*, pages 337–348, 2003.
2. C.H. Chang and S.C. Lui. Iepad: information extraction based on pattern discovery. In *Proceedings of the* $10th$ *International World Wide Web Conference*, pages 681–688, 2001.
3. V. Crescenzi, G. Mecca, and P. Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of* 27^{th} *International Conference on Very Large Data Bases*, pages 109–118, 2001.
4. V. Crescenzi, G. Mecca, and P. Merialdo. Wrapping-oriented classification of web pages. In *Proceedings of the* 17^{th} *ACM Symposium on Applied Computing (SAC)*, pages 1108–1112, 2002.
5. S. Grumbach and G. Mecca. In search of the lost schema. In *ICDT '99*, pages 314–331, 1999.
6. N. Kushmerick. Wrapper verification. *World Wide Web*, 3(2):79–94, 2000.
7. B. Liu, R. L. Grossman, and Yanhong Zhai. Mining data records in web pages. In *Proceedings of the* 9^{th} *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–606, 2003.
8. X. Meng, D. Hu, and C. Li. Schema-guided wrapper maintenance for web-data extraction. In *the* 5^{th} *ACM CIKM International Workshop on Web Information and Data Management*, pages 1–8, 2003.
9. D.C. Reis, P.B. Golgher, A.S. Silva, and A.H.F. Laender. Automatic web news extraction using tree edit distance. In *Proceedings of the* 13^{th} *international conference on World Wide Web*, pages 502–511, 2004.
10. J. Wang and F. H. Lochovsky. Data extraction and label assignment for web databases. In *Proceedings of the* 12^{th} *International World Wide Web Conference*, pages 187–196, 2003.
11. Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In *Proceedings of the* 14^{th} *international conference on World Wide Web*, pages 76–85, 2005.
12. H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. T. Yu. Fully automatic wrapper generation for search engines. In *Proceedings of the* 14^{th} *international conference on World Wide Web*, pages 66–75, 2005.

XML 数据管理

(XML Data Management)

XML查询优化研究

孟小峰, 王 宇, 王小锋

软件学报, 卷17(10):2069-2086, Oct. 2006

基于直方图的Xpath含值谓词路径选择性代价估计

王宇， 孟小峰，王珊

计算机研究与发展，卷43 (2):2069-2086:288-294, Oct.2006

OrientX: an Integrated, Schema-Based Native XML Database System

Xiaofeng Meng, Xiaofeng Wang, Min Xie, Xin Zhang, Junfeng Zhou

Wuhan University Journal of Natural Sciences,11(5):1192-1196, Nov., 2006.(The Third Web Information System and Application(WISA2006)

XML 数据流上的有序XPath 查询处理

谢敏，王小锋， 张新，孟小峰，周军锋

计算机研究与发展，卷43(增刊)：464－470，2006,11. (第23届中国数据库学术会议，广州.)

软件学报, 卷17(10):2069-2086, Oct. 2006

XML查询优化研究*

孟小峰[1+], 王 宇[2], 王小锋[1]

[1](中国人民大学 信息学院,北京 100872)

[2](河北大学 计算中心,河北 保定)

Research on XML Query Optimization

MENG Xiao-Feng[1+], WANG Yu[2], WANG Xiao-Feng[1]

[1](Information School, Renmin University of China, Beijing 100872, China)

[2](Computer Center, Hebei University, Baoding , China)

+ Corresponding author: Phn: +86-10-62519453, E-mail: xfmeng@ruc.edu.cn, http://www.ruc.edu.cn

Meng XF, Wang Y, Wang XF. Research on XML query optimization. *Journal of Software*, 2006,17(10): 2069−2086. http://www.jos.org.cn/1000-9825/17/2069.htm

Abstract: XML has become the de-facto standard for data representation and exchange on the World-Wide Web. Due to the nature of information on the Web and the inherent flexibility of XML, we expect that much of the data encoded in XML will be semi-structured. Data on the internet is increasingly presented in XML format which enables researches on various kinds of XML storage model. Meanwhile, XML query optimization has become a hot research topic in database field. This paper gives an overview of the current status of technology for XML query optimization. The features of XML query optimization and key problems of research are also discussed deeply. Main aspects of current work on XML query optimization include XML algebra, cost model, complex path selectivity estimation, statistics information, and so on. Finally, we prospect future research directions and present some viewpoints of XML query optimization.

Key words: XML; query optimization

摘 要: XML 已经成为网络上信息描述和信息交换的标准.由于网络上信息的本质特性和 XML 数据内在的灵活性,很多用 XML 编码的数据都是半结构化的.随着 XML 应用越来越广泛,人们提出了多种 XML 数据的存储模型.与此同时,XML 的查询优化也是数据库界研究的一个重要课题.综合论述了 XML 数据查询优化技术的现状,指出了 XML 查询优化的特点和研究的关键性问题.描述了查询优化技术的各个方面的重要的研究成果和存在的问题,进一步展望了未来的研究方向,并在此基础上提出了对 XML 查询优化方法的一些观点.

关键词: XML;查询优化

中图法分类号: TP311 **文献标识码**: A

XML 已经成为网络上信息描述和信息交换的标准.早期的 XML 数据以文档方式存储,以关键字查询等信息检索手段查询,简单易用.由于缺乏系统的存储和查询机制的支持,造成查询能力低,不能满足复杂条件的查

* Supported by the National Natural Science Foundation of China under Grant Nos.60073014, 60273018 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2003CB317000 (国家重点基础研究发展规划(973)); the Key Project of Chinese Ministry of Education under Grant No.03044 (国家教育部科学技术重点项目); the Program for New Century Excellent Talents in University (教育部新世纪优秀人才支持计划)

Received 2006-01-19; Accepted 2006-04-17

询,更谈不上查询优化.一些现有的商业数据库系统扩充了处理 XML 数据的功能,利用现有数据库成熟的技术,把 XML 查询要求转变为数据库查询表达,由查询优化器优化查询表达并执行,再将查询的结果转变为 XML 数据.这种方法在一定程度上解决了查询复杂性的要求,但多级转换带来的问题是效率的降低和查询语义的混淆.

与传统的数据库数据相比,XML 数据具有下述特点:

- 数据是自描述的,内容与结构混杂在一起;
- 数据具有完整的嵌套层次;
- 数据是有序的.

XML 数据的不规则性是对传统统计信息方法的重要挑战,其数据分布情况使得一些传统的分布假设难以成立.为了达到所需的代价估计精度,需要更多的统计信息.而结构的复杂性又为获得相对精确的统计信息带来存储和计算上的困难.XML 的有序性制约了转换规则的灵活性.XML 数据的上述问题对无论是关系数据库或是面向对象数据库的现有查询优化技术都是严峻的挑战.

与传统的查询需求相比,XML 查询具有如下特点:

- 以长路径表达式为查询的核心语句,路径复杂,包含分支路径;
- 嵌套的查询表达,查询表达式中加入编程语言的嵌套和条件判断思想;
- 路径中包含不确定因素,这在之前的查询需求中未出现过;
- 查询对象和返回结果类型不确定.

面向对象数据库已有一些处理复杂长路径表达式的经验,但无法处理 XML 查询中的路径表达式中的不确定情况;关系数据库中已有很多处理嵌套查询的方法,但对掺杂编程语言风格的 XML 查询语言却难以适应.

综上所述,来自数据结构和查询需求两方面的问题导致基于关系和面向对象数据库的查询处理和查询优化技术均不能适应 XML 查询的需要.目前,对 XML 查询优化的研究正在成为热点.本文的内容就是对 XML 查询优化技术现状的综合论述,指出 XML 查询优化的特点和研究的关键性问题,描述了查询优化技术的各个方面的重要的研究成果和有待近一步解决的问题.

1 XML 查询优化研究问题

查询优化是数据库技术中重要的研究问题,是实现高效查询的关键性因素.对传统数据库查询优化的研究已经形成相对成熟技术和方法,其中基于代价的优化是主流.查询语言首先被转换成为一种内部表达形式(通常是某种代数,如关系代数等),根据变换规则得到等价表达式,计算不同形式的表达式的执行代价,然后选择一个最小的执行方案.当把这种方法用于 XML 查询优化时,研究者遇到如下问题:

(1) 完善的查询代数标准

众所周知,关系数据库统治数据管理领域长盛不衰的法宝就是描述性查询语言 SQL 和其运行基础关系代数.关系代数的目的之一是给出明确的查询语义,之二是用于支持查询优化.关系代数的优势来自简单明确的数据模型——关系,具有完善的数学基础和系统的转换规则.后来的数据模型都以关系代数为蓝本,定义了不同的运算,如面向对象数据模型等,但效果并不尽如人意.XML 数据模型本身具有的半结构化特点是定义完善的代数运算的最大障碍.而 XML 查询语言中的不确定性和一些编程思想的引入是另一个难以克服的困难.

(2) 精确的代价估计

关系模型中,表中的记录是无序的、大小相等的,代价计算时依据的一些分布假设是稳定的.而且,由于其记录大小相等,对时间的估计可以转换为对 I/O 次数的估计,进而转换为对中间结果大小的估计.而在 XML 模型中,数据是有序的,数据聚集的方式不定,每个数据的大小相差悬殊,中间结果大小与 I/O 次数之间的对应关系没有明显的规律.简单地沿用传统的代价计算方法必然导致误差的产生,从而影响精确的代价估计.

(3) 足够的统计信息

足够精确的统计信息是保证查询优化有效性的基础;缺乏足够的统计信息,是造成估计与实际情况误差的重要因素.传统的统计信息多是对值的统计,如对平均值、最值、记录个数等的统计.这些对 XML 查询是不够

的.XML 数据本身缺乏模式的支持,对数据结构信息的统计显得更加重要.XML 数据中的数值分布在类似树状结构的树叶上,即使相同类型的数据,由于半结构化特点,其分布情况也可能完全不同.因此,需要把对结构的统计信息和对值的统计信息结合到一起,才能得到足够精确的统计信息.

2 XML 查询处理结构

与传统的关系数据库系统的查询处理结构类似,我们可以将 XML 查询处理分为 4 个大的阶段.如图 1 所示:中间方框表示查询处理步骤;左侧方框为使用的相关技术或方法;右侧方框为查询优化和执行时需要的信息.

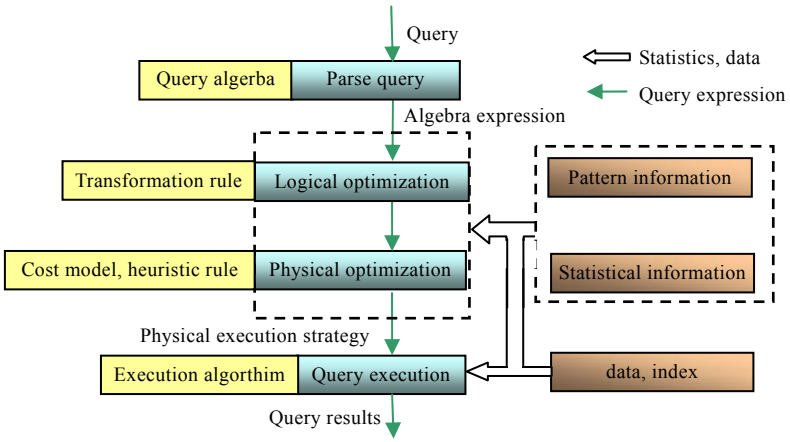


Fig.1 Query Processing

图 1 查询处理过程

第 1 阶段查询解析,将查询转换为某种内部表达方式,以便于机器处理,并为下一步的优化过程铺平道路.这种内部表达方式通常以一种抽象语法树或者查询树的形式出现.以传统数据库为基础的查询引擎的做法是转换成关系代数,然后由关系数据库优化器完成剩下的优化工作.而 Native 的数据库系统则采用不同的 XML 代数系统.我们在第 3 节中介绍目前流行的几种 XML 代数.

第 2 阶段逻辑优化,利用模式信息,规范和简化内部表达式.在这一阶段中,系统不考虑实际数据的值和数据的存储情况.同一查询请求,可以转换成不同的等价表达方式,其中有一些比原有查询更高效.为了进行这种转换,优化器需要一些转换规则,我们将在第 4 节中讨论这些转换规则.

第 3 阶段物理优化,利用代价模型和统计信息计算不同表达式,不同算法的执行代价,选择最低代价的查询计划.在这一阶段中,需要解决两个问题:确定表达式执行顺序和决定每步操作的具体算法.对于 XML 查询树而言,需要首先将查询表达式分解为可执行的片断,然后选择合适的执行顺序和执行算法并执行.中间结果集的大小,是决定执行策略是否高效的关键因素,与实际的数据分布密切相关.综合考虑数据的存储、索引和数据值的分布情况,准确地估计复杂路径选择性是其中的难点.对于一个给定的执行策略,通常会有多个可能的执行算法,产生所有的执行算法的组合造成选择本身的代价过大.因此,会有一些启发式规则用来控制其空间规模,并采用一些空间搜索技术加速选择的过程.我们在第 5 节中详细讨论.

第 4 阶段查询执行,根据物理优化确定的执行策略和算法,访问数据并得到查询结果.由于 XML 数据复杂和变化的结构,需要高效的数据访问算法.

3 XML 代数

XML 代数是对遵循一定数据模型的 XML 文档集合的操作集.XML 代数提供根据请求在文档集合中选择一个或多个文档或者文档片段的能力.XML 代数应支持对查询结果的重构.

目前对 XML 代数的研究主要集中在对查询代数的定义和从查询语言到查询代数的转换方面.查询代数定

义查询对象的类型,可以执行的操作和不同操作之间的转换规则.查询语言经分解转换为由查询代数的操作表达构成的操作树或者操作序列.不同的代数表达可以有相同的语义和执行结果,构成代价空间.

3.1 XML代数定义

目前产生很多种 XML 代数,风格各异,其主要思想来源于关系代数、面向对象代数、半结构化代数和功能化编程语言等.由于篇幅有限,不能在这里一一介绍,我们介绍具有代表性和影响力的几种.表 1 列出了不同代数之间特点的比较.

Table 1 Comparisons among XML query algebra

表 1 XML 查询代数比较

	Data structure	Document order	Node order	Reference supported	Logical operation	Physical operation	Transformation rule
AT&T	Directed graph	×	√	√	√	×	×
IBM	Directed graph	√	√	√	√	×	×
FS	Xquery data model	√	√	√	√	Seldom	√
Lore	OEM	×	√	√	√	√	√
TAX	Tree	×	√	×	×	√	×

Oracle,IBM 和 MS 联合提出的一个 XML 代数标准是文献[1].该标准把 XML 文档看作有向标记图(如果忽略引用,可以看作有向标记树).用五元组 $G(V,E,A,R,O)$表示.其中:V 表示结点,有两种类型:element 和 value.E 表示 element 到 element;A 表示 element 到 value,即属性;R 表示引用.上述 3 种为边的类型.O 表示次序.在这个模型上,规定了导航、选择、连接、构造等操作,其导航操作提供在有向标记图中的遍历操作,包括正向遍历和反向遍历;其连接操作语义类似关系代数中的连接操作,根据相同的值连接不同的文档.该代数采用类似关系代数的表达形式.

Bell Labs和AT&T Labs的Mary等人提出的XML查询代数[2],基于简化的XSLT数据模型[3],增加了引用结点,合并了属性和元素结点,删除了注释结点.其主要思想来源于曾用于半结构化和面向对象数据库的代数——嵌套关系代数,并增加了对正则表达式的操作.在嵌套关系中,数据由多个元组和多个链表组成,并可多级嵌套,其采用list comprehension方法表达导航、笛卡尔积、嵌套和连接操作.List comprehension根据一系列过滤和generator操作,得到满足条件的结果链表.Generator操作与导航操作相对应.该代数还支持结构递归等程序语言特点,用Haskell程序语言作为表达形式.

上述两种代数停留在逻辑层次,没有考虑与之相对应的物理代数和查询优化策略,其优势是具有较高的描述性和丰富的语义,与查询语言有密切的转换关系;其操作中对路径的处理并不完善,形成过多的递归结构[3]或者遍历操作[2],给下一步的优化带来困难,但其处理XML查询的方法和思路被后来的XML代数规则大量采用.

基于文献[2]、文献[3]等,W3C于 2001 年公布了一个XML查询代数标准XQuery 1.0 Formal Semantics[4],用于规范查询语言语义.该标准遵循简化的XQuery 1.0 and Xpath 2.0 Data Model[6].比照关系代数给出了XML数据模型的投影、选择和连接等操作的定义,还引入结构递归、条件判断等编程语言的概念.Formal Semantics的一个特点是代数表达与XQuery查询语言相同,成为XQuery的核心语法;另一个特点是操作由不同的层次,高层次的操作可以转换为低层次的操作.标准中还给出了少量表达式转换规则,有待近一步扩充.目前已经有了应用该标准的XML查询引擎,如Galax[6].

Standford大学开发的XML数据库Lore系统[7]针对系统本身提供的访问操作和索引情况,提出的一套独特的代数操作,包括逻辑代数、物理代数和相应的转换规则.Lore以该代数系统为基础提出了查询优化方法[8],但由于代数操作定义过多地依赖于Lore本身独特的数据存储和索引技术,该代数很难应用于其他系统.

Timber数据库[9]中应用的TAX代数[10],其数据模型为无序的树的集合,树中的数据是有序的.直接针对树和树枝规定了一系列操作无需中间结构的转换.把XML数据模式看作树,把查询语句也看作树,二者之间作模式匹配,得到满足查询树条件的结果树集合.TAX在处理连接操作时对操作的顺序未做明确规定,不适应严格要求文档顺序的情况.

XAL[11]基于集合概念构造了逻辑代数操作集,其操作分为 3 种:抽取操作从XML文档中获得必要的数据,如选择、投影等;元操作控制表达式求值过程,并非针对XML数据的抽取或者构造操作,而是为其他操作符准备输入或者控制其他操作的操作,如映射、迭代等;构造操作用于构造查询结果.XAL为查询优化提供了一组启发式转换规则.

其他如XOM代数[12],是完整的操作集,包含6种对象操作,但不支持优化;OPAL代数[13]基于半结构化数据模型,操作对象为多个链表,将有限状态自动机用于生成执行计划;还有SAL[14]等[15−17].根据代数的操作方式,可将上述不同的方法分为两种:一种是面向集合的代数,其操作对象是某种类型的集合,如树的集合、值的集合等.这种方法具有很好的优化基础,但可能丢失数据的顺序;另一种称为导航的代数,其操作对象是单个的数据.这种方法不利于进一步的优化.代数定义应是逻辑操作与物理操作的有机结合.但目前的XML代数研究或是把他们混合在一起,或是虽然分开但缺乏相应的转换规则.

早期对 XML 代数的研究工作重点在于规范 XML 查询语义,并未考虑查询优化因素,这些代数具有明显的程序化思想,很难进一步优化,只能利用遍历方法求解查询,造成查询效率的低下,不适应大规模 XML 数据的查询需求.而基于数据库思想提出的一些面向“集合”的代数,具有很好的优化基础.因此,目前查询优化的研究工作也多以这些代数为背景,但也存在问题:

首先,表达式嵌套问题.在 XQuery 查询中,由于表达式可以任意嵌套,谓词可以出现在任意地方.谓词是有作用域的,同一个谓词在不同的地方,会产生不同的查询结果.基于集合的代数需要将嵌套的查询转换为逻辑树形式,不可避免地面临嵌套结构的非嵌套化问题.虽然在关系数据库中有一些方法可以借鉴,但由于 XML 数据结构的复杂性,解决这个问题变得更加困难.

其次,XML 数据的有序性.除非查询语句特别指定,否则应该保持结点在源文档中的结点顺序.而原来一些处理连接的算法,比如排序连接、Hash 连接等,会打乱结点顺序,在连接完成后,需要对连接结果做一个额外的根据结点顺序的排序操作.如果用 nest-loop 连接算法,则可以省去这趟额外的排序.在进行代价估计的过程中,这个额外的排序操作要被考虑在内,这就给进行查询优化的过程带来新的考虑因素.一个可能的解决方法是在所有操作中,忽略结点的有序性,在最后构造结果的时候再对结点按照文档顺序排序.究竟是使用 nest-loop,还是最后增加额外的排序的方法,这是查询优化的一个研究点.

最后,不同的代数标准.在 XML 代数研究中一个值得重视的问题是:目前已经出现了一些查询代数标准,这些标准在风格上相去甚远,很难有共通性.而对执行方法的研究还远远不够,不能形成完整的系统.而且从逻辑代数到物理代数的转换也将是未来研究的一个重要的问题.

3.2 复杂路径表达式分解

目前的XML查询语言很多,有XPath[18],XQuery[19],XML-QL[20],XQL[21],Quilt[22]等.它们的一个共同的特点就是对复杂路径表达式的支持.路径表达式分解是根据一定的转换规则,把用查询语句表达的、复杂的、不确定的路径表达式转换为简单的、明确的、系统可识别的方式,如XML代数.路径表达式分解是查询转换的难点,也是查询优化的重要一步,是代价估计的前提条件.根据不同的规则,路径表达式可能分解为不同的等价形式,其中有的代价高,有的代价小,形成代价空间.路径分解的原则是能够产生有限的代价空间,有利于利用分解的结果搜索代价最小的执行方案.

在路径分解时,有两种不同的思路:一种思路是把路径细分为两两的祖先后代或父子对,如lore,XISS[23]等.这样做分解算法简单,可利用数据物理存储信息,分解的结果容易转换为结构连接运算或者运用系统提供的各种索引.如果查询语句中出现通配符(如*,?,//),可以利用索引直接定位数据,也可以借助模式信息确定通配符所代表的各种可能情况扩展路径.经过分解,路径表达式转换为连接、投影、选择、导航等不同的代数运算;另一种思路[24]是将复杂路径表达式用树的方式表示:从根开始,在树中搜索最长的确定路径(不含*或//)称为一次分解,其路径构成树的一个子串.以这个点为起点,用上述原则再分解路径,得到确定路径(子串)的集合,称为一个最小分解.在XML文档的查询中,确定路径的查询是相对容易的;而不确定路径的查询是比较困难的,尤其是在没有模式或索引的情况下,可能要将中间结果合并才能得到全部的结果.将复杂的、不确定的路径分解为确定的、

简单的路径处理.这种分解方法在没有模式信息的情况下处理不确定路径具有优势.

4 逻辑优化

传统数据库技术中,逻辑优化是指通过一系列转换规则,将原始的查询表达式转换为等价且更高效的形式.关系代数表达式求解时,操作顺序是影响效率的关键.因此,逻辑优化研究的重点并不在于对冗余操作的分析,而是对操作顺序的调整.而 XML 代数的核心是由路径表达式转换的查询树,查询效率依赖于查询树的规模.因此,查询树的最小化是 XML 逻辑优化研究的重点,也是目前研究的热点问题.而对操作顺序的调整因为更多地依赖于物理存储的情况,而与物理优化相关联.

从层次上,逻辑优化可分为两个层次:语法层次和语义层次.语法层次的优化是指不依靠任何其他信息,独立地分析查询表达式中分支或结点间的逻辑包含关系,删除冗余部分;语义层次的优化是指通过数据库提供的模式信息,如 DTD, XML Schema 等,或者语义包含、结构包含等完整性约束,查找查询表达式中的冗余分支或结点.下面的例子进一步说明二者之间的区别.若有如下的 XQuery 查询表达式:

for $*a* in reference/book,

for $*b* in $*a*/author, $*c* in $*a*/author/ name, $*d* in $*a*/author/email

where $*b* and $*c* and $*d*

return ⟨book⟩$*a*⟨book⟩

其Pattern Tree(以下简称PTQ)如图 2(a)所示,其中实心圆为查询返回结点.若数据满足$*c*,同时必然满足$*b*,$b分支对查询返回结果没有任何影响,是冗余结点.我们称这种冗余结点为语法冗余结点.经语法优化后的PTQ如图 2(b)所示.若从模式可知任一author均有name,则v_6为冗余结点.我们称这种冗余结点为语义冗余结点.经语义优化后的PTQ如图 2(c)所示.

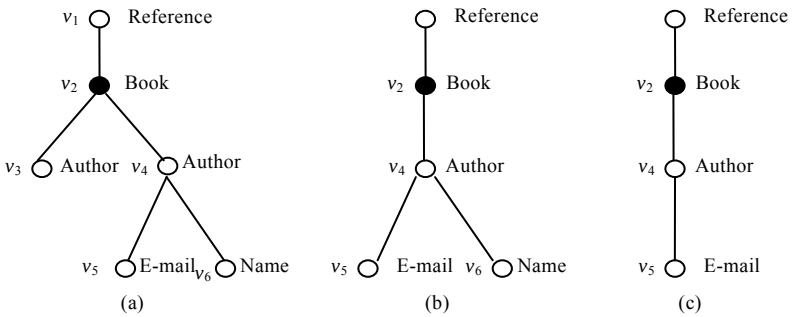


Fig.2 Syntax and semantic optimization

图 2 语法优化和语义优化

下面我们分别介绍语法优化和语义优化的研究现状.

4.1 语法优化

对PTQ语法优化问题基于对路径等价性问题[25−28]的研究.最早提出XPath最小化的Wood[29]指出:一个Xpath可以表示为合取范式,对XPath等价性检查的复杂度等价于对合取范式的等价性检查.而这已经证明是一个NP完全问题.如果对XPath路径表达式的复杂程度加以一定的限制,XPath最小化问题的复杂度可以达到多项式级.目前已经提出了对不同类型的PTQ的最小化方法.

(1) 简单 PTQ{/,[],*}

文献[29]提出了对只包含{/,[],*}的简单路径的最小化方法.其思想为:设原始 PTQ 为 *P*,在所有与之等价的 PTQ 中找到 *Q*,使得 *Q* 中的结点个数|*NQ*|最小,则 *Q* 为 *P* 的最小化 PTQ.

这种方法的关键是通过对包含映射(containment mapping)的判断完成 PTQ 的等价性判断.不存在祖先后代关系(//)的简单 PTQ 的最小化 PTQ 为其子树.查找等价 PTQ 的范围限制在其子树的范围内,是保证最小化算法

的复杂度为多项式级的关键因素.文献[23]中给出了复杂性证明,虽然并未给出算法细节.

(2) PTQ{/,//,[]}

文献[30]提出了PTQ的CIM (constraint independent minimization)算法.与文献[29]相同,其算法的基础也是包含映射.路径中缺少“*”的PTQ的最小化问题,仍旧可以在其子树范围内解决.CIM算法的思想为:从叶结点开始,判断结点在PTQ中是否冗余:若某结点是冗余结点,则删除这个结点,继续处理其他叶结点直到所有叶结点判断完毕;若PTQ中结点个数为n,则CIM算法的最大复杂度为$O(n^4)$.

文献[31]提出了一种最大复杂度为$O(n^2)$的改进算法,通过结点间的二元关系Simulation判断冗余性.改进CIM算法与CIM最大的不同点在于:前者只关心后代结点之间是否相同;而后者还要关心祖先结点之间是否相同.改进CIM算法通过正向遍历查找冗余结点,如果某结点的一个儿子结点与另外儿子结点之间有Simulation关系,则这个儿子结点为冗余结点.这样,在一次遍历可以对多个结点的冗余性进行判断,从而提高了PTQ最小化的效率.

(3) PTQ{/,//,[],*}

与限制条件下的 PTQ 不同的是,普遍意义下的 PTQ 在语法优化时遇到的一个关键问题是:最小化 PTQ 并非原始 PTQ 的子树,而是由以原始 PTQ 的根为根,连接多个 PTQ 子树构成的 PTQ.其中每个子树均为原始子树的最小化部分.这个问题也是导致其复杂度上升的关键.

文献[32]给出普遍意义下的 PTQ 最小化算法.其算法思想是:递归的在原始 PTQ 的子树中查找最小子树并连接它们,在这个过程中冗余的分支被删除了.文献[32]证明其算法的复杂度为 NP 完全.并指出:在对 PTQ 的分支个数加以一定限制的情况下,改进的算法复杂度可以达到多项式级.但我们有理由相信,这样的改进意义不大,因为这要求用户在写查询语句时必须注意查询的分支情况,否则将导致某些查询无法优化.

目前,语法优化都以判断结点之间的包含映射关系为基础,分析路径等价性.在查询树中不断的修剪冗余的分支或结点,达到减少查询树规模的目的.普遍意义的 PTQ 语法优化是一个 NP 完全问题,研究者通过对 PTQ 复杂度加以一定限制,提出了多种高效的算法.语法优化不涉及 XML 模式信息,可以利用模式信息进一步简化 PTQ,这种优化称为语义优化.

4.2 语义优化

最早提出语义优化的是关系数据库系统,利用表格属性值之间的约束关系把查询表达式转换为等价但更高效的形式.chase方法[33]是其中的代表.其思想为:把完整性约束作为冗余条件插入到查询表达式中,与已存在的冗余操作合并,使得组合后的条件符合某些事先定义好的等价转换规则,利用这些等价转换规则重写查询表达式达到优化的目的.这是一个巧妙的先膨胀再收缩的方法.

但是,应用 chase 方法于 XML 查询的语义优化时面临下述严重的挑战:

数据结构的变化:与平面表结构不同,XML 数据具有嵌套性.原有的主键、外键等完整性约束不能表达结构上的嵌套关系,缺乏匹配的转换规则.

数据类型的变化:关系模型中,数据有严格的类型;而在 XML 半结构化模型中,数据没有严格的类型约束,同名的结点可以出现在不同的位置,可以有不同的子结点.传统的转换规则的应用方法不适应这种情况,引发的问题就是产生递归的转换,导致路径的无限增长.

查询语句的复杂性:SQL 语句清晰明确,关系代数操作均为输入参数明确的一元或二元运算.而 XML 查询语句中包含//, *等不确定因素,并以包含多个分支长路径为特点,原有的转换规则不适应于 XML 语义优化.

基于上述挑战,一些研究者提出改进的 Chase 方法,而另一些研究者从对图分析处理的角度出发,研究 XML 查询的语义优化问题.

(1) 改进的 Chase 方法

Wood等[29]最早将chase方法引入简单XPath语义优化.文献[29]在DTD上定义了 3 种结构约束关系,分别为儿子约束、父亲约束和兄弟约束.若某个查询树中结点n为上述约束关系中的主体,且其约束的结点不在查询树中,则在查询树中相应位置加入客体结点.当所有约束关系应用完毕,再用语法优化的方法对查询树进行修剪,

得到最小化查询树.为了讨论的简单性,文献[29]中方法只适用于不包含“*”和“//”的简单路径,其复杂度为多项式级.

文献[30]则认为XML数据中的结构完整性约束可用儿子约束、后代约束和类型约束概括.为了得到正确的优化结果,他们对chase方法做了 3 个方面的改进:首先,假设约束集合是闭包的;其次,为了保证优化能够完成,约束条件只应用于PTQ中原有结点,如果某结点是由于应用某约束条件加入PTQ的,则不对其应用任何约束条件;最后,由于应用约束条件加入的结点是冗余的,因此,需要在算法的结束时删除这些临时结点.ACIM算法分为 3 个步骤:首先,应用约束集合中的约束条件放大PTQ;然后,应用CIM算法语法删除冗余结点,在删除时保证不检查被加入临时结点的冗余性;最后,删除所有在第一步中加入的临时结点.若PTQ中结点个数为n,则ACIM的最坏计算复杂度为$O(n^6)$.一些冗余结点是容易识别的,如果提前删除这些容易识别的冗余结点然后再应用ACIM算法,可以有效地提高优化的效率.算法CDM在PTQ中遍历地查找并删除这样的冗余结点,其计算复杂度为$O(n^2)$.实验证明:在使用ACIM之前使用CDM,比直接应用ACIM可有效地节省时间.

文献[31]在文献[30]的基础上扩充了子类约束,并利用语法优化中的TPQSimulation和TPQMinimization改进了ACIM算法,使计算复杂度达到$O(n^4)$.

(2) 基于 DTD 的路径等价类方法

文献[34]提出了一种基于模式的 XPath 路径表达式的语义优化方法.其思想为:把 XML 文档模式(DTD)划分为若干个路径等价类,每个类中的路径等价;将 XPath 转换为由简单路径构成的合取范式形式,利用路径等价类中的最短路径代替表达式中的路径.通过这种方法,可以实现 3 个方面的优化:首先,删除冗余的谓词条件;其次简化路径;最后,判断表达式的条件是否满足.如果某个分支的条件不满足模式中的约束关系,则整个表达式的查询结果为空.整个优化过程分为 4 部分:分解、扩展、优化和重构.在分解过程中,XPath 表达式被转换为合取树(XCT);重构 XPath 表达式则将优化的 XCT 转换为 XPath 路径表达式.

改进的 chase 方法的优点是:语法优化与语义优化相结合,优化过程无须对 PTQ 的转换.问题是难以保证彻底的优化:首先,PTQ 中存在非叶冗余结点,而语法优化只能在删除叶结点后,让非叶结点变为叶结点的情况下才能判断其冗余性.加入约束条件后的 PTQ 语法优化,不能在不删除叶结点的情况下,判断非叶结点的冗余性;其次,膨胀后的 PTQ 难以压缩,采用对转换规则加以一定限制的方法限制膨胀后 PTQ 规模的方法,会导致优化的不彻底.目前的改进 chase 方法都是针对一定复杂程度的 PTQ 的优化策略,普遍意义上的 PTQ 的语义优化研究还需深入;最后,从 DTD 中获得的约束条件并不充分,这也是导致优化不彻底的一个因素.如何抽取更多的语义约束条件,是未来研究的一个重要问题.

基于 DTD 的优化方法的优点是:不但能够删除冗余分支,还能缩短路径长度和直接判断路径是否满足.问题主要是:首先,需要对 PTQ 进行转换,占用大量优化时间;其次,需要不确定路径的确定化,这实际上也是一种路径膨胀,难以保证优化的结果小于优化前.

两种方法都需要首先扩大路径规模,造成优化的不彻底和效率的丧失,这是目前语义优化面临的一个重要问题.

5 物理优化

逻辑优化的结果是一个或多个查询树.如何确定其中不同查询片断的执行次序,是XML物理优化的核心问题.确定执行次序的主要因素是中间结果集的大小.复杂路径表达式选择性分析就是用来估计结果集规模的.其主导思想是:统计 XML 数据的分布情况,基于一定假设估计路径目标结点中间结果集的大小.这种方法一般忽略执行算法的不同和数据的物理存储.本节从统计信息抽取、存储、压缩、维护和统计信息计算等几个方面论述目前这一技术的发展情况和面临的问题.

5.1 代价估计方法研究

代价估计是对查询物理运算时间的估计.目前,代价计算方法主要有 3 种:基于参数的方法、基于取样的方法和基于统计信息计算的方法.基于参数的方法[35]无需统计数据信息,根据数据分布情况假定其满足包含某些

参数的分布函数,通过计算函数的值估计查询计划的执行代价.这种方法在处理有规律分布的数据(如学生成绩)时,可节省大量的统计信息空间和I/O代价,但对分布无明显规律的数据会有很大的误差;基于取样的方法[36,37]也无需统计数据信息,做法是从数据集中提取具有代表性的样本,比较不同的查询计划的执行情况,获得代价最小的方案.这种方法的精确性决定于取样的代表性.最简单的方法是随机取样,一些优化的方法根据数据的密度取样.取样方法的缺点是代价估计本身占用时间可能很大;最常用也是研究最多的方法是基于统计信息的方法[38],需要统计估计所用的各种信息,利用统计信息计算不同方案的执行代价.这种方法的精确性取决于统计信息的正确性.但是,统计信息过大又会导致统计时间过长.利用有限的时间和空间得到相对小的执行方案,是代价统计的基本原则.不同的系统支持的物理运算算法不同,代价模型也不同.

在关系模型中,进行代价估计时有两个通用的前提:独立性假设和均匀分布假设.前者是指各谓词之间没有相互依赖关系;后者是指如果关系在某个属性上没有直方图,则认为该属性的各值在数据库中均匀出现.

XML 数据的不规则性是对传统统计信息方法的重要挑战.其数据分布情况使得一些传统的分布假设难以成立.在 xml 中,相同名字的结点可能在同一个文档的不同部分出现但却具有截然不同的语义.如同为 name 结点,在 person 下和在 city 下出现意义就完全不同,这可以称之为元素之间的结构依赖性;同时,在 xml 文档中,嵌套在不同祖先下的同类结点的个数差别也很大,如 book 结点下的 author 个数是不确定的,这可以称之为元素之间的结构相关性.为了达到所需的代价估计精度,需要更多的统计信息.而结构的复杂性又为获得相对精确的统计信息带来存储和计算上的困难.XML 的有序性制约了转换规则的灵活性.所有这些问题,都使得在 xml 中采用传统的代价估计方法不切实际,会带来很大的误差.针对 xml 数据的特点,我们应该寻求一种新的代价估计方法.

5.1.1 代价模型

查询计划的执行代价主要来自 3 方面的因素:CPU 计算代价、I/O 代价和数据传输代价.CPU, 磁盘和网络的速度差距悬殊.当不考虑数据的分布性因素时,影响代价的决定性因素是 I/O 代价.I/O 代价受众多因素制约,主要来自 3 个方面:一是数据库系统参数,如页面大小、内存使用情况等;二是数据集本身因素,如数据存储空间大小、索引情况、每个元素占用空间情况和元素的聚集情况等;三是查询请求因素,如查询条件的选择性等.

目前,对XML代价模型的研究并不充分,代价模型相对简单,这也是造成代价估计误差的一个原因.Lore的代价模型没有考虑聚集情况,不能判定不同的数据是否在同一页面上,因此,其假设每次I/O操作只能获得一个对象,把对I/O时间的估计转换为对中间结果大小的估计.由于Lore中数据本身无聚集,这种方法可以获得较好的效果,但对其他XML数据库系统参考意义不大.文献[39]提出了一种新的代价模型,其基本思想是利用查询反馈信息来调整参数.如图 3 所示,在用户提交查询之前,先人工找出影响查询执行时间的特征,再利用线性回归模型计算出各个特征对查询代价的影响的系数,即得到一个形如$cost(q)=f(v_1,v_2,\ldots,v_d)$的函数模型,当用户提交查询时,利用函数和事先统计的特征值进行计算.但是,文中提出的方法只是针对CPU代价估计的,没有扩展到I/O代价的估计;而且只考虑了一个XNav操作符,至于如何扩展到其他的情况,文中并没有提及.人工抽取特征具有主观性,如何让系统自动地抽取特征是下一步研究的重点.

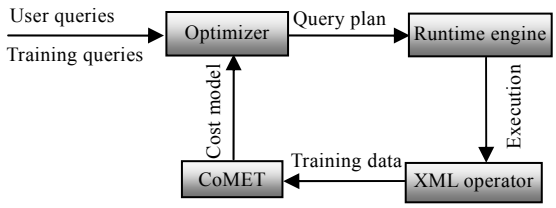


Fig.3 CoMET optimization system

图 3 CoMET 优化系统

5.1.2 代价空间搜索技术

代价空间搜索算法首先通过某种计算方法量化代价空间、构造搜索函数,根据函数值的变化判断是否继续搜索.在众多的空间搜索技术中,最简单的是随机搜索方法,随机的或者按照某个顺序在搜索空间中计算代价.

这种方法效率低下,在实际的系统中很少被采用.爬山算法是应用广泛的一种搜索算法,在以某步长在搜索函数中按照逐步接近的方向,定位局部最优执行计划,搜索的效率与初始值和步长相关.当搜索函数非单调时,这种方法找到的是局部极值,而并非全局最值.遗传算法是解决局部最优的一种新颖的空间搜索方法.用杂交的方法搜索不同的最优执行计划,适用于有多个极值的搜索函数.这种方法在关系数据库查询优化中有应用意义,在XML 查询优化的代价空间搜索技术中应用遗传算法的难点在于适应度函数的构造.

单个查询物理计划形成的代价空间可能非常庞大,尤其是路径很长的情况,其代价空间成幂次级增长.为了减少代价估计时间,需要利用启发性规则约束代价空间.Lore 的做法是:分别将每一个逻辑操作转换为最优物理子计划,并在转换时应用启发性规则.例如:TargetSet 操作只在路径表达式的起始结点是标记名并且只在路径结束结点上有变量约束时使用;当查询中有多个路径表达式时,不改变其间的顺序.值得注意的一条规则是选择操作总在最后做.这条规则和关系查询优化的启发性规则正好相反.这是由于在 Lore 中,选择运算总是基于变量绑定运算.

在 XML 代价估计研究中,路径表达式选择性代价估计是核心问题,也是在 XML 查询优化研究中份量最重的一个领域,值得我们特别关注.在 5.2 节中我们将做专门的论述.

5.2 路径表达式选择性代价估计

XML路径表达式可视为一棵树,其中的一个主枝为从起点到目标点的主路径,其余分支为约束主枝的谓词条件(如图 4 所示),表示为$P=t_1[p_1]/t_2[p_2]/\ldots./t_n[p_n]$.其中:$t_i$为结点名;$p_i$为谓词,默认存在量词布尔表达式.路径表达式的选择性估计是对满足分支条件的主枝的数据个数的估计.对XML路径表达式的估计需要数据结构的统计信息与分布在结构内部的值的统计信息的结合,计算路径的选择性.

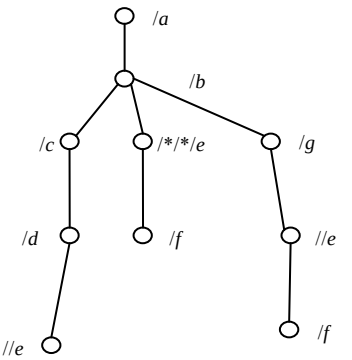


Fig.4 /a/b[c/d//e][g//e/f]//*/*/e/f

图 4 /a/b[c/d//e][g//e/f]//*/*/e/f

5.2.1 数值统计

Chen等人[40]把数值结点作为普通结点看待,这样,估计a=3 与a.3 是等价的,简化统计结构,适用于数值量较少的情况.如果数值量庞大,统计每一个数值的个数会占据大量的空间,导致统计信息过多,影响查询代价估计的效率.这种方法对等值的定点查询可以使用,却很难估计范围查询的代价.如估计a>3 的代价,需要遍历统计信息中所有同类数值结点,判断其值是否满足条件.

Freire等人[41]用直方图等方法分别统计不同类型的数值信息,如最大值、最小值、平均值、个数等信息.即适用于范围查询,也适用于点查询.其缺点是:数据类型过多会导致统计信息的膨胀;并且,XML数据的特点是自描述性,其值和结构有密切的语义关系,分开统计必然导致分别估计,导致估计误差,在文献[41]中有关于这方面的详细论述.

5.2.2 数据结构抽取

XML数据为有序有向图.对图的结构抽取有两种极端的方法[42]:

标记分裂图(label-split graph)方法粗略地统计模式信息,其思想是合并所有的标记名相同的结点为一个结点,记录合并结点的个数作为标记的统计信息.这种方法占用空间相对较小,但不能精确的反应数据分布的真实情况,尤其是同名结点出现在不同位置上时,可能包含错误的路径或者圈信息;

B/F 类似图(B/F-bisimiar graph)中,只有所有入边和出边集合相同时合并同名的结点,保证准确地统计路径表达式所有的分支情况.这种方法的缺点是占用空间大.

查询优化的统计信息控制在一定的范围内,现有系统的抽取方法都是介于两个极端的情况之间.

Lore的DataGuide[43]抽取模式的方法是保证每条路径只出现一次,其最小模式是标记分裂图的一个例子.文献[43]指出:这种方法统计路径信息能够精确判断路径是否存在,但并不能更精确地统计不同结点的值在不同路径中的分布情况.如图 5 所示:图 5(c)为图 5(a)的最小DataGuide.如果对结点 19 的统计信息为t,根据图 5(c)

无法判断是路径A.C或者B.C的结点个数;图 5(b)为图 5(a)的强壮DataGuide.如果对结点 12 和 13 的统计信息为t_{12}和t_{13},根据图 5(b)判断路径A.C的个数为t_{12}, B.C的个数为t_{13}.

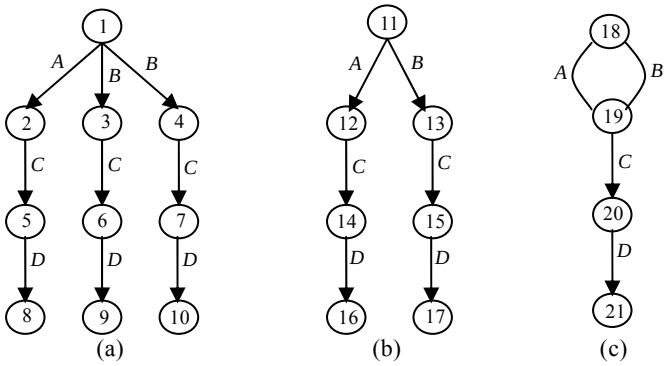


Fig.5 XML data and its corresponding DataGuides

图 5 XML 数据和 DataGuides

StatiX[41]根据XML Schema统计结构信息,是一种折衷的方法.出边不同但同类型的结点合并为一个,系统对不同类型的结点标记以不同区域的值,按照区域分别统计不同路径的结点个数,保留了分支结点的路径分布信息.儿子结点的分布情况保留在父亲结点的统计信息中.每个结点的统计信息不再是单个的值,而是一个复杂的结构.导致的问题一是代价信息的增长,二是代价估计算法的复杂性.

Xsketch[42]也是一种折衷的方法,但只统计结构中每个结点对应数据中结点的个数信息.其特别之处在于对结点入边和出边的信息的统计.如果对任意结点$v\in V$,存在$u\in U$,在数据集中都有边(u,v),则V对U向后固定.如果对任意结点$u\in U$,存在$v\in V$,在数据集中都有边(u,v),则U对V向前固定.这种统计信息用于在合并不同分枝与主枝之间选择性代价估计的计算.

上述方法统计的只是路径中父子之间的关系,而没有统计同父的兄弟之间的相关性.为了便于计算相对路径的代价和统计路径之间的相关性,Zhiyuan Chen等人[40]吸收了信息检索中在文档中检索子串的采用后缀树的方法[44]统计路径信息,称为相关子路径树(correlated subpath tree).从XML数据中抽取所有到叶子的可能路径,形成路径集合,其中间结点名为不可分割子串;叶结点为数值或者字符串值,为可分割子串.每个结点存储子路径在数据中出现的次数和路径特征矢量.我们将在后面的选择性计算部分介绍后缀树方法的代价计算方法;在信息统计部分介绍后缀树信息的存储和修剪维护方法.

XSketches[45,46]在文献[42]的基础上增加了对边的信息和值的信息的统计,从而在一定范围内(TSN)能够处理结构和值或值和值的相关性的问题.但增加的信息同时,也使得构造XSketches结构代价很大.在XSEED[47]中提到:为 100M的XMark[47]文档构造一个XSketches结构需要超过一天的时间,这在实际中变得不可接受.

针对XSketches的缺点,XSEED[47]提出了一种新的处理思路,即抽取最粗略的信息,称为XSEED内核,一般只占文档大小的 2%左右;然后,再利用查询反馈的信息把误差最大的那部分路径选择率的精确值存储起来,而这部分存储的信息的多少是根据内存大小来确定的.但是,这种方法只能处理XPath.

如果把XML数据看作树,对树中的结点按照(start,end)[49]编码,以start为横坐标,end做纵坐标,则树中所有结点可看作是平面空间上的点,路径上的祖先和后代关系满足:$x_{ancestor}<x_{descendant}<y_{descendant}\leq y_{ancestor}$,把整个空间区域分成多个方格,统计结点在每个格中的分布个数;Wu等[50]的思想是:把不同结点映射为一维坐标上的线段,祖先线段和后代线段的起点和长度满足某种偏序关系,把整个区间分成多个小区间,分别计算落在不同区间内的线段之间的关系.这种方法减少了稀疏分布时的误差,但不能完全避免.这两种统计方法适用于对结构连接运算的代价估计.

5.2.3 选择性计算

当 XML 数据结构复杂、每个文档的数据量很大时,精确地统计信息是不可能的.在计算查询代价时,需要用

一些假设来弥补统计信息的不足.目前的路径选择性计算方法分为 3 种:

(1) 基于马尔科夫链的方法

Lore[43]的方法基于马尔科夫链思想,用于计算没有分支条件的简单路径.系统中只保留短路径的选择性统计信息,基于父子结点分布的独立性假设计算长路径选择性.如有长路径$t_1/t_2/t_3/\ldots/t_n$,其选择性计算公式可为

$$\hat{\sigma}(t_1t_2...t_n)=\left(\prod_{i=1}^{n-1}\frac{f(t_i,t_{i+1})}{f(t_i)}\right)\times f(t_{n-1},t_n).$$

此时,只需保留长度为 1 和 2 的路径信息.也可为

$$\hat{\sigma}(t_1t_2...t_n)=\frac{f(t_1t_2...t_{n-1})\times f(t_2t_3...t_n)}{f(t_2t_3...t_{n-1})},$$

则需保留长度为 n–2 和 n–1 的路径信息.路径信息越长,组合个数越多,占用空间越大,计算值越精确.实验表明:当路径信息较短时,加长路径信息导致明显的计算精确性,且空间代价增长相对缓慢;当路径信息较长时,加长路径信息导致空间代价的爆炸性增长,而精确性提高缓慢.如果在路径的某个结点上有对简单值的选择,则根据兄弟分布独立性原则,计算不同选择性再做交集运算.

Aboulnaga等[51]对Lore的方法进行了改进,提出了路径(path)树和Markov表来估计简单路径的选择性.用路径树可以表示原文档的结构,树中的每一个结点代表了从文档的根结点开始的路径,并记录了相应结点出现的次数.但当树变得很大以至于不能放在内存的时候,就需要对树进行剪枝,根据一定的算法,删去那些出现不频繁的结点,然后在这个剪枝过的树上进行选择率的估算.Markov表存储长度为m (m>=2)的不同路径,如果查询的长度和m相等,直接就可以从表中读出相应的值,误差为 0;当长度大于m时,用公式

$$f(t_1/t_2/.../t_n)=f(t_1/t_2/.../t_m)\times\prod_{i=1}^{n-m}\frac{f(t_{1+i}/t_{2+i}/.../t_{m+i})}{f(t_{1+i}/t_{2+i}/.../t_{m+i-1})}$$

进行计算.同样的,当表不能放入内存时,删除那些不频繁路径.

Xsketch[42]增加了对边的固定性统计信息,并对Lore的方法做了改进,以适用于更一般的路径表达式代价估计.如果主路径是向后固定的,则统计信息为精确信息,无须进一步计算;如果主路径中有些点不是向后固定的,则在这些点上把主路径分为多个子路径,根据路径独立性假设,用类似Lore的公式,以子路径统计信息为参数,计算长路径的选择性.如果分支路径是向前固定的,则其选择性为 1,无需参与计算;如果分支路径中有些点不是向前固定的,则在这些点上把分支路径分为多个子路径,计算不同选择性,再做交集运算.为了提高计算的精确性,其代价路径分解方法为最大交叉方法.

XSketches[45,46]对文献[42]的工作进行了扩展,可以计算twig匹配的个数.XSketch在原来的模型上增加了边的分布信息,从而能够从细节上把握数据的分布.这种方法的特点是:先抽取XML文档的结构建立XSketches,然后利用边的稳定性和边的分布概率来估计twig的匹配个数,如果边的确是分布均匀的话,那么这种方法的准确率就比较高.

XSEED[47]方法在XSEED核结构上增加了对递归结点的处理,递归结点是指在DTD中有类似$A(A+,B^*)$的定义,则A结点是一个递归结点.XSEED核结构在边上记录了在递归的不同层相应的父亲、孩子的结点个数.因此,这种方法可以很好地处理带有递归表达式的XPath查询.

马尔科夫链思想中的一个关键性假设是父子结点分布独立性和兄弟结点分布独立性假设.而实际上,很多XML 数据的父子结点、兄弟结点之间的相关性非常强,应用上述计算方法会导致误差.集合哈希方法统计相同分支之间的相关性信息,更准确地计算分支路径的选择性.

(2) 集合哈希方法

集合哈希方法[40]的核心思想来自蒙特卡罗技术的Min-wise independent permutations[44].这种方法用于估计两个集合之间的相似性,集合的特征通过设置哈希函数的集合获得.其优势在于集合的哈希函数值占用存储空间很小.对集合A,其特征矢量为

$$sigA=(\min\{\pi_1(A)\},\min\{\pi_2(A)\},\ldots,\min\{\pi_l(A)\}).$$

其中:$U=\{1,\ldots,n\}$; π是 U 的排列;l 是哈希函数的个数.

$$sig_{A_1\cup\ldots\cup A_k}[i]=\min\{sig_{A_1}[i],\ldots,sig_{A_k}[i]\}.$$

假设A_j为势最大的集合,则 $\hat{\gamma}=\dfrac{|A_j|}{|A_1\cup.....\cup A_k|}$,而

$$|A_1\cup\ldots\cup A_k|=\frac{|\{i\,|\,\min\{\pi_i(A_1)\}=\ldots=\min\{\pi_i(A_k)\}\}|\bullet|A_j|}{\hat{\gamma}}.$$

关于集合哈希函数的构造方法在文献[44]中有详细的论述.

利用集合哈希方法计算路径表达式的代价的方法为:用后缀树统计所有可能出现在查询中的子路径的特征矢量;分解查询为多个相关子路径;应用公式计算选择性.文献[40]中提供了 3 种路径分解的方法,并比较了不同方法之间的优劣.这种代价计算的精度对路径的长度敏感:随路径长度的增长,统计精度下降.并且,特征矢量本身是一种信息压缩的方法,用来计算相关性时的精确性是值得商榷的.

上述两种算法,都是根据确定路径上的父子关系统计信息计算路径表达式中后代结点的选择性,没有直接计算后代,也没有根据某些后代估计满足条件的祖先结点的选择性.在执行计划中,存在大量的向前遍历的算法.如何估计这些算法的代价,是一个需要解决的问题.

上面所提到的各种方法的处理能力各有不同,仅从方法能支持的各种情况(Xpath,Xquery,//,*,value)来看,可以总结为表 2.

Table 2 Comparisons among various methods' processing

表 2 各种方法处理能力比较

	Query	//	*	Branch	Value	Correlation
Statix	Simple twig + value	N	Y	Y	Y	N
Lore	Simple path	N	Y	N	N	N
CST	Simple twig + prefix string matching	N	Y	Y	Prefix string matching	Y (set hashing)
Path tree	Simple path	N	Y	N	N	N
Markov table	Simple path	N	Y	N	N	N
Xsketch	Simple twig	N'	N'	Y	N	N
Xsketches	Simple twig + value	N'	N'	Y	Y	Y (md methods)
XSEED	Path	Y	Y	N	N	N

(3) 位置直方图方法

Wu等人[50]采用位置直方图的方法统计组先后代的分布信息,其代价计算分为基于祖先的代价估计和基于后代的代价估计.基于祖先的代价估计根据每一个祖先的格,累计其满足条件的后代的格中的结点个数;基于后代的代价估计则与其正好相反.根据不同区域的祖先后代结点的偏序关系,分别计算.如图 6 所示:A为某祖先格,则B区域中所有的格的所有结点均为A中所有结点的后代;C, E区域中所有格的部分结点为A中部分结点的后代;而D,F区域中只有左上半角区域中存在结点且部分结点为A中部分结点的后代.如果考虑自身的嵌套,A中部分结点是A中部分结点的后代.根据上述,满足条件P_1的A的满足条件P_2的后代个数估计公式为

$$Est_{P_{1,2}}[A]=Hist_{P_1}[A]\times\{\frac{1}{4}\times Hist_{P_2}[A]+Hist_{P_2}[B]+Hist_{P_2}[C]+Hist_{P_2}[E]+\frac{1}{2}\times(Hist_{P_2}[D]+Hist_{P_2}[F])\}.$$

对每个格而言,其后代格限定在较小区域内,避免多余的统计和计算.但在计算区域的边缘,并非所有结点满足上述关系.根据平均分布的假设给计算结果加以某个系数是产生误差的主要因素,误差在空间结点分布稀疏时变得很大.一个改进的方法是分别统计不同类型的结点的分布情况,但统计信息占用空间很大.这种方法解决的问题是已知集合间的祖先后代关系的估价,未涉及路径中单个谓词的选择性计算问题.

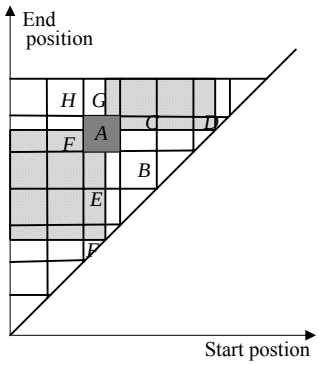


Fig.6 Two-Dimention histogram

图 6 二维直方图示例

Jiang等人[52]的思想是:把不同的结点映射为一维坐标上的线段,祖先线段和后代线段的起点和长度满足某种偏序关系.把整个编码空间[cmin,cmax]分成多个小区间,然后在每个小区间中估计覆盖的线段/起始点的对数.如图 7 所示:统计信息*n*表示每个桶中的线段/起始点的对数;*wss*表示桶的起始坐标;*wse*表示桶的终点坐标;*l*表示桶中线段的平均长度;匹配的祖先-后代个数在图中用*X*表示.这种方法减少了稀疏分布时的误差,但不能完全避免.

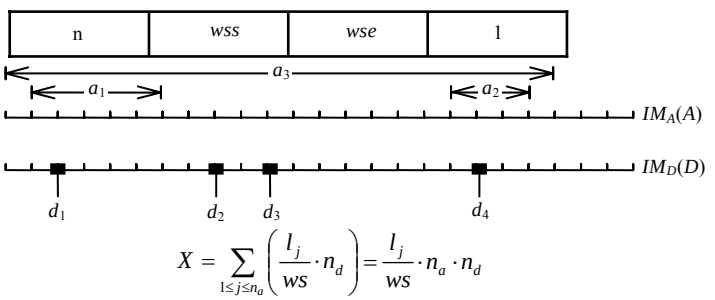


Fig.7 PL histogram and statistics

图7 PL直方图及统计信息

上述方法在计算路径选择性时,只考虑有谓词约束的结点,越过其他结点,直接估计后代或祖先的代价,简化了计算的复杂度,在对模糊路径的代价估计中有优势.与马尔科夫方法相同的是:当路径中出现多个谓词时,假设不同的谓词条件是独立的,没有计算不同的谓词之间的相关性.集合哈希方法计算不同谓词之间的相关性,但后缀树占用空间过大,尤其是XML 树据中包含大量数值时.而查询优化的一个原则就是在有限的时间和空间内精确地估计代价.这时,必须对统计信息进行压缩,以保证优化的效率.

5.3 统计信息研究

XML 数据结构复杂,分布不均匀.数据量庞大时,在有限的空间内统计足够多的信息是统计信息研究的难点.当数据更新频繁时,高效的信息维护技术显得更为重要.

5.3.1 统计信息存储

在前文介绍数值统计和数据结构抽取时,已经涉及到统计信息的存储问题,但讨论集中在逻辑层,并未涉及存储的形式.XML 数据统计信息的存储结构主要有以下几种:

树或图[40−42]:用于存储模式信息,如在文献[42]中模式树中每个结点的个数,或者文献[40]中结点之间的固定关系等.由于其数据结构与XML原始数据相同,可用对数据本身的存取方法存取统计信息.如果XML数据模

式结点个数为n,则树方法存储的代价为$O(n)$;后缀树的存储代价为$O(n!)$.

马而科夫表[51]:用于存储路径信息,不同的路径对应其在数据中出现的次数.为了查找方便,一般在表上再加以一定的索引.如果只统计小于k长度的路径,其存储代价为$O(nk)$.

直方图[38]:关系数据库中普遍应用的一种统计信息方法.将某属性的值域划分为多个连续或不连续的部分,分别统计不同部分区域内数据的分布情况.对于简单数据,采用一维直方图方法;如果统计相关的不同属性的分布情况,需要二维或者更多维数的直方图.在对XML数据统计时,主要采用直方图和树相结合的方法.如果某个结点的值为树值型,则用直方图统计这个结点数值的分布情况.文献[41]中,对于结点之间的父子关系也采用直方图的方法统计.必须首先唯一地标记所有的数据结点,并且,不同类型结点标记的值域没有交叉.当路径中的某些点有谓词约束时,通过统计信息无法知道那些满足条件的结点的标记,从而无法进一步估计其后代的个数.

位置直方图[50]:这是一种二维直方图,其值是离散的,把结构嵌套转换为平面位置关系.如果按类型每维分为k格,则其存储代价为$O(nk^2)$.文献[52]等采用一维直方图方法保存结点的位置信息,但需要根据结点在连接过程中是祖先或后代保存不同的统计信息,实际上冗余很大.

5.3.2 统计信息压缩

对直方图压缩方面的研究在关系数据库查询优化中已经有一定的研究基础,普遍采用的是小波压缩[53]和DCT(discrete cosine transform).两种方法均能把直方图中大量的“桶”压缩为少量的系数,同时丢失很少的信息.小波压缩方法在代价估计时,要重新构造与查询相关的部分直方图.Yan[35]等采用密度函数方法压缩直方图,用数据密度函数直接模拟实际的数据的分布情况.这种方法适用于离散的或是连续的数值型数据.

树的修剪和马尔科夫表的压缩思想是从统计信息中删除对代价估计结果影响相对较少结点.结点在数据中出现次数越少,越缺乏统计价值.文献[51]中提供 4 种不同的方法,并通过实验分析了几种方法在不同的数据分布情况和查询情况时的占用空间和代价估计精度关系,但并未得到相对稳定的方法.

5.3.3 统计信息维护

据我们所知,目前对XML统计信息维护的研究很少见.文献[54]提出一种马尔科夫表的在线维护方法,其思想是:在查询开始时没有统计信息,利用查询反馈逐步生成和细化统计信息.其细化规则有两个:重尾规则(heavy-tail)和增量规则(delta).重尾规则是在计算选择性时,给接近查询路径末端的路径的选择性计算加以较高的权重.这样做基于如下考虑:接近路径末端的路径选择性对整个路径的选择性影响更大;增量规则是一种错误减少学习方法.文献[54]的优点在于在线维护统计信息.文献[55]提出了基于StatiX[41]框架IMAX增量维护算法,包括结构信息和值信息的增量维护,但是不易扩展到其他系统上.

6 总结及展望

随着 Internet 的发展,XML 数据规模与日剧增,准确、高效地查询 XML 数据成为目前研究的热点问题.近两年来,XML 查询优化研究方兴未艾,主要集中在几个方面:对 XML 代数的研究、对根据 XML 代数分解查询语句的研究、对路径选择性估计的研究、对结构连接代价估计的研究等.XML 查询优化是一门综合性强的研究领域,需要吸收众多其他技术的思想,其中对 XML 查询优化影响深刻的主要有:关系数据库查询优化技术、面向对象数据库查询优化技术、信息检索技术、数据仓库和数据挖掘技术、图像处理和压缩技术、人工智能技术、概率论和统计技术等.

从查询优化的过程来讲,XML 查询优化和其他数据库查询优化技术并无不同之处.从优化的不同环节的技术上,XML 查询优化具有其独特的方面:既要适应更复杂的数据结构和更灵活的变化,又要适应更丰富的查询语义.目前对 XML 查询优化的研究工作还远未成熟,仍有众多尚待解决的问题或需要完善的技术.因此,未来的XML 查询优化研究将以更广泛、更深入的方式展开.

在XML代数研究中,一个值得重视的问题是逻辑操作与物理操作的分工不明确.大量的工作在于制定不同的代数标准,这些标准在风格上相去甚远,很难有共通性.而对真正执行的物理代数的研究还远远不够,而且不能形成完整的系统.而且,从逻辑代数到物理代数的转换也将是未来研究的一个重要的问题.

关系数据库中一些约定俗成的启发式转换规则是在大量的实践基础上形成的.而目前对 XML 数据查询的各种不同的执行方法之间的优劣比较的工作还刚刚开始,并未形成共识性的规则.由于 XML 数据本身的灵活性,找到一些普遍适用的规律是很困难的.在今后的一段时间内,相信会有更多的研究工作在这方面展开.

复杂路径表达式选择性代价估计是XML查询优化研究的核心问题,目前已有大量的成果.这些研究或对数据分布进行大量的假设,或对查询表达式的复杂性加以一定的约束.尤其是在相关路径选择性的研究方面,仍有一些尚待解决的关键性问题.

一直以来,统计信息维护是代价估计的基础.但是,关于 XML 树据统计信息的维护问题的研究处于起步状态.由于XML树据统计信息在数据结构上与传统的统计信息有本质的不同,很难直接利用现有的统计信息维护的技术.又由于目前在 XML 统计信息研究中采用了大量的压缩技术,为统计信息维护增加了难度.

Native XML Database 的研究是目前在 XML 研究领域的一个热点,已经出现一批相对独立的系统,这些系统采用的查询和处理方法也将对 XML 查询优化技术产生越来越重要的影响.

References:

[1] Beech D, Malhotra A, Rys M. A formal data model and algebra for XML. 1999. http://www-db.stanford.edu/infoseminar/Archive/FallY99/malhotra-slides/malhotra.pdf

[2] Fernandez M, Simeon J, Suciu D, Wadler P. A data model and algebra for XML query. 1999. http://www.cs.bell-labs.com/wadler/topics/xml.html#algebra

[3] Kay M. XSL transformations (XSLT), Version 1.0. W3C Recommendation, 1999. http://www.w3.org/TR/xslt

[4] Fankhauser P, Fernandez M, Malhotra A, Rys M, Simeon J, Wadler P. XQuery 1.0 formal semantics. W3C Working Draft, 2002. http://www.w3.org/ TR/query-semantics/

[5] Fernandez M, Robie J. XQuery 1.0 and XPath 2.0 data model. W3C Working Draft, 2002. http://www.w3.org/TR/query-datamodel/

[6] Mary FF, Jérôme S, Byron C, Amélie M, Gargi S. Implementing xquery 1.0: The galax experience. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases. Berlin: Morgan Kaufmann Publishers, 2003. 1077−1080.

[7] McHugh J, Abiteboul S, Goldman R, Quass D, Widom J. Lore: A database management system for semistructured data. In: Franklin MJ, ed. SIGMOD Record, 1997,26(3):54−66.

[8] McHugh J, Widom J. Query optimization for XML. In: Atkinson MP, Orlowska ME, Valduriez P, Zdonik SB, Brodie ML, eds. Proc. of the 25th Int'l Conf. on Very Large Data Bases. Edinburgh: Morgan Kaufmann, 1999. 315−326.

[9] Jagadish VH, Al-Khalifa S, Lakshmanan L, Nierman A, Paparizos S, Patel J, Srivastava D, Wu YQ. Timber: A native XML database. The VLDB Journal, 2002,11(4):274−291.

[10] Jagadish VH, Al-Khalifa S, Lakshmanan L, Srivastava D, Thompson K. Tax: A tree algebra for XML. In: Ghelli G, Grahne G, eds. Database Programming Languages. Frascati: Springer, 2001. 149−164.

[11] Frasincar F, Houben GJ, Pau C. XAL: An algebra for XML query optimization. In: Zhou XF, ed. Proc. of the 13th Australasian Database Conf. (ADC 2002). Melbourne: Monash University, 2002.

[12] Zhang D, Dong YS. A data model and algebra for the web. In: Trevor JM, Bench C, Giovanni S, A. Min Tjoa, eds. Proc. of the 10th Int'l Workshop on Database & Expert Systems Applications. Florence: IEEE Computer Society, 1999. 711−714.

[13] Liefke H. Horizontal query optimization on ordered semistructured data. In: Cluet S, Milo T, eds. Proc. of the ACM SIGMOD Workshop on the Web and Databases. Philadelphia: ACM Press, 1999. 61−66.

[14] Mukhopadhyay P, Papakonstantinou Y. Mixing querying and navigation in MIX. In: Agrawal R, Dittrich K, Ngu AHH, eds. Proc. of the 18th Int'l Conf. on Data Engineering. San Jose: IEEE Computer Society, 2002. 245-254

[15] Paparizos S, Al-Khalifa S, Jagadish HV, Nierman A, Wu YQ. A physical algebra for XML. Technical Report, University of Michigan, 2002.

[16] Christophides V, Cluet S, Moerkotte G. Evaluating queries with generalized path expressions. In: Jagadish HV, Mumick IS, eds. Proc. of the '96 ACM SIGMOD Int'l Conf. on Management of Data. Montreal: ACM Press. 1996. 413−422.

[17] Buneman P, Fan W, Simen J, Weinstein S. Constraints for semistructured data and XML. In: Buneman P, ed. ACM SIGMOD Record, 2001,30(1):47−45.

[18] World Wide Web Consortium. XML path language (XPath) Version 1.0. W3C Recommendation, 1999. http://www.w3.org/TR/xpath.html

[19] Chamberlin D, Clark J, Florescu D, Robie J, Sim´eon J, Stefanescu M. XQuery 1.0: An XML query language. Technical Report, World Wide Web Consortium. W3C Working Draft, 2001.

[20] Deutsch A, Fernandez M, Florescu D, Levy A, Suciu D. A query language for XML. http://www8.org/w8-papers/1c-xml/query/query.html.2003

[21] Robie J, Lapp J, Schach D. XML query language (XQL). http://www.w3.org/TandS/QL/QL98/pp/xql.html. 1999

[22] Chamberlin D, Robie J, Florescu D. Quilt: An XML query language for heterogeneous data sources. In: Suciu D, Vossen G, eds. Proc. of the 3rd International Workshop on the Web and Databases. Dallas: Springer, 2001. 1−25.

[23] Li Q, Moon B. Indexing and querying XML data for regular path expressions. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases. Roma: Morgan Kaufmann, 2001. 361−370.

[24] Chan C, Felber P, Garofalakis M, Rastogi R. Efficieng filtering of XML documents with Xpath expressions. In: Agrawal R, Dittrich K, Ngu AHH, eds. Proc. of the 18th Int'l Conf. on Data Engineering. San Jose: IEEE Computer Society, 2002.. 235-244.

[25] Wood PT. On the equivalence of XML patterns. In: John W L, Verónica D, Ulrich F, Manfred K, Kung-K L, Catuscia P, Luís M P, Yehoshua S, Peter J S, eds. Proc. of the 1st Int'l Conf. on Computer on Computation Logic. Berlin: Springer, 2000. 1152−1166.

[26] Florescu D, Levy AY Suciu D. Query containment for conjunctive queries with regular expressions. In: Popa L, ed. Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. Seattle: ACM Press, 1998. 139−148.

[27] Calvanese D, Giacomo GD, Lenzerini M. On the decidability of query containment under constraints. In: Popa L, ed. Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. Seattle: ACM Press. Seattle, 1998. 149−158.

[28] Wadler P. A formal semantics of patterns in XSLT. Markup Languages archive, 1999, 2(2): 183-202.

[29] Wood PT. Minimizing simple XPath expressions. In: Mecca G, Siméon J, eds. Proc. of the 4th Int'l Workshop on the Web and Databases. Santa Barbara: ACM Press, 2001. 13−18.

[30] Amer-Yahis S, Cho S, Lakshmanan LV, Srivastava D. Minimization of tree pattern queries. In: Aref WG, ed. Proc. of 2001 ACM SIGMOD Conf. on Management of Data. Santa Barbara: ACM Press, 2001. 497−508.

[31] Ramanan P. Efficient algorithms for minimizing tree pattern queries. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data. Madison: ACM Press, 2002. 299−309.

[32] Flesca S, Furfaro F, Masciari E. On the minimization for Xpath queries. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases. Berlin: Morgan Kaufmann, 2003. 153-164

[33] Popa L, Deutsch A, Sahuguet A, Tannen V. A chase too far? In: Chen WD, Naughton JF, Bernstein PA, eds. Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. Dallas: ACM Press, 2000. 273−284.

[34] Kwong A, Gertz M. Schema-Based optimization of XPath expressions. Technical Report, University of California, 2001.

[35] Lynch CA. Selectivity estimation and query optimization in large databases with highly skewed distributions of column values. In: Bancilhon F, DeWitt DJ, eds. Proc. of 14th Int'l Conf. on Very Large Data Bases. Los Angeles: Morgan Kaufmann, 1988. 240−251.

[36] Haas PJ, Swami AN. Sequential sampling procedures for query size estimation. SIGMOD Record, 1992,21(2):341−350.

[37] Ling Y, Sun W. A supplement to sampling based methods for query size estimation in a database system. ACM SIGMOD Record, 1992, 21(4). 12−15.

[38] Muralikrishna M, DeWitt DJ. Equi-Depth histograms for estimating selectivity factors for multi-dimensional queries. SIGMOD Record, 1988,17(3):28−36.

[39] Zhang N, Hass PJ, Josifovski V, Lohman GM, Zhang C. Statistical learning techniques for costing XML queries. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson PK, Ooi BC, eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases. Trondheim: ACM Press, 2005. 289−300

[40] Chen ZY, Jagadish HV, Korn F, Koudas N, Muthukrishnan S, Ng RT, Srivastava D. Counting twig matches in a tree. In: Young DC, ed. Proc. of the 17th Int'l Conf. on Data Engineering. Heidelberg: IEEE Computer Society, 604ˆ2001. 595

[41] Freire J, Haritsa JR, Ramanath M, Roy P, Siméon J. StatiX: Making XML count. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data. Madison: ACM Press, 2002. 181−191.

[42] Polyzotis N, Garofalakis MN. Statistical synopses for graph-structured XML databases. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data. Madison: ACM Press, 2002. 358−369.

[43] Goldman R, Widom J. DataGuides: Enabling query formulation and optimization in semistructured databases. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA, eds. Proc. of the 23rd Int'l Conf. on Very Large Data Bases. Athens: Morgan Kaufmann, 1997. 436-445.

[44] Chen ZY, Korn F, Koudas N, Muthukrishnan S. Selectivity estimation for Boolean queries. In: Popa L, ed. Proc. of the 19th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems. Dallas: ACM Press, 2000. 216−225.

[45] Polyzotis N, Garofalakis M. Structure and value synopses for XML data graphs. In: Bressan S, Chaudhri AB, Lee ML, Yu JX, Lacroix Z, eds. Proc. of the 28th Int'l Conf. on Very Large Data Bases. Hong Kong: Morgan Kaufmann, 2002. 466-477.

[46] Polyzotis N, Garofalakis M, Iosnnidis Y. Selectivity estimation for XML twigs. In: Titsworth F, ed. Proc. of the 20th Int'l Conf. on Database Engineering. Boston: IEEE Computer Society, 2004. 264-275.

[47] Zhang N, Ozsu MT, Aboulnaga A, Ilyas IF. XSEED: Accurate and fast cardinality estimation for XPath queries. In: Ling L, Andreas R, Kyu-Y W, Jianjun Z, eds. Proc. of the 22nd International Conference on Data Engineering. Atlanta: IEEE Computer Society, 2006. 61

[48] Schmidt AR, Waas F, Kersten ML, Florescu D, Manolescu I, Carey MJ, Busse R. The XML benchmark project. Technical Report, INS-R0103: CWI. 2001.

[49] Zhang C, Naughton JF, DeWitt DJ, Luo Q, Lohman GM. On supporting containment queries in relational database management systems. In: Aref WG, ed. Proc. of the 20th ACM SIGMOD Int'l Conf. on Management of Data. Santa Barbara: ACM Press, 2001.425-436.

[50] Wu YQ, Patel JM, Jagadish HV. Estimating answer sizes for XML queries. In: Jensen CS, Jeffery KG, Pokorný J, Saltenis S, Bertino E, Böhm K, Jarke M, eds. Proc. of 8th International Conference on Extending Database Technology. Prague: Springer, 2002. 590−608.

[51] Aboulnaga A, Alameldeen AR, Naughton JF. Estimating the selectivity of XML path expressions for Internet scale applications. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases. Roma: Morgan Kaufmann, 2001. 591−600.

[52] Jiang HF, Lu HJ, Wang W, Yu JX. Containment Join Size Estimation: Models and Methods. In: Halevy AY, Ives ZG, Doan A, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data. San Diego: ACM Press, 2003. 145−156.

[53] Matias Y, Vitter JC, Wang M. Wavelet-Based histograms for selectivity estimation. In: Haas LM, Tiwary A, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Seattle: ACM Press, 1998. 448−459.

[54] Lim L, Wang M, Padmanabhan S, Vitter JS, Parr R. Xpath learner: An on-line self-tuning Markov histogram for XML path selectivity estimation. In: Bressan S, Chaudhri AB, Lee ML, Yu JX, Lacroix Z, eds. Proc. of the 28th Int'l Conf. on Very Large Data Bases. Hong Kong: Morgan Kaufmann Publishers, 2002.442-453.

[55] Ramanath M, Zhang LZ, Freire J. Incremental maintenance of schema-based XML statistics. In: Donald F. Shafer, ed . Proc. of the 21st IEEE Int'l Conf. on Data Engineering . Tokyo: IEEE Computer Society, 2005. 273−284.

计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
43(2):288~294, 2006

基于直方图的 XPath 含值谓词路径选择性代价估计

王 宇[1] 孟小峰[2] 王 珊[2]
[1](河北大学计算中心 保定 071002)
[2](中国人民大学信息学院 北京 100872)
(sandpiperwy@yahoo.com.cn)

Using Histograms to Estimate the Selectivity of XPath Expression with Value Predicates

Wang Yu[1], Meng Xiaofeng[2], and Wang Shan[2]
[1](*Computer Center*, *Hebei University*, *Baoding* 071002)
[2](*Information School*, *Renmin University of China*, *Beijing* 100872)

Abstract Selectivity estimation of path expressions is the basis of XML query optimization and also intense research interest. A path expression may contain multiple branches with value predicates. Some of the values and the nodes of the XML data are highly correlated. Previous methods of selectivity estimation rarely take that relation into consideration, and assume, instead, that the selectivity of attribute values on different nodes and structures is independent and uniform. In this paper, a novel value histogram is proposed, which captures the correlation between the structures and the values in the XML data. Also defined are six operations on the value histograms as well as on the traditional histograms that capture nodes positional distribution. Based on such operations, the selectivity of any node (or branch) in a path expression can be estimated. Experimental results indicate that the method provides accuracy especially in cases where the distribution of the value or structure of the data exhibit a certain correlation without any independent assumption.

Key words XML; query optimization; selectivity; histogram; predicate

摘 要 路径选择性代价估计是 XML 查询优化的基础,也是研究的热点. 目前的方法采用大量正态分布和独立性分布假设是造成误差的根本原因. 定义了一种新颖的值-位置直方图用于统计 XML 数据中的结构和值的分布情况,并提出了 6 种直方图运算. 在此基础上,给出用直方图计算估计路径中任一结点选择性的方法. 实验证明,这种方法无需独立性分布假设,也能在数据结构和数值分布不均匀的情况下,精确地估计路径选择性代价.

关键词 XML;查询优化;选择性;直方图;谓词

中图法分类号 TP311.13

1 引 言

用 XPath 表示的多谓词复杂路径是 XQuery 的核心表达式,也是影响 XML 查询执行效率的关键因素. 如何优化执行多谓词复杂路径是人们关注的焦点问题. 复杂路径表达式包含多个谓词分支,如何精确地估计位于不同分支结点的选择性成为研究

收稿日期:2004-10-09;修回日期:2005-06-22
基金项目:国家自然科学基金项目(60073014,60273018);国家"八六三"高技术研究发展计划基金项目(2002AA116030);教育部科学技术重点基金项目(03044);河北大学博士科研启动基金项目(Y2005050)

的热点.

在含值谓词复杂路径中,既隐含数据之间的嵌套结构,更有对分散在结构中的值的计算. 为了精确地计算结点的选择性,需要综合考虑表达式中所有结点对该结点的影响. 传统方法在计算时需使用正态分布和独立性分布等假设. XML 数据有复杂的层次结构,相关结点的分布以及结点值的分布很难满足传统代价计算常用的分布假设,导致代价估计的误差.

我们设计了一种新颖的二维直方图,正确地反映 XML 数据中值与结构的关系,提出了一种基于直方图计算的复杂路径选择性代价计算方法,把值与结构的相关性隐含在直方图的计算中. 构造了模式与直方图相结合的统计信息模型,给出了利用模式信息生成高效代价计算树的方法. 通过与 XSketch 方法的比较实验,从存储代价、计算效率和计算精确性 3 个方面证明了本文提出方法的有效性和先进性.

2　相关工作比较

目前 XML 信息统计和代价计算方法主要分为两种:一种基于模式,从宏观角度掌握数据的整体分布情况,基于一定的分布假设计算复杂路径的查询代价[1~5];另一种用直方图存储数据之间的位置关系,计算路径或者结构连接的代价[6,7].

Lore[1]方法从数据中提取模式信息(DataGuide),基于父子结点分布的独立性假设计算长路径的选择性. 其计算方法基于 Markov 链思想,只适用于没有分支条件的简单路径. 文献[2]采用 Markov 表保存路径信息,相当于 Lore 方法的改进和丰富. 文献[3]等采用后缀树方法保存结构信息,需将查询分解为简单的子查询分别计算,然后按照独立性假设组合. 不支持数值等原子类型,只能够处理等值谓词.

二维位置直方图(PH)[6]或一维位置直方图(PLH)[7]方法可用于计算结构连接运算代价. 位置直方图只保存数据的位置关系,不能直接用于计算含值路径的选择性. 本文在计算中引用了上述结构,但计算方法有本质的不同. 并且,本文提出的 PD 和 PA 运算解决了用直方图计算父子关系代价的问题.

已知的综合利用模式信息和直方图的工作有两个,StatiX[5]和 XSketch[4,8,9]. StatiX[5]用一维直方图方法统计分支结点的路径分布信息,仅能根据父亲计算儿子的分布情况. XSketch[4,8,9]利用模式信息统计数据分布的整体情况,采用传统多维直方图保存多个值对某类结点的影响. 但这种方法需要预先分析数据之间的相关性. 当路径中包含多个含值谓词时,仍然需要独立性分布假设综合计算结果.

3　值-位置直方图

XML 数据通常被模型为树状结构,XML 树中的每个结点可对应惟一的一个编码($start$, end),称为位置标识. 位置标识满足如下特点:①祖先位置标识严格地包含后代位置标识;②兄弟位置标识互不重叠. 根结点的位置标识区间最大,我们称之数据编码区间,如图 1 所示:

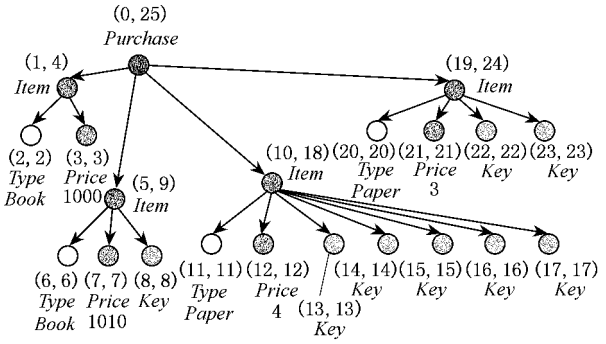


Fig. 1　Example data tree and its region codes.

图 1　实例数据与位置标识

设 XML 数据编码区间为[min, max],某类型结点内容的值域为[$V_{\min}$, $V_{\max}$],在平面坐标系中,x 轴为值所属结点的编码区间,y 轴为值域. 某类结点集合在坐标系上的分布称为值-位置分布图.

图 1 中 *Price* 的值-位置分布如图 2(a)所示. 点 a 的坐标为(7,1010),表明其位置标识的 *Start* 为 7,*Price* 值为 1010.

定义 1. 将值-位置分布图的 x 轴等分为 g 个格,y 轴根据不同类型和值域范围划分为 m 格,分别统计结点在每个格中的结点个数,构成值-位置直方图,记为 ***VH***.

值-位置直方图的每一格统计的是结点位置标识中 *Start* 和结点的值包含在本格区间中的结点个数,如图 2(d)所示. ***VH***[1][3] = 1 表示 *Price* 值在[757.5,1010], *Start* 值在[6.25,12.5]区域范围内的结点个数为 1 个.

值-位置直方图体现值的分布与结构之间的关系. 若有谓词约束某值域,可以通过值-位置直方图很容易地得到满足谓词的结点位置标识的 *Start* 分

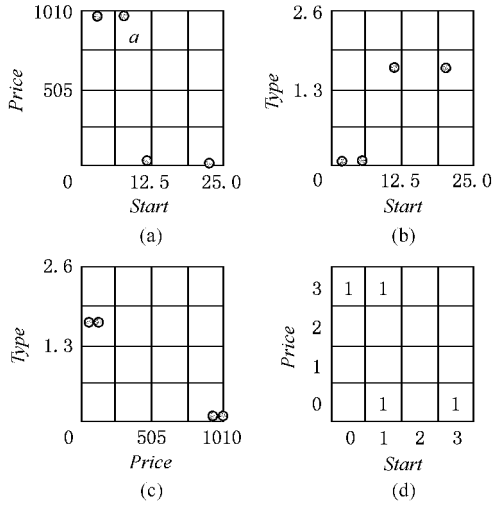


Fig. 2 *VH* vs. two dim. value distribution. (a) Value/code distribution of *Price*; (b) Value/code distribution of *Type*; (c) Value distribution of *Type* and *Price*; and (d) ***VH*** of *Price*.

图 2 值-位置直方图与二维值分布比较.(a) *Price* 值-位置分布;(b) *Type* 值-位置分布;(c) 二维值分布;(d) *Price* 值-位置直方图

布情况. 而 *Start* 值对应结点在树中的先序遍历值,隐含结点的位置分布情况. 因此,根据值-位置直方图可以获得满足谓词的结点的位置分布. 如查询(例 1):/purchase/item[*Type* =" book "][*Price* > 500]/*Key* 中求 *Item* 的选择性. 通过 *Price* 和 *Type* 的值-位置分布(图 2)可以看出,满足谓词 *Price* > 500 与 *Type* =" *book* "的 *Start* 值范围重合,均在[0, 12.5]内. 这个结果还可用于进一步计算 *Key* 的选择性. 而如果采用 XSketch[8]的方法,用二维值直方图反映不同 *Price* 和 *Type* 值域内 *Item* 的个数,如图 2(c)所示. 也可得到 *Item* 的个数为 2,但 *Item* 的位置是未知的,只能根据独立性假设进一步计算 *Item* 对 *Key* 的影响. 如果相关结点大于两个,则需要更高维的直方图,存储和计算代价会成幂级增长甚至导致方法的不可行. 这实际上只是孤立计算方法的一种改进. 而值-位置直方图分别统计不同值的位置分布,利用计算组合分布情况,真正建立了值与结构之间的桥梁,既保证的存储和计算效率,又不会丢失相关性信息.

体现结构分布的位置直方图有两种,Wu 等人[6]提出的 PH 和 Wang 等人[7]提出的 PLH. 后面我们将以 PH 为代表进行论述. 图 3 是一个 PH 实例图. 图 3(a)是图 1 中除 *Key* 外其他结点的二维分布图,图 3(b)为 *Price* 类结点的 4×4 ***PH***. 关于位置直方图的详细介绍请参考具体文献.

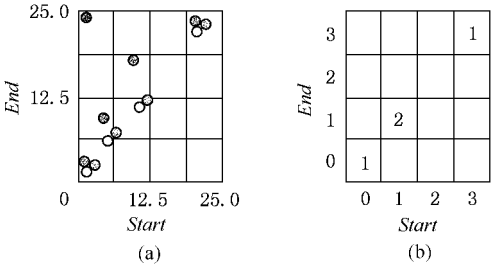


Fig. 3 An example of PH. (a) Distribution graph and (b) ***PH*** of *Price*.

图 3 位置直方图实例.(a) 二维数据分布;(b) Price 位置直方图

4 直方图基本运算

直方图运算有 6 种:值选始位置(V)和始位置转换(S)用于谓词计算,选后代(D)和选祖先(A)用于结构嵌套计算,自除后代(PD)和自除祖先(PA)用于计算父子关系时去掉祖先或后代. 参与运算的是直方图,运算的结果也是直方图. 考虑到篇幅问题,我们以 ***PH*** 为参数简要说明运算. PLH 与其思想相同,但计算方法稍有差别.

值选始位置运算(V):在 ***VH*** 不同列中分别统计满足值域条件的结点个数,构成一维始位置直方图 ***SH***.

始位置转换运算(S):统计 ***PH*** 每列结点之和,与 ***SH*** 对应格之比作为谓词的选择性;乘以 ***PH*** 对应列的格中结点个数,得到满足谓词的位置直方图 PH^P.

选后代运算(D):对后代位置直方图中每个格,根据其祖先位置直方图 PH^A 对应区域的格中结点个数,确定其选择性 $SelPH[i][j]$. 根据选择性,得到满足祖先的后代位置直方图.

选祖先运算(A):对祖先位置直方图中每个格,根据其后代位置直方图 PH^D 对应区域的格中结点个数,确定其选择性 $SelPH[i][j]$. 根据选择性,得到满足后代的祖先位置直方图.

自除后代运算(PD):在位置直方图中去除嵌套的后代结点.

自除祖先运算(PA):在位置直方图中去除嵌套祖先结点.

6 种运算的计算复杂度随直方图格数变化. 设 ***VH*** 格数为 $m \times g$,***PH*** 格数为 $g \times g$,则 V 运算复杂度为 $O(m \times g)$,S 运算复杂度为 $O(g^2)$,A,D,PA,PD 运算均为两个二维直方图的嵌套迭代,算法复杂度为 $O(g^4)$.

5　路径选择性计算(PM)

计算某结点在路径中的选择性需构造代价树(CT).后根序遍历执行 CT 得到的位置直方图即反映满足路径约束的结点的位置分布情况.在本节中,首先介绍简单谓词和简单路径中结点的选择性计算,然后通过实例分析多谓词复杂路径表达式中结点的选择性计算.

5.1　简单谓词选择性计算

我们称形如[$Type$="book"]的谓词表达式为简单谓词.计算简单谓词对结点的选择性的方法为以结点的 VH 为参数进行 V 运算,得到始位置直方图 SH.与结点 PH 进行 S 运算,得到满足谓词的位置直方图 PH^P.

5.2　简单路径选择性计算

若路径表达式为 a/b 或者 $a//b$,其中 a,b 为结点名,则该路径表达式为简单路径.简单路径中没有谓词出现.

若表达式为 $a//b$,通过 A 运算或 D 运算即可获得满足路径的 a 结点或 b 结点的 PH.若路径表达式为 a/b,需要在相应运算之前,先进行 PA 或 PD 运算以去掉祖先或者后代结点.

简单路径和简单谓词是两种基本的代价计算.所有复杂路径表达式均可分解为简单路径和简单谓词.

5.3　多谓词复杂路径选择性计算

多谓词复杂路径形如:$n_1[p_1^1][p_1^2]\cdots[p_1^{k1}]\phi n_2[p_2^1][p_2^2]\cdots[p_2^{k2}]\phi\cdots\phi n_m[p_m^1][p_m^2]\cdots[p_m^{km}]$,其中 n 为结点名,p 为谓词路径,ϕ 为"/"或"//".估计路径中某结点 n_x 的选择性,需从 3 方面计算:祖先对其的选择性,后代对其的选择性和谓词对其的选择性.下面分别论述.

计算来自祖先的选择性需从路径中的最左结点开始沿路径做 D 运算,如果结点之间的关系为父子,还应在子结点参加运算之前做 PD 运算.如果某结点有谓词约束,则应首先计算谓词约束对该结点的选择性,利用满足谓词约束的位置直方图参加运算.例如,若计算路径(例 2)$n_1/n_2//n_3//n_4[a>v]$ 中祖先结点对 n_3 的选择性,首先对 $PH(n_2)$ 做 PD 运算,然后与 $PH(n_1)$ 做 D 运算,再与 $PH(n_3)$ 做 D 运算.

计算来自后代的选择性则相反,从路径中最右结点开始沿路径做 A 运算.若结点之间的关系为父子,还需加入 PA 和 PD 运算.若某后代结点有谓词约束,则也应首先计算谓词约束对该后代结点的选择性.如计算例 2 中后代结点对 n_3 的选择性,首先计算谓词对 n_4 的选择性,利用满足谓词的 n_4 的 PH 沿路径做 A 运算,计算 n_4 对 n_3 的选择性,如图 4(a)所示:

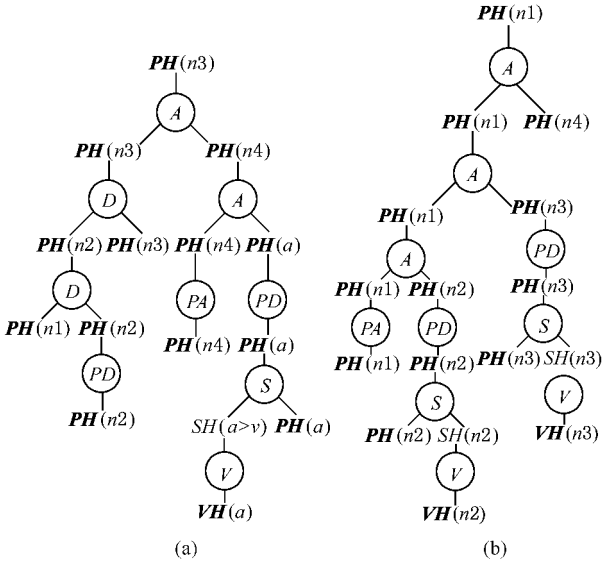


Fig. 4　An example CT. (a) $n_1/n_2//\underline{n_3}//n_4[a>v]$ and (b) $\underline{n_1}[n_2>v][n_3>w]//n_4$.

图 4　CT 实例.(a) $n_1/n_2//\underline{n_3}//n_4[a>v]$;(b) $\underline{n_1}[n_2>v][n_3>w]//n_4$

计算谓词对某结点的选择性与计算后代对其的选择性相似.若存在多个谓词,求分支结点的选择性是关键问题,需要按顺序分别计算不同谓词的选择性.如(例 3):$n_1[n_2>v][n_3>w]//n_4$ 中,n_1 为分支结点,其选择性 CT 如图 4(b)所示.首先计算谓词 n_2 对其的选择性,然后计算谓词 n_3,最后计算后代 n_4 的选择性.

PM 方法的中心思想是利用一系列直方图运算,综合路径中所有结点对估计结点选择性的影响.路径中结点间结构和值的相关性反映在直方图的不同区域计数的变化中.因此 PM 方法无需事先对结点的相关性进行分析,也无需任何独立性分布假设.PM 方法既可用于计算路径中任一点在整个路径中的选择性,也可用于计算任一点在部分路径中的选择性.但是,这种方法的缺点在于运算次数多,导致效率的下降.如例 3 中结点个数为 4,但 CT 中的运算却高达 10 个.为了提高代价计算的效率,我们采用一种改进的方法——模式与直方图相结合的方法(SGM).

6 模式指导的路径选择性计算(SGM)

6.1 统计信息模型

SGM 方法计算复杂路径选择性所需的统计信息由 3 部分组成:模式信息、不同类型结点的位置直方图和含值结点的值-位置直方图.

定义 2. 统计信息模型(*SI*)用三元组[N,E,T]表示. N 为结点类型集合,$\forall n \in N$,有 $tagName(Eid,C,R,V,L)$,$tagName$ 为结点名称,Eid 惟一地标识该结点;C 为该类型结点对应数据结点的个数;R 为递归嵌套标记,标记结点是否在递归嵌套的圈中;V 为其内容值域范围,若 n 不含值则 V 为空;L 为指向直方图的指针,每个结点对应一个位置直方图,如果该结点含值,则还有值-位置直方图. E 为边集合,表示结点之间的嵌套关系,为了简单起见,不区分属性边. T 为直方图与 Eid 对照表. 为了方便查找,我们以 Eid 为关键字在表上建立了 $B+$ 树索引.

我们采用绝对路径方法生成模式图[4]. 在 XML 树中同名的结点如果入边路径不同,则在模式图中视为不同类型的结点. 图 5 为统计信息模型实例:

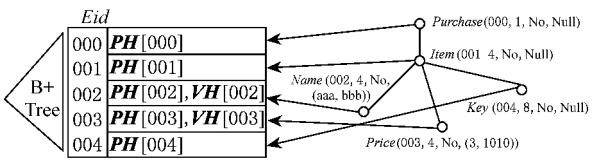


Fig. 5 An example of statistic information model.

图 5 统计信息实例

影响统计信息存储代价的因素有 4 个:模式的结点个数(n);含值结点个数(nv);直方图格数(g,m)和存储方法. 数据模式的结点个数决定了位置直方图的个数,含值结点个数决定了值-位置直方图的个数. 直方图格数决定每个直方图的规模. PH+VH 统计信息存储代价为 $n+g^2\times n+g\times m\times nv$. PLH+VH 统计信息存储代价为 $n+g\times 3\times n+g\times m\times nv$. 两种存储总代价均随格数呈幂级增长. 当格数很大时,存储代价是不能忍受的. 作为统计信息主要组成部分的直方图信息可表示为矩阵,通过对不同数据集统计信息实验,我们认识到矩阵中的大部分格的值为零,也就是说,矩阵是稀疏的.为此我们可以采用压缩存储的方法减少统计信息的存储代价,即只存储值不为零的格. 这种方法的压缩率非常高,使存储代价与格数增长呈线形关系.

6.2 模式指导代价树生成

采用绝对路径方法生成的模式信息,确定了结点的结构嵌套关系,可以用来判断表达式中的结点是否影响其他结点的选择性,从而跳过不必要的直方图运算.

定理 1. 若路径中某中间结点无谓词约束,无嵌套标记,则该结点对其祖先、后代结点的选择性为 100%.

证明略.

定理 2. 若路径表达式某中间结点 n_2 无谓词约束,有嵌套标记,则在形如路径 $\phi n_1[p_1]//n_2/n_3[p_3]$ 中对祖先,或在 $\phi n_1[p_1]/n_2//n_3[p_3]$ 中对后代的选择性为 100%. 证明略.

根据上述定理,SGM 方法在构造 CT 时,通过对结点谓词和嵌套标记的判断,跳过那些对其他结点选择性为 100%的中间结点.

假设路径中结点均无嵌套标记,SGM 方法生成例 2 和例 3 的 CT 如图 6(a),(b)所示. 与图 4(a)比较减少运算 5 次. 采用 SGM 方法生成例 3 的 CT 与图4(b)比较减少运算 3 次.

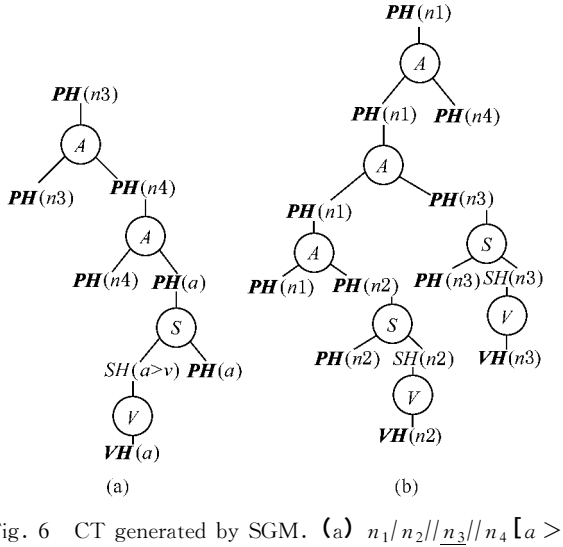


Fig. 6 CT generated by SGM. (a) $n_1/n_2//\underline{n_3}//n_4[a>v]$ and (b) $\underline{n_1}[n_2>v][n_3>w]//n_4$.

图 6 SGM 方法生成 CT.(a) $n_1/n_2//\underline{n_3}//n_4[a>v]$;(b) $\underline{n_1}[n_2>v][n_3>w]//n_4$

7 实验和算法分析

我们在 Win XP 中用 VC++ 编码实现本文提出的方法,实验机器的 CPU 为 PⅣ 1.6GHz,内存为 256MB. 测试集有两个,一个是 Xmark[10],数据结构嵌套关系复杂,数据和结构分布相对合理(相关性

不大). 另一个是由 XML SPY[11] 生成 Bib 数据集,数据模式含 11 个不同类型的结点,其中 6 个结点包含值. 结构相对简单,数据和结构分布扭曲,并且其结点值的分布高度相关.

7.1　统计信息存储代价

Bib 数据集在不同情况下的统计信息存储代价如图 7 所示,存储代价最小的是压缩 PLH + VH 方法. 其次是压缩 PH + VH 方法和无压缩的 PLH + VH 方法,成线性发展. 无压缩的 PH + VH 在方法存储代价稍高. XSketch 中常用的值分布直方图为二维. 对 6 个两两相关的值建立二维直方图的代价随格数平方级增长,当格数为 100×100 时,达到 625KB,与无压缩的 PH + VH 接近. 当用三维直方图统计相关值之间的相关性时,其存储代价远远超过无压缩的 PH + VH 方法. 当格数为 30×30 时,其存储代价超过 2MB.

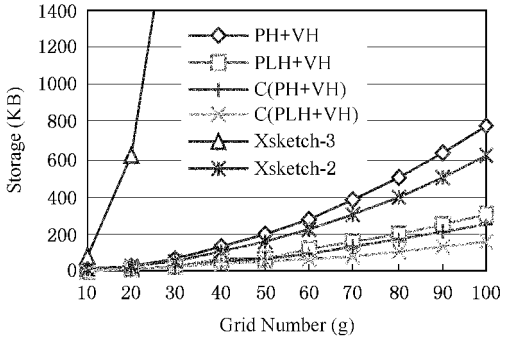


Fig. 7　Storage cost.

图 7　存储代价

7.2　选择性计算代价

我们设计了两组不同特征的查询,一组为单个谓词,但路径较长(>4). 另一组路径较短,但含值谓词较多(>2). 每组查询有 5 个实例. 我们以 2M 规模 XMark 数据集查询时间为基准计算代价估计时间与查询执行时间之间的比值.

图 8 比较了 PM,SGM 与二维直方图 XSketch 方法的对上述两组查询的平均代价计算效率. 与 PM 方法相比,由于有效地减少了结构选择计算,SGM 方法的效率提高一倍. 而 XSketch 方法无需多个直方图运算,计算代价较小.

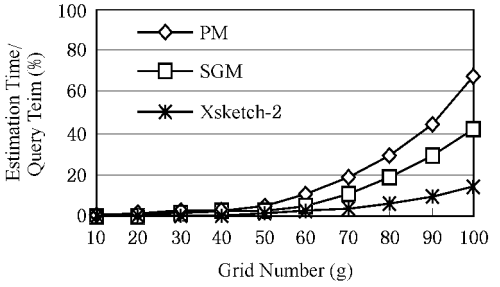


Fig. 8　Selectivity estimation cost.

图 8　选择性计算代价

7.3　选择性计算精度

我们采用相对误差计算方法. 查询数据集采用 2MB 数据集和 10MB XMark 数据集. 图 9 显示路径表达式中只有单个值谓词时的计算误差情况. 由于准确地抓住了结构与值之间的相关性信息,无论在数据分布非常扭曲的 Bib 数据集,还是在数据分布相对规则的 XMark 数据集,PH 方法和 PLH 方法在格数大于 15×15 时均能准确地估计查询选择性. 而这时存储空间不过 30KB,代价估计的时间不到查询时间的 1%,说明了本文方法的高效性和准确性. 而 XSkerch 二维直方图方法的误差很大,这在 Bib 数据集上表现得非常明显. 而且格数的增长不能有效减少误差.

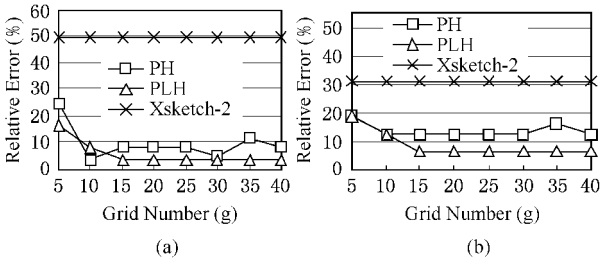


Fig. 9　Estimation accuracy with one predicate. (a) Bib DataSet and (b) XMark DataSet.

图 9　单个谓词表达式选择性计算精度. (a) Bib 数据集;(b) XMark 数据集

当格数很小时(<10),直方图计算误差较大,尤其是 PH 方法,这是由于格内计算误差造成的. 经过大量的实验我们发现,把格数控制在 20 左右能够获得存储、效率和准确性的最佳组合.

多个谓词的代价估计误差如图 10 所示. PH 与

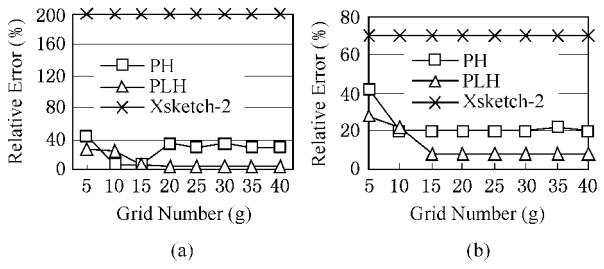


Fig. 10　Estimation accuracy with multi-predicates. (a) Bib DataSet and (b) XMark DataSet.

图 10　多个谓词表达式选择性计算精度. (a) Bib 数据集;(b) XMark 数据集

PLH 方法的平均误差较单谓词路径相比略有增长,其中,PH 方法增长较明显. 而 XSketch 基于独立性分布假设,导致误差的成倍增长.

8 总 结

XML 数据中普遍存在结构和数值的相关性. 以前的研究工作采用独立性分布假设,割裂地讨论结构相关和值相关问题,是计算误差的根本原因. 本文提出一种直方图计算方法,实验证明本文提出的方法在数据分布扭曲和数据结构复杂的情况下,均能获得有效精确的估计结果. 这种方法既可用于最终结果大小的估计,也可用于计算中间结果选择最优执行计划.

参 考 文 献

1 J. McHugh, J. Widom. Query optimization for XML. In: Proc. 25th VLDB Conf. San Francisco: Morgan Kaufmann, 1999. 315~326

2 A. Aboulnaga, R. A. Alameldeen, J. Naughton. Estimating the selectivity of XML path expressions for internet scale applications. In: Proc. 27th VLDB Conf. San Francisco: Morgan Kaufmann, 2001. 591~600

3 Z. Chen, V. H. Jagadish, F. Korn, *et al*. Counting twig matches in a tree. In: Proc. 17th ICDE Conf. Los Alamitos, CA: IEEE Computer Society Press, 2001. 595~604

4 N. Polyzotis, M. Garofalakis. Statistical synopses for graph-structured XML databases. In: Proc. 2002 ACM SIGMOD Conf. New York: ACM Press, 358~369

5 J. Freire, R. Jayant, M. Ramanath, *et al*. StatiX: Making XML count. In: Proc. 2002 ACM SIGMOD Conf. New York: ACM Press, 2002. 181~191

6 Y. Wu, J. Patel, H. Jagadish. Estimating answer sizes for XML queries. In: Proc. 8th EDBT Conf. Berlin: Springer, 2002. 590~608

7 W. Wang, H. Jiang, H. Lu, *et al*. Containment join size estimation: Models and methods. In: Proc. 2003 ACM SIGMOD Conf. New York: ACM Press, 2003. 145~156

8 N. Polyzotis, M. Garofalakis. Structure and value synoposes for XML data graphs. In: Proc. 28th VLDB Conf. San Francisco, CA: Morgan Kaufmann, 2002

9 N. Polyzotis, M. Garofalakis, Y. Ioannidis. Selectivity estimation for XML twigs. In: Proc. 20th ICDE Conf. Los Alamitos, CA: IEEE Computer Society Press, 2004

10 CWI. Xmark. http://monetdb.cwi.nl/xml, 2003-01

11 ALTOVA, Inc. XML Spy. http://www.xml.com/pub/p/15, 2004

Wang Yu, born in 1973. Ph. D. Her main research interests include Web data management and XML database.

王宇,1973 年生,博士,主要研究方向为 Web 数据管理、XML 数据库.

Meng Xiao Feng, born in 1964. Professor and Ph. D. supervisor. His main research interests include Web data integration, XML database and mobile database.

孟小峰,1964 年生,教授,博士生导师,主要研究方向为 Web 数据集成、XML 数据库、移动数据管理等.

Wang Shan, born in 1944. Professor and Ph. D. supervisor. Her main research interests include data warehousing and business intelligent technology, mobile database system and Web data management.

王珊,1944 年生,教授,博士生导师,主要研究方向为数据仓库、商务智能、移动数据库、Web 数据管理等.

Research Background

This work is supported by the 863 High Technology Foundation of China under grant number 2002AA116030, the Natural Science Foundation of China (NSFC) under grant number 60073014, 60273018, the Key Project of Chinese Ministry of Education (No.03044), and the Doctoral Foundation of Hebei University.

As XML becomes the standard for data exchanging over the Internet, much research has been devoted to the efficient support of XML queries. Compared with traditional query optimization technologies, XML query optimization has complex data model, weak schema information supporting, and insufficient relative basic research. So it needs some particular technologies. An XML query, expressed by a path expression, may contain branches with predicates, and each branch may have different impact on the selectivity of the entire query. If the branch with less selectivity is queried first, the intermediate result will have a relatively small size and the query efficiency can be improved. XML data is a hybrid of structures and values, some of which are highly correlated. Previous methods for selectivity estimation deal with the distribution of the value and the structure by using independent assumption, and have not taken into consideration the predicate value correlation among the different branches. We design a novel method connecting the isolated estimation together like a bridge. The method is of great accuracy especially when the distribution of the value or structure of the data is very skew.

Wuhan University Journal of Natural Sciences,11(5):1192-1196, Nov., 2006.
(The Third Web Information System and Application(WISA2006)

OrientX: an Integrated, Schema-Based Native XML Database System

☐ **MENG Xiaofeng, WANG Xiaofeng, XIE Min, ZHANG Xin, ZHOU Junfeng**

School of Information Renmin University of China，100872, Beijing

Abstract: The increasing number of XML repositories has stimulated the design of systems that can store and query XML data efficiently. OrientX, a native XML database system, is designed to meet this requirement. Compared with other native XML databases, OrientX has two main features: First, XML schema is fully supported in OrientX, because schema has been proved to be of great importance to XML data storage, Indexing, query optimization and access control; Second, OrientX is an integrated system that meets various requirements on XML data repository, which includes a native storage subsystem, several XML query evaluators, a composite index manager, a cost-based XQuery Optimizer, an Access Control module and an extension to XQuery1.0 for XML update. The main contributions of OrientX are: a)We have implemented an integrated native XML database system, which supports native storage of XML data, and based on it we can handle XPath& XQuery efficiently; b)In our OrientX system, Schema Information is fully explored to guide the storage, optimization and query processing, which we show can boost the system performance remarkably.

Keywords: XML; Database
CLC Number: TP 391

Received date: 2006-03-25
Foundation item: supported by the grants from the Natural Science Foundation of China (60573091, 60273018)
Biography: MENG Xiaofeng(1964-), male, Professor, research direction: web integration, XML Database and mobile data management. Email: xfmeng@ruc.edu.cn

0 Introduction

XML is a self-describing language, and has become the new de facto standard for data representation and exchange on Internet. The increasing number of XML repositories has stimulated the design of systems that can store and query XML data efficiently. Many systems have been designed to meet the goal. All of these systems can be divided into two categories: one falls into the Relational way, which utilizes the table-based storage model of existing DBMS, and needs to map XML data into two-dimension tables(e.g.[1]); The other is a native way, which develops a tree-based storage strategy for XML data, and doesn't need an additional mapping.

The relational strategy introduces an additional transform between the logical XML data and its physical relational storage. As hierarchy of XML data is complex, and optional or repeatable sub-elements are allowed in it, the mapping from XML data to relational data often results in large amount of tables. Due to this storage strategy, the complexity of XQuery is beyond the capacity of SQL expressiveness. None of the existing traditional DBMS could be adequately customized to support XML, despite all claims of their vendors.

In a native XML database, XML data is stored directly, which retain XML data's natural tree structure (for short, we call it data tree, and call elements of XML document in it as nodes). The query processing engine can handle query languages such as XQuery&XPath directly on the tree structure. As it reserves the tree structure of XML data, native strategy can avoid the mapping operations. Some native XML databases have appeared, for example, Timber[2], Natix[3], tamino[4] and so on. Most of these systems are schema-independent, that is to say, the schema is not a necessity of the system. But OrientX believes that schema plays a crucial role in XML data management and is indispensable. XML schema describes the tree structure and the data type definitions of XML data, i.e. nodes in data tree are organized together based on its schema. We argue that making good use of schema information could improve the efficiency of storing and retrieving XML data remarkably. XML schema is of great use in the following aspects of a native XML database:

- As related(neighbor) nodes in XML data tree are

likely to be queried at the same time, it is better to store them together, and XML schema provides the information of how nodes are related.

- Path index is very important to XML query evaluation, and schema makes good support to it.
- Schema can be used in validation of query and update processing.
- Query optimizer can collect statistical information by integrating the schema.
- Schema is the precondition to access control of XML data.

OrientX stores XML data using its own storage subsystem---OrientStore, in which XML data is stored in pages on disk, the granularity of storage can be changed according to schema. When stored data are loaded into memory, tree structure of the related nodes is built dynamically, therefore OrientStore offers a DOM-like navigation interface to the upper modules. Besides OrientStore, there is a holistic Index Structure called SUPEX, in which all path index and some value index are put together. XPath queries can be handled efficiently through fast navigation and join operation with the help of SUPEX. There are two query Evaluators for XQuery in OrientX. One is navigation-based, the other is algebra-based.

1 Motivation

The goal of OrientX is to build a native database system, which can manage XML repositories conveniently and efficiently.

Among all criterions of such a system, we first focus on the two main aspects: storage and query, for a well-designed storage and an efficient query evaluator form the basis of a good database system.

Two kinds of information need to be stored to preserve the structure of the document tree: node and edge. Node denotes element's data type and content. Edge denotes the direct relationship (Parent-Child in data tree) between nodes.

Another issue in XML data management is query, on which much work has been done too. A group of automata-based approaches are proposed for XPath and twig query processing, but they become incapable when facing complex XQuery statement. However, XQuery has become more and more popular because of its expressiveness and flexibility. There are two main methods to process XQuery, one is navigating through the data tree, and the other is using an algebra.

Besides storage and query, many problems remain in XML data management, such as indexing, query optimization, updating and accessing control on data repository. As XML is more and more popular today, both the industry and research communities show close interest to these problems.

2 An overview of OrientX

The overall architecture of OrientX is shown in Fig.1, it can be divided into three layers: data storage, access interface and execution engine.

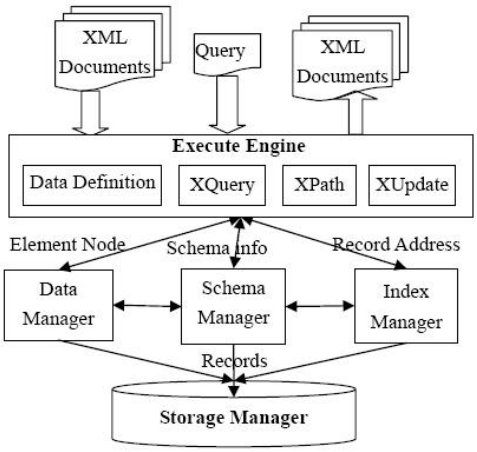


Fig. 1 OrientX Architecture

- Data storage

In OrientX, XML documents are organized in datasets according to their schema, which is a main feature of data repository. Interfaces for user to create and/or delete dataset are supplied. Documents should first be imported into certain datasets before any other jobs being done. Validation of document is done during importing process, if data does not conform to schema, a relevant error massage is reported and importing process will be interrupted. OrientX also supports document export operation. The detail of the storage strategy will be discussed in section IV.

- Access interface

The access interface consists of data manager, schema manager and index manager, which is uniform to upper level applications and data storage is hidden by it.

Data manager offers a uniform interface to access the storage subsystem. The only entity type the query evaluator can obtain is a logical element, which is the same as node. And some navigation methods are bound to logical elements to support navigation on data tree.

OrientX system is based on XML schema, and schema manager is a key module in the system. The idea and implementation of schema manager are discussed later in section V.

- Execution engine

The execution engine composes of several modules. The data definition defines a document in the dataset. XPath and XQuery evaluators are the two main parts of execution engine. The basic strategy of XPath processing is navigation, however, in co-operating, we also develop the index-based strategy that acts effectively, the detail has been described in our previous work[5].

Our previous work [5,6] has discussed storage module, schema management, index module and optimization module in detail, so here we only briefly describe the above four modules, we will focus on the new modules of our OrientX system, which include the query processing engine, update handling strategy and the access control module.

3 OrientStore: A multi-granularity storage strategy

Several native storage strategies have been developed[2,3,4,6]. According to the granularity of the records, these storage methods can be classified into Element-Based(EB),Subtree-Based(SB)and document-Based (DB). We observe that schema plays a key role in designing effective storage strategies for XML management systems. OrientX exploits schema information in the design and implementation of two storage strategies [7]: Clustering Element-Based (CEB), and Clustering Subtree-Based strategies. OrientX also implements the above schema-independent storage strategies DEB and DSB. Detailed description can be found in our previous work[6,8].

4 Schema Management

OrientX is schema-based. XML Schema strictly constrains the type and structure of data. So, data storing, retrieving and updating are all under the schema's guidance. Schema information can be used in data layout, in choice of index, in type checking, in user access control, and in query optimization. Schema in OrientX is consistent with the XML Schema standard. Detailed description can be found in our previous work[9].

5 SUPEX: A Schema-Guided Index Structure

SUPEX[5] consists of two structures: structural graph (SG), and element map (EM). SG is constructed according to the schema, and represents the structure summary of XML data; each node in it represents a list of elements in XML document with the same tag name as the schema node. And EM provides fast entries to the nodes in SG. Further information can be found in our previous work[5].

6 Query Processing

We have implemented several kinds of Query Processing Engines in our OrientX system based on the original work[6]. Currently two XML query engines have been used extensively in OrientX, one for XPath, the other for XQuery. Here we only introduce the techniques utilized in our new navigation-based XQuery evaluator[9].

- **Combine continuous steps in one XPath into a single path.** An XPath fragment has been separated into a series of step expressions, the navigation processing on it should be nested loop. But the only element we want is end step of the long XPath, all traversal for its ancestors is redundant. With the help of OrientStore and the SUPEX index, we can access any elements directly.
- **Reform syntax tree into reduced execution plan.** Structure of execution plan of navigation processing is similar to the structure of the syntax tree of XQuery statement. Especially, the key FLW(O)R structure in syntax tree is also the key operation in processing. Therefore, we use a "reduced"[1] syntax tree to denote the execution plan. Note some tricks in the reducing and reforming process:
  1) multi-processing units may be put together into one syntax node, for example, the FOR-Var binding, LET-Varbinding, WHERE-Predicate and the RETURN-EleConstruct form the FLWR node.
  2) multi-syntax nodes may be put into one processing unit; this is one kind of reducing operation

[1] Reduce here means omitting some hierarchies and nodes in the syntax tree.

We explain the reforming from syntax tree to execution plan with an XQuery example Q_1:

```
<results>{
  let $doc := document("bib.xml")
  for $t in distinct-values($doc//book/title)
  let $p := $doc//book[title = $t]/price
  return
    <minprice title="{ $t }">
      <price>{ min($p) }</price>
    </minprice>
}</results>
```

Q1's execution plan is shown in Fig 2. We can see that it is pretty small and is similar to Q1 in the structure. Most transforming actions are straightforward.

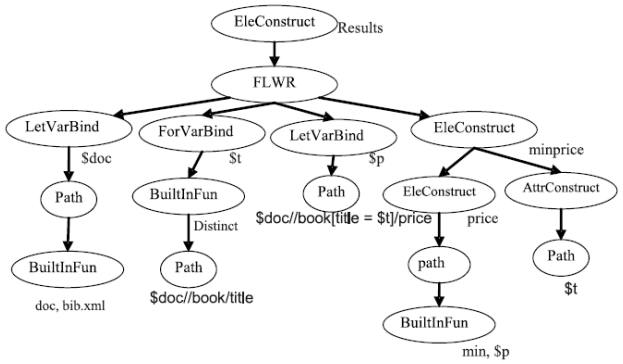


Fig. 2 Execution plan for Q_1

In execution, a pointer pointing to a node in the execution plan is hold during the whole query process. It points to the root in the beginning, and it travels through the plan tree in the top-down and left-right manner, if it encounters a source data node, for example, a ForVarBind node, then proper data is located by navigating on source document tree, on its way going, operations on subsequent "action" node are done on current located data. When the end of a FLWR is encountered, the pointer is redirected to beginning to this FLWR node except that there is no proper source data available. Query execution stops at the same time when the pointer stops.

The whole query process in OrientX is shown in Fig. 3, the white rectangle denotes processing modules and the rectangle in grey is auxiliary data or data structure. The chain composed of thin solid line is the data flow of XQuery statement, the chain composed of thick solid

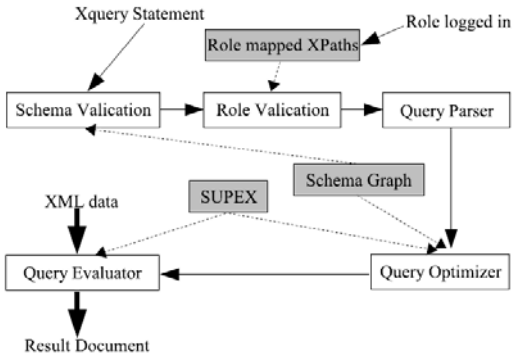


Fig. 3 Query process

Line is data flow of source XML data. The broken line denotes the fact that the auxiliary data is used in the target module.

7 Update

With the extensive use of XML in application over the Web, how to update XML data is becoming an important issue. So we currently implement a new update engine to the users. We extend XQuery with a FOR ...LET...WHERE...UPDATE structure for updates.

For XML documents with schema, we should check whether the update request violates the schema constraint. This is called update validation.

As elements are all encoded with region code in OrientX, we have to encode the newly inserted data right after inserting process, some efficient strategy is used in code encoding for updating during update/inserting[10]. Currently, update for index is not implemented, it seems that it is similar to the update for common relational index.

The update process is as follows: Firstly we locate the referred element in document. Secondly we validate the update request according to its schema. The validation includes the check of the new data's inner constraint and whether the new data as a whole can appear at the specified location. The request being valid means that all constraints are fulfilled, if not, it is rejected. Finally, valid request is evaluated.

8 Node-Mapping Role Based Access Control

Access control module is an essential part for an integrated database system. Because of the different data model between relation data and XML data, the access control mechanism in relational database is not capable of managing XML data any more. Some important aspects need to be reconsidered, such as the granularity of access control, the semantic of authority, the relation among the rights on relative node in XML structure. At the same time, large data capacity and alteration of the data also should be taken care of. In this section, we will discuss a new Node-Mapping Role Based Access Control module for XML data that is utilized in the OrientX system.

The module is excited from the fact that the part of

dataset one can access in an XML document being best described by one node or several nodes in schema graph. It is true that if role *A* is the superior to role *B*, then the part of dataset that role *A* can access should be the superset of the part of dataset that role *B* can access. Therefore, we can map the two roles to two nodes in XML schema so that ancestor is the superior and descendant is the junior. In this way we can achieve both the excellence of Role and the convenience of XML data access controlling.

We give the definition of role here: A role is a set of triples R = {*Node*, *Context*, *Action*}, where *Node* denotes the tag name of the root of the subtree in schema graph; *Context* is an XPath locating the unique position of the node in the schema, and the *Action* represents a collection of allowed operation on the node, including reading, inserting, deleting and updating.

Access rules can be defined on roles(nodes), for example, we have used the Dynamic Separation of Duty Relation(DSD) characteristic to solve the problem of illegal association information accessing. Roles can be assigned to the user in two ways: positive and negative. The positive roles assign the actions user can do, and the negative roles assign the actions user can not do. In OrientX, general roles and DSD roles are all supported for compatibility. A user can choose many general roles during one session, and only one DSD role during one session.

9 Conclusion and Expectation

In this paper, we described the system structure and design of OrientX, an integrated, schema-based native XML database proposed by Renmin University of China. We have explored many issues on XML data management and proposed some new ideas. We also proved that schema plays a crucial role in XML data management system. Right now, the storage, query and index parts mentioned in this paper have already been implemented, and the query optimization, access control parts are being integrated and will be completed soon.

References

[1] Florescu D, Kossman D. Storing and Querying Xml Data Using an Rdbms [J]. *IEEE Data Eng Bull* , 1999,**22**(3):27-34

[2] Jagadish H V, Divesh S, Wu Yuqing *,et al.* Timber: A Native Xml Database[J]. *The VLDB Journal*, 2002, **11**(4):274-291.

[3] Kanne C and Moerkotte G. *Efficient Storage of Xml Data*[M]. California: IEEE Computer Society,2000.

[4] McHugh J, Abiteboul S, Goldman R, Quass D, *et al*. Lore: A Database Management System for Semi-structured Data [J]. *SIGMOD Record*, 1997,**26**(3):54-66 .

[5] Wang Jing, Meng Xiaofeng, and Wang Shan. Supex: A Schema-guided Path Index for Xml Data[C] //*Proceedings of 28th International Conference on Very Large Data Bases(VLDB)*, Hong Kong ,2002.8.

[6] Meng Xiaofeng and Wang Yu. OrientX: A Native XML Database System[C].//*Proceedings of 20th NDBC*,Chang Sha: 2003.10(Ch)

[7] Li Quanzhong and Moon B. Indexing and Querying Xml Data for Regular Path Expressions[C]//*Proceedings of 27th International Conference on Very Large Data Bases(VLDB)*, Roma, 2001.9

[8] Meng Xiaofeng, Luo Daofeng, An Jing *,et al.* OrientStore: A Schema Based Native XML Storage System[C]// *Proceedings of 29th International Conference on Very Large Data Bases(VLDB)*,Berlin,2003.9

[9] Lu Shichao, Meng Xiaofeng, Lin Can , *et al.* Navigation Implementation for XQuery in OrientX [C]// *Proceedings of 20th NDBC* ,XiaMen,2004.10(Ch).

[10] Jiang Yu, Luo Daofeng, Meng Xiaofeng, *et al.* Dynamically Updating Xml Data:Numbering Scheme Revisited[J]. *World Wide Web*, 2005, **8**(1):.5-26

计算机研究与发展，卷43(增刊)：464－470，2006,11.
(第23届中国数据库学术会议，广州.)

XML 数据流上的有序 XPath 查询处理

谢敏[1] 王小锋[1] 张新[1] 孟小峰[1] 周军锋[1,2]

[1](中国人民大学信息学院，北京 100872)

[2](燕山大学计算机系，河北 秦皇岛 066004)

(xiemin@ruc.edu.cn)

摘要 XML 数据流上的查询处理是最近研究工作的一个热点，如何高效地处理 XML 数据流上的 XPath 查询是其中的核心问题。之前的相关工作主要考虑了无序 XPath 查询处理的情况，而在股票信息监控，新闻信息订阅等很多的 XML 数据流应用中常常需要对有序 XPath 查询进行有效的支持。对于有序 XPath 查询的处理，之前的方法需要将查询进行分解，然后通过连接将分解后的子查询得到的中间结果合并。针对有序 XPath 查询自身的特点，本文提出了在查询树上引入顺序和位置标记，记录查询结点之间的顺序关系，并在此基础上提出了一种创新的 XML 数据流上的 XPath 查询处理算法 OrderedXP。相比之前的工作，OrderedXP 能够大量地减少缓存的中间结果数目，而且不需要分解原来的查询，避免了额外的连接操作。详细的实验数据验证了 OrderedXP 能够显著地提高有序 XPath 查询在 XML 数据流上的执行效率。

关键字 XML 数据流；XPath；查询处理

中图法分类号 TP391

Ordered XPath Query Processing on XML Stream

Xie Min[1], Wang Xiaofeng[1], Zhang Xin[1], Meng Xiaofeng[1], and Zhou Junfeng[1,2]

[1](Information School, Renmin University, Beijing 100872)

[2](College of Computer Science, Yanshan University, HeBei Qinghuangdao 066004)

Abstract Recently, query processing on the XML stream is a hot topic in research community, in which how to efficiently handle XPath query on the XML stream is a core problem. On dealing with this problem, the previous work mainly concerns how to efficiently handle unordered XPath query, but in the application of XML stream like stock information monitoring, news feed monitoring and etc., we often need to handle ordered XPath query. On concerning these requests, the previous methods often break the XPath into some query Fragments and execute them separately, at the last stage these algorithms refer to an explicitly join to get the final results. On observing the characteristic of the ordered XPath query, we bring order specification and position specification into the query tree which may record the order relationship between the query nodes, and then we propose a novel XPath query processing method OrderedXP for XML stream, which can to the maximum extent reduce the number of intermediate results we cache in the memory compared to the previous work with the additional benefits of no decomposition and final join step. Extensive experiments show that our OrderedXP algorithm can handle all the ordered XPath query efficiently.

Keywords XML Stream; XPath; Query Processing

1. 引　言

可扩展标记语言(XML)[1]由于其易用性和强大的复杂数据描述能力，已经逐渐成为因特网上数据交换的标准。很多流行的数据库引擎已经开始通过增加对于这种新数据类型以及相关操作的支持来满足新的需求。XML 数据流上的应用是 XML 研究的一个很重要的方面，在新的基于 Web 的应用场景下，有研究已经提出，有效地处理 XML 数据流上的 *XPath* [2]，*XQuery*[3]查询将会成为下一代信息系统的特点[4，5]。

XPath[2]是 W3C 推荐的 XML 上的查询语言，用于抽取满足条件的 XML 文档片段。之前在 XML 数据流上

收稿日期：2006-05-25
基金项目：国家自然科学基金项目（60573091，60273018）

有很多关于 *XPath* 查询处理的工作[4，5，6，7]，这些工作主要集中在讨论XML数据流上处理XPath中结点间的父子(*Parent-Child*)关系，祖先后代(*Ancestor-Descendant*)关系，以及 *XPath* 中的谓词，分支结点。这些之前工作处理的 *XPath* 查询都不涉及数据上的顺序关系。我们把这些不涉及 XML 数据顺序关系的 *XPath* 查询称作***无序 XPath 查询***(*Unordered XPath Query*)。

但是在XML数据流的实际应用，比如股票信息监控和新闻信息过滤等，有很多的查询会对数据的出现顺序有一定的要求[8]。如图 1 所示，查询Q_1需要查找满足先后顺序要求的两个交易的情况，查询Q_3需要查找位置满足某一要求的特定新闻，而查询Q_2和Q_4则需要查找同时满足先后顺序和位置要求的特定交易信息和新闻信息。这些查询无论有先后顺序的要求还是有位置的要求，一个共同的特点就是对XML数据流上查询涉及到的结点的出现顺序有一定的限制。

Q_1	查找在IBM和联想交易发生之后的微软的交易
Q_2	查找IBM在与联想交易后发生的第1笔交易
Q_3	查找关于NBA火箭队的头10条新闻
Q_4	查找北京申奥成功新闻发布后出现的头10条新闻

图 1 XML 数据流上的例子查询

我们把这些查询中涉及结点上的先后顺序或者位置的要求称作查询的***有序要求***(*Order Request*)。这些有序要求直接转化成 *XPath* 中的 *Following* 关系，*Following-Sibling* 关系和 *Position* 谓词。我们把这种含有 *Following* (*Preceding*)关系，*Following-Sibling*(*Preceding-Sibling*)关系或者 *Position* 谓词的 *XPath* 查询称为有顺序要求的 *XPath* 查询，在文章里为了描述方便，简称为***有序 XPath 查询***(*Ordered XPath Query*)。下面的讨论中，我们将假设查询中不含有 *Preceding* (*Preceding-Sibling*)轴，可以通过之前的工作[4]将其转换成 *Following*(*Following-Sibling*)轴进行处理。

之前的工作在处理XML数据流上的 *XPath* 查询的时候，当查询中出现 *Following* 或者是 *Following-Sibling* 轴时，往往需要从 *Following*，或者是 *Following-Sibling* 轴处将查询分解成为多个查询，然后将其各自的结果通过一个额外的连接操作来得到最后的查询结果[4]。这种分解的方法会造成两个方面的问题：首先，因为将查询分解成了几个部分，会造成缓存很多无用的中间结果，比如当查询是 *//A/B/Following-Sibling::C*，而 XML 数据如图 2(*a*)所示，我们将查询分解成两个查询，*A/B* 和 *A/C* 将会造成缓存 *n* 个无用的中间结果 *b* 和 *c*，而当查询是 *//A/Following::B*，XML 数据如图 2(*b*)所示，我们将查询分解成为两个查询*//A* 和*//B* 将会造成缓存无用的中间结果 *b*；其次，在得到分解后的子查询的中间结果后，我们需要一个额外的连接操作才能得到最后的结果。而对于 *Position* 谓词，之前所有的方法都没有考虑XML数据流上处理含有这种谓词的 *XPath* 查询。

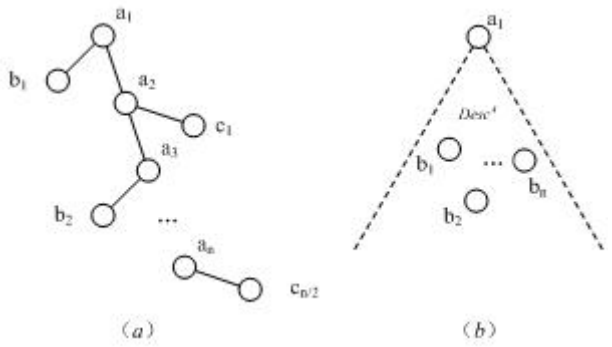


图 2 两个 XML 文档的例子

针对上面提到的XML数据流上有序 *XPath* 查询的需求以及之前工作中存在的问题，我们创新性地提出了一种在 XML 数据流上处理有序 *XPath* 查询的方法。本文的贡献如下：

- 分析了 XML 数据流上的三种类型有序关系 *Following*，*Following-Sibling*，*Position* 各自的特点，在查询树上引入了 O-Spec(顺序标记)和 P-Spec(位置标记)，通过这些标记的组合可以完整地描述 *XPath* 查询的有序要求涉及的结点之间关系。
- 结合顺序标记 O-Spec 和位置标记 P-Spec，本文给出了一种创新算法 **OrderedXP**，相对于之前的工作可以有效地减少缓存的中间结果的数目，并且避免了额外的连接操作。
- 详细的实验数据验证了本文提出的 OrderedXP 查询算法能非常有效地处理 XML 数据流上的有序 *XPath* 查询。

本文后面部分的组织如下：第 2 部分描述有序 *XPath* 查询，及各种有序关系在流处理情况下的特点；第 3 部分阐述 OrderedXP 算法；第 4 部分讨论实验情况；第 5 部分介绍相关工作；最后一部分是我们的工作总结和展望。

2. XML 数据流上有序 XPath 查询及其分析

2.1 XML 数据流模型和有序 XPath 查询

XML 是具有有序特性的树结构，结点包括元素(*Element*)结点和文本结点。XML 上结点的先序遍历顺序反映了该结点在 XML 文件中的位置，对应到结点

的文档顺序(*Document Order*)。

XML 数据流是由一系列连续的 SAX 事件流组成。而这些 SAX 事件流由 SAX 分析器分析 XML 文件流的时候产生，对每个 XML 上的元素结点，开始和结束标签都会对应到一个 SAX 事件。

本文处理的有序 *XPath* 查询是 W3C 推荐的 *XPath* 查询[2]的一个子集，如图 3 所示。如引言中所述，我们不考虑 *Following*，*Ancestor* 等逆向轴，在查询中涉及到这些轴的时候，我们可以用现有的工作[4]首先将其进行转换，再用本文的方法进行处理。

```
PathExpr ::=  '/' Path | '//' Path
Path     ::= Path Step Path | Path '[' Pred ']' | label
Step     ::=  '/' | '//' | 'following' | 'following-sibling'
Pred     ::= Path | PosPred
PosPred  ::=  'Position()' Oprel PosDef
Oprel    ::=  '<' | '<=' | '>' | '>=' | '='
PosDef   ::= Dec-Number | 'Last()' - Dec-Number
```

图 3 本文处理的 XPath 查询

在详细分析各种有序关系的特点之前，我们在这里首先给出在 XML 数据流上的查询处理一些基本假设和定义。

定义 1：*对于一个数据结点 T 来说，假设 T 对应查询结点* Q_T，*那么如果对于* Q_T *在查询树上的所有后代结点* Q_T^C，*都能在数据中找到一个匹配结点* T^C，*而且 T 以及所有的这些匹配结点* T^C *满足* Q_T *为根的查询子树上的所有的边要求(假设* Q_T *对应的子树为* SQ_T，$\forall x,y \in \{T \cup T^C\}, x \neq y$，*如果* x，y *对应的查询结点* Q_T，Q_T' *在* SQ_T *上有边关系 Edge-Req，那么* x，y *也满足 Edge-Req)，我们就说数据结点 T 有一个****子查询匹配****。*

引理 1：*XML 数据流上无序 XPath 查询处理过程中，遇到一个数据结点 T 的结束标签</T>时，能够判断：这个结点 T 要么有一个子查询匹配，要么没有。*

这个引理的证明可以由之前工作[4]中得到。但是当查询中涉及 *Following-Sibling* 或者是 *Following* 关系以及 *Position* 谓词时，很显然，我们在遇到</*T*>时，不能确定 *T* 是否有子查询匹配。

有序 *XPath* 查询在遇到结点结束标签时不能确定结点的子查询匹配，这种情况是之前的工作所不能处理的。在这里，我们提出 XML 数据流上处理这种有序 *XPath* 查询时，一种好的策略应该具有的特征如定义 2 所述。

定义 2： *我们说 XML 数据流上一个有序 XPath 查询处理是****最优的****(Optimal)，当且仅当因为查询中的有序关系结点 T 在遇到结束标签被缓存之后：1.如果结点 T 是有用的，能够在遇到第一个子查询匹配时对结点 T 做出判断；2.如果结点 T 没有匹配，也可以在遇到第一个能够判断 T 没有子查询匹配的标签流过时对 T 做出判断。*

我们称最优处理第一方面的条件为**最早肯定条件**，第二方面条件为**最早否定条件**，如果一个查询处理策略同时满足最早肯定条件和最早否定条件，那么这个查询处理策略是**最优的**。

如果一个有序 *XPath* 查询处理是最优的，那么算法必然能够以最少的缓存消耗完成 XML 流上的查询。下面两个小节我们将分别分析不同的有序要求在处理时要达到最优目标所需要的条件。

下文中为了描述方便，*Following-Sibling* 简称为 F-S， *Ancestor-Descendant* 简称为 A-D，而 P-C 代指 *Parent-Child*。对于查询中的 *A/Axis::B*，称 *A* 为轴关系的上下文(*Context*)结点，*B* 为轴关系的目标(*Target*)结点。

2.2 Following-Sibling,Following 分析

F-S 在处理时会遇到两种情况。第一种情况，F-S 关系的上下文结点跟其父查询结点是父子边关系，如图 4(*a*)所示。决定 F-S 关系中上下文结点 *B* 是否有子查询匹配就需要看在 *B* 结束之后结束的 *C* 中是否有 *B* 的右兄弟。而在 *B* 结束标签之后结束的 *C* 结点从图 4(*b*)中可以看到有 4 种情况。对应最优处理策略的两个条件，对于 *B* 而言：如果 *B* 是有子查询匹配的，那么在遇到第一个 *C* 兄弟结束标签时，我们就能对 *B* 做出判断；而如果 *B* 没有右兄弟，我们能够最早决定 *B* 结点没有子查询匹配的事件是 *A* 结点的结束事件。

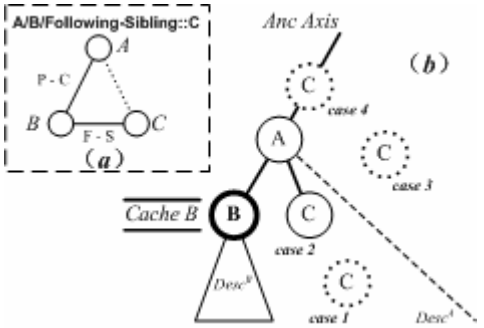


图 4 F-S 关系对数据结点的影响(1)

另一种情况，当 F-S 上下文结点 *B* 和其查询父结点是 A-D 关系的时候，找到一个最优的处理策略会比较困难。如图 5 (a)所示的 *XPath* 查询，我们可以看到在 *B* 结点结束之后结束的 *C* 的情况增加到 6 种。如果我们还是用上面的策略来处理的话，我们就不能保证查询处理最优要求的最早否定条件。原因是我们应该在数据中遇到 *B* 的父亲 *D* 的结束标签时就可以判断 *B* 没有子查询匹配，在 *A* 结束时判断就会将决定 *B* 的时机延后。但是因为我们在数据流过之前并不知道 *B* 的父亲结点是 *D*，所以需要在查询时动态做记录，才

能保证处理策略的最优。在本文里，我们暂时用之前的方法来处理这种上下文结点和其父查询结点是 A-D 关系的 F-S 的情况，在 *A* 结束时清除 *B* 结点的缓存。

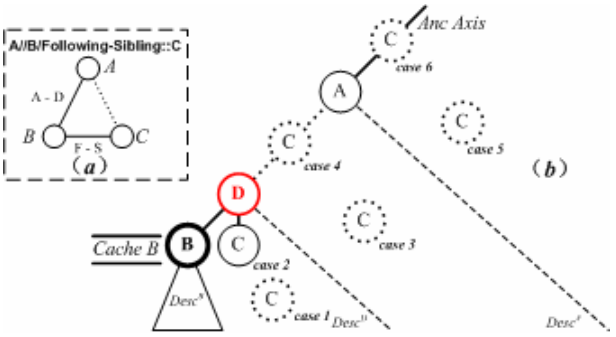


图 5 F-S 关系对数据结点的影响(2)

Following 的情况与 *Following-Sibling* 有所不同。对于一个查询 *A/Following::B* 来说，从图 6(*a*)可以看出，在 *A* 结束之后结束的 *B* 一共有两种情况：*A* 的祖先；*A* 的 *Following*。就查询处理策略最优的第一个条件而言，如果 *A* 有子查询匹配的话，我们需要缓存 *A* 直到 *A* 的有子查询匹配的 *Following* 结点出现；而对于第二个条件，如果 *A* 是没有子查询匹配的，那么我们需要将 *A* 缓存直到 XML 文档流结束，之前的任意时间我们释放 *A* 的缓存都会造成丢失解。

在查询中出现 *Following* 轴时还会导致**级联缓存**的问题。考虑查询 *C/A/Following::B*，如图 6(b)所示，我们需要对 *Following* 轴的做特殊的级联缓存来处理这种情况。

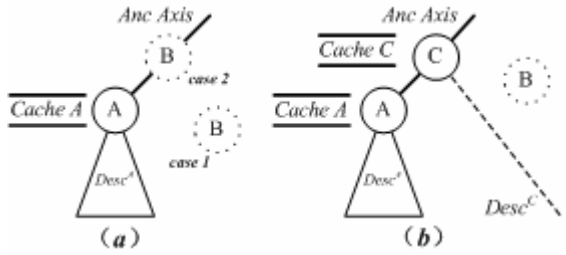


图 6 Following 关系对数据结点的影响

2.3 Position 谓词分析

Position 谓词(位置谓词)可以分为两种情况，不含有 *Last()*的简单位置谓词和含有 *Last()*的复杂位置谓词。对于两种谓词我们都可以通过在缓存的数据结点上增加一个计数器和缓冲队列进行处理，较为简单，在这里我们将不详细描述这部分的工作。

2.4 查询结点的 O-Spec 和 P-Spec

根据上面对于两种有序关系的分析，我们在查询树上为有序关系相关的查询结点引入 O-Spec(顺序标记)和 P-Spec(位置标记)，通过这些标记的组合我们可以将一个有序 XPath 查询的所有的有序要求反映到查询树的结点，这样可以根据 2.2，2.3 节的描述通过在处理结点开始或结束事件时，作相应的操作来支持有序 *XPath* 查询。

O-Spec(*Order-Specification* 顺序标记)用于标记先后顺序关系要求涉及的结点及其相互的关系。O-Spec 由一个三元组{*Type*， *Relation*， *Target*}构成。根据先后顺序关系类型的不同(*Following*，F-S)，我们分为两种情况来考虑如何给结点设置合适的 O-Spec 标签：

1. 对查询中的一个先后顺序关系 F-S，为 F-S 的上下文结点 $Q_{context}$ 设置 O-Spec {FS，ls，-}，为 F-S 的目标结点 Q_{target} 设置 O-Spec {FS，rs，-}。如果 F-S 的上下文结点和其父查询结点 Q_{parent} 之间是 P-C 或者 A-D 关系，那么我们在结点 Q_{parent} 上设置 O-Spec {FS，p，$Q_{context}$}
2. 对查询中的一个先后顺序关系 *Following*，为 *Following* 关系的上下文结点 $Q_{context}$ 设置 O-Spec {F，1，-}，为 *Following* 关系的目标结点 Q_{target} 设置 O-Spec {F，r，-}。级联缓存的情况由 *Relation* 为 *c* 来表示。

根据结点上的 O-Spec 标签，我们可以依据 2.2 节所述，对查询中的先后顺序关系进行相应的处理。

P-Spec(*Position-Specification* 位置标记)用于标记位置关系涉及结点之间的相互关系。P-Spec 由一个四元组{*Type*， *Target*， *Op*， *Num*}构成。根据结点上的 P-Spec 标签，我们可以依据 2.3 节所述，结合计数器和缓存队列，对查询中的 Position 进行相应的处理。

通过将查询中的有序关系转化成查询树上结点的 O-Spec 标记和 P-Spec 标记，我们可以将原来的有序查询问题简化为在遇到数据结点的开始或结束标签时根据结点的标记来做适当的缓存和检查操作。从下节的 OrderedXP 算法可以看出，通过这种方式，我们可以在一遍遍历 XML 数据流的情况下完成有序 *XPath* 查询。

3. 一种有序 XPath 查询算法 OrderedXP

3.1 相关的数据结构

在详细阐述有序 *XPath* 查询处理算法 OrderedXP 之前，下面先介绍算法涉及的数据结构。

我们的基础数据结构同之前的工作[4]类似，对给出的 *XPath* 查询，首先将其转化成一个查询树，根据有序 *XPath* 查询的需要，我们扩展了两部分的数据结构：首先，对于查询结点对应的数据结构，我们增加了相应的 O-Spec 和 P-Spec 标签，如果查询结点 *A* 的 O-Spec 有{*F，1，*-}，{*FS，1s，*-}，我们给其查

询结点 DS 增加一个缓存队列CQ_A；然后，对于每个缓存的数据结点，如查询结点 *A* 有 P-Spec{*Complex，target，op，num*}，给数据流流过时给 *A* 对应的每个数据结点 *a* 的数据结点 DS 增加一个 *target* 的队列Q_{target}。

下面是一个查询的数据结构的例子。对于查询 *A//B[C//Following-Sibling:: D]/E[position() = Last()]*而言，其对应的查询树结构如图 7 所示。

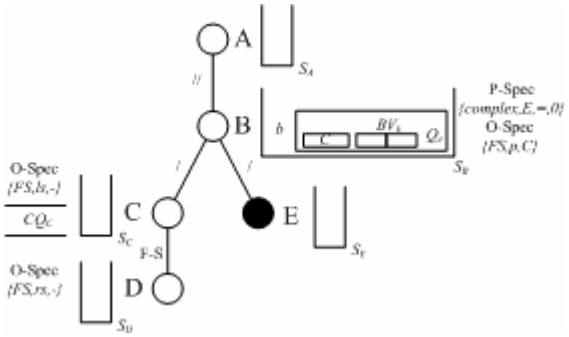


图 7 XPath 查询对应的查询树

3.2 OrderedXP 算法

在描述完 OrderedXP 算法涉及的数据结构之后，下面将会详细描述算法的流程。

StartElement 比较简单,同之前的工作[4]类似，根据结点的标签到查询树上找到数据结点 *n* 对应的查询结点Q_n和其查询的父结点Q_{parent}(2-3 行)。注意这里如果父结点不存在，表示对应查询的根结点，直接将数据结点压栈并返回(4-6 行)。然后判断当前的结点 *n* 是否在查询父结点的栈$Stack_p$或者缓存(如果边关系是 F-S 或者 *Following*)中有对应的满足边关系的数据结点(9-10 行)，如果有的话，将 *n* 放入当前查询结点Q_n的栈中。

Algorithm 1: StartElement(n, QT)

```
Input: n is the label of the element
Input: QT is the Query-Tree structure
1  begin
2      Q_n = QT.findQueryNode(n)
3      Q_parent = QT.getParentNode(Q_n)
4      if Q_parent is NULL then
5          Q_n.getStack().push(n)
6          RETURN
7      Edge = Q_n.getEdge()
8      Stack_p = Q_parent.getStack()
9      if Stack_p.existValidParent(n, Edge) then
10         Q_n.getStack().push(n)
11 end
```

算法 1

EndElement 由下面的算法 2 可以看出分为 6 个步骤：1.检查查询结点的 P-Spec，如果当前结点是某个 F-S 关系上下文结点的父结点，检查该 F-S 关系上下文结点的缓存，删除所有缓存的结点中该 F-S 关系不满足的结点；2.如果当前结点有 P-Spec 标签，检查当前结点的关于孩子(/)或后代(//)子查询结点的 *Counter* 和缓存队列，如果有子查询结点的 Position 谓词不满足条件，当前结点必然没有子查询匹配，做适当清理后返回；3.如果有子查询结点不满足，且该子查询结点不造成当前结点级联缓存，则肯定当前结点不含有子查询匹配，做适当的清理然后返回；4.如果结点的 O-Spec 标签含有{*F, 1, -*}或者是{*FS, 1s, -*}，将结点放入缓存 *CQ* 中并返回；5.如果结点的 O-Spec 有标签{*F, c, target*}，则查看是否已经有合适的 *target* 结点流过，如果没有，将结点放入缓存并返回；6.表示当前结点有一个子查询匹配，通知父查询结点检查缓存结点的相应信息。

Algorithm 2: EndElement(n, QT)

```
Input: n is the label of the element
Input: QT is the Query-Tree structure
1  begin
2      Q_n = QT.findQueryNode(n)
3      item = Q_n.getStack.pop()
4      /*STEP 1*/
5      if SOME Q_n.P-Spec LIKE {FS,p,*} then
6          ClearFSCache(QT, item)
7      /*STEP 2*/
8      if Q_n.P-Spec NOT NULL then
9          if ¬ StructurePosPredicateSatisfy(Q_n, item) then
10             Clear and Return
11     /*STEP 3*/
12     BV_stru = item.getStructureBV()
13     if ¬ AllSatisfy(BV_stru) then
14         Clear and Return
15     /*STEP 4*/
16     if SOME Q_n.O-Spec LIKE {F,l,-} OR {FS,ls,-} then
17         Cache item into Q_n.CQ and Return
18     /*STEP 5*/
19     if SOME Q_n.O-Spec LIKE {F,c,target} then
20         if ¬ has match target item then
21             Cache item into Q_n.CQ and Return
22     /*STEP 6 SUCCESSFUL MATCH*/
23     Q_parent = QT.getParentNode(Q_n)
24     NotifyParentQueryNode(Q_parent,item)
25 end
```

算法 2

通知父查询结点的函数 *NotifyParentQuery Node* 是系统一个比较重要的操作，主要功能是在当前查询结点找到一个子查询匹配之后通知父查询结点，修改相应的状态向量 *BV*，缓存结点的状态等等。

从第 2 节的讨论可以看出，我们的方法可以在一遍扫描数据的情况下，完整地支持有序 *XPath* 查询。

4. 实验

为了验证本文提出的算法的性能和相对于之前工作效率的提升，实验我们将分为两部分：一部分是 OrderedXP 算法和之前工作(TwigM[4])的比较(这里

我们在原 TwigM 的算法基础上实现了 TwigMO 算法用于有序要求的处理，根据原文献[4]的描述，我们采用了先查询分解最后连接过滤的方式)；另一部分是算法在数据规模增长的情况下性能的分析。

实验的环境配置：CPU 为 P4 2.0G，内存 768M，80G ATA 硬盘，操作系统是 Windows XP SP2，SAX 分析器使用了 Apache Xerces C++ Parser (Version 2.7.0)[9]。

这里我们使用了通用的XMark数据集作为我们的实验数据集。在数据集上用到的查询Q_1是只含有顺序关系的查询，Q_2是只含有位置谓词的查询，Q_3是既含有顺序关系又含有位置谓词的查询。

Q_1	//person/name[/fs::phone]
Q_2	//item/incategory[position()=last()-3]
Q_3	//item/location[/fs::incategory[2]]

表 1

表 2 是查询Q_1，Q_2，Q_3在XMark3 上的查询实验数据。可以看到，OrderedXP 相对于之前的方法(TwigMO)，大大减少了中间缓存结点的数目，这是因为我们在数据流经过的同时检查缓存数据结点的有效性，将所有缓存的无用中间结点清除，避免了最后的连接操作。同时也可以看到OrderedXP相对于之前的方法也有较好的时间性能表现。

XMark3	算法	缓存结点数目	结果数目	时间 ms
Q_1	OrderedXP	2	6368	41958
	TwigMO	19118	6368	43771
Q_2	OrderedXP	9	4545	41110
	TwigMO	40925	4545	41400
Q_3	OrderedXP	3	9235	41520
	TwigMO	52800	9235	43100

表 2

图 8 是OrderedXP算法在XMark1(5M)到XMark5(200M)上执行同时含有有序要求和位置谓词查询Q_3的查询时间(ms)分析。从图 8 可以看出，OrderedXP 算法的耗时随XML数据流大小增长线型增加，容易得到我们提出的OrderedXP算法有较好的扩展性。

5. 总结和未来工作

本文提出了在 XML 数据流上做有序 *XPath* 查询的问题，详细地分析了各种有序关系在数据流情况下处理可能的最优策略。根据这些分析将有序 *XPath* 查询中的有序要求转化成了查询树上结点上的 O-Spec 和 P-Spec 标签，结合这些标签，我们提出了 OrderedXP 算法，保证在一遍扫描 XML 数据流的基础上就给出查询所有的解。在今后的工作中我们将考虑怎样结合模式信息来对 XML 数据流上的 XPath 查询做各种优化。

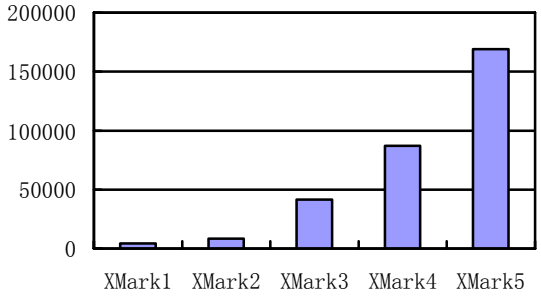


图 8

参考文献

[1] http://www.w3.org/TR/REC-xml/, 2006-4-10

[2] http://www.w3.org/TR/xpath, 2006-4-10

[3] http://www.w3.org/TR/xquery, 2006-4-10

[4] Y. Chen, S.b. Davidson, Y. Zheng. An Efficient XPath Query Processor for XML Streams. In: Proceedings of ICDE. Atlanta:IEEE Press, 2006. 79-79

[5] V. Josifovski, M. Fontou, A. Barta. Querying XML streams. In VLDB Journal, 2005, 14(2):197-210

[6] C. Barton, P. Charles, M. Fontoura, V. Josifovski. Streaming XPath Processing with Forward and Backward Axes. In: Proceedings of ICDE. Bangalore:IEEE Press, 2003. 455-466

[7] A.K. Gupta, D. Suciu. Stream Processing of XPath Queries with Predicates. In: Proceedings of SIGMOD. San Diego:ACM Press, 2003. 419-430

[8] A. Demers, J. Gehrke, M. Hong, M. Riedewald, W. White. Towards Expressive Publish/Subscribe Systems. In: number 3896 in Lecture Notes in Computer Science. Munich: Springer-Verlag, 2006. 627-644

[9] http://xerces.apache.org, 2006-4-10

谢敏，男，1983 年生，硕士研究生，研究方向：XML 数据库。

王小锋，女，1980 年生，硕士研究生，研究领域：XML 数据库。

张新，男，1983 年生，硕士研究生，研究领域：XML 数据库。

孟小峰，男，1964 年生，教授（博导），研究领域：Web数据管理，XML数据库，移动数据管理。

周军锋，男，1977 年生，博士研究生，研究方向：XML 数据库。

本体数据管理

(Ontology Data Management)

HStar - a Semantic Repository for Large Scale OWLDocuments

Yan Chen, Jianbo Ou, Yu Jiang, and Xiaofeng Meng

In Proceedings of the First Asian Semantic Web Conference (ASWC2006), page 415-428, Beijing, China, September 3-7, 2006. Lecture Notes in Computer Science 4185, Springer

Abox Inference for Large Scale OWL-Lite Data

Xiaofeng Wang, Jianbo Ou, Xiaofeng Meng, Yan Chen

In Proceedings of The 2th International Conference on Semantics, Knowledge, and Grids(SKG2006), Guilin, China, Oct. 31 - Nov. 3, 2006

In Proceedings of the First Asian Semantic Web Conference (ASWC2006),
page 415-428, Beijing, China, September 3-7, 2006.

HStar - a Semantic Repository for Large Scale OWL Documents

Yan Chen, Jianbo Ou, Yu Jiang, and Xiaofeng Meng

{chenyan8, oujianbo, jiangyu, xfmeng}@ruc.edu.cn
School of Information, Renmin Univ. of China, China, 100872

Abstract. HStar is implemented to support large scale OWL documents management. Physical storage model is designed on file system based on semantic model of OWL data. Inference and query are implemented on such physical storage model. Now HStar supports characters of OWL Lite and we try to adopt strategy of partial materializing inference data, which is different from most of existing semantic repository systems. In this paper we first give the data model which HStar supports, then give an analysis of our inference strategy; storage model and query process are discussed in detail; experiments for comparing HStar and related systems are given at last.

1 Introduction

RDF(S) standard is firstly proposed by W3C to support research and application of semantic web. It can be used to describe Ontology and metadata with very limited express ability. To support more complicated application of semantic web, OWL standard, which is built on RDF(S), is brought forward. OWL imports more vocabu-laries and rules and is divided into three sub languages: OWL-Lite, OWL-DL and OWL-Full based on the express ability. Semantic web needs high performance se-mantic repository for OWL documents. Now there are many prototype systems, most of which depend on relation database and are designed for RDF(S) documents. From RDF(S) to OWL, more semantic rules make performance of these systems depraved dramatically. This can be proved by our experiment of Sesame [1] using database storage model. Relation database has a single storage model, which cannot satisfy complex data model of OWL data, e.g. the hierarchy relation in OWL data cannot be represented by relation table directly. Relation database can only use logical pointer, not physical pointer, to link different entity in OWL data. Most systems completely materialize inference data to reduce join operation of logical pointers. For such method, more complicated storage strategy is needed to support update operation and this will affect system performance seriously, e.g. Sesame [1] constructs large de-pendent relations among entities of OWL data after loading operation and this is a waste of time. Completely materializing inference data is not fit for large scale OWL documents, because large redundancy data will be produced and this will also affect system performance, especially for loading operation. This has been proved by our experiment discussed in section 6. HStar designs physical storage model, which is independent of relation database and based on characters of OWL Lite data. Most of inference is processed during query processing

time to avoid storing large scale inference data. Our aim is to improve performance of semantic repository and provide the possibility of managing large scale semantic data.

2 Relate Work

Along with more and more popular research of semantic web, many semantic repositories have been developed. All of them can be divided into three categories based on the persistent strategy they use: RDB (Relational Database)-based, File system-based and Memory-based. Because RDB has been fully studied these years, RDB-based systems are in the majority, like Sesame [1], DLDB-OWL [6], RStar [5] and so on. Sesame provides a general storage interface and implements storage method on MySQL, Oracle and so on. File system-based and memory-based storage methods have also been implemented. Sesame provides two logical storage models: RDF schema and RDFS schema. No inference is supported for RDF schema. For RDFS schema, user can use default inference function defined by Sesame, but this is limited to inference rules defined in RDFS. Moreover, user can also use self-defined inference rules, which makes Sesame have good extensibility. From RDF(S) to OWL, only the self-defined inference rules change. But from experiment, we can observe that large number of rules is needed to express complete OWL semantics and when loading data, performance is very bad for doing complete inference based on such rules. Therefore, it can't be used to manage large scale OWL data. DLDB-OWL uses MS Access as its persistent platform and uses inference engine FaCT. It declares high performance for large scale OWL data, but has limited inference ability. From ex-periment we can observe that DLDB-OWL cannot get any answers for some queries. OWLim [3] is a typical memory-based system. It supports more semantic rules than any other systems. OWLim uses Sesame's general storage interface and it has higher performance than Sesame's own memory-based storage module. To support persis-tent storage of semantic data, OWLim uses a simple file format, named "N-triples" and provides backup function. But when do query and inference processing, all data will be read from hard disk into memory. From experiment, we can observe that OWLim cannot handle OWL documents which size is larger than 100MB on general computer hardware. Because OWLim supports most of the semantic rules in OWL Lite, we use it as benchmark of query completeness in our experiment. To support large scale semantic data management, HStar is built on file system and do query and inference processing on physical storage model. Very small part of inference data is materialized and almost the same semantic rules are supported as OWLim. Only one query has less answers than OWLim when do test queries of Lehigh University Benchmark.

3 Data Model

To give better description of HStar's functions, we formalized data model of OWL supported by HStar. This data model has summarized most of the characters of OWL Lite. Our storage, inference and query processing strategies are all based on the data model.

D: all data in OWL document as format<subject property object>

$L = \{C, P, I, R_C, R_P, R_{CP}, R_I, R_{CI}, T_P\}$;

$C = \{URI_i | \exists < URI_i$ rdf:type owl:Class $> \in D\}$;

$P = \{URI_i | \exists < URI_i$ rdf:type owl:ObjectProperty $> \vee$
$\exists < URI_i$ rdf:type owl:DatatypeProperty $> \in D\}$;

$I = \{URI_i | URI_i \notin C$ and $URI_i \notin P\}$;

$R_C = \{C_i \prec C_j | C_i, C_j \in C \wedge \exists < C_j$ rdfs:subClassOf $C_i > \in D\} \cup$
$\{C_i \equiv C_j | \exists < C_i$ owl:equivalentClass $C_j > \in D\}$;

$R_P = \{P_i \prec P_j | P_i, P_j \in P \wedge < P_j$ rdfs:subPropertyOf $P_i > \in D\} \cup$
$\{P_i \equiv P_j | \exists < P_i$ owl:equivalentProperty $P_j > \in D\} \cup$
$\{P_i \leftrightarrow P_j | \exists < P_i$ owl:inverseOf $P_j > \in D\}$;

$R_{CP} = \{[P_i, C_j] | \exists < P_i$ rdfs:domain $C_j > \in D \vee \exists < P_i$ rdfs:range $C_j > \in D\}$;

$R_I = \{[URI_i, URI_j] | \exists < URI_i \, P_x \, URI_j >, P_x \in P \in D\} \cup$
$\{URI_i \equiv URI_j | \exists < URI_i$ owl:sameAs $URI_j > \in D\}$;

$R_{CI} = \{[URI_i, C_j] | \exists < URI_i$ rdf:type $C_j > \in D\}$;

$T_P = P_T \cup P_S \cup P_F \cup P_{IF}$;

$P_T = \{P_i | P_i \in P \wedge < P_i$ rdf:type owl:TransitiveProperty $> \in D\}$;

$P_S = \{P_i | P_i \in P \wedge < P_i$ rdf:type owl:SymmetricProperty $> \in D\}$;

$P_F = \{P_i | P_i \in P \wedge < P_i$ rdf:type owl:FunctionalProperty $> \in D\}$;

$P_{IF} = \{P_i | P_i \in P \wedge < P_i$ rdf:type owl:InverseFunctionalProperty $> \in D\}$;

OWL data has been divided into three categories by this data model: one consists of C, P, I, which respectively represent OWL Class, OWL Property and Individual Resource; one consists of R_C, R_P, R_I, R_{CP}, R_{CI}, which respectively represent relation of elements in C, relation of elements in P, relation of elements in I, relation between elements in C and elements in P, relation between elements in C and elements in I; the last one is T_P, which represents characters defined on OWL Property, including transitive P_T, symmetric P_S, functional P_F and inverse functional P_{IF}. C, P, R_C, R_P, R_{CP} and T_P are used to define Ontology and they are always stable. Most of OWL data focus on R_I, which use Ontology to describe type and relation information of elements in I. Completeness of inference includes two aspects: one is to get complete relation of R_C and R_P, the other is to get complete relation of R_I and R_{CI}. The former represents complete ontology and the latter represents complete ontology instances.

4 Analysis of Inference Completeness

We mentioned above that inference completeness is to get complete R_C, R_P, R_I and R_{CI}. Below we give a detailed discussion for them respectively:

4.1 Completeness Analysis of R_C, R_P

Elements in R_C have two relations: $C_i \prec C_j$ represents inheritance; $C_i \equiv C_j$ represents equivalence. Inheritance is transitive. Equivalence is transitive and symmetric. Because equivalence affects inheritance, complete equivalence relation should be computed first. Complete R_C should satisfy:

(a) $\forall C_i \in C$, can get all $\{C_j | C_j \in C \wedge \exists$ (explicit or implicit) $C_i \equiv C_j\}$;
(b) $\forall C_i \in C$, can get all $\{C_j | C_j \in C \wedge \exists$ (explicit or implicit) $C_j \prec C_i\}$;
(c) $\forall C_i \in C$, can get all $\{C_j | C_j \in C \wedge \exists$ (explicit or implicit) $C_i \prec C_j\}$;

There are three methods to guarantee requirement (a): The first is to store all explicit $C_i \equiv C_j$ and get all relevant $\{C_i \equiv C_j\}$ to construct equivalent set when query; The second is to store all explicit and implicit $C_i \equiv C_j$. There will be no implicit data left and equivalent set does not need to be built. The third is to store equivalent set directly on hard disk. The second method uses redundancy data to improve search performance but adds maintenance cost. The third method not only avoids redundancy data, but also can get equivalent set directly. It is suitable for managing large equivalence relation. In general, equivalence relation in OWL is quite few. So HStar adopts the first method.

For inheritance relation $C_i \prec C_j$, transitive character makes $C_i \prec C_j \prec C_k \Rightarrow C_i \prec C_k$. Requirement (b) and (c) are all related with it. There are also two methods for these two requirements: One is for every transitive chain, compute all implicit inheritance relation and put them into storage system. E.g. if use $C_i \leftarrow C_j$ represents $C_i \prec C_j$ and suppose there are inheritance relations in fig.1.

There are four transitive chains in fig.1: $C_i \prec C_j \prec C_l$, $C_i \prec C_j \prec C_m$, $C_i \prec C_k \prec C_m$, $C_i \prec C_k \prec C_n$. We can compute three implicit inheritance relations from these chains: $C_i \prec C_l$, $C_i \prec C_m$, $C_i \prec C_n$. For large inheritance relation, such method will produce too much redundancy data. Computation complexity is $O(n^2)$ (n is the number of elements in inheritance relations) and it will be a hard work to maintain the redundancy data. The other method is using tree storage structure to represent inheritance relations.

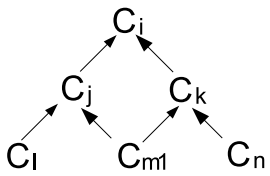


Fig. 1. An example of inheritance relation

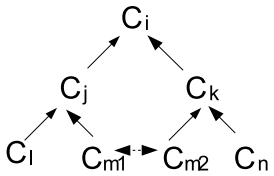


Fig. 2. Tree structure for Fig.1

Node C_m in fig.1 splits into nodes C_{m1} and C_{m2}. Node C_{m1} copies all information of Cm and node C_{m2} is a reference of C_{m1}. If it is required to find all C_x, which satisfy $C_x \prec C_m$, first locate node C_{m1} in Fig.2, get all ancestors of C_{m1}, i.e. $\{C_i, C_j\}$, and then get all ancestors of C_{m2}, i.e. $\{C_i, C_k\}$, union the two result sets and remove the duplicate, we can get $\{C_i, C_j, C_k\}$. Such method avoids computing redundancy data, but needs native tree storage on hard disk. HStar adopts this method.

Inheritance relation and equivalence relation defined in R_P are same as those in R_C in essence. HStar uses same method to deal with them. Besides these, there is another relation defined, i.e. $\{P_i \leftrightarrow P_j | < P_i$ owl:inverseOf $P_j > \in D\}$. This relation will only

bring implicit data in R_l according to OWL semantic definition. So we will discuss it later.

4.2 Completeness Analysis of R_I

From definition of R_I, we can see there are two sub-relations in it. We give their definitions below:

$$R_{I1} = \{[URI_i, URI_j] | \exists < URI_i\, P_x\, URI_j > \in D\}$$
$$R_{I2} = \{URI_i \equiv URI_j | \exists < URIi \text{ owl:sameAs } URI_j > \in D\}$$

R_{I2} defines equivalence relation which affects completeness of R_{I1}, just as equivalence relation in R_C does. Besides that user can directly define R_{I2}, property that is element of P_F or P_{IF} can also infer R_{I2} relation. The inference rules are defined below:

$$< URI_i P_x URI_k > \wedge < URI_j P_x URI_k > \wedge P_x \in P_{IF} \Rightarrow URI_i \equiv URI_j$$
$$< URI_k P_x URI_i > \wedge < URI_k P_x URI_j > \wedge P_x \in P_F \Rightarrow URI_i \equiv URI_j$$

So complete R_{I2} needs to apply rules above to every element in P_F and P_{IF}. And the process needs to do iteratively. E.g. suppose $P_x \in P_F$, a, b, c, d respectively represent an URI and $\exists\{< a\, P_x\, b >, < a\, P_x\, c >, < b\, P_x\, d >, < c\, P_x\, a >\} \in D$. According to rules above, we can get $< a\, P_x\, b > \wedge < a\, P_x\, c > \Rightarrow b \equiv c$. But the process cannot terminate now, because $b \equiv c$ also affects existed data. With this consideration, we can get $< b\, P_x\, d > \wedge < c\, P_x\, a > \Rightarrow d \equiv a$. The process needs to do iteratively until no new equivalence relations are generated. Storage method of R_{I2} is the same as equivalence relation of R_C.

Completeness of R_{I1} is mainly determined by characteristic of P_x. If $P_x \in P_T$ or $P_x \in P_S$ or P_x has inheritance or equivalence relation in R_p, it will bring implicit data into R_{I1}. If there exist P_x satisfying $P_x \in P_T \wedge P_x \in P_S$, we treat such P_x as an equivalent relation.

There is a condition that is not defined definitely in OWL semantic. If $\{P_x \prec P_y \in R_p$ or $P_y \prec P_x \in R_p\}$ and $\{P_x \in P_T$ or $P_x \in P_S$ or $P_x \in P_F$ or $P_x \in P_{IF}\}$, whether $P_y \in P_T$ or $P_y \in P_S$ or $P_y \in P_F$ or $P_y \in P_{IF}$ is not defined. So HStar does not consider the interaction effect between R_p and T_p.

Under the precondition above, completeness of R_{I1} can be considered from P_T, P_S and R_P respectively:

1. P_T defines transitive character which is equivalent to inheritance relation of R_C in essence. HStar adopts the same method to deal with P_T.
2. PS defines symmetric relation and related rule in OWL is $P_x \in P_S \wedge < URI_i\, P_x\, URI_j > \Rightarrow < URI_j\, P_x\, URI_i >$. Two methods can guarantee the completeness of P_s: One is to store all implicit data brought by P_s. E.g. when user inserts $< URI_i\, P_x\, URI_j >$, both $< URI_i\, P_x\, URI_j >$ and $< URI_j\, P_x\, URI_i >$ will be stored. There is no need to consider P_S character when

query with this method. But the volume of such data will be doubled. The other method only stores the explicit data and use query rewriting to satisfy P_S requirement. E.g. suppose $P_x \in P_S$, query $< URI_i\, P_x\, ? >$ should be rewritten as $< URI_i\, P_x\, ? >$ and $<?\, P_x\, URI_i >$. When data volume that has P_S character become larger, performance of the second method will be better than the first one.

3. Rule $P_i \prec P_j \wedge < URI_x\, P_j\, URI_y > \Rightarrow < URI_x \quad P_i \quad URI_y >$makes R_p may bring implicit data. Considering query $< URI_x\, P_i\, ? >$, if there is only $< URI_x\, P_j\, URI_y >$ in R_I, no result will be returned if don't use rule above. As we have mentioned in section 4.1, relation $P_i \prec P_j$ in R_p is stored as a tree structure in HStar. For any P_i in this structure, all its descendants can be accessed directly. So when processing query $< URI_x\, P_i\, ? >$, HStar will search all data in D which have P_i or P_i's descendants as their Property. Special storage design in HStar makes such operation can be processed efficiently. We will give detailed analysis in section 5. Relation $\{Pi \leftrightarrow P_j | < P_i$ owl:inverseOf $P_j > \in D\}$, which is defined in R_p, has rule $< P_i$ owl:inverseOf $P_j > \wedge < URI_x\, P_i\, URI_y > \Rightarrow < URI_y\, P_j\, URI_x >$ defined in OWL semantic. Like P_s, completely materializing implicit data brought by this rule will double such data volume. Query rewriting can also be used here and its performance will be better when data volume is larger.

4.3 Completeness Analysis of R_{CI}

R_{CI} describes type information of URI and it is the most complex part of OWL data. Both R_C and R_{CP} affect completeness of R_{CI} and the related rules are list below:

1. $< URI_x$ rdf:type $C_j > \wedge C_i \prec C_j \Rightarrow < URI_x$ rdf:type $C_i >$
2. $< P_x$ rdfs:domain $C_y > \wedge < URI_i P_x URI_j > \Rightarrow < URI_i$ rdf:type $C_y >$
3. $< P_x$ rdfs:range $C_y > \wedge < URI_i P_x URI_j > \Rightarrow < URI_j$ rdf:type $C_y >$

As we have mentioned in section 4.1, relation $C_i \prec C_j$ is stored as a tree structure in HStar. When processing query $< URI_x$ rdf:type $? >$, we first get C_i if there is explicit data $< URI_x$ rdf:type $C_i >$ in D; then get all ancestors of C_i and return them as the result. For rules 2 and 3, if we don't get complete R_{CI} relation when loading OWL documents, the whole data space search will be required when query processing. HStar materializes all implicit data brought by rules 2 and 3.

From discussion above, we can observe that HStar only materializes implicit data brought by P_F and P_{IF}, implicit data in R_{CI} brought by Property's domain and range.

5 Storage Design

From the third section, we can see that the main part of OWL data is five kinds of relations, R_C, R_P, R_{CP}, R_I and R_{CI}. How to organize these relations on hard disk is the task of storage design. Considering the characteristics of both OWL data and queries against it, we designed a special storage model for OWL data, which is built on file system rather than RDB, ORDB and etc. In the rest of this section, we will first describe the inner identifier of entities, and then present the storage method of different relations.

5.1 Inner Identifier for Entities: OID

In OWL data, entities are identified by URI, which is usually a long string. Storing original URI takes considerable space; therefore we use inner identifier OID to replace URI in storage. OID consists of two members: *id*, which occupies four bytes, *flag*, which occupies one byte, indicates whether entity has equivalent resources. Thus an OID totally occupies five bytes, which is much smaller than a URI. The relationship between OID and URI is one to one and is saved in two global hash tables.

5.2 Storage of R_C and R_{CI}

Inheritance relation in R_C is stored in tree structure. We named it C-Tree. E.g. the relation in fig.2 is stored as C-Tree structure in fig.3. Each tree node keeps addresses (represented by page number and offset in physical page) of its first child, parent, left and right siblings. It is easy to access the ancestors and descendents of a tree node by these addresses.

Non-tree nodes in inheritance relation graph split into multiple copies. One is primary (P-Node), and the others are references. E.g. node C_m has been divided into C_{m1}, C_{m2}. They are linked in the Same Entity List (SE-List), with the primary one as head. Only primary node stores the address of child and Individual List.

Locating arbitrary C_x in C-Tree structure is an indispensable operation for inheritance relation query. C-index is built to improve the performance of this operation, which is a B+ tree structure and uses identifiers of P-Nodes as keys. As showed in fig.3, C-index record addresses of nodes in C-Tree. Using C-index and SE-List, all the nodes responding to C_x in C-Tree can be accessed quickly.

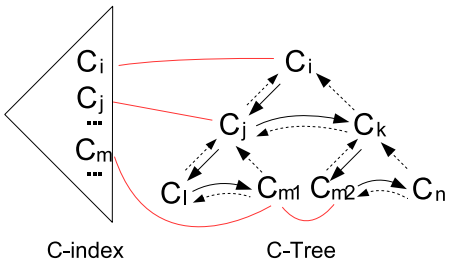


Fig. 3. C-Tree and C-Index for R_C Storage

Equivalence relation in R_C is stored in B+ tree. E.g. suppose C_i is equivalent to C_j and the id of C_i's OID is smaller than the id of C_j's OID, and then take C_i as key and C_j as value. Only explicit equivalence relations are stored. Equivalence sets are built in memory to facilitate query processing, each set corresponding to a memory list. Updating equivalence relation needs to maintain both B+ tree and lists in memory.

Individuals related to the same C_x are stored in one Individual List (I-List), whose start address is saved in C_x's P-Node of C-Tree. E.g. in fig.4, individuals I_i and I_j have

type of C_m. They are stored in an I-List, with the start address kept in node C_{m1} of C-Tree. This structure is to facilitate querying individuals of given Class, which is the most frequent query about R_{CI}.

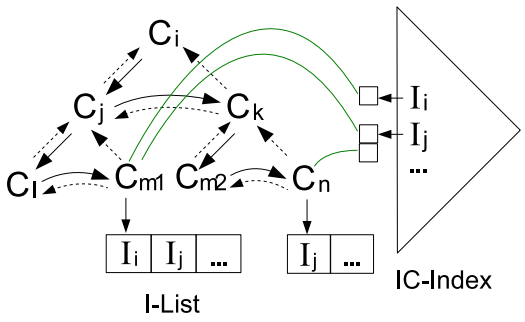


Fig. 4. I-List and IC-index for R_{CI} Storage

Queries for type of given individual are less frequent but necessary. IC-index is built to facilitate these queries. It is a B+ tree index, which uses OID of individual as key. Leaf node contains all the Classes to which the individual belongs. E.g. IC-index in fig.4 records that individual I_i belongs to C_m and I_j belongs to C_m and C_n.

Only explicit R_{CI} relations are stored in I-List and IC-index. To guarantee the inference completeness, we need to combine I-List and IC-index with C-Tree structure. That is the reason why we store addresses of I-Lists in P-Nodes of C-Tree. E.g. in fig.4, to find individuals of C_i, I-List of both C_i and its descendants need to be returned. Here, I_i, I_j is the result. To query type of I_i, we find C_{m1} through IC-index, then C_m and its ancestors are returned. Here, C_i, C_j, C_k, C_m is the result.

5.3 Storage of R_p, R_{CP}, R_I and T_p

Inheritance relation and equivalence relation in R_p are stored in the same way as those relations in R_C. P-Tree, and P-index are built as C-Tree and C-index. Inverse relations in R_p are stored as data members of P-Nodes in P-Tree (Property Tree); for $P_i \leftrightarrow P_j$, store P_j in P_i's P-Node, and store P_i in P_j's P-Node.

T_p and R_{CP} are also stored as data members of P-Nodes. T_p is represented by a byte and the first four bits are used to indicate whether P_x has transitive, symmetric, function and inverse-function characters. R_{CP} is stored as two arrays, which store entities having $rdfs:domain$ or $rdfs:range$ relation with P_x.

Equivalence relation in R_I (namely R_{I2} in section 4.2) is stored as same as that relation in R_C. Individual pairs of R_{I1}, which are related to same transitive P_x, are stored in one Individual Tree (I-Tree). I-Tree adopts the same structure as C-Tree, including I-index and SE-List structures. E.g. in fig.5, P_n is transitive. Pairs (I_k, I_m), (I_k, I_n) relate to P_n, and are stored in its I-Tree. Individual pairs related to same non-transitive P_x are stored in two Individual B+ trees (IB-Tree). One is SB-Tree (S-Key B+ tree),

taking subject as key. The other is OB-Tree (O-Key B+ tree), taking object as key. E.g. in fig.5, P_m is not transitive. Pair (I_i, I_j) relates to P_m and is stored in its IB-Trees. SB-Tree takes Ii as key and OB-Tree takes I_j as key. The root addresses of I-Tree and IB-Trees are kept in P_x's P-Node.

IP-index is built similarly to IC-index. The difference is that IP-index records how an individual relates to different properties (as subject or object). E.g. the IP-index in fig.5 records that I_i relates to P_m as subject (represented by solid lines), I_j relates to P_m as object (represented by dashed line), and so on.

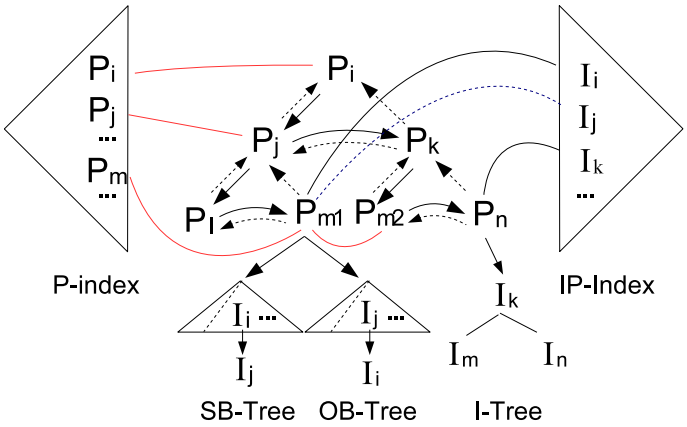


Fig. 5. Storage of R_p and R_{I1}

Queries against R_{I1} can be processed in a similar way with R_{CI}. The difference is that queries against R_{I1} may need further search in P_x's I-Tree or IB-Tree. For transitive P_x, search in I-Tree in the same way as in C-Tree. For non-transitive P_x, search in SB-Tree with given subject, or in OB-Tree with given object.

6 Query Processing

HStar supports queries in SPARQL language, which is proposed by W3C and likely to be the standard query language for OWL. When we mention "OWL query" later, it means SPARQL query. Here we give a query example, which queries all the facts related to "students take courses".

```
PREFIX p:<http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl>
SELECT ?x ?y
WHERE {
       ?y rdf:type p:Course.
       ?x rdf:type p:Student.
       ?x p:takesCourse ?y
}
```

We call triple with variable(s) "Query Triple", QT for short. E.g. "?y rdf:type p:Course" in the query example is a QT, in which "?y" represents variable to be evaluated during

query processing. From query example above, we can observe that QT is the basic unit in OWL query. Query processor first evaluate all QTs to get middle results and then choose some order to join all middle results to get final results. Different join orders produce different sizes of middle results and this affects query performance. Such problem has also been encountered in SQL query processing. For OWL data is different from data in relational database, new solution needs to be proposed. In the next section, we give our intuition for this problem and describe several possible solutions that can be used for OWL query optimization.

6.1 Query Optimization

1. Remove possible redundant QTs based on Ontology.

R_{CP} is part of Ontology and it defines domain and range of a Property. Rules

$$< P_x rdf:domain C_y > \wedge < URI_i P_x URI_j > \Rightarrow < URI_i \text{ rdf:type } C_y > \text{ and}$$
$$< P_x rdf:range C_y > \wedge < URI_i P_x URI_j > \Rightarrow < URI_j \text{ rdf:type } C_y >$$

have been mentioned in section 4.3. These rules not only bring implicit data, but also define restrictions. That means if there is $< URI_i P_x URI_j >$ in D and P_x has domain C_m, has range C_n, then URI_i must be an instance of Class C_m and URI_j must be an instance of Class C_n. We can make full use of such restrictions to optimize some type of queries. E.g. suppose there are properties "StudentNumber", "Teach" and two disjoint classes "Student", "Teacher". We know only Class "Student" can have "StudentNumber" and only Class "Teacher" can do "Teach" in real world. These facts will be defined by R_{cp}. Now if user issues query $<?s$ StudentNumber $?n >< ?s$ Teach $?c >$,we can immediately judge that such query has no result because "Student" can not "Teach" and "Teacher" has no "StudentNumber". Another example is that if user issues query $<?s$ rdf:type Student $> <?s$StudentNumber $?n >$, we can remove QT $<?s$ rdf:type Student $>$ because only "Student" has "StudentNumber".

2. Choose Join order based on statistic data.

Choosing join order needs a method to estimate mid-result size of two QTs' join. E.g. query $< s\, p_1\, ?x >, <?x\, p_2\, ?y >, <?y\, p_3\, o >$ contains three QTs. There are two possible join orders: $(< s\, p_1\, ?x >$ join $<?x\, p_2\, ?y >)$ join $<?y\, p_3\, o >$ or $< s\, p_1\, ?x >$ join $(<?x\, p_2\, ?y >$ join $<?y\, p_3\, o >)$. If we can estimate middle results' size of $(< s\, p_1\, ?x >$ join $<?x\, p_2\, ?y >)$ and $(<?x\, p_2\, ?y >$ join $<?y\, p_3\, o >)$, then we can choose the join order which has smaller middle result size. Here we suggest borrowing idea for such problem from relational database. When loading data into HStar, we can compute how many triples there are for every Property, we named this number as N_{tp}; and compute how many different instances there are for every Property's subject and object, we named the two numbers as N_{sp} and N_{op}, then the middle result size of $< s\, p_1\, ?x >$ join $<?x\, p_2\, ?y >$ can be computed by $min\{N_{tp1}/N_{op1}, N_{tp2}/N_{sp2}\}$.

7 Experiment

Experiments in [2] give detailed compare among semantic repositories, DLDB-OWL [6], Sesame-DB [1], Sesame-Memory [1] and OWLJessKB [4]. The experiments test performance of data loading, query processing and query completeness. In our experiment, we do test on systems DLDB-OWL, Sesame-DB, OWLim [2] and HStar. OWLim is a memory-based system, implemented under Sesame general architecture and has better performance on data loading, query processing and query completeness than Sesame's original memory-based system. OWLJessKB [4] is also a memory-based system. [2] points out that it has implemented incorrect inference strategy. So OWLim can be treated as the best memory-based system and we ignore Sesame-Memory and OWLJessKB system in experiment.

Our experiment uses an extension of Lehigh University Benchmark, which has been described in [2]. Four test data sets are generated by tool provided by [2]. They are univer1, univer5, univer10 and univer20. The smallest data set is 8MB including 15 OWL documents. The largest is 218MB including 402 OWL documents. We get "OutOfMemory" error when loading univer10 into OWLim system. Sesame-DB uses user-defined inference rules and costs about 13 hours to load univer5. "OutOfMemory" error occurred when loading univer20 into HStar for a memory-based hash map is used. This will be improved in the next version. DLDB-OWL costs more than 13 hours to load univer10, but it still doesn't finish loading work, which is different from that discussed in [2]. So we just give out the test result for first three data sets.

7.1 Experiment Environment

Hardware: CPU P4.3G, 512MB of RAM, 40GB of hard disk; Software: Windows XP, Java JDK1.5, MySQL4.1.4, MS Access2003, DLDB-OWL(04-03-29 release), Sesame(1.2.2), OWLim(2.8). For all test systems, we set maximum heap size as 256MB.

	Data set	Instance number	Load time(ms)	Repository size(KB)
OWLim	LUBM(1, 0)	103,074	2,985	17,311
Sesame-DB			1,206,141	48,333
HStar			98,641	19,922
DLDB-OWL			183,937	15,876
OWLim	LUBM(5, 0)	645,649	47,578	107,809
Sesame-DB			47,131,655	283,967
HStar			982,875	77,082
DLDB-OWL			994,157	89,156
OWLim	LUBM(10,0)	1,316,322	-	-
Sesame-DB			-	-
HStar			2,135,453	154,656
DLDB-OWL			-	-

Table 1. Description of test data sets and data loading performance

From left of fig.6, we can observe that OWLim has the best data loading performance for the first two data sets. HStar has almost the same performance with DLDB-OWL. Sesame-DB has the worst performance. [2] pointes out that Sesame-DB constructs dependent relation among OWL data elements when loading data. This is very time consumed but is very useful for update performance. DLDB-OWL doesn't consider update problem. HStar just materialize a little part of inference data and it's easy to maintain their relation.

[2] gives 14 query test cases. They are used to test query performance and query completeness. In our experiment, OWLim supports the most semantic rules and we use OWLim query answers as benchmark to evaluate other systems' query completeness.

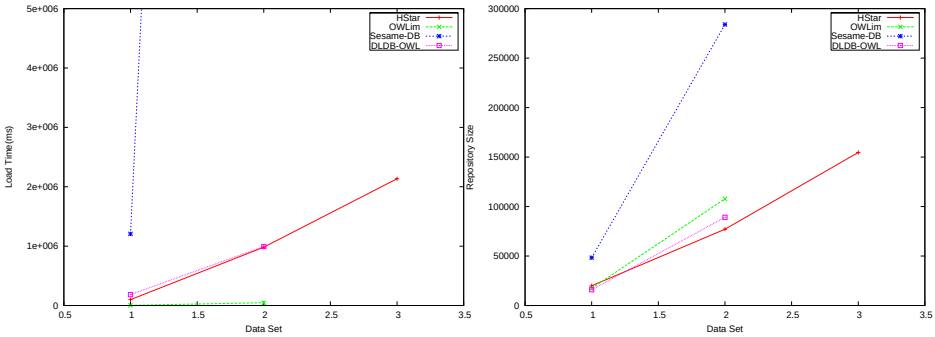


Fig. 6. Data loading performance and repository size

From fig.7, we can observe that HStar has different answers with OWLim only for the 12th query. DLDB-OWL has no answers for the 11, 12, 13th queries. Sesame-DB has incompleteness answers for the 6, 7, 8, 9th queries and has no answers for the 10, 12th queries. We can sort them by answer completeness as below: OWLim $>$ HStar $>$ DLDB-OWL $>$ Sesame-DB.

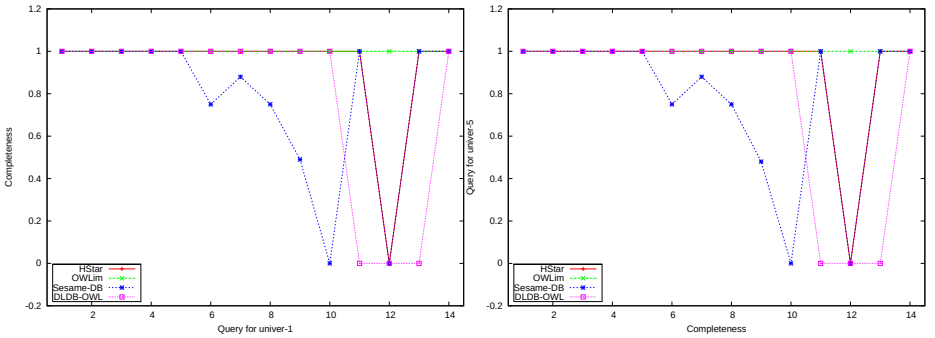


Fig. 7. Query completeness

Fig.8 describes the query response time for 14 queries. To avoid impact of OS buffer, we test 10 times for every query and compute the average time. Only OWLim is memory-based, so it has the best query performance. HStar, DLDB-OWL and Sesame-DB have different query process strategies, so they have owned preponderance for different queries. E.g. HStar has better performance for queries 6, 8, 10, 11, 12, 14 than DLDB-OWL and Sesame-DB, has better performance for query 3 than DLDB-OWL but worse than Sesame-DB, has worse performance for queries 1, 4, 7 than DLDB-OWL and Sesame-DB, has better performance for query 2 than Sesame-DB but worse than DLDB-OWL. For queries 5, 9 and 13, the performance is related with data sets.

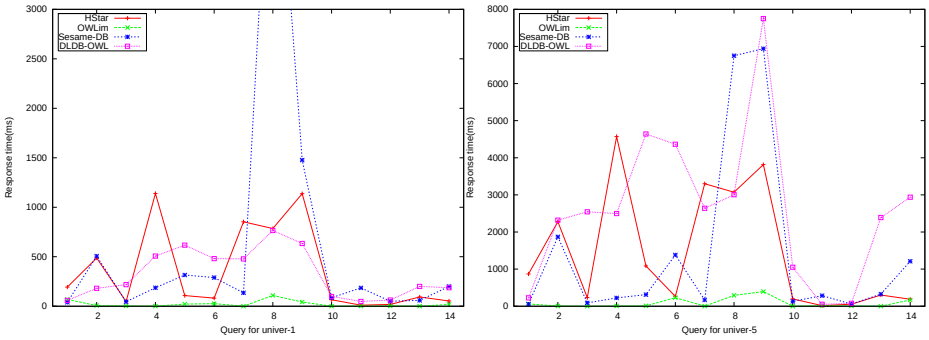


Fig. 8. Query response time

From experiments described above, we can summarize that HStar has an ideal performance for data loading, query processing and provides the highest query completeness among all hard disk based systems.

8 Conclusion

This paper introduced a semantic repository system called HStar, which is based on file system. We first formalized OWL data model supported by HStar and then gave detailed discussion for completeness problem of OWL data, gave detailed discussion of storage design on file system and query processing strategy. At last, we used extensional Lehigh University Benchmark to test HStar and compared it with DLDB-OWL, Sesame-DB, which use relational database, and OWLim, which is memory-based. From experiment, we observed that HStar has an ideal performance for data loading, query processing and provides the highest query completeness among all hard disk based systems. Because HStar has used a memory-based hash map module, ”OutOfMemory” error occurred when loading data set univer20. We plan to design a hard disk based hash structure to replace it in next version of HStar.

9 Acknowledgments

This research was partially supported by the grants from the Natural Science Foundation of China under grant number 60573091, 60273018; the National 973 Basic Research Program of China under Grant No.2003CB317000 and No.2003CB317006; the Key Project of Ministry of Education of China under Grant No.03044 ; Program for New Century Excellent Talents in University(NCET).

References

1. Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In Ian Horrocks and James A. Hendler, editors, *International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2002.
2. Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. An evaluation of knowledge base systems for large owl datasets. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2004.
3. Atanas Kiryakov, Damyan Ognyanov, and Dimitar Manov. Owlim - a pragmatic semantic repository for owl. In Mike Dean, Yuanbo Guo, Woochun Jun, Roland Kaschek, Shonali Krishnaswamy, Zhengxiang Pan, and Quan Z. Sheng, editors, *WISE Workshops*, volume 3807 of *Lecture Notes in Computer Science*, pages 182–192. Springer, 2005.
4. Joseph Kopena and William C. Regli. Damljesskb: A tool for reasoning with the semantic web. *IEEE Intelligent Systems*, 18(3):74–77, 2003.
5. Li Ma, Zhong Su, Yue Pan, Li Zhang, and Tao Liu. Rstar: an rdf storage and query system for enterprise resource management. In David Grossman, Luis Gravano, ChengXiang Zhai, Otthein Herzog, and David A. Evans, editors, *CIKM*, pages 484–491. ACM, 2004.
6. Zhengxiang Pan and Jeff Heflin. Dldb: Extending relational databases to support semantic web queries. In Raphael Volz, Stefan Decker, and Isabel F. Cruz, editors, *PSSS*, volume 89 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

In Proceedings of The 2th International Conference on Semantics, Knowledge, and Grids(SKG2006),
Guilin, China, Oct. 31 - Nov. 3, 2006

Abox Inference for Large Scale OWL-Lite Data*

Xiaofeng Wang, Jianbo Ou, Xiaofeng Meng, Yan Chen
Renmin University of China
{zzuwxf,oujianbo,xfmeng,chenyan8}@ruc.edu.cn

Abstract

Abox inference is an important part in OWL data management. When involving large scale of instance data, it can not be supported by existing inference engines. In this paper, we propose efficient Abox inference algorithms for large scale OWL-Lite data. The algorithms can be divided into two categories: initial inference and incremental inference. Initial inference is used in situation where only raw data exists in storage system, and for this category we propose Rule Static Association Based (RSAB), Rule Dynamic Association Based (RDAB) and Rule Grouped-Sorted Based (RGSB) inference methods. Incremental inference algorithm is used in situation where large volume inference data exists in storage system, and for this category we extend the initial inference algorithm and propose Rule Pattern-Sharing Based(RPSB) method. At last, extensive experiments show that our methods are efficient in practice.

1. Introduction

OWL[1] (Web Ontology Language) is proposed by W3C and it is used to describe Web resource. OWL-Lite is a sublanguage of OWL. It was disigned for easy implementation and to provide users with a functional subset that will get them started in the OWL. Inference for OWL-Lite can be divided into two categories: Tbox and Abox. Tbox inference concerns Ontology and generally can be supported by existing inference engines[6, 7, 12, 5]. Abox inference focuses on large scale instance data, and can not be supported by existing inference engines.

As the research of semantic web is becoming more and more popular,many semantic repositories have been developed,which can be divided into three categories based on the storage strategy they use: RDB (Relational Database)-based[8, 10, 9],File system-based[4] and Memory-based[3].For the RDB has been studied extensively these years, RDB-based systems are in the majority, like Sesame[8], DLDB-OWL[10], RStar[9] and so on. Sesame provides a general storage interface and implements storage method on MySQL, Oracle and so on. Sesame aims at storage for RDF files, and doesn't take the feature of OWL data into consideration.From experiment, we observe that large number of rules are needed to express complete OWL semantics and loading process is also very inefficient for doing complete inference based on such rules. Therefore, it can't be used to manage large scale OWL data. DLDB-OWL uses MS Access as its persistent platform and uses inference engine FaCT[7]. It declares high performance for large scale OWL data, but has limited inference ability. From experiment we observe that DLDB-OWL cannot get any answers for some queries. OWLim[3] is a typical memory-based system. It supports more semantic rules than any other systems. OWLim uses Sesame's general storage interface and has higher performance than Sesame's own memory-based storage module.But when do query and inference processing, all data will be read from hard disk into memory. Because OWLim supports most of the semantic rules in OWL Lite, we use it as benchmark of query completeness in our experiment. To support large scale semantic data management, we develop HStar system, which is built on file system and do query and inference processing on physical storage model. In this paper, we mainly concern the inference engine in HStar system.

This paper discusses Abox inference for large scale OWL-Lite data and divides the algorithms into two categories: initial inference and incremental inference. Initial inference is used in situation where only raw data exists in storage system. Because inference about one single rule has high computation cost, it is important to remove redundant rule inference. From this point, we propose Rule Static Association Based (RSAB), Rule Dynamic Association Based (RDAB) and Rule Grouped-Sorted Based

*Supported by the National Natural Science Foundation of China under Grant Nos.60073014, 60273018; China National Basic Research and Development Program's Semantic Grid Project No. 2003CB317000; the Key Project of Chinese Ministry of Education under Grant No.03044 ;Program for New Century Excellent Talents in University

(RGSB) inference algorithms.Compared to basic inference algorithm, RSAB algorithm removes some redundant rule inference procedures. But there still exist some redundant procedures. RDAB takes data set and rules together into consideration, any redundant procedures can be removed by this algorithm. The whole Rules of OWL-Lite Abox inference has lattice-like relation,based on which we can group rules and sort such groups. Do inferences based on Group-Sorted Rules can make inference more focused and then reduce the number of data and pattern matching.

Incremental inference algorithm is used in situation where large volume inference data existing in storage system.So we propose Rule Pattern Sharing Based(RPSB)increment inference algorithm to reduce number of data and pattern matching. Compared to old data, new data is generally small in scale. So buffering new data in the sharing patterns can avoid redundant inference and reduce number of data and pattern matching.

The contributions of this paper are:

(1) We summarize general Abox inference methods for large scale OWL-Lite data and divide the algorithms into two categories: initial inference and incremental inference.

(2)For initial dataset, we propose Rule Static Association Based (RSAB), Rule Dynamic Association Based (RDAB) and Rule Grouped-Sorted Based (RGSB) inference algorithms besides basic inference algorithm.

(3) We extend initial inference algorithm and design Pattern-Sharing Based incremental inference(PSB) algorithm to handle incremental inference.

(4) We present an extensive performance evaluation of different inference algorithms and analyze the results in detail.

2. Preliminaries

Before introducing any specific inference algorithms, first we will explain some symbols used in this paper. [2]describes the features of OWL-Lite, according to the semantic of which we list rules related to Abox inference shown in table 1.

The basic unit of rule is binary relation related to Ontology data and ternary relation related to Instance which are described by RDF syntax. We call them rule pattern, written by P, whose formal format is: P = P_{onto} |P_{Ins}. P_{Ins} = (S P O), in which S, P and O can be variables. P_{onto}=Rel(Class |Property |Class, Property), in which Rel(Class) describes the relationship between Class, Rel(Property) describes relationship between Property. Rel(Class, Property) describes the relationship between Class and Property. Inference Rule in Abox can be formalized as below:

R = [P_{onto},] P_{Ins-1},..., P_{Ins-n} :- P_{Ins-x}, the part before symbol ':-' is called rule premise, and the part after symbol ':-' is called rule conclusion. If we use { V_1, V_2, ..., V_m }to represent variable sets of P_{Ins-1},...,P_{Ins-n}, then the variables which belong to P_{Ins-x} is in the subset of { V_1, V_2, ..., V_m }.

1	EquivalentClass(u,v),(x type u):-(x type v)
2	SubPropertyOf(u, v),(x u y):-(x v y)
3	FunctionalProperty(p),(u p y),(u p x):-(y sameAs x)
4	InverseFunctionalProperty(p),(x p u),(y p u):-(x sameAs y)
5	TransitiveProperty(p),(x p y),(y p z):-(x p z)
6	SymmetricProperty(p), (x p y) :- (y p x)
7	InverseOf(p, q), (x p y) :- (y q x)
8	EquivalentProperty(p, q), (x p y) :- (x q y)
9	Domain(p, c), (x p y) :- (x type c)
10	Range(p, c), (x p y) :- (y type c)
11	SubClassOf(u, v), (x type u) :- (x type v)
12	(u sameAs v) :- (v sameAs u)
13	(u sameAs v), (u p x) :- (v p x)
14	(u sameAs v), (x p u) :- (x p v)
15	AllValuesFrom(c, p, d), (x p y), (x type c) :- (y type d)

Table 1 inference rule in OWL-Lite

3. Initial Abox Inference Algorithms

To begin, we give the definition of complete inference dataset, which defines complete results that do not omit any inference procedures, then we talk about basic rule inference, at last we will explain rule association based inference algorithms.

3.1. Basic Inference Method

Before introducing any inference methods,first,we want to explain inference complete dataset,for it is critical to the final results.

Definition 1: Complete Inference Dataset *Given the dataset D which is declared by users explicitly, we call D* the complete inference dataset after inference procedure, if and only if it satisfies the following condition: no new data can be generated in D* using any rule.*

According to the above definition, we can get D* from D using the rules shown in table 1 repeatedly. We first discuss inference methods focusing on single rule.

The rule premise consists of several patterns, and the association between patterns is connected by the same variable. For example, the association between patterns (a p b) and (b q c) is connected by variable b. Generally speaking, there are two methods to compute all instance data from the rule premise. One is to use join method and the other is navigation. The former method is to match the pattern with the data, then for each pattern we can get a result set, at last these sets are joined to get the final set. The latter method is to match a certain pattern with the data, then use its result set to query the pattern associated with it, this

procedure continues until all the patterns have been handled. In most cases,the navigation method is better than the join method.When many variables in the pattern are unknown,the join method would probably match all instances in the dataset,which is unaccepted for large scale data.

Based on definition 1, we propose single rule inference algorithm using navigation method below.({I} represents value sets of instance data matching the pattern P_i)

Algorithm 1: SingleRuleReason(R,D)

```
begin
    select rule R whose pattern P_i has known element;
    {I}← Query(P_i);
    repeat
        Select a pattern P_j which has least unknown
        elements associated to P_i;
        Substitute corresponding variables in P_j with
        the value in instance sets I and obtain P_j';
        {J} ← Query(P_j');
        if J= NULL then
            delete corresponding data from {I};
        else
            combine I and J and get {I,J};
    until the rule premise of R has been scanned once
    ;
    Substitute corresponding variable in rule
    conclusion of R with the value in instance set I and
    obtain result set D';
    foreach item e in D' do
        if e doesn't exists in D then
            insert it into D; return TRUE;
    return FALSE;
end
```

Based on Algorithm 1,it is easy to think of a method which do inference straightforward.When new data is generated by using certain rule, in order to assure the completeness of the inference, we need do inference for all rules over again, for newly generated data may match the rule premise, and the rule may lead to new data. When we have no prior association in advance, it is an effective way to do inference iteratively to ensure the completeness of the result.

Algorithm 2: BasicReason(D)

```
begin
    BOOLEAN hasNewData ← FALSE;
    repeat
        foreach R_i in RuleSet do
            hasNewData
            ← SingleRuleReason(R_i, D);
    until hasNewData is FALSE ;
end
```

3.2. Association Based Inference Methods

Algorithm 2 doesn't consider the association of rules, so we may do redundant inference. When a certain rule R generates new data, which triggers other rules {R'}, then R and {R'} have association relationship. If we can get all R and its association rules{R'}, we only need to do inference on {R'} to avoid redundant inference.

Definition 2: Pattern Match *P_{Ins-A}(a p b) match P_{Ins-B}(x q y) if and only if x is a variable or a=x, and q is a variable or p=q, and y is a variable or b=y.*

Above definition ignores the case when pattern (x p u) matches pattern(x type u), for this kind of match is related to instance data, we call it *conditional pattern match*. If the rule conclusion of R_1 conditional matches the rule premise of R_2, we call R_1 conditional association match R_2.

Based on the above discussion, we can improve basic rule inference algorithm.The main idea is to provide a waiting rule set.In an iterative inference procedure, if rule R generates new data,then we add rules that are associated to R to waiting rule set,after that, we check whether new generated data satisfies conditional association rules, if so, we add the rules to the waiting rule set too. In next iterative inference procedure, we do inference according to rules in the waiting rule set. This method can avoid redundant inference. Improved algorithm called Rule Static Association Based Inference (RSAB)algorithm is shown as Algorithm 3.

Algorithm 3: ReasonAlgorithm_1(D)

```
begin
    Put all rules into WaitSet;
    clear WaitSet_1;
    repeat
        foreach R_i in WaitSet do
            if SingRuleReason(R_iD) then
                Add rules associated to R_i to
                WaitSet_1
                if new generated data satisfies
                conditional association rules then
                    add rules to WaitSet_1;
        WaitSet = WaitSet_1;
        clear WaitSet_1;
    until WaitSet is NULL ;
end
```

Algorithm 3 avoids some redundant inference procedures, but not thorough.For example, in table 2, new data generated by rule 1 can trigger rule 2, but when matching instance data, the case is not always true. Say, rule 1 generates (x' type v') , when it is applied to rule 2, we can get SubProperty(type v), (x' type v'), if SubProperty(type, v) has no corresponding instance data, rule 2 won't generate new inference procedure, so we need not add rule 2 to WaitSet. Determination of dynamic association between rules

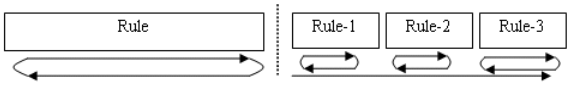


Figure 1 RGSB and RAB inference algorithm

is similar to that of conditional association rule. We use instance data to test if there exists matched data for non-matched pattern.Now we give Rule Dynamic Association Based Inference algorithm shown as Algorithm 4.

Algorithm 4: ReasonAlgorithm_2(D)

```
begin
    put all rules into WaitSet;
    clear WaitSet_1;
    repeat
        foreach R_i in WaitSet do
            if SingleRuleReason(Ri,D) then
                Add rules associated to R_i which
                match instance data generated by R_i
                Based on table 1;
                if new generated data satisfies
                conditional association rules then
                    add rules to WaitSet_1;
        WaitSet = WaitSet_1;
        clear WaitSet_1;
    until WaitSet is NULL ;
end
```

Algorithm 3 and Algorithm 4 describe association based inference algorithms, which avoid redundant inference procedure by association relationship, but do not consider inference order.

Let's come back to table 1 again, and assume that there are only rule 5, 6 and 9, if we do inference in such order [{5,6} → {9}] , we can assure that rule 9 need only be executed once, for newly generated data by rule 9 won't trigger rule 5 and rule 6, so we can put them in one group, and do inference before rule 9, or else we should do inference for rule 9 repeatedly. for example, if we do inference in such order [{5}→{9}→{6}], and if rule 6 generates new data, it will trigger rule 9 , probably trigger rule 5, we need a new run of inference [{5}→{9}→{6}].When we group rules in advance, the inference iteration faces rule group, not the whole rule sets.Figure 1 illustrates the difference between Rule Grouped-Sorted Based inference algorithm and Rule Association Based inference algorithm.

After the discussion, we propose rule grouped-sorted based inference algorithm in Algorithm 5:

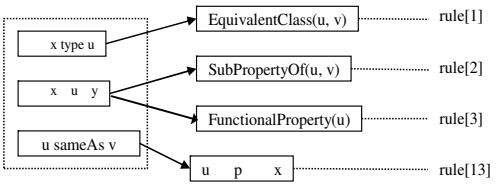


Figure 2 Pattern Sharing Structure

Algorithm 5: ReasonAlgorithm_3(D)

```
begin
    assume the order after grouping is
    [{R1}→{R2}→...→{Rn}];
    for i ← 1 to n do
        do inference on rule set{R_i};
        using ReasonAlgorithm_2(D);
end
```

4. Incremental Abox Inference Algorithms

In section 3, we described Initial inference, now let's consider another scenario: inference complete data has already existed in storage system, now new data comes and needs to be inserted into the system. If we use the methods introduced in section 3 directly, we will do redundant inference on old data repeatedly, for those algorithms do not differentiate new data and old data.

When new data and old data coexist,obviously,the old data is inference complete and we want to avoid inference procedure on it. But inference on new data may have relation with the old data, for new data and old data probably match certain rule premise.

In figure 2,we show design pattern sharing structure. we test each pattern on the new data, if rules share the same pattern, the times that rules match on new data can be greatly reduced. Take rule {1,2,3,13}in table 1 for example, in the rule premise P_{Ins}, the common pattern is[(x u y)x, u, y are variables].

Box drawn with dashed line in figure 2 contain all the rule premise(P_{Ins}), we can see that rule 2 and rule 3 share the same pattern (x u y). When rules are represented in such structure, we need only test the patters enclosed by the box in dashed line, then we can complete the inference procedure along the pointer of the pattern.we propose Rule Pattern Sharing Based(RPSB) incremental inference algorithm shown as Algorithm 6.

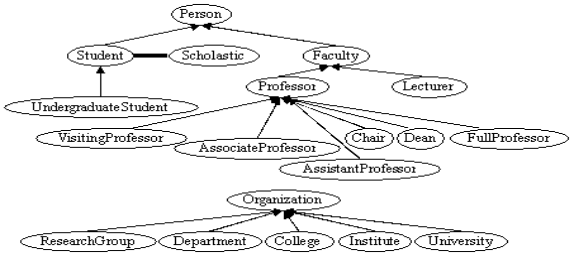


Figure 3 Class Hierarchy

Algorithm 6: ReasonAlgorithm_4(Δ D, D)

begin
  Assume the sharing patterns are $\{P_1, \ldots, P_m\}$;
  clear WaitSet;//waiting to be inferred
  foreach d_i *in* Δ *D* **do**
    foreach P_x *in* $\{P_1, \ldots, P_m\}$ **do**
      if d_i *matches* P_x **then**
        Assign corresponding variables in P_x and the pattern P_x pointing to;
        for each pattern P_x pointing to, query in new and old data and get inference result;
        if *new data exists in inference result* **then**
          add association rules based on ReasonAlgorithm_2(D) to the WaitSet;
end

5. Experiments

We now experimentally evaluate the techniques presented in this paper. First, we present the performance of different algorithms in the initial dataset. Second,In the case of incremental inference, we compare the performance between RPSB and RGSB.

Dataset: We use Benchmark[11] developed by LEHIGH university, which provides test data and its corresponding Ontology, and also defines 14 typical queries reflecting OWL semantic features. The details of the dataset will be discussed later.

Figure 3 and figure 4 separately depict the Class and Property hierarchy of the Benchmark.

Environment: All experiments were conducted on a Pentium IV 1.7GHz machine with 512M memory, and 40 G hard disk , running the WindowXP. We conducted our experiments on HStar system, which is an extention of OrientX developed by Renmin university of China. All experiments were repeated 10 times and the average processing time was calculated.

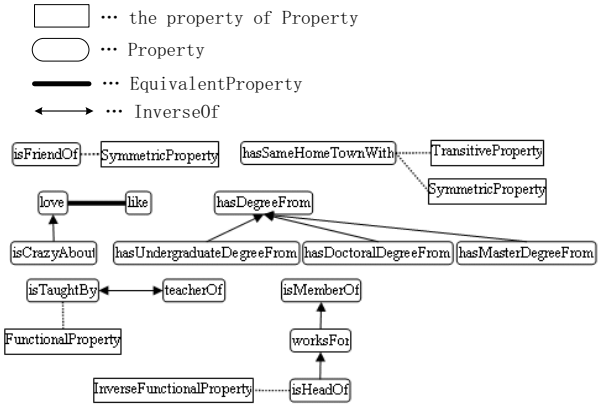


Figure 4 Property Hierarchy

Queries: We design the following 6 queries in order to test the completeness of inference. These queries cover all the inference rules.
Query-1. (? rdf:type Professor)
Query-2. (? rdf:type Student)
Query-3. (uri rdf:type ?)
Query-4. (uri isFriendOf ?)
Query-5. (uri hasSameHomeTownWith ?)
Query-6. (uri love ?)

In this experiment, we use three instance datasets and their corresponding Ontology. The number of triples and file size are shown in table 2:

dataset	tripple numbers	file size(KB)
Ontology	428	42
Instance 1	10694	992
Instance 2	19321	2483
Instance3	30181	4658

Table 2 Triple numbers and file size

Because the basic inference algorithm is based on integrity definition, its results can be used as the standard sets. table 3-5 shows the results of different inference algorithms on the above 6 queries, from which we can make a conclusion that the initial inference algorithms are complete and correct.(BI represents Basic Inference, None represents No Inference)

Query	None	BI	RSAC	RDAB	RGSB
Query-1	0	36	36	36	36
Query-2	0	666	666	666	666
Query-3	2	8	8	8	8
Query-4	2	5	5	5	5
Query-5	2	4	4	4	4
Query-6	1	1	1	1	1

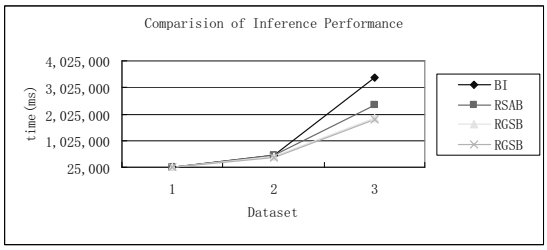


Figure 5 efficiency comparisons among different inference algorithms

Table 3 completeness comparison on dataset 1

Query	None	BI	RSAC	RDAB	RGSB
Query-1	0	44	44	44	44
Query-2	0	666	666	666	666
Query-3	2	18	18	18	18
Query-4	3	14	14	14	14
Query-5	5	70	70	70	70
Query-6	0	1	1	1	1

Table 4 completeness comparison on dataset 2

Query	None	BI	RSAC	RDAB	RGSB
Query-1	0	52	52	52	52
Query-2	0	666	666	666	666
Query-3	5	18	18	18	18
Query-4	9	730	730	730	730
Query-5	7	141	141	141	141
Query-6	0	18	18	18	18

Table 5 completeness comparison on dataset 3

Next we compare the efficiency of the four inference algorithms, figure 5 shows the performance of the different approaches. RGSB performs best, while the performance of basic inference declines sharply as the dataset increases.

At last, we evaluate the performance two incremental inference algorithms, which are RGSB and RPSB.We evaluate the performance from two aspects: processing time and results completeness.We omit the completeness evaluation due to limited space.

Figure 6 is about the processing time between RGSB and RPSB. Not surprisingly, RPSB inference algorithm outperforms RGSB.

6. Conclusion

We have presented algorithms for Abox inference on large OWL-Lite data in initial dataset, including basic inference, RSAC, RSAB and RGSB. From the experiment, we can see that these inference algorithms are complete, among which RGSB performs best. When large amount of inference data exists, we provide incremental inference,

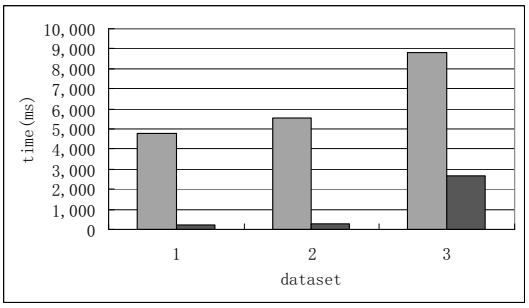


Figure 6 processing time between RGSB and RPSB

which not only ensure the inference completeness but also improve the efficiency. At last,extensive experiments show that our methods are efficient in practice.

References

[1] Owl,web ontology language. *http://www.w3.org/2004/OWL*.

[2] Rdf,resource description framework. *http://www.w3.org/RDF/*, 2004.

[3] D. O. D. M. Atanas Kiryakov. Owlim - a pragmatic semantic repository for owl. *In Proc. of Int. Workshop on Scalable Semantic Web Knowledge Base Systems*, 2005.

[4] P. G. David Wood and T. Adams. Kowari: A platform for semantic web storage and analysis. *In WWW*, 2005.

[5] V. Haarslev and R. Moller. High performance reasoning with very large knowledge bases: A practical case study. *In B. Nebel, editor, Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 161–166, 2001.

[6] V. Haarslev and R. R. Moller. A core inference engine for the semantic web. *In Workshop on Evaluation on Ontology-based Tools*, pages 27–36, 2003.

[7] I. Horrocks. The fact system. *In Automated Reasoning with Analytic Tableaux and Related Methods International Conference*, pages 27–30, 1998.

[8] A. K. J. Broekstra and F. Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. *In Proc. of the 1st International Semantic Web Conference(ISWC)*, pages 54–68, 2002.

[9] Y. P. L. Z. Li Ma, Zhong Su and T. Liu. Rstar: An rdf storage and query system for enterprise resource management. *In CIKM*, 2004.

[10] Z. Pan and Heflin. J. dldb: Extending relational databases to support semantic web queries. *In Workshop on Practical and Scalable Semantic Systems*, 2003.

[11] Z. P. Y. Guo and J. Hefin. An evaluation of knowledge base systems for large owl datasets.

[12] T. F. Youyong Zou and H. Chen. F-owl: an inference engine for the semantic web. *In Book Formal Approaches to Agent-Based Systems*, pages 238–248, 2004.

受限网络移动对象管理

(Network-Constrained Moving Objects Management)

Update-effcient Indexing of Moving Objects in Road Networks

Jidong Chen, Xiaofeng Meng, Yanyan Guo, Zhen Xiao

In Proceedings of the Third Workshop on Spatio-Temporal Database Management in conjunction with VLDB 06 (VLDB-STDBM2006), Seoul, Korea, September 11, 2006

Tracking Network-Constrained Moving Objects with Group Updates

Jidong Chen, Xiaofeng Meng, Benzhao Li, Caifeng Lai

In Proceedings of the Seventh International Conference on Web-Age Information Management (WAIM2006), page 158-169, Hong Kong, China, 17-19 June, 2006. Lecture Notes in Computer Science 4016, Springer 2006.

Modeling and Predicting Future Trajectories of Moving Objects in a Constrained Network

Jidong Chen, Xiaofeng Meng, Yanyan Guo, Stephane Grumbach, Hui Sun

In Proceedings of the 7th International Conference on Mobile Data Management (MDM 2006), Nara, Japan, May 9-13, 2006. IEEE Computer Society 2006: 156

In Proceedings of the Third Workshop on Spatio-Temporal Database Management
in conjunction with VLDB 06 (VLDB-STDBM2006), Seoul, Korea, September 11, 2006

Update-efficient Indexing of Moving Objects in Road Networks

Jidong Chen Xiaofeng Meng Yanyan Guo Zhen Xiao

School of Information, Renmin University of China
{chenjd, xfmeng, guoyy, xiaozhen}@ruc.edu.cn

Abstract

Recent advances in wireless sensor networks and positioning technologies have boosted new applications that manage moving objects. In such applications, a dynamic index is often built to expedite evaluation of spatial queries. However, development of efficient indexes is a challenge due to frequent object movement. In this paper, we propose a new update-efficient index method for moving objects in road networks. We introduce a dynamic data structure, called *adaptive unit*, to group neighboring objects with similar movement patterns. To reduce updates, an adaptive unit captures the movement bounds of the objects based on a prediction method, which considers the road-network constraints and stochastic traffic behavior. A spatial index (e.g., R-tree) for the road network is then built over the adaptive unit structures. Simulation experiments, carried on two different datasets, show that an adaptive-unit based index is efficient for both updating and querying performance.

Keywords Spatial-Temporal Databases, Moving Objects, Index Structure, Road Networks

1 Introduction

Recent advances in wireless sensor networks and positioning technologies have enabled a variety of new applications such as traffic management, fleet management, and location-based services that manage continuously changing positions of moving objects [11, 12]. In such applications, a dynamic index is often built to expedite evaluation of spatial queries. However, existing dynamic index structures (e.g. B-tree and R-tree) suffer from poor performance due to the large overhead of keeping the index updated with the frequently changing position data. Development of efficient indexes to improve the update performance is an important challenge.

Current work on reducing the index updates of moving objects mainly contains three kinds of approaches. First, most efforts [4, 9, 10, 15] focus on the update optimization of the existing multi-dimensional index structures especially the adaptation and extension of the R-tree [6]. To avoid the multiple paths search operation in the R-tree during the top-down update, recent work proposes the bottom-up approach [9, 10] and memo-based [15] structure to reduce the updates of the R-tree. Another method [4] exploits the change-tolerant property of the index structure to reduce the number of updates that cross the MBR boundaries of R-tree.

However, the indexes based on MBRs exhibit high concurrency overheads during node splitting, and each individual update is still costly. Therefore, some index methods based on a low-dimensional index structure (e.g. B^+-tree) are proposed [7, 16], which construct the second category of index methods. They combine the dimension reduction and linearization technique with a single B^+-tree to efficiently update the index structure.

The third kind of approaches use a prediction method with a time-parameterized function to reduce the index updates [12, 13, 14]. They describe a moving object's location by a linear function and the index is updated only when the parameters of the function change, for example, when the moving object changes its speed or direction. The MBRs of the index vary with the time as a function of the enclosed objects. However, the linear prediction is hard to reflect the movement in many real application and therefore leads to low prediction accuracy and frequent updates.

Though these index structures solve the problem of index updates to some extent, they are designed to index objects performing free movement in a two-dimensional space. We focus on the index update problem in real life environments, where the objects move within constrained networks, such as vehicles on roads. In such setting, the spatial property of objects' movement is captured by the network. Therefore, the

**Proceedings of the third Workshop on STDBM
Seoul, Korea, September 11, 2006**

spatial location of moving objects can be indexed by means of the road-network index structure. For example, moving objects can be accessed by each road segment indexed by the R-tree. Since the road network seldom change and objects just move from one part to the other part of the network, the R-tree in this case remains fixed. Existing index work that handles network-constrained moving objects [1, 5, 11] is based on this feature. They separate spatial and temporal components of the moving objects' trajectories and index the spatial aspect by the network with a R-tree. However, they are mostly concerned with the historical movement and therefore they do not consider the problem of index updates.

In this paper, we address the problem of efficient indexing of moving objects in road networks to support heavy loads of updates. We exploit the constraints of the network and the stochastic behavior of the real traffic to achieve both high update and query efficiency. We introduce a dynamic data structure, called *adaptive unit* (AU for short) to group neighboring objects with similar movement patterns in the network. A spatial index (e.g., R-tree) for the road network is then built over the adaptive units to form the index scheme for moving objects in road networks. The index scheme optimizes the update performance for the following reasons: (1) An AU functions as a one-dimensional MBR in the TPR-tree [13], while it minimizes expanding and overlaps by considering more movement features. (2) The AU captures the movement bounds of the objects based on a prediction method, which considers the road-network constraints and stochastic traffic behavior. (3) Since the movement of objects is reduced to occur in one spatial dimension and attached to the network, the update of the index scheme is only restricted to the update of the AUs. We have carried out extensive experiments based on two datasets. The results show that an adaptive-unit based index not only improves the efficiency of each individual update but also reduces the number of updates and is efficient for both updating and querying performance.

The main contributions of this paper are:

- The introduction of Adaptive Units that optimize for frequent index updates of moving objects in road networks.
- An experimental evaluation and validation of the efficient update as well as query performance of the proposed index structure.

The rest of the paper is organized as follows. Section 2 surveys related work and introduces underlying model. Section 3 describes the structure and algorithms of adaptive units for efficient updates. Section 4 contains algorithm analysis and experimental evaluation. We conclude and propose the future work in Section 5.

2 Related Work and Underlying Model

2.1 Related Work

There are lots of efforts at reducing the need for index updates of moving objects. In summary, they can be classified into three categories.

First, most work focuses on the update optimization of existing multi-dimensional index structures especially the adaptation and extension of the R-tree [6]. The top-down update of R-tree is costly since it needs several paths for searching the right data item considering the MBR overlaps. In order to reduce the overhead, Kwon et al. [9] develop the Lazy Update R-tree, which is updated only when an object moves out of the corresponding MBR. With adding a secondary index on the R-tree, it can perform the update operation in the bottom-up way. Recently, by exploiting the change-tolerant property of the index structure, Cheng et al. [4] present the CTR-tree to maximize the opportunity for applying lazy updates and reduce the number of updates that cross MBR boundaries. [10] extends the main idea of [9] and generalizes the bottom-up update approach. However, they are not suitable to the case where consecutive changes of objects are large. Xiong and Aref [15] present the RUM-tree that processes R-tree updates in a memo-based approach, which eliminates the need to delete the old data item during an index update. Therefore, its update performance is stable with respect to the changes between consecutive updates. In our index structure, however, the R-tree remains fixed since it indexes the road network and only the adaptive units are updated.

The second type of methods are based on the dimension reduction technique [11] and a low-dimensional index [7, 16] (e.g. B^+-tree). The B^x-tree [7, 16] combine the linearization technique with a single B^+-tree to efficiently update the index structure. They uses space filling curves and a pre-defined time interval to partition the representation of the locations of the moving objects. This makes the B^+-tree capable to index the two-dimensional spatial locations of moving objects. Therefore, the cost of individual update of index is reduced. However, the B^x-tree imposes discrete representation and may not keep the precise values of location and time during the partitioning. For our setting, the two-dimensional spatial locations of moving objects can be reduced to the 1.5 dimensions [8] by the road network where objects move.

The techniques in third category use a prediction method represented as the time-parameterized function to reduce the index updates [12, 13, 14]. They store the parameters of the function, e.g. the velocity and the starting position of an object, instead of the real positions. In this way, they update the index structure only when the parameters change (for example, the speed or the direction of a moving object changes). The Time-Parameterized R-tree (TPR-tree) [13] and its variants (e.g. TPR*-tree) [12, 14] are

the examples of this type of index structures. They all use a linear prediction model, which relates objects' positions as a linear function of time.However, the linear prediction is hard to reflect the movement in many real application especially in traffic networks where vehicles change their velocities frequently. The frequent changes of the object's velocity will incur repeated updates of the index structure. Our technique also fall into this category and apply an accurate prediction method we proposed in [3] by considering more transportation features.

Several methods have been proposed for indexing moving objects in spatially constrained networks. Pfoser et al. [11] propose to convert the 3-dimensional problem into two sub-problems of lower dimensions through certain transformation of the networks and the trajectories. Another approach, known as the FNR-tree [5], separates spatial and temporal components of the trajectories and indexes the time intervals that each moving object spends on a given network link. The MON-tree approach [1] further improves the performance of the FNR-tree by representing each edge by multiple line segments (i.e. polylines) instead of just one line segment. However, they all focus on the historical movement and cannot support frequent index updates. To the best of our knowledge, there is no current index method to support efficient updates of moving objects in road networks.

2.2 Underlying Model

We use the GCA model we proposed in [3] to model the network and moving objects. A road network is modeled as a graph of cellular automata (GCA), where the nodes of the graph represent road intersections and the edges represent road segments with no intersections. Each edge consists of a cellular automaton (CA), which is represented, in a discrete mode, as a finite sequence of cells.

In the GCA, a moving object is represented as a symbol attached to the cell and it can move several cells ahead at each time unit. Intuitively, the velocity is the number of cells an object can traverse during a time unit. The motion of an object is represented as some (time, location) information. Generally, such information is treated as a trajectory.

3 The Adaptive Unit

3.1 Structure and Storage

Conceptually, an adaptive unit is similar to a one-dimensional MBR in the TPR-tree, that expands with time according to the predicted movement of the objects it contains. However, in the TPR-tree, it is possible that an MBR may contain objects moving in opposite directions, or objects moving at different speeds. As a result, the MBR may expand rapidly, which may create large overlaps with other MBRs. The AU avoids this problem by grouping objects having similar moving patterns. Specifically, for objects in the same network edge, we use a distance threshold and a speed threshold to cluster the adjacent objects with the same direction and similar speed. In comparison, the AU has no obvious enlargement because objects in the AU move in a cluster.

We now formally introduce the AU. An AU is a 8-tuple:

$$\begin{aligned} AU \quad = \quad & (\texttt{auID}, \texttt{objSet}, \texttt{upperBound}, \texttt{lowerBound}, \\ & \texttt{edgeID}, \texttt{enterTime}, \texttt{exitTime}, \texttt{auInitLen}) \end{aligned}$$

where `auID` is the identifier of the AU, `objSet` is a list that stores objects belonging to the AU, `upperBound` and `lowerBound` are upper and lower bounds of predicted future trajectory of the AU. The trajectory bounds will be explained in details in Section 3.3. We assume the functions of trajectory bounds as follows:

$$\begin{aligned} \texttt{upperBound}: \quad & D(t) = \alpha_u + \beta_u \cdot t \\ \texttt{lowerBound}: \quad & D(t) = \alpha_l + \beta_l \cdot t \end{aligned}$$

`edgeID` denotes the network edge that the AU belongs to, `enterTime` and `exitTime` record the time when the AU enters and leaves the edge and `auInitLen` represents the initial length of the AU.

In the road network, multiple AUs are associated with a network edge. Since AUs in the same edge are likely to be accessed together during query processing, we store AUs by clustering on their `edgeID`. That is, the AUs in the same edge are stored in the same disk pages. To access AUs more efficiently, we create an in-memory, compact summary structure called the *direct access table* for each edge. A direct access table stores the summary information of each AU on an edge (i.e. number of objects, trajectory bounds) and pointers to AU disk pages. Each AU corresponds to an entry in the direct access table, which has the following structure `(auID, upperBound, lowerBound, auPtr, objNum)`, where `auPtr` points to a list of AUs in disk storage and `objNum` is the number of objects included in the AU. In order to minimize I/O cost, we use the direct access table to filter AUs and only access the disk pages when necessary.

3.2 The Index Scheme

We build a spatial index (e.g., R-tree) for the road network over the adaptive units to form the index scheme for the network-constrained moving objects. The AU index scheme is a two-level index structure. At the top level, it consists of a 2D R-tree that indexes spatial information of the road network. On the bottom level, its leaves contain the edges representing road segments included in the corresponding MBR of the R-tree and point to the lists of adaptive units. The top level R-tree remains fixed during the lifetime of the index scheme (unless there are changes in the network). The index scheme is developed with the R-tree

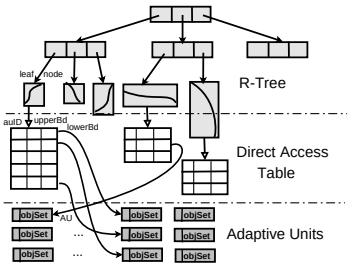
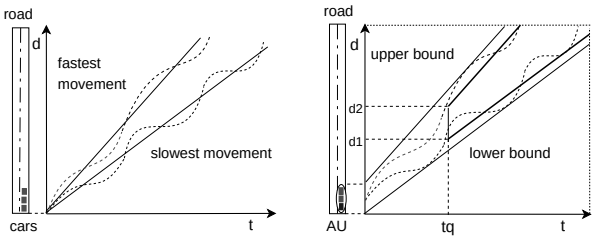


Figure 1: Structure of the AU index scheme



(a) Simulated trajectories (b) Trajectory bounds

Figure 2: The simulation-based prediction

in this paper, but any existing spatial index can also be used without changes.

Figure 1 shows the structure of the AU index scheme, which also includes the direct access table. The R-tree and adaptive units are stored in the disk. However, the direct access table is in the main memory since it only keeps the summary information of adaptive units. In the index scheme, each leaf node of the R-tree can be associated with its direct access table by its `edgeID` and the direct access table can connect to corresponding adaptive units by `auPtr` in its entries. Therefore, we only need to update the direct access table when AUs change, which greatly enhances the performance of the index scheme.

3.3 Optimizing for Updates

An important feature of the AU is that it groups objects having similar moving patterns. The AU is capable of dynamically adapting itself to cover the movement of the objects it contains. By tightly bounding enclosed moving objects for some time in the future, the AU alleviates the update problem of MBR rapid expanding and overlaps in the TPR-tree like methods.

For reducing the updates further, the AU captures the movement bounds of the objects based on a prediction method we proposed in [3], which considers the road-network constraints and stochastic traffic behavior. Since objects in an AU have similar movement, we then predict the movement of the AU, as if it were a single moving object. In the following, we describe the application and adaptation of the prediction method to the AU.

We use GCAs not only to model road networks, but also to simulate the movements of moving objects by the transitions of the GCA. Based on the GCA, the *Simulation-based Prediction (SP)* method to anticipate future trajectories of moving objects is proposed. The SP method treats the objects' simulated results as their predicted positions. Then, by the linear regression, a compact and simple linear function that reflects future movement of a moving object can be obtained. To refine the accuracy, based on different assumptions on the traffic conditions we simulate two future trajectories to obtain its predicted movement function. Specifically, we extend the CA model used in traffic flow simulation for predicting the future trajectories of objects by setting $P_d(i)$ to values that model different traffic conditions. In this setting, $P_d(i)$ is treated as a random variable to reflect the stochastic, dynamic nature of traffic system. By giving $P_d(i)$ two values (e.g. 0 and 0.1 in our experiments), we can derive two future trajectories, which describe, respectively, the fastest and slowest movements of objects. Finally, we translate the two regression lines, until all estimated future positions fall within to obtain the predicted trajectory bounds. The SP method is shown in Figure 2. Through the SP method, we obtain two predicted future trajectory bounds of objects. We apply this technique to the AU - a set of moving objects that have similar movement and are treated as one object.

The future trajectory bounds are predicted as soon as AU is created. The trajectory bounds will not be changed along the edge that the AU moves on until the objects in the AU move to another edge in the network. It is evident that the range of predicted bounds of AU will become wider with the time, which leads to lower accuracy of future trajectory prediction. However, if we issue another prediction when the predicted bounds are not accurate any more, the costs of simulation and regression are high. Considering that the movement of objects along one network edge is stable, we can assume the same trends of the trajectory bounds and adjust only the initial locations when the prediction is not accurate. Specifically, when the predicted position exceeds its actual position above the predefined accuracy, the AU treats its actual locations (the locations of the boundary objects) at that time as the initial locations of the two trajectory bounds and follow the same movement vector (e.g. slope of the bounds) as the previous bounds to provide more accurate predicted trajectory bounds. In this way, the predicted trajectory bounds can be effectively revised with few costs. Figure 2(b) shows the adaptation of the trajectory bounds. t_q is the time slice when actual locations of boundary objects in the AU exceeds the predicted bounds of the AU above precision threshold and the d_1,d_2 are the actual locations of the first object and last object respectively in the AU. The trajectory bounds are revised according to the actual locations and the original bounds' slopes. Therefore, without executing more prediction, the prediction accuracy of the objects' future trajectories can be kept high.

Since the R-Tree indexes the road network, it remains fixed, and the update of the AU index scheme restricts to the update of adaptive units. Specifically, an AU is usually created at the start of one edge and dropped at the end of the edge. Since the AU is a one-dimensional structure, it performs update operations much more efficiently than the two-dimensional indexes. We will describe these operations in details.

3.4 Update Operations

The update of an AU can be of the following form: creating an AU, dropping an AU, adding objects to an AU and removing objects from an AU.

Creating an AU

To create an AU, we first compose the $objSet$ – a list of objects traveling in the same direction with similar velocities, and in close-by locations. We then predict the future trajectories of the AU by simulation and compute its trajectory bounds. In fact, we treat the AU as one moving object (the object closest to the center of the AU) and predict its future trajectory bounds by predicting this object. The prediction starts when the AU is created and ends at the end the edge. Finally, we write the created AU to the disk page and insert the AU entry to its summary structure.

Dropping an AU

When objects in an AU move out of the edge, they may change direction independently. So we need to drop this AU and create new AUs in adjacent edges to regroup the objects. When the front of an AU touches the end of the edge, some objects in the AU may start moving out of the edge. However, the AU cannot be dropped because a query may occur at that time. Only after the last object in the AU enters another edge and joins another AU, can the AU be dropped. Dropping an AU is simple. Through its entry in direct access table, we find the AU and delete it.

Adding and removing objects from an AU

When an object leaves an AU, we remove this object from the AU and find another AU in the neighborhood to check if the object can fit that AU. If it can, the object will be inserted into that AU, otherwise, a new AU is created for this object. Specifically, when adding an object into an AU, we first find the direct access table of the edge that the object lies and, by its AU entry in the table, access the AU disk storage. Finally, we insert into the objects list of the AU and update the AU entry in the direct access table. Removing an object from an AU has the similar process.

Therefore, when updating an object in the AU index scheme, we first determine whether the object is leaving the edge and entering another one. If it is moving to another edge, we delete it from the old AU (if it is the last object in the old AU, the AU is also dropped) and insert it into the nearest AU or create a new AU in the edge it is entering. Otherwise, we do not update the AU that the object belongs to unless its position exceeds the bounds of the AU. In that case, we execute the same updates as those when it moves to another edge or only revise the predicted trajectory bounds of the AU. Factually, we find, from the experiment evaluation, that the chances that objects move beyond the trajectory bounds of its AU on an edge are very slim. The algorithm 1 shows the update algorithm of AUs.

Algorithm 1: Update($objID, position, velocity, edgeID$)

input : $objID$ is the object identifier, $position$ and $velocity$ are its position and velocity, $edgeID$ is the edge identifier where the object lies

Find AU where $objID$ is included before update;
if $AU.edgeID \neq edgeID$ *or (position* $<$ $AU.lowerBound$ *or position* $>$ $AU.upperBound$*)*
then
  `// The object moves to a new edge or exceeds bounds of its original AU`
  Find the nearest AU AU_1 for $objID$ on $edgeID$;
  if $GetNum(AU_1.objSet) < MAXOBJNUM$ *and* $ObjectFitAU(objID, position, velocity, AU_1)$ **then**
    InsertObject($objID, AU_1.auID, AU_1.edgeID$);
  else $AU_2 \leftarrow$ CreateAU($objID$,$edgeID$);
  if $GetNum(AU.objSet) > 1$ **then**
    DeleteObject($objID, AU.auID, AU.edgeID$);
  else DropAU($AU.edgeID, AU.auID$);
end

In summary, updating the AU-based index is easier than updating the TPR-tree. It never invoke any complex node splitting and merging. Moreover, thanks to the similar movement features of objects in an AU and the accurate prediction of the SP method, the objects are seldom removed or added from their AU on an edge, which reduces the number of index updates.

3.5 Query Algorithm

Query processing in the AU index scheme is straightforward. Given a query, we use the top level R-tree to get the edges involved and then scan the direct access tables of the edges. With the `upperBound` and the `lowerBound` in the direct access table, we can easily find AU entries that intersect the query, and then visit the disk pages to get more information about these AUs. For space limitation, we just take window range query for example. Given a range with (X_1, Y_1, X_2, Y_2), we first perform a spatial range search in the top level R-Tree to locate the edges (e.g. $e_1, e_2, e_3, \ldots$). For each selected edge e_i, we transform the original search (X_1, Y_1, X_2, Y_2) to a 1D search range (S_1, S_2) $(S_1 \leq S_2)$, where S_1 and S_2 are the relative distances from the start vertex along the edge e_i. In the case of multiple intersecting edges, we can divide the query range into several sub-ranges by edges and apply the transformation method to each edge. The method is also applicable to the various modes

that the query and edges intersect. Here, we only illustrate the case when the query window range only intersects one edge and compute its relative distances S_1 and S_2. It can be easily extended to other cases. Suppose $X_{start}, Y_{start}, X_{end}, Y_{end}$ are the start vertex coordinates and the end vertex coordinates of the edge e_i. According to Thales Theorem about similar triangles, we obtain S_1 and S_2 as follows:

$$\begin{aligned} r &= \sqrt{(X_{start} - X_{end})^2 + (Y_{start} - Y_{end})^2} \\ S_1 &= \frac{X_1 - X_{start}}{X_{end} - X_{start}} r \\ S_2 &= \frac{Y_1 - Y_{start}}{Y_{end} - Y_{start}} r \end{aligned}$$

The transformed query (S_1, S_2) is then executed in each of the AUs in the direct access table of the corresponding edge e_i. By the trajectory bounds of the AU, we can determine whether the transformed query intersects the AU, thus filtering the unnecessary AUs quickly. Finally, we access the selected AUs in disk storage and return the objects satisfying the query window. In summary, the query processing is efficient due to the grouping of similar objects in AUs and the dimensionality reduction of the query.

4 Performance Analysis

We evaluate the AU index scheme (denoted as "AU index") by comparing it with the TPR-tree and the AU index scheme when the direct access table is not used (denoted as "AU index without DT"). We measure their their update performance with the individual update, update frequency and total update costs and their query performance.

4.1 Datasets

We use two datasets for our experiments. The first is generated by the CA simulator, and the second by the Brinkhoff's Network-based Generator [2]. We use the CA traffic simulator to generate a given number of objects in a uniform network of size 10000×10000 consisting of 500 edges. Each object has its route and is initially placed at a random position on its route. The initial velocities of the objects follow a uniform random distribution in the range $[0, 30]$. The location and velocity of every object is updated at each time-stamp. The Brinkhoff's Network-based Generator is used as a popular benchmark in many related work. The generator takes a map of a real road network as input (our experiment is based on the map of Oldenburg including 7035 edges). The positions of the objects are given in two dimensional X-Y coordinates. We transform them to the form of (`edgeid`, `pos`), where `edgeid` denotes the edge identifier and `pos` denotes the object relative position on the edge. The generator places a given number of objects at random positions on the road network, and updates their locations at each time-stamp.

Table 1: Parameters and their settings

Parameters	Settings
Page size	4K
Node capacity	100
Numbers of queries	200
Numbers of mo(cars)	10K, ... , 50K, ... , **100K**
Numbers of updates	**100K**, ... , 500K, ... , 1M
Dataset Generator	CA Simulator, Network-based Generator

We implemented both the AU index scheme and the TPR-tree in Java and carried out experiments on a Pentium 4, 2.4 GHz PC with 512MB RAM running Windows XP. To improve the performance of the index structure, we employed a LRU buffer of the same size as the one used in the TPR-tree [13]. We summarize workload parameters in Table 1, where values in bold are default values.

4.2 Update Cost

We compare the cost of index update for the AU index and the TPR-tree in terms of the average individual update cost, update frequency and total update cost.

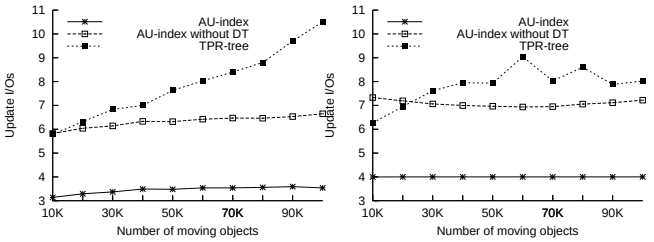
Individual Update Cost

We study the individual update performance of the index while varying the number of moving objects and updates. Figure 3 shows the average individual update cost when increasing the data size from 10K to 100K. Figure 4 shows how the performance varies over time. Clearly, updating the TPR-tree tends to be costly, and the problem is exacerbated when the data size increases. In each case of different data size and different number of updates, the AU index has much lower update cost than the TPR-tree. The main reason can be explained as follows. Each update of the TPR-tree involves the search of an old entry and a new entry, as well as the modification of the index structure (node splitting, merging, and the propagating of changes upwards). The cost increases with larger data size due to more overlaps among MBRs. The changes of index structure with the increase of data updates also affect the performance of the TPR-tree. However, the AU index has better performance because its update only restricts to the AU's update and as a one-dimensional access structure, the AU has few overlaps and incurs no cost associated with node splitting and the propagation of MBR updates.

The direct access table of the AU index has a significant contribution in improving update performance. This is because the search of the specific AU is accelerated by the in-memory structure.

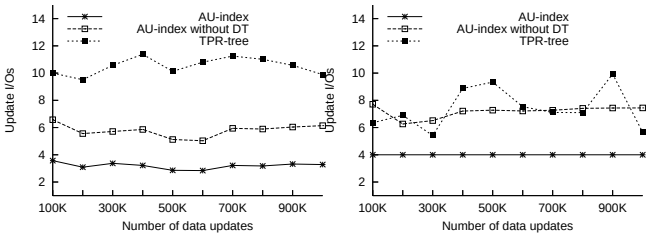
Update Frequency

Frequent updates of moving objects (a.k.a. data updates) may lead to frequent updates of index. When an object's position exceeds the MBR or AU, the index needs to be updated to delete the object from the old MBR or AU and insert it to another one. In this experiment, we measure the index update rate, which is



(a) Brinkhoff (b) CA

Figure 3: Individual Update Cost with Different Datasize



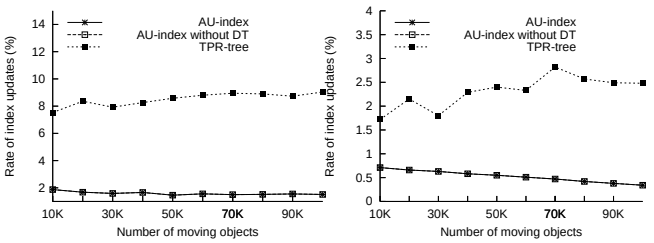
(a) Brinkhoff (b) CA

Figure 4: Individual Update Cost over Time

the ratio between number of index updates and number of data updates, for every 100K data updates and different data size. Figure 5 and 6 show that the update rate of the TPR-tree is nearly 4 to 5 times more than that of the AU index. The index update rate depends on the prediction method. In the AU index, the future positions of the object are predicted more accurately, so the object is likely to remain in its AU, which leads to fewer index updates.

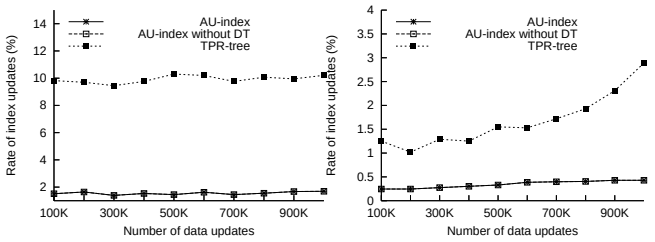
Total Update Costs

The total update costs depend on the update frequency and the average individual update cost, and it can reflect the index update performance more accurately. From both Figure 7 and 8, we can see that although the AU index has to deal with the creation and dropping of AUs, the TPR-tree incurs much higher update costs than the AU index and its performance deteriorates dramatically as data size increases. This is mainly due to the inaccuracy of the linear prediction model and the complex reconstruction of the TPR-tree (e.g. splitting and merging).



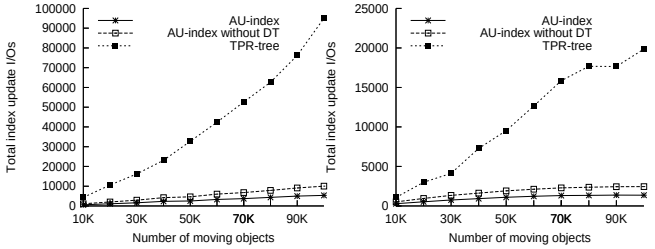
(a) Brinkhoff (b) CA

Figure 5: Index Update Frequency with Different Datasize



(a) Brinkhoff (b) CA

Figure 6: Index Update Frequency over Time



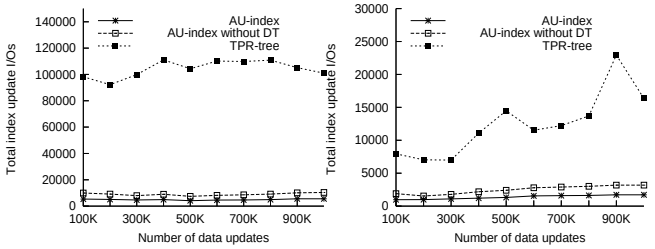
(a) Brinkhoff (b) CA

Figure 7: Total Update Cost with Different Datasize

For each data size, the update costs of the two indexes in the Brinkhoff's dataset are both higher than those in the CA dataset due to the higher complexity of road network and skewed spatial distribution of objects in the Brinkhoff's dataset.

4.3 Query Cost

We study the window range query performance of the TPR-tree and the AU index with different update settings. We increase the number of updates from 100K to 1M to examine how query performance is affected. We issued 200 range queries for every 100K updates in a 1M dataset. Figure 9 shows that the cost of the TPR-tree increases much faster as the number of updates increases. The cost of the AU index is considerably lower and is less sensitive to the number of updates. This is because the adaptive units in the AU index have much less overlaps than the MBRs in the TPR-tree, and the overlaps to a large extent determine the range query cost. Besides, as objects move apart, the amount of dead space in the TPR-tree increases,

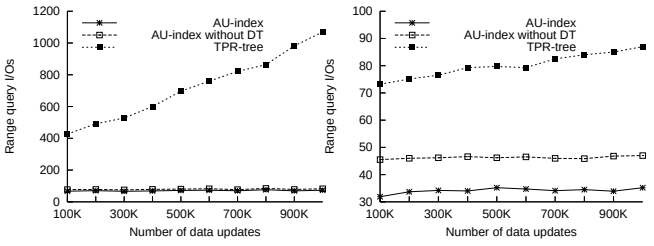


(a) Brinkhoff (b) CA

Figure 8: Total Update Cost over Time

which makes false hits more likely. Also, updates lead to the expanding and overlaps of MBRs, which further deteriorate the performance of the TPR-tree. For the AU index, the increase of the updates hardly affect the total number of AUs, and the chances of overlaps of different AUs are very slim.

We also study the query performance while varying the number of moving objects and query window size. For the space limitation, we do not report the experimental results. Also, in each case, the AU index has lower query cost than the TPR-tree and scales well.



(a) Brinkhoff (b) CA

Figure 9: Effect of Updates on Query Performance

5 Conclusions and Future Work

Indexing objects moving in a constrained network especially the road network is a topic of great practical importance. We focus on the index update issue for the current positions of network-constrained moving objects. We introduce a new access structure, adaptive unit that exploits as much as possible the characteristics of the movements of objects. The updates of the structure are minimized by an accurate prediction method which produces two trajectory bounds based on different assumptions on the traffic conditions. The efficiency of the structure also results from the possible reduction of dimensionality of the trajectory data to be indexed. Our experimental results performed on two datasets show that the efficiency of the index structure is one order of magnitude higher than the TPR-tree.

In the future, we will compare the update performance with the work of the R-tree-based updating optimization such as RUM-tree [15] and CTR-tree [4]. On the other hand, since the adaptive units contain the predicted future trajectories of moving objects, the predictive query algorithms can be developed naturally based on the adaptive unit-based index. Furthermore, we will extend the query algorithms to support the KNN query and continuous query for moving objects in the road network.

Acknowledgments

This research was partially supported by the grants from the Natural Science Foundation of China under grant number 60573091, 60273018; the Key Project of Ministry of Education of China under Grant No.03044; Program for New Century Excellent Talents in University (NCET); Program for Creative PhD Thesis in University. The authors would like to thank Jianliang Xu and Haibo Hu from Hong Kong Baptist University and Stéphane Grumbach from CNRS, LIAMA in China for many helpful advice and assistance.

References

[1] V. T. Almeida, R. H. Güting. Indexing the Trajectories of Moving Objects in Networks (Extended Abstract). In SSDBM, 2004, 115-118.

[2] T. Brinkhof. A framework for generating network-based moving objects. In GeoInformatica, 6(2), 2002, 153-180.

[3] J. Chen, X. Meng, Y. Guo, S. Grumbach, H. Sun. Modeling and Predicting Future Trajectories of Moving Objects in a Constrained Network. In MDM, 2006, 156 (MLASN workshop).

[4] R. Cheng, Y. Xia, S. Prabhakar, R. Shah. Change Tolerant Indexing for Constantly Evolving Data. In ICDE, 2005, 391-402.

[5] E. Frentzos. Indexing objects moving on Fixed networks. In SSTD, 2003, 289-305.

[6] A. Guttman. R-trees: A Dynamic Index Structure for Spatial Searching. In SIGMOD, 1984, 47-57.

[7] C. S. Jensen, D. Lin, B. C. Ooi. Query and Update Efficient B+-Tree Based Indexing of Moving Objects. In VLDB, 2004, 768-779.

[8] G. Kollios, D. Gunopulos, V. J. Tsotras. On indexing mobile objects. In PODS, 1999, 261-272.

[9] D. Kwon, S. J. Lee, S. Lee. Indexing the current positions of moving objects using the lazy update R-tree. In MDM, 2002, 113-120.

[10] M. L. Lee, W. Hsu, C. S. Jensen, B. Cui, K. L. Teo. Supporting Frequent Updates in R-Trees: A Bottom-Up Approach. In VLDB, 2003, 608-619.

[11] D. Pfoser, C. S. Jensen. Indexing of network constrained moving objects. In ACM-GIS, 2003, 25-32.

[12] S. Saltenis, C. S. Jensen. Indexing of Moving Objects for Location-Based Service. In ICDE, 2002, 463-472.

[13] S. Saltenis, C. S. Jensen, S. T. Leutenegger, M. A. Lopez. Indexing the Positions of Continuously Moving Objects. In SIGMOD, 2000, 331-342.

[14] Y. Tao, D. Papadias, J. Sun. The TPR*-Tree: An Optimized Spatiotemporal Access Method for Predictive Queries. In VLDB, 2003, 790-801.

[15] X. Xiong, W. G. Aref. R-trees with Update Memos. In ICDE, 2006, 22.

[16] M. L. Yiu, Y. Tao, N. Mamoulis. The $B^{dual-Tree}$: Indexing Moving Objects by Space-Filling Curves in the Dual Space. To appear in Very Large Data Base Journal, 2006.

In Proceedings of the Seventh International Conference on Web-Age Information Management (WAIM2006), page 158-169, Hong Kong, China, 17-19 June, 2006. Lecture Notes in Computer Science 4016, Springer 2006.

Tracking Network-Constrained Moving Objects with Group Updates

Jidong Chen, Xiaofeng Meng, Benzhao Li, and Caifeng Lai

School of Information, Renmin University of China,
Beijing, 100872, China,
{chenjd, xfmeng, bzli, laicf}@ruc.edu.cn

Abstract. Advances in wireless sensors and position technologies such as GPS enable location-based services that rely on the tracking of continuously changing positions of moving objects. The key issue in tracking techniques is how to minimize the number of updates, while providing accurate locations for query results. In this paper, for tracking network-constrained moving objects, we first propose a simulation-based prediction model with more accurate location prediction for objects movements in a traffic road network, which lowers the update frequency and assures the location precision. Then, according to their predicted future functions, objects are grouped and only the central object in each group reports its location to the server. The group update strategy further reduces the total number of objects reporting their locations. A simulation study has been conducted and proved that the group update policy based on the simulation prediction is superior to traditional update policies with fewer updates and higher location precision.

1 Introduction

The continued advances in wireless sensors and position technologies such as GPS enable new data management applications such as traffic management and location-based services that monitor continuously changing positions of moving objects [2, 7]. In these applications, large amounts locations can be sampled by sensors or GPS periodically, then sent from moving clients to the server and stored in a database. Therefore, continuously maintaining in a database current locations of moving objects namely tracking technique becomes a fundamental component of these applications [1, 2, 9, 10]. The key issue is how to minimize the number of updates, while providing precise locations for query results.

The number of updates from moving objects to the server database depends on both the update frequency and the number of objects to be updated. To reduce the location updates, most existing works are proposed to lower the update frequency by a prediction method [1, 9, 10]. They usually use the linear prediction which represents objects locations as linear functions of time. The objects do not report their locations to the server unless their actual positions exceed the predicted positions to a certain threshold. This provides a general principle for the location update policies in a moving object database system.

However, few research works focus on improving the update performance from the aspect of reducing the number of objects to be updated. We observe that in many applications, objects naturally move in clusters, including vehicles in a congested road network, packed goods transmitted in a batch, animal and bird migrations. It is possible that the nearby objects are grouped and only one object in the group reports its location to the server to represent all objects within it. Considering real life applications, we focus on objects moving on a road network. Figure 1 gives an example of grouping vehicles on a part of road network. Due to the grouping of vehicles in each road segment, the total location updates sent to the server are reduced from 9 to 5.

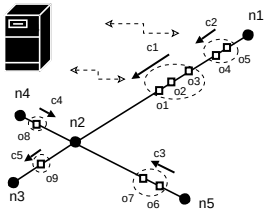


Fig. 1. Group location updates

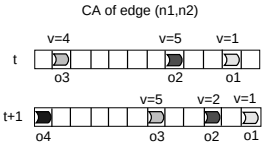


Fig. 2. A transition of the CA on an edge

The idea of grouping objects for location updates is similar to the GBL proposed in [6], but the GBL groups objects by their current locations and predicted locations after a time parameter τ. In fact, it obtains the predicted locations also by the linear prediction model assuming the linear movement with current velocity. However, in the urban road network, due to complex traffic conditions, cars may update their velocities frequently even for each timestamp. In this case, the linear prediction used in the GBL and other location update methods is inapplicable because the inaccurate predicted locations result in frequent location updates and lots of group management. In this paper, for the purpose of improving the performance of tracking for network-constrained moving objects, we focus on the both two factors affecting location updates and propose our solutions. One is a better prediction model to lower update frequency, and the other is a group update strategy to reduce the total number of objects reporting their locations. The accurate prediction model also reduces the maintenance of the groups and assures the location precision for querying.

Therefore, we first propose a simulation-based prediction (SP) model which captures traffic features in constrained networks. Specifically, we model road networks by graphs of cellular automata, which are also used to simulate vehicles future trajectories in discrete points in accordance with the surrounding traffic conditions. To refine the accuracy, we simulate two future trajectories to obtain the predicted movement function, which correspond to the fastest and slowest possible movements. We then propose a group location update strategy based on the SP model (GSP) to minimize location updates. In the GSP, for each edge in

the road network, the objects with their predicted movement functions similar are grouped or clustered and only the object nearest to its group center needs to report the location of the whole group. Within a certain precision, the locations of other objects can be approximated to their group location. Finally, through the experimental evaluations, we show that the GSP strategy has more efficient update performance as well as higher location precision.

The rest of the paper is organized as follows. Section 2 surveys related work by classifying the existing tracking techniques. In Section 3, a road network modeled as a graph of cellular automata is represented and our simulation-based prediction model is proposed. Section 4 describes our group update strategy. Section 5 contains an experimental analysis, and finally Section 6 concludes.

2 Related Work

Research on tracking of moving objects has mainly focused on location update policies. Existing methods can be classified according to the threshold, the route, the update mode or the representation and prediction of objects future positions.

Updates differ in threshold and route

Wolfson et al.[9] first proposed the dead-reckoning update policies to reduce the update cost. According to the threshold, they are divided into three policies, namely the Speed Dead Reckoning (SDR) having a fixed threshold for all location updates, the Adaptive Dead Reckoning (ADR) having different thresholds to different location updates and the Disconnection Detection Reckoning (DTDR) having the continuously decreasing threshold since last location update. The policies also assume that the destination and motion plan of the moving objects is known a priori. In other words, the route is fixed and known. In [4], Gowrisankar and Nittel propose a dead-reckoning policy that uses angular and linear deviations. They also assume that moving objects travel on predefined routes. Lam et al. propose two location update mechanisms for further considering the effect of the continuous query results on the threshold [7]. The idea is that the moving objects covered by the answers of the queries have a lower threshold, leading to a higher location accuracy. Zhou et al. [11] also take the precision of query results as a result of a negotiated threshold by the Aqua location updating scheme proposed.

Updates differ in representation and prediction of future positions

Wolfson and Yin [10] consider tracking with accuracy guarantees. They introduce the deviation update policy for this purpose and compare it with the distance policy. The difference between the two polices lies in the representation of future positions respectively with the linear function in the former and constant function in the latter. Based on experiments with artificial data generated to resemble real movement data, they conclude that the distance policy is outperformed by the deviation policy. Similarly, Civilis et al. [1, 2] propose three update policies: a point policy, a vector policy, and a segment-based policy, which differ in how they predict the future positions of a moving object. In fact, the first and third policy are the good representatives of the policies in

[10]. They further improve the update policies in [2], by exploiting the better road-network representation and acceleration profiles with routes. It should also be noted that Ding and Guting [3] have recently discussed the use of what is essentially segment-based tracking based on their proposed data model for the management of road-network constrained moving objects. In paper [8], the non-linear models such as the acceleration are used to represent the trajectory which is affected by the abnormal traffic such as traffic incident.

Updates based on individual object and their group

Most existing update techniques are developed to process individual updates efficiently [1, 2, 9, 10]. To reduce the expensive uplink updates from the objects to the location server, Lam et al. [6] propose a group-based scheme in which moving objects are grouped so that the group leader will send location update on behalf of the whole group. A group-based location update scheme for personal communication network is also proposed in [5]. The aim is to reduce location registrations by grouping a set of mobile objects at their serving VLRs.

Our work improves the tracking technique from the aspect of prediction model and update mode, and focuses on the accuracy of the predicted positions of the objects in urban road networks. Based on their predicted movement functions, we groups objects to further reduce their location updates. To the best of our knowledge, there exists no proposal for tracking of moving objects that combines the simulation based prediction and grouping of objects by exploiting the movement features of objects in traffic systems.

3 Data Model and Trajectory Prediction

We model a road network with a graph of cellular automata (GCA), where the nodes of the graph represent road intersections and the edges represent road segments with no intersections. Each edge consists of a cellular automaton (CA), which is represented, in a discrete mode, as a finite sequence of cells. The CA model was used in this context by [12].

In the GCA, a moving object is represented as a symbol attached to the cell and it can move several cells ahead at each time unit. Intuitively, the velocity is the number of cells an object can traverse during a time unit. Let i be an object moving along an edge. Let $v(i)$ be its velocity, $x(i)$ its position, $gap(i)$ the number of empty cells ahead (forward gap), and $P_d(i)$ a randomized slowdown rate which specifies the probability it slows down. We assume that V_{max} is the maximum velocity of moving objects. The position and velocity of each object might change at each transition of the GCA according to the rules below (adapted from [12]):

1. if $v(i) < V_{max}$ and $v(i) < gap(i)$ then $v(i) \leftarrow v(i) + 1$
2. if $v(i) > gap(i)$ then $v(i) \leftarrow gap(i)$
3. if $v(i) > 0$ and $random() < P_d(i)$ then $v(i) \leftarrow v(i) - 1$
4. if $(x(i) + v(i)) \leq l$ then $x(i) \leftarrow x(i) + v(i)$

The first rule represents linear acceleration until the object reaches the maximum speed V_{max}. The second rule ensures that if there is another object in front

of the current object, it will slow down in order to avoid collision. In the third rule, the $P_d(i)$ models erratic movement behavior. Finally, the new position of object i is given by the fourth rule as the sum of the previous position with the new velocity if it is in the CA. Figure 2 shows a transition of the cellular automaton of edge (n_1, n_2) in Figure 1 in two consecutive timestamps. We can see that at time t, the speed of the object o_1 is smaller than the gap (i.e. the number of cells between the object o_1 and o_2). On the other hand, the object o_2 will reduce its speed to the size of the gap. According to the fourth rule, the objects move to the corresponding positions based on their speeds at time $t+1$.

We use GCAs not only to model road networks, but also to simulate the movements of moving objects by the transitions of the GCA. Based on the GCA, a *Simulation-based Prediction (SP)* model to anticipate future trajectories of moving objects is proposed. The SP model treats the objects simulated results as their predicted positions. Then, by the linear regression, a compact and simple linear function that reflects future movement of a moving object can be obtained. To refine the accuracy, based on different assumptions on the traffic conditions we simulate two future trajectories to obtain its predicted movement function. Figure 3 and Figure 4 show the comparison of the SP model and the linear prediction (LP) model. We can see from Figure 3 that the LP model cannot predict accurately the future trajectories of objects due to the frequent changes of the object velocity in traffic road networks.

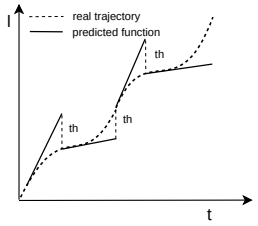


Fig. 3. The *Linear* Prediction

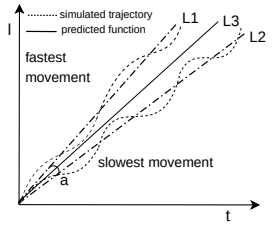


Fig. 4. The *Simulation Based* Prediction

Most existing work uses the CA model for traffic flow simulation in which the parameter $P_d(i)$ is treated as a random variable to reflect the stochastic, dynamic nature of traffic system. However, we extend this model for predicting the future trajectories of objects by setting $P_d(i)$ to values that model different traffic conditions. For example, laminar traffic can be simulated with $P_d(i)$ set to 0 or a small value, and the congestion can be simulated with a larger $P_d(i)$. By giving $P_d(i)$ two values, we can derive two future trajectories, which describe, respectively, the fastest and slowest movements of objects. In other words, the object future locations are most probably bounded by these two trajectories. The value of $P_d(i)$ can be obtained by the experiences or by sampling from the given dataset. Our experiments show one of methods to choose the value of $P_d(i)$. It is proved that 0 and 0.1 are realistic values of $P_d(i)$ in our cases.

For getting the future predicted function of an object from the simulated discrete points, we regress the discrete positions to a linear function by the Least Square Estimation (LSE) in Statistics. It can be calculated efficiently with low data storage cost. Let the discrete simulated points be $(t_0, l_0), (t_1, l_1), ..., (t_i, l_i), ..., (t_{n-1}, l_{n-1})(i \geq 0, n > 0)$, where t_i is the time at $i+1$ timestamp, l_i is the relative distance of the moving object in an edge at timestamp t_i, n is the total time units for the simulation, a linear function of time variable t can be obtained as follows:

$$l = a_0 + a_1 t \tag{1}$$

where the slope a_1 and the intercept a_0 can be calculated in Statistics

$$a_1 = \frac{n \sum_{i=0}^{n-1} t_i l_i - \sum_{i=0}^{n-1} t_i \sum_{i=0}^{n-1} l_i}{n \sum_{i=0}^{n-1} t_i^2 - (\sum_{i=0}^{n-1} t_i)^2} \tag{2}$$

$$a_0 = \frac{1}{n} \sum_{i=0}^{n-1} l_i - \frac{a_1}{n} \sum_{i=0}^{n-1} t_i \tag{3}$$

After regressing the two simulated future trajectories to two linear function denoting L_1 and L_2 in Figure 4, we can compute the middle straight line L_3, the bisector of the angle a between L_1 and L_2 as the final predicted function $L(t)$.

Through the SP model, we obtain a compact and simple linear prediction function for the moving object. However, this is different from the linear prediction in that the simulation-based prediction method not only considers the speed and direction of each moving object, but also takes correlation of objects as well as the stochastic behavior of the traffic into account. The experimental results also show it is a more accurate and effective prediction approach.

4 Group Location Update Strategy

As the number of updates from moving objects to the server database depends on both the update frequency and the number of objects updated, we propose a group location update strategy based on the SP model (GSP) to minimize location updates. In the GSP, for each edge in a road network, the objects are grouped or clustered by the similarity of their predicted future movement function and their locations are represented and reported by the group (Figure 1). It means that the nearby objects with similar movement during the future period on the same edge are grouped and only the object nearest to its group center needs to report the location of the whole group. Within a certain precision, the locations of other objects can be approximated to their group location.

The idea of grouping objects for location updates is similar to the GBL proposed in [6]. The main differences are that the GSP groups the objects by their future movement function predicted from the SP model instead of their current locations and predicted locations after a time parameter τ obtained by

current velocity. Grouping by objects predicted movement function can insure the validity of the groups. The accurate prediction from the SP model can also reduce the maintenance of the groups. Due to the constraint of the road network, each group in the GSP has its lifetime in accordance to the edge. A group only exists on one edge and will be dissolved when objects within it leave the edge. Furthermore, unlike the GBL in which objects have to send a lots of messages to each other and compute the costly similarities for grouping and leader selection, the GSP executes the grouping on the server after predicting. This alleviates the resource consumption of moving clients and overloads of wireless communication.

The similarity of two objects simulated future trajectories in the SP model has to be computed by comparing a lot of feature points on the trajectories. A straightforward method is to select some of the simulated points to sum their distance difference. However, the computation cost for simulated trajectories is very high. For simplicity and low cost, we group objects by comparing their final predicted linear functions. Therefore, the movement similarity of two objects on the same edge can be determined by their predicted linear functions and the length of the edge. Specifically, if both the distance of their initial locations and their distance when one of the objects arrives the end of the edge are less than the given threshold (corresponding to the update threshold ε), we group the two objects together. These distances can be easily computed by their predicted functions. Figure 5 shows the predicted movement functions (represented as $L1, L2, L3, L4, L5$) of the objects o_1, o_2, o_3, o_4, o_5 on the edge (n_1, n_2) from Figure 1. le is the length of the edge and $t1, t2, t3, t4$ are respectively the time when the objects o_1, o_2, o_3, o_4 arrive the end of the edge. Given the threshold is 7, for objects o_1, o_2, the location difference between them at initiate time and t_1 are not larger than 7, therefore, they are clustered in one group c_1. We then compare the movement similarities of o_3 and o_1 as well as o_3 and o_2. The location differences are all not larger than 7, so o_3 can be inserted to c_1. Although at the initiate time, o_3 and o_4 are very close with the distance less than 7, they move far away each other in the future and their distance exceeds 7 when o_3 arrives the end of the edge. They cannot be grouped in one cluster. In the same way, o_4 and o_5 form the group c_2. Therefore, given a threshold, there are three cases of the objects predicted linear function when they are grouped together on one edge. These cases can be seen in the Figure 5 respectively labeled by a ($L2$ and $L3$ with objects moving close), b ($L1$ and $L3$ with objects moving far away) and c ($L1$ and $L2$ with one object exceeding another one).

In a road network, we group objects on the same edge. When objects move out of the edge, they may change direction independently. So we dissolve this group and regroup the objects in adjacent edges. Each group has its lifetime from the group formation to all objects within it leaving the edge. For each edge, with the objects predicted functions, groups are formed by clustering together sets of objects not only close to each other at a current time, but also likely to move together for a while on one edge. We select the object closest to the center of its group both the current time and some period in future on the edge to represent the group. The central object represents its group and is responsible for reporting

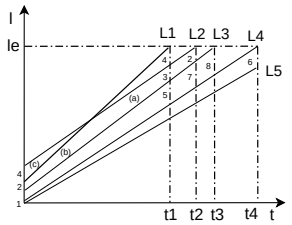


Fig. 5. Grouping objects by their predicted functions

the group location to the server. For reselecting the central object, according to objects predicted future functions, we can choose the objects close to the center of the group during its lifetime as the candidates of the central object. We can also identify when the central object will move away from the group center and choose another candidate as a new central object. A joining from a moving object to a group must be executed as follows. The system first finds the nearby groups according to the edge the object lies and then compares the movement similarity of the object and the group by their predicted functions. If the object cannot join to the nearby groups, a new group will be created with only one member. When a moving object leaves a group, the central object of the group needs to be reselected. However, for the object leaving an edge, to reduce the central object reselection of its group, we just delete it from its group and do not change the central object until the central object leaves the edge.

In the GSP, the grouping method assures the compactness and movement similarity of the objects within a group. Given the precision threshold ε, the objects locations in a group may be approximated by the location of the group (i.e. location of its central object). Only the location update from the central object of the group to the location server is necessary. After the server makes predictions for objects in a road network and initiates their groups, the client of the central object measures and monitors the deviation between its current location and predicted location and reports its location to the server. Other objects do not report their locations unless they enter the new edge. The prediction and grouping of objects are executed in the server and the group information (including the edge id, the central object id, its predicted function and a set of objects within the group) is also stored in the database of the server. The update algorithm in the server is described in Algorithm 1.

5 Performance Evaluation

In this section, we experimentally measure the performance of the *point-based*, *segment-based* [1], and our *GSP* update policies. We also evaluate the simulation based prediction (SP) method used in the *GSP* update policy with the simulation parameter P_d and prediction accuracy compared to the linear prediction (LP) method. We implemented the three update policies in Java and carried out experiments on a Pentium 4, 2.4G PC with 256MB RAM running Windows XP.

Algorithm 1: GroupUpdate($objID, pos, vel, edgeID, grpID$)

input : $objID, edgeID$ and $grpID$ are respectively the identifier of the object to be updated, its edge and group, pos, vel are its position and velocity
Simulate two future trajectories of $objID$ with different P_d by the CA;
Compute the future predicted function $l(t)$ of $objID$;
if *$objID$ does not enter the new edge* **then**
  if *$objID$ is the central object of $grpID$* **then**
    Update the current position pos and predicted function $l(t)$ of $grpID$;
    Send the predicted function $l(t)$ of $grpID$ to the client of $objID$;
  end
else
  if $GetObjNum(grpID) > 1$ **then**
    Deletes $objID$ from its original group $grpID$;
    if *$objID$ is the central object of $grpID$* **then**
      Reselect the central object of $grpID$, update and send its group info;
    end
  else Dissolve the group $grpID$;
  Find the nearest group grp_1 for $objID$ on $edgeID$;
  Compute the time t_e when $objID$ leaves $edgeID$ by $l(t)$ and $edgeID$ length;
  if *Both distances between $objID$ and grp_1 at initiate time and $t_e \leq \varepsilon$* **then**
    Insert $objID$ into grp_1 and send grp_1 identifier to the client of $objID$;
    Reselect the central object of grp_1, update and send its group info;
  else Create a new group grp_2 only having $objID$ and send its group info;
end

5.1 Datasets

The datasets of our experiments are generated by Thomas Brinkhoff Network-based Generator of Moving Objects [13], which is used as a popular benchmark in many related work. The generator takes a map of a real road network as input and may simulate the moving behaviors of various kinds of moving objects in real world. Our experiment is based on the real map of Oldenburg city with 7035 segments. For modeling the road network, we associate those adjacent but not crossed segments together to form edges of the graph. After that, the total number of edges is 2980 and their average length is 184. We set the generator the parameter "maximum time" to be 20, "maximum speed" 50 and the number of initial moving objects 100000. The generator places these objects at random positions on the road network, and updates their locations at each time-stamp. The positions of the objects are given in two dimensional X-Y coordinates. We transform them to the form of (`edgeid`, `pos`), where `edgeid` denotes the edge identifier and `pos` denotes the object relative position.

5.2 Update Performance

For evaluating update performance and accuracy, we consider two metrics, namely, the number of updates (for 100000 moving objects during 20 time-stamps) and average error of the location of each object at each times-tamp as following.

$$average\_error = \frac{1}{mn} \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} |l_{ij} - l_{rij}| \tag{4}$$

where l_{ij} is the predicted location of mo_j or approximated location by its group at the timestamp t_i, l_{rij} is the real location of mo_j at timestamp t_i, m is total update time-stamps and n is the number of moving objects.

Figure 6 and 7 show the update number and average error of three update policies respectively with different update thresholds. We observe that with increase of the threshold, the update number will decrease and the average error will increase in any one of these three policies. This is because the larger the threshold is, the larger the allowable deviation between the predicted location and its real location, and the less updates it causes. However, the GSP update policy outperforms the other two policies for fewer number of update and average error. Specifically, the GSP only causes 30%-40% updates of segment-based policy and 15%-25% of point-based policy, while improves the location accuracy with lower average error. This owns to the accurate prediction of the SP method and the technique of grouping moving objects. For the GSP policy, larger threshold results in more objects in one group and therefore fewer group updates and higher location average error. In addition, notice that the largest performance improvement of the GSP policy over other policies is for smaller thresholds. For thresholds below 10, the GSP policy is nearly three times better than the segment-based policy and four times than the point-based policy.

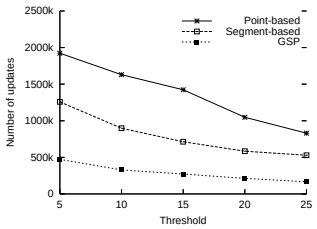


Fig. 6. Number of Updates

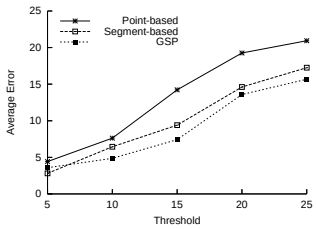


Fig. 7. Average Error of Updates

5.3 Prediction Performance

The Slowdown Rate P_d We study the effect of the choices of different P_d, which determines two predicted trajectories corresponding to the fastest and slowest movements. We use P_d from 0 to 0.5 and measure the prediction accuracy by the average error and overflow rate. The overflow rate represents the probability of the predicted positions exceeding the actual positions. The purpose of this metric is to find the closest two trajectories binding the actual one as future trajectories. In this way, we choose the P_d with both the lower average error and overflow

rate, which can also be treated as one of methods to set the proper values of P_d in a given dataset. Figure 8 and Figure 9 show the prediction accuracy of the SP method with different P_d. We can see that when P_d is set to 0 and 0.1, both the average error and overflow rate are lower than others. Therefore, we use them in the experiments to obtain better prediction results.

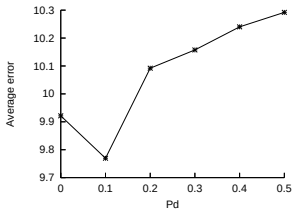


Fig. 8. Average Error with Different P_d

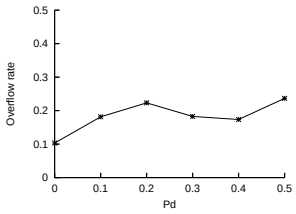


Fig. 9. Overflow Rate with Different P_d

Prediction Accuracy and Cost Finally, we compare the prediction accuracy of the SP method with the LP method. We measure the average error for predicted locations (without grouping) with different thresholds. From Figure 10, we observe that the average error will increase when the threshold increases. This is tenable in both the LP and SP method. However, the SP method predicts more accurately than the LP method with any threshold. For the costs of SP method, as its time complexity depends on many factors, we compute average CPU time when simulating and predicting the movements of one object along the edge with length 1000. The results show that the average cost of one prediction is about 0.25ms. This is acceptable even for large number of moving objects.

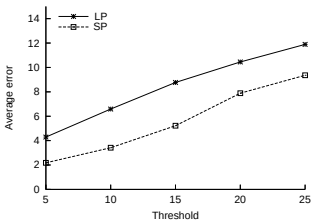


Fig. 10. Comparison of Prediction Accuracy

6 Conclusion

Motivated by the features of vehicles movements in traffic networks, this paper presents new techniques to track network-constrained moving objects. Our contribution is twofold. First we propose a prediction model, based on simulation,

which predicts with a great accuracy the future trajectories of moving objects. This lowers location update frequency in tracking. Then, based on the prediction, we propose a group update strategy which further reduces location updates and minimizes the cost of wireless communication. The experiments show that the update strategy has much higher performance and location accuracy.

Acknowledgments

This work was partially supported by the grants from the Natural Science Foundation of China with grant number 60573091, 60273018; China National Basic Research and Development Program's Semantic Grid Project (No.2003CB317000); the Key Project of Ministry of Education of China under Grant No.03044; Program for New Century Excellent Talents in University(NCET); Program for Creative PhD Thesis in University.

References

1. A. Civilis, C. S. Jensen, J. Nenortaite, S. Pakalnis. Efficient Tracking of Moving Objects with Precision Guarantees. In MobiQuitous 2004: 164-173.
2. A. Civilis, C. S. Jensen, S. Pakalnis. Techniques for Efficient Road-Network-Based Tracking of Moving Objects. In IEEE Trans. Knowl. Data Eng. 17(5): 698-712 (2005).
3. Z. Ding, R. H. Guting. Managing Moving Objects on Dynamic Transportation Networks. In SSDBM 2004: 287-296.
4. H. Gowrisankar, S. Nittel. Reducing Uncertainty In Location Prediction Of Moving Objects In Road Networks. In GIScience 2002: 228-242.
5. Y. Huh, C. Kim. Group-Based Location Management Scheme in Personal Communication Networks. In ICOIN 2002: 81-90.
6. G. H. K. Lam, H. V. Leong, S. C. Chan. GBL: Group-Based Location Updating in Mobile Environment. In DASFAA 2004: 762-774.
7. K. Y. Lam, O. Ulusoy, T. S. H. Lee, E. Chan, and G. Li, An Efficient Method for Generating Location Updates for Processing of Location-Dependent Continuous Queries. In DASFAA 2001: 218-225.
8. G. Trajcevski, O. Wolfson, B. Xu, Peter Nelson: Real-Time Traffic Updates in Moving Objects Databases. In DEXA 2002: 698-704.
9. O. Wolfson, A. P. Sistla, S. Camberlain, Y. Yesha. Updating and Querying Databases that Track Mobile Units. In Distributed and Parallel Databases 7(3): 257-387 (1999).
10. O. Wolfson and H. Yin. Accuracy and Resource Consumption in Tracking and Location Prediction. In SSTD 2003: 325-343.
11. J. Zhou, H. V. Leong, Q. Lu, K. C. Lee. Aqua: An Adaptive QUery-Aware Location Updating Scheme for Mobile Objects. In DASFAA 2005: 612-624.
12. K. Nagel and M. Schreckenberg, A Cellular Automaton Model for Free Traffic, In physique I, 1992, 2: 2221-2229.
13. T. Brinkhoff. A Framework for Generating Network-based Moving Objects, In GeoInformatica 6(2): 153-180 (2002).

In Proceedings of the 7th International Conference on Mobile Data Management (MDM 2006),
Nara, Japan, May 9-13, 2006. IEEE Computer Society 2006: 156

Modeling and Predicting Future Trajectories of Moving Objects in a Constrained Network

Jidong Chen† Xiaofeng Meng† Yanyan Guo† Stéphane Grumbach‡ Hui Sun†
†Information School, Renmin University of China, Beijing, China
{chenjd, xfmeng, guoyy, hsun}@ruc.edu.cn
‡CNRS, LIAMA, Beijing, China
grumbach@liama.ia.ac.cn

Abstract

Advances in wireless sensor networks and positioning technologies enable traffic management (e.g. routing traffic) that uses real-time data monitored by GPS-enabled cars. Location management has become an enabling technology in such application. The location modeling and trajectory prediction of moving objects are the fundamental components of location management in mobile location-aware applications. In this paper, we model the road network and moving objects in a graph of cellular automata (GCA), which makes full use of the constraints of the network and the stochastic behavior of the traffic. A simulation-based method based on graphs of cellular automata is proposed to predict future trajectories. Our technique strongly differs from the linear prediction method, which has low prediction accuracy and requires frequent updates when applied to real traffic with velocity changes. The experiments, carried on two different datasets, show that the simulation-based prediction method provides higher accuracy than the linear prediction method.

1 Introduction

The continued advances in wireless sensor networks and position technologies enable traffic management and location-based services that track continuously changing positions of moving objects. For example, moving cars on a road network can be monitored and their locations are sampled by sensors or GPS periodically, then sent to the server and stored in a database. According to the real-time locations and predicted future trajectories of cars, we can forecast traffic jams and route the traffic intelligently. Timely location information is becoming one of the key features in these applications. In this paper, we focus on the the location modeling and future trajectory prediction of moving objects, which are the foundations for efficient location management in mobile location-aware applications.

Many models and algorithms have been proposed to handle the continuously changing positions of moving objects. Wolfson et al. in [16, 21] firstly proposed a Moving Objects Spatio-Temporal (MOST) model, which represents the location as a dynamic attribute. Later, the model based on linear constrain [17], abstract data types [9] and Space-Time Grid Storage [4] for moving objects have been proposed. However, in most real life applications, objects move within constrained networks, especially the transportation networks (e.g., vehicles move on road networks). These works ignore the interaction between moving objects and the underlying transportation networks.

In fact, the interaction is very important to manage network-constrained moving objects. For example, in the location tracking, the road-network representation of moving objects can be exploited to reduce the number of updates from moving objects to the database server [5]. For indexing moving objects in road networks, the temporal aspect can be distinguished and related to the road network to save considerable index storage space [2, 8] since the spatial property of objects' movement is already captured by the network. In addition, using the network constraints, the query processing can also be improved [10, 15].

More recently, the models connecting moving objects with the road network representation have been proposed [7, 13, 14, 20]. Most of them represent road networks as graphs and moving objects as moving graph points with their speed in order to capture objects' movement. However, the models assume linear movement and can not reflect the real movement feature of moving objects in a road network where objects frequently change their velocity. This limits their applicability in a majority of real applications.

In this paper, we propose a new graph of cellular automata (GCA) model to integrate the traffic movement fea-

tures into the model of moving objects and the underlying road network. The GCA model exploits the stochastic behavior of the real traffic by the cellular automaton which is used in the traffic simulation [12]. It also combines the road network model with the real movement of objects and therefore improves the efficiency of managing network-constrained moving objects.

Considering the new feature of the GCA model, it can be efficiently used to simulate future trajectories of moving objects, where objects' movement follows traffic rules. We further propose a simulation-based prediction method based on the GCA model. Since the GCA exploits features of traffic systems, the method can predict future trajectories of moving objects in road network more accurately than the linear prediction method widely used in the predictive indexing and query processing.

The framework built on the GCA model and simulation-based prediction forms the foundation of the efficient storage and management of network-constrained moving objects. Specifically, it is capable of reducing the number of updates in tracking and indexing and supporting the predictive queries on moving objects in a road network.

In summary, this paper makes the following contributions:

1. We present the graphs of cellular automata (GCA) model to integrate the trajectory representation of moving objects and the transportation network with intrinsic movement features in the real traffic.

2. Based on the GCA model, we propose a simulation-based prediction (SP) method, which improves the accuracy of predicting future trajectories of objects moving in a traffic network.

3. The experiments show that the simulation-based prediction method obtains higher accuracy than the linear prediction method widely used.

The rest of the paper is organized as follows. Section 2 surveys related work. In Section 3, the graph of cellular automata model is introduced to model the road network and movement of objects. Section 4 presents the simulation-based prediction method. Section 5 contains experimental evaluation. We conclude in Section 6.

2 Related Work

The modeling of moving objects attracts a lot of research interests. Wolfson et al. in [16, 21] firstly proposed a Moving Objects Spatio-Temporal (MOST) model which is capable of tracking not only the current, but also the near future position of moving objects. Su et al. in [17] presented a data model for moving objects based on linear constraint databases. Chon et al. in [4] proposed a Space-Time Grid Storage model for moving objects. In [9], Güting et al. presented a data model and data structures for moving objects based on abstract data types. However, nearly none of these works have treated the interaction between moving objects and the underlying transportation networks in any way.

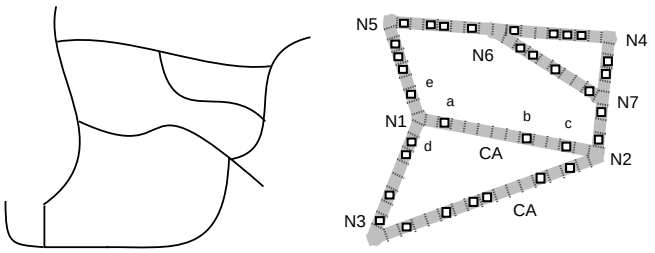
In 2001, Vazirgiannis and Wolfson [20] first introduced a model for moving objects on road networks, which connects the moving object's trajectory model with the road network representation. In the model, the road network is represented by an electronic map and the trajectory of a moving object is constructed by the map and its destination. In [14], the authors presented a computational data model for network constrained moving objects in which the road network has two representations namely a two-dimensional representation and a graph representation to obtain both expressiveness and efficient support for queries. In this model, the moving objects treated as query points are represented by graph points located on segments or edges. Ding et al. [7] proposed a MOD model, based on dynamic transportation networks. They model transportation networks as dynamic graphs and moving objects as moving graph points. In addition, Papadias et al. in [13] presented a framework to support spatial network databases. However, these models capture movement information of objects only by their speed and assume the linear movement, which limit applicability in a majority of real applications.

Prediction methods for future trajectories of moving objects play an important role in indexing and querying current and anticipated future positions. Most existing prediction methods, used in the indexing and querying, assume linear movement, which cannot reflect the real movement. Aggarwal et al [1] introduced a non-linear model that uses quadratic predictive function, Tao et al [18] proposed a prediction method based on recursive motion functions for objects with unknown motion patterns, and Cai et al [6] used Chebyshev polynomials to represent and index spatio-temporal trajectories. In [19], Tao et al developed Venn sampling (VS), a novel estimation method optimized for a set of pivot queries that reflect the distribution of actual ones. These prediction methods improve the precision in predicting the location of each object, but they ignore the correlation of adjacent objects when they move in traffic networks, and thus may not reflect the realistic traffic scenario.

Despite the wide use of traffic simulation rules in transportation GIS domain [12, 3], their integration to a database model for objects in constrained networks has never been done before.

3 Graphs of Cellular Automata Model

We model a road network with a graph of cellular automata (GCA), where the nodes of the graph represent road intersections and the edges represent road segments with



(a) A Road Network (b) An instance of GCA

Figure 1: An example of a road network and its GCA model

no intersections. Different from the general graph model, each edge in the GCA consists of a cellular automaton (CA), which is represented, in a discrete mode, as a finite sequence of cells. Each cell corresponds in practice to some road segment of about 7.5 m.

Figure 1 shows an example of a road network and its GCA model. Each node has a label which represents an intersection of the road network. The wide lines represent edges and each edge treated as one CA connects many cells.

The CA model was used in this context by [12]. We first recall the definition of cellular automaton.

Definition 1 *A cellular automaton consists of a finite oriented sequence of cells. In a configuration, each cell is either empty or contains a symbol. During a transition, symbols can move forward to subsequent cells, symbols can leave the CA and new symbols can enter the CA.*

An example of cellular automaton corresponding to edge (N_1, N_2) in Figure 1(b) with a transition between two configurations is given in Figure 2. We now formally define a graph of cellular automata.

Definition 2 *The structure of a GCA is a directed weighted graph $G = (V, E, l)$ where V is a set of vertices (i.e., nodes), E is a set of edges and $l : E \rightarrow \mathbb{N}$ is a function which associates to each edge the number of cells of the corresponding cellular automaton.*

We assume a countably infinite alphabet $\Omega : \{\alpha, \beta, \gamma, \cdots\}$, denoting moving objects' names. Let C be the set of cells of a GCA.

A configuration or an instance of a GCA, is a mapping from the cells of the GCA to constants in Ω together with a given velocity. Intuitively, the velocity is the number of cells an object can traverse during a time unit.

Definition 3 *An instance I of a GCA is defined by two functions:*

$\mu : C \rightarrow \Omega \bigcup \{\varepsilon\}$ *(1-1 mapping)*
$v : \Omega \rightarrow \mathbb{N}$.

A moving object is represented as a symbol attached to the cell in the GCA and it can move several cells ahead at each time unit. Figure 1(b) is an instance of the GCA corresponding to the road network of Figure 1(a). In Figure 1(b), moving objects are denoted by squares. A moving object lies on exactly one cell of the edge and its location can be obtained by computing the number of cells relative to the start node. For instance, object α lies on the edge (N_1, N_2) and there are two cells away from N_1 along the edge. Therefore, its position can be expressed by $(N_1, N_2, 2)$.

The motion of an object is represented as some (time, location) information. Representing such information of a moving object as a trajectory is a typical approach [20]. In the GCA model, the trajectory of a moving object can be divided two types: the in-edge trajectory for the object's movement in one edge (CA) and the global trajectory for the object that may move cross several edges (CAs) during its movement. The in-edge trajectory of an object is a polyline in two-dimensional space (one-dimensional relative distance, plus time), which can be defined as follows:

Definition 4 *The in-edge trajectory of a moving object in a CA of length L is a piece-wise function $f : T \rightarrow \mathbb{N}$, represented as a sequence of points $(t_1, l_1), (t_2, l_2), \ldots, (t_n, l_n)(t_1 < t_2 < \ldots < t_n, l_1 < l_2 < \ldots < l_n \leq L)$.*

When an object moves across multiple edges, its global trajectory is defined as functions mapping the time to the edge and relative distance.

Definition 5 *The global trajectory of a moving object in different CAs is a piece-wise function $f : T \rightarrow (E, \mathbb{N})$, represented as a sequence of points $(t_1, e_1, l_1), \ldots, (t_i, e_j, l_k), \ldots, (t_z, e_m, l_n)(t_1 < t_2 < \ldots < t_z)$.*

In the sequel, we will be interested by deterministic paths in the GCA i.e., path with source nodes of out degree 1. The successive CAs in a deterministic path can be then seen as a unique CA.

Let i be an object moving along an edge. Let $v(i)$ be its velocity, $x(i)$ its position, $gap(i)$ the number of empty cells ahead (forward gap), and $P_d(i)$ a randomized slowdown rate which specifies the probability it slows down. We assume that V_{max} is the maximum velocity of moving objects. The position and velocity of each object might change at each transition as shown definition 6 adapted from [12].

Definition 6 *At each transition of the GCA, each object changes velocity and position in a CA of length L according to the rules below:*

1. *if* $v(i) < V_{max}$ *and* $v(i) < gap(i)$ *then* $v(i) \leftarrow v(i)+1$
2. *if* $v(i) > gap(i)$ *then* $v(i) \leftarrow gap(i)$

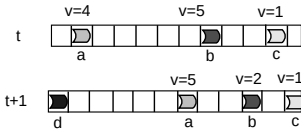


Figure 2: Transition of the GCA

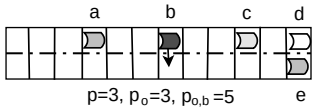


Figure 3: An example of changing lane in transition of the two lane GCA

3. *if* $v(i) > 0$ *and* $rand() < P_d(i)$ *then* $v(i) \leftarrow v(i) - 1$

4. *if* $(x(i) + v(i)) \leq L$ *then* $x(i) \leftarrow x(i) + v(i)$

The first rule represents linear acceleration until the object reaches the maximum speed V_{max}. The second rule ensures that if there is another object in front of the current object, it will slow down in order to avoid collision. In the third rule, the $P_d(i)$ models erratic movement behavior. Finally, the new position of object i is given by the fourth rule as the sum of the previous position with the new velocity if it is in the CA. Note that it is easy to extend the definition of transition to deterministic paths. Because of deterministic path, the objects move to a new position in a subsequent CA. Figure 2 shows the simulated movement of objects on a cellular automaton of the GCA in two consecutive timestamps. We can see that at time t, the speed of the object a is smaller than the gap (i.e. the number of cells between the object a and b). On the other hand, the object b will reduce its speed to the size of the gap. According to the fourth rule, the objects move to the corresponding positions based on their speeds at time $t + 1$.

However, objects in real traffic have different desired speed. With the transitions of the GCA of one lane CA mentioned above, it can be found that slow objects being followed by faster ones, and the average speed reduced to the free-flow speed of the slowest object [11]. In view of this, we extend the one lane GCA to two lane GCA in which a CA consists of two parallel single lane. Therefore, each cell in two lane GCA is composed of two parallel single lane and each lane may contain one symbol namely a moving object. The function μ in a GCA instance I will change to the 1-2 mapping accordingly.

For the transition of GCA with one lane, we extend it to the two lane by attaching an additional rule that models the changing of lanes of the object. Suppose the objects move only sideways, the transition of GCA happens on both lanes according to the previous four rules and then the exchange of objects between two lanes is checked according to the additional conditions for changing lane as follows:

5. *object* i *changes lane with probability* P_c *if*

$$gap(i) < p, gap_o(i) > p_1, \text{ and } gap_{o,b}(i) > p_{o,b}$$

where $gap(i)$ is the number of empty cells ahead in the same lane, $gap_o(i)$ is the forward gap on the other lane, $gap_{o,b}(i)$ is the backward gap on the other lane, p, p_o and $p_{o,b}$ are the parameters which decide how far the object looks ahead on the current lane, ahead on the other lane, and back on the other lane, respectively.

In fact, the changing lane rule is based on the following observation: the car looks ahead if some car is in its way; the car looks on the other lane if it is any better there; the car looks back on the other lane if it would get in other cars way. Generally, in the above rule, both p and p_o are essentially proportional to the velocity, whereas looking back depends mostly on the expected velocity of other objects, not on one's own. An example of invoking the rule of changing lane with $p = v + 1, p_o = p, p_{o,b} = v_{max}, P_c = 1$ is given in Figure 3. The object b with $p = 3, p_o = 3, p_{o,b} = 5$ changes to the other lane in the GCA due to satisfying the fifth rule mentioned above.

4 Trajectory Prediction

In the management of moving objects, the trajectory prediction method is usually used to improve the performance of the location update strategy and to support the predictive index and queries. In this part, we first review some linear prediction methods and analyze their problem in handling moving objects in constrained networks, and finally present our simulation-based prediction method.

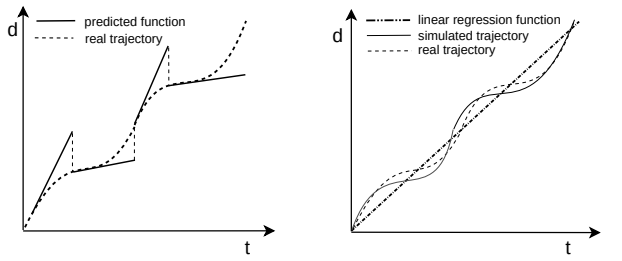
4.1 The Linear Prediction (LP)

Most current index and query processing approaches use the linear prediction method for its simplicity and capability of approximating any curve of free movement by piecewise linear segments. Suppose the trajectory function for an object between time t_0 and t_1 is

$$\vec{X}_t = \vec{X}_{t_0} + \vec{V}(t - t_0) \quad (t_0 \leq t \leq t_1)$$

where $\vec{X}_{t_0}$ denotes the position of the object at time t_0 and $\vec{V}$ denotes the velocity of the object, which is assumed to remain fixed between t_0 to t_1.

General LP The general linear prediction method uses the object's current position $\vec{X}_{t_0}$ and current velocity $\vec{V}$ to predict its position in the near future. When the prediction is deemed inaccurate, that is, its deviation from the actual position is beyond a predefined threshold, we revise our prediction by resetting $\vec{X}_{t_0}$ and $\vec{V}$. In situations where object's velocity remains largely constant, this method enables us to



(a) General Linear Prediction (b) Simulation-based Prediction

Figure 4: Linear Prediction VS. Simulation-based Prediction

make future prediction with high precision. However, when objects move with changing velocity, their trajectory functions have to be revised frequently.

Road Segment Based LP If objects move in a constrained environment such as a transportation network, we can use the road segments of the network to help model the object's movement. In other words, we assume objects move at constant speed along a road segment, that is, their trajectory functions will not change until they move out of a road segment. When an object enters a new road segment, we reset the velocity $\vec{V}$ in its trajectory function. The frequency of revising the trajectory function depends on the average length of the road segments.

Route Based LP If objects have regular and known routes in the transportation network (e.g., one takes the same route from home to work), we can use the routes instead of the road segments to reduce the number of updates needed to maintain the objects' position. If the route is predicted incorrectly, we simply make an additional update.

However, any real traffic system has a stochastic, dynamic and fuzzy nature. The accuracy of linear prediction methods mentioned above is inadequate because linear methods can hardly reflect the movement of objects constrained by road networks. For example, in urban road networks, because of traffic conditions, a vehicle may travel at a constant speed, decelerate to stop, wait, accelerate and travel again at a constant speed. Vehicles may often repeat the above movement in modern urban road networks.

We use Figure 4 to demonstrate the inadequacy of the linear prediction method for real road networks. Figure 4(a) shows the predicted (linear) trajectory and the actual trajectory of an object. We can see that each time the change of the object's velocity is above a certain threshold, an update is triggered and the trajectory is revised by a new velocity vector. The frequent changes of the object's velocity will incur repeated update and prediction.

4.2 The Simulation-based Prediction (SP)

Considering the simulation feature of the GCA model, we use GCAs not only to model road networks, but also to simulate future trajectories of moving objects by the transitions of GCAs, where objects' movement follows traffic rules. Based on the GCA, a *Simulation-based Prediction (SP)* method to anticipate future trajectories of moving objects is proposed. The SP method treats the object's simulated results as its predicted positions to obtain its future in-edge trajectory. To refine the accuracy, based on different assumptions on the traffic conditions we simulate two future trajectories in discrete points for each object on its edge. Then, by linear regression and translating, the trajectory bounds that contain all possible future positions of a moving object on that edge can be obtained. When the object moves to another edge in the GCA or the predicted position exceeds its actual position above the predefined accuracy, another simulation and regression will be executed to predict new future trajectory bounds. The process of the simulation-based prediction can be seen in Figure 5.

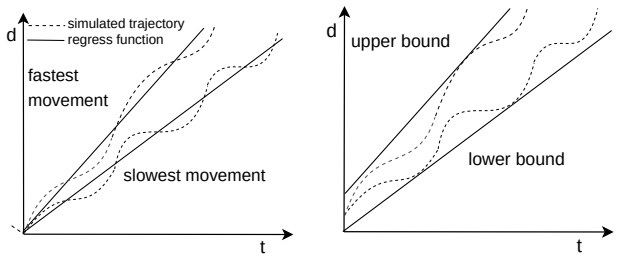
Most existing work uses the CA model for traffic flow simulation in which the parameter $P_d(i)$ is treated as a random variable to reflect the stochastic, dynamic nature of traffic system. However, we extend this model for predicting the future trajectories of objects by setting $P_d(i)$ to values that model different traffic conditions. For example, laminar traffic can be simulated with $P_d(i)$ set to 0 or a small value, and the congestion can be simulated with a larger $P_d(i)$. By giving $P_d(i)$ two values, we can derive two future trajectories, which describe, respectively, the fastest and slowest movements of objects as showed in Figure 5(a). In other words, the object's future locations are most probably bounded by these two trajectories. The value of $P_d(i)$ can be obtained by the experiences or by sampling from the given dataset. Our experiments show one of methods to choose the value of $P_d(i)$. It is proved that 0 and 0.1 are realistic values of $P_d(i)$ in our cases.

For getting the future trajectory function of an object from the simulated discrete points, we need to regress the discrete positions. We find that in most cases the linear regression (as shown in Figure 4b) fits the prediction well and at low cost. The OLSE (Ordinary Least Square Estimation) method, for example, can be calculated efficiently with low data storage cost. Let the discrete simulated points be $(t_1, d_1), \ldots, (t_i, d_i), \ldots, (t_n, d_n)$, where d_i ($i \in [1, n]$) denotes the relative distance in an edge and the average value be $\overline{t}$ and $\overline{d}$. After regression, the trajectory function of the moving object is:

$$D(t) = \hat{\beta}_0 + \hat{\beta}_1 \cdot t$$

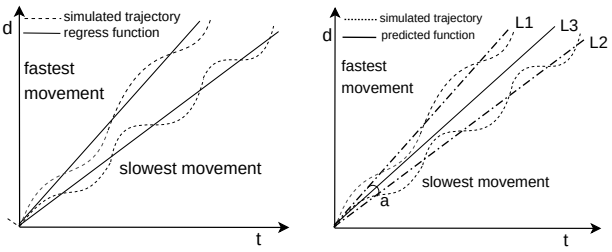
where $\hat{\beta}_o$ and $\hat{\beta}_1$ are given by:

$$\hat{\beta}_0 \quad = \quad \overline{d} - \hat{\beta}_1 \cdot \overline{t}$$



(a) Simulated trajectories (b) Two predicted bounds

Figure 5: Two Predicted Bounds of Future Trajectories



(a) Simulated trajectories (b) Single predicted function

Figure 6: Singe Predicted Future Trajectory

$$\hat{\beta}_1 \quad = \quad \frac{\sum_{i=1}^{n} t_i d_i - n\bar{t} \cdot \bar{d}}{\sum_{i=1}^{n} t_i^2 - n(\bar{t})^2}$$

In Figure 5(a), the dashed curves show two future trajectories, which are the slowest and the fastest movements simulated by using different P_d. Applying the OLSE algorithm to the two trajectories generates two linear functions, which are shown in solid lines.

$$\texttt{fastTrj}: \quad D(t) = \alpha_f \cdot t + \gamma_f$$
$$\texttt{slowTrj}: \quad D(t) = \alpha_s \cdot t + \gamma_s$$

Finally, in order to find the bounds of the area that contains all estimated future positions, we translate the two regression lines, until all estimated future positions fall within. More specifically, we translate the upper line (fastest movement) upwards until it touches the point with the max residual (denoting ϵ_1 the distance translated upward), and similarly, we translate the lower line (slowest movement) downwards (denoting ϵ_2 the distance translated downward). This minimizes the loss of information and errors brought by the OLSE algorithm.

We now define the two bound lines as the upper bound and lower bound of objects' future trajectory.

Definition 7 *The upper bound of an object trajectory* `upperBound` *is the upper bound line of its fastest future trajectory, and the lower bound* `lowerBound` *is the lower bound line of its slowest future trajectory. They are linear functions of the following form:*

$$\texttt{upperBound}: \quad D(t) = \alpha_f \cdot t + \lambda_f$$
$$\texttt{lowerBound}: \quad D(t) = \alpha_s \cdot t + \lambda_s$$

where $\lambda_f = \gamma_f + \epsilon_1, \lambda_s = \gamma_s - \epsilon_2$.

The two bound lines are shown in Figure 5(b). we can treat the two predicted lines as the bounds of the possible future positions of one object. The predicted trajectory bounds can be used in the predictive index structure and query processing in road network to reduce the index updates and filter unnecessary query results to improve the performance of predictive query. For example, given a predictive range query with the specified region R during time interval $[t_1, t_2]$ in the future, we can filter the objects in the result during the pre-process phase if the area between their upper and lower trajectory bounds can not intersect the R during $[t_1, t_2]$.

However, for other applications such as the tracking of moving objects, a single predicted function is needed to obtain the specific future positions of the object. For example, to lower update frequency from moving objects to server database, a general principle for location update policies is as follows: the moving objects equipped by GPS receiver do not report their locations to the server unless their actual positions exceed the predicted positions to a certain threshold. Their predicted positions need to be computed by a single predicted function. In this case, we can also adapt the SP method to obtain a compact and simple linear prediction function. The process can be seen in Figure 6. After regressing the two simulated future trajectories to two linear function denoting L_1 and L_2 , we compute the middle straight line L_3, the bisector of the angle a between L_1 and L_2 as the final predicted function $L(t)$.

Although the predicted function obtained by the SP method is a simple linear function, it is different from the linear prediction in that the SP method not only considers the speed and direction of each moving object, but also takes correlation of objects as well as the stochastic behavior of the traffic into account. The experimental results also show it is a more accurate and effective prediction approach.

As the prediction of in-edge trajectory only use the GCA to simulate the movement of objects in an edge, we have to consider the cases when objects move across the nodes in order to make the global trajectory prediction. If the out degree of a node in the GCA is one, the behavior of the object in the adjacent edge is the same. However, if the out degree of the node is bigger than one, we can not trace the objects cross the different edges. In this case, we could use the probability of objects changing the edges according to the historical data. In this paper, we only predict the in-edge trajectory of the object moving in one edge of the

GCA. When the object moves to another edge or its prediction accuracy of the future positions cannot arrive the given accuracy requirement, we issue another prediction based on the current traffic conditions.

5 Experimental Evaluation

We evaluate the simulation-based prediction method by comparing it with the general linear prediction method. Using two datasets (generated by the CA simulator and by the Brinkhoff's Network-based Generator [3]), we measure their prediction accuracy when applied to predict the near anticipated future positions in the real map network. We also study the effect of the choice of different values of the parameter P_d on the simulation-base prediction.

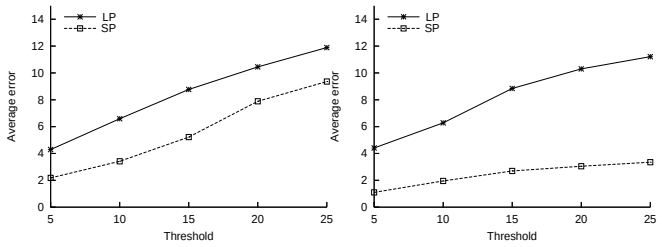
Datasets

We use two datasets for our experiments. The first is generated by the CA simulator, and the second by the Brinkhoff's Network-based Generator [3]. We use the CA traffic simulator to generate a given number of objects in a uniform network of size 10000×10000 consisting of 500 edges. Each object has its route and is initially placed at a random position on its route. The initial velocities of the objects follow a uniform random distribution in the range $[0, 30]$. The location and velocity of every object is updated at each time-stamp.

The Brinkhoff's Network-based Generator has been used as a popular benchmark in the related work of the MOD. The generator takes a map of a real road network as input (our experiment is based on the map of Oldenburg including 7035 edges). The positions of the objects are given in two dimensional X-Y coordinates. We transform them to the form of $(\texttt{edgeid}, \texttt{pos})$, where `edgeid` denotes the edge identifier and `pos` denotes relative position on the edge. The generator places a given number of objects at random positions on the road network, and updates their locations at each time-stamp. Each object has its own destination, and it moves toward its destination along a given route.

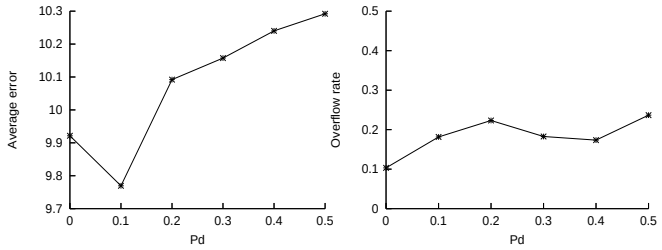
Prediction Accuracy and Cost

We compare the precision of the SP method with the LP method. We measure the prediction accuracy by "average error" but with different threshold. The threshold represents the maximum deviation between the predicted locations of a moving object and its real locations allowed in the prediction. That means when the deviation exceeds the threshold, we make another prediction. From Figure 8, we observe that average error will increase when threshold increases. This is because the larger the threshold is, the larger the deviation becomes, which leads to the more errors. This is tenable in both the LP and SP method. However, the SP method predicts more accurately than the LP method with any threshold.



(a) Brinkhoff (b) CA

Figure 7: Prediction Accuracy with Different Threshold



(a) Average Error (b) Overflow Rate

Figure 8: Prediction Accuracy with Different P_d

The time complexity of the simulation-based prediction depends on many factors. We compute the average CPU time when simulating and predicting the movement of one object along the edge with length 1000 in different dataset sizes. The results show that the average cost of one simulation-based prediction is about 0.25ms. This is quite acceptable.

The Slowdown Rate P_d

The CA simulation has an important effect on the accuracy of the simulation-based prediction. We study the effect of the choice of different P_d, which determines the two predicted trajectories corresponding to the fastest and slowest movement. We test on the Brinkhoff dataset with different data size and use P_d from 0 to 0.5 and measure the average prediction accuracy by "average error" and "overflow rate". The average error is the average absolute error between the predicted and actual positions, and the overflow rate represents the probability of predicted positions exceeding the actual positions. The purpose of this metric is to find the closest two trajectories binding the actual one as future trajectories. In this way, we can choose the P_d both with lower average error and overflow rate. Figure 8 shows the prediction accuracy of the SP method with different slowdown rates. We can see that when P_d is set to 0 and 0.1, both the average error and overflow rate are lower than others. Therefore, we use the value 0 and 0.1 as slowdown rates for the fastest movement bound and the slowest movement bound to obtain better prediction results.

6 Conclusion

Managing moving objects in a constrained network is a challenging task as well as of great practical importance in mobile location-aware applications. It is necessary to represent and predict the future trajectories of moving objects more accurate. In this paper, we first combine road network representation and the movement model of objects in a traffic network to introduce a new model - GCA for network-constrained moving objects. And then we propose a prediction method, based on the GCA, which predicts with a great accuracy the future trajectories of moving objects. The accuracy results from the fact that the GCA model exploits the constraints of the network and models the stochastic aspect of urban traffic. Our experimental results performed on two datasets show that the prediction accuracy of our simulation-based prediction is higher than the linear prediction used in the predictive indexing and query processing.

Acknowledgment

This research was partially supported by the grants from the Natural Science Foundation of China under grant number 60573091, 60273018; the Key Project of Ministry of Education of China under Grant No.03044; Program for New Century Excellent Talents in University (NCET); Program for Creative PhD Thesis in University and performed in the framework of a joint project with INRIA. The authors would like to thank Zhen Xiao, Benzhao Li from Information school, Renmin University of China for the assistance of experimental evaluation and Karine Zeitouni from PRISM, Versailles Saint-Quentin University in France for many helpful advice and assistance.

References

[1] C. Aggarwal, D. Agrawal. On Nearest Neighbor Indexing of Nonlinear Trajectories. In PODS, 2003, 252-259.

[2] V. T. Almeida, R. H. Güting. Indexing the Trajectories of Moving Objects in Networks (Extended Abstract). In SSDBM, 2004, 115-118.

[3] T. Brinkhof. A framework for generating network-based moving objects. In GeoInformatica, 6(2), 2002, 153-180.

[4] H. D. Chon, D. Agrawal, A. E. Abbadi. Using Space-Time Grid for Efficient Management of Moving Objects. In MobiDE 2001, 59-65.

[5] A. Civilis, C. S. Jensen, S. Pakalnis. Techniques for Efficient Road-Network-Based Tracking of Moving Objects. In IEEE Trans. Knowl. Data Eng. 17(5): 698-712 (2005).

[6] Y. Cai, N. Raymond. Indexing spatiotemporal trajectories with chebyshev polynomials. In SIGMOD, 2004, 599-610.

[7] Z. Ding, R. H. Güting. Managing Moving Objects on Dynamic Transportation Networks. In SSDBM, 2004, 287-296.

[8] E. Frentzos. Indexing objects moving on Fixed networks. In SSTD, 2003, 289-305.

[9] R. H. Güting, M. H. Böhlen , M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. In TODS 25(1), 1-42(2000).

[10] M. Kolahdouzan, C. Shahabi. Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases. In VLDB 2004, 840-851.

[11] T. Nagatani. Bunching of cars in asymmetric exclusion models for freeway traffic, Phys. Rev. E 51(N2), 922 (1995).

[12] K. Nagel, M. Schreckenberg. A cellular automaton model for freeway traffic. Journal Physique I 2, 1992, 2221-2229.

[13] D. Papadias, J. Zhang, N. Mamoulis, Y. Tao. Query Processing in Spatial Network Databases. In VLDB, 2003, 790-801.

[14] L. Speicys, C. S. Jensen, A. Kligys. Computational Data Modeling for network-Constrained Moving Objects. In ACM-GIS, 2003, 118-125.

[15] C. Shababi, M.R. Kolahdouzan, M. Sharifzadeh. A Road Network Embedding Technique for K-Nearest Neighbor Search in Moving Objects Databases. In GeoInformatica 7(3), 2003, 255-273.

[16] P. Sistla, O. Wolfson, S. Chamberlain, S. Dao. Modeling and Querying Moving Objects. In ICDE 1997, 422-432.

[17] J. Su, H. Xu, O. Ibarra. Moving Objects: Logical Relationships and Queries. In SSTD 2001, 3-19.

[18] Y. Tao, C. Faloutsos, D. Papadias, B. Liu. Prediction and Indexing of Moving Objects with Unknown Motion Patterns. In SIGMOD, 2004, 611-622.

[19] Y. Tao, D. Papadias, J. Zhai, Q. Li. Venn Sampling: A Novel Prediction Technique for Moving Objects. In ICDE, 2005, 680-691.

[20] M. Vazirgiannis, O. Wolfson. A Spatiotemporal Model and Language for Moving Objects on Road Networks. In SSTD, 2001, 20-35.

[21] O. Wolfson, B. Xu, S. Chamberlain, L. Jiang. Moving Object Databases: Issues and Solutions. In SSDBM 1998, 111-122.

2006年学术交流活动

WAMDM 实验室 2006 年学术交流活动一览

一 孟小峰教授 2006 年担任学术职务

Steering Committee member, International Conference on Mobile Data Management(MDM)

Steering Committee member, International Conference on Web-Age Information Management(WAIM)

Publicity Co-Chair，The 25th ACM SIGMOD International Conference on Management of Data / Principles of Database Systems(SIGMOD2006), June 26-29, 2006, Chicago, Illinois, USA

Publicity Co-Chair ，The 8th International Conference on Web-Age Information Management(WAIM 2006), June 17-19, 2006, Hong Kong, China

Program Committee member, The 11th International Conference on Database Systems for Advanced Applications (DASFAA 2006), April 12-15, 2006, Singapore

Program Committee member, The 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006), April 9-12, 2006, Singapore

Program Committee member, The 8th Asia-Pacific Web Conference(APWeb2006), January 16-18, Harbin, China

Program Committee member, The Seventh International Conference on Web-Age Information Management, 17-19 June, 2006, Hong Kong, China
Program Committee member, The 5th International Conference on Web-based Learning (ICWL 2006), July 19-21, 2006, Penang, Malaysia

Program Committee member, The 7th International Conference on Web Information Systems Engineering(WISE2006), October 23-26, 2006, Wuhan, China

Program Committee member, ICDE The Second International Workshop on Databases for Next-Generation Researchers(SWOD2006),April 7, 2006, Atlanta, Georgia, USA

Program Committee member, 19th Australian Joint Conference on Artificial Intelligence 2006(AIDM)

二 孟小峰教授 2006 年学术交流及出访活动

2006年1月16-18， 哈尔滨，参加国际会议APWeb2006。

与会期间，孟小峰教授主持了会议分组报告

2006年3月1-5日　日本冲绳，参加日本数据库会议DEWS2006。
孟小峰教授作题为:RecipeCrawler: Collecting Recipe Data from WWW Incrementally的报告

2006年6月16-17日，香港，参加大中华数据库研究峰会。
孟小峰教授作题为Trend in Database Research in China的特邀报告

2006年6月17-19日，香港，参加国际会议WAIM2006。
孟小峰教授主持会议分组报告

2006年6月26日-30日　美国芝加哥，参加SIGMOD2006会议，WebDB2006。
孟小峰教授作题为Vision-based Web Data Records Extraction,分组报告

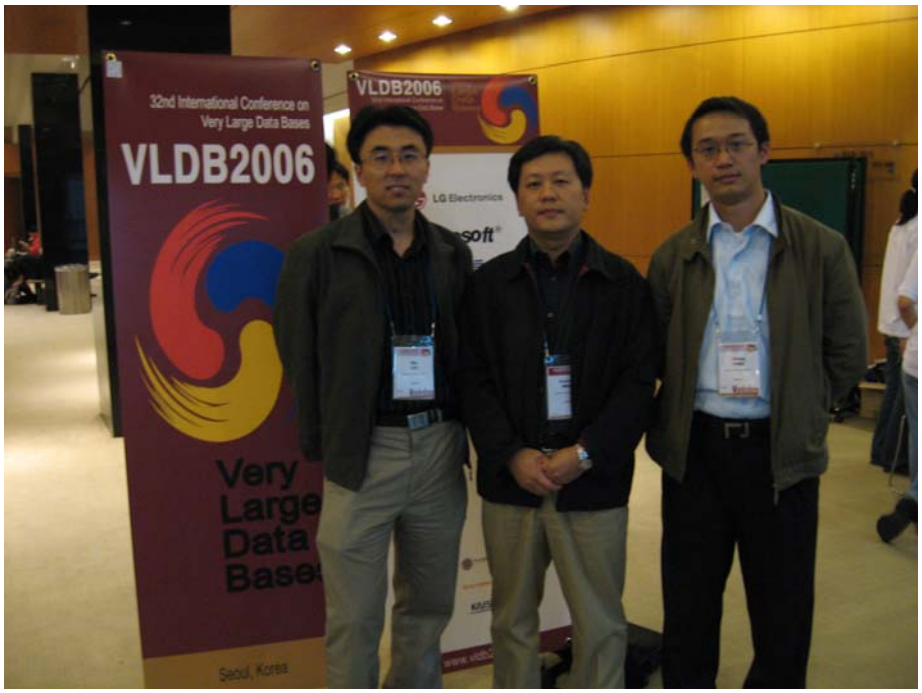
2006年7月1-13日，访问美国IBM T J Watson研究中心
访问期间，孟小峰教授作题为Web Database Integration,的报告

2006年7月20-22日，峨眉山，为教师进修班讲课“数据库课程教学研讨”

2006年7月23-28日，北京，参加2006年国家自然基金信息学部评审会议

2006年9月4日， 北京，参加First Asian Semantic Web Conference (ASWC2006)
孟小峰教授主持分组报告

2006年9月11-14日，韩国首尔，参加VLDB2006。
孟小峰教授和两位博士研究生参加了这次数据库盛会，与同行进行了交流，两位博士研究生就发表的workshop文章进行了报告和讨论。



2006年10月，北京，参加中国计算机学会王选奖的评审会议

2006年10月23-24日，武汉，参加国际会议WISE2006。
会上孟小峰教授主持分组报告

2006年11月1-3日，桂林，参加第二届SKG2006会议。
在The 2th International Conference on Semantics, Knowledge, and Grids(SKG2006)会议上，主持分组报告

2006年11月3-5日，南京，参加WISA2006。
主持特邀报告

2006年11月10-13日，广州，参加NDBC2006。
孟小峰教授担任会议秘书长，并主持分组报告

2006年11月19-22日，德国，参加Dagstuhl Seminar。
作题为OrientX: an Schema-Based Native XML Database System的报告

2006 年 11 月 19 日至 22 日，孟小峰教授应邀参加了在德国举办的“Dagstuhl Seminar on XQuery Implementation Paradigms”。

此次研讨会是关于 W3C XQuery 查询语言实现技术。所有参加者都是是受邀参加，且在本领域研究活跃。过去五年孟小峰教授领导的研究小组自主开发了 Native XML 数据库系统 OrientX，在本领域内得到广泛关注。会议主席德国慕尼黑理工大学的 Torsten Grust 教授在发来的邀请中特别指出”Your native XML database system OrientX is clearly recognized as a highly significant contribution in this research area and the seminar organizers are looking forward to your attendance(你们的 Native XML 数据库系统 OrientX 在本研究领域被认为具有突出的贡献，会议组织方希望你的参加)”。共有 36 位来自世界各地的 XQuery 查询语言实现的研究高手参加了本次会议，孟小峰教授是唯一来自亚洲的参加者。

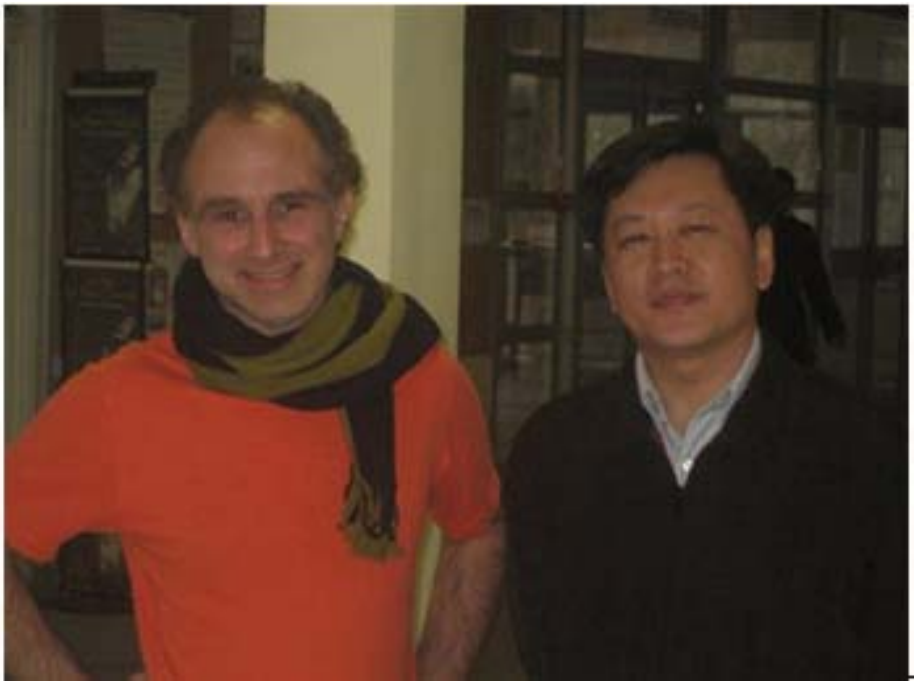
会议两天半的时间，但效率非常高，首先大家分别介绍各自的研究成果，然后分组讨论本领域的热点问题。会议讨论问题深入，具有一定的前瞻性，澄清了本领域的许多关键问题。孟小峰教授感到直接参加这种小范围高水平的研讨对提升创新性研究非常有必要，但前提是必须持之以恒地做出让世人关注的研究成果，这样才有讨论的资本。

Dagstuhl School 是专门面向计算机科学领域的学术研讨机构，常年举办一些 Seminar 或 Workshop。其规模一般不大，但水平较高，在计算机领域非常有影响。

与会人员合影。

2006年11月22-29日，法国巴黎，访问法尔赛大学和INRIA

孟小峰教授及博士生周军锋于 11 月 22 日至 29 日顺访了法国凡尔赛大学和法国 INRIA（法国乃至欧洲最大的信息技术研究机构）Philippe Pucheral 教授和 Karine Zeitouni 博士。在访法期间，双方就共同合作的题目“Moving Objects Management in Road Networks”展开了深入的讨论，孟小峰教授作了题为“Key Techniques of Moving Object Databases”的报告。这次在法访问期间，还幸遇纽约大学计算机系的 Dennis Shasha 教授，其《Database Tunning 》一书在数据库界非常有影响，孟小峰教授曾翻译了该书，双方进行了一个下午的深入交流，Dennis Shasha 教授接受了孟小峰教授的邀请，同意明年适当的时候来人大讲学。与 Dennis Shasha 教授合影



2006年11月29-12月9日，希腊雅典，访问雅典国立理工大学

依中希国际合作项目“基于 context 的 XML 数据管理研究”的需要，于 11 月 29 至 12 月 9 日访问希腊雅典国立理工大学 Timos Sellis 教授，他是著名的 R+树的发明人。11 月 30 日双方进行第一次会议，Timos Sellis 教授目前是雅典国立理工大学知识与数据库系统实验室的主任，他代表希腊方介绍该实验室的研究工作。孟小峰教授随后作了题为“Web and Mobile Data Management”的报告，介绍了人民大学及数据库方面的研究工作，并介绍国内数据库界的一些情况，引起合作方的极大兴趣。随后几天是深入的工作讨论，围绕“基于 context 的 XML 数据管理研究”这一课题进行了深入的交流。

这次工作访问取得圆满的成效，并确定了双方共同感兴趣的问题和今后的合作计划和人员互访计划。

2006 年 12 月 19 日，参加 863 信息技术领域部分专题战略研讨会

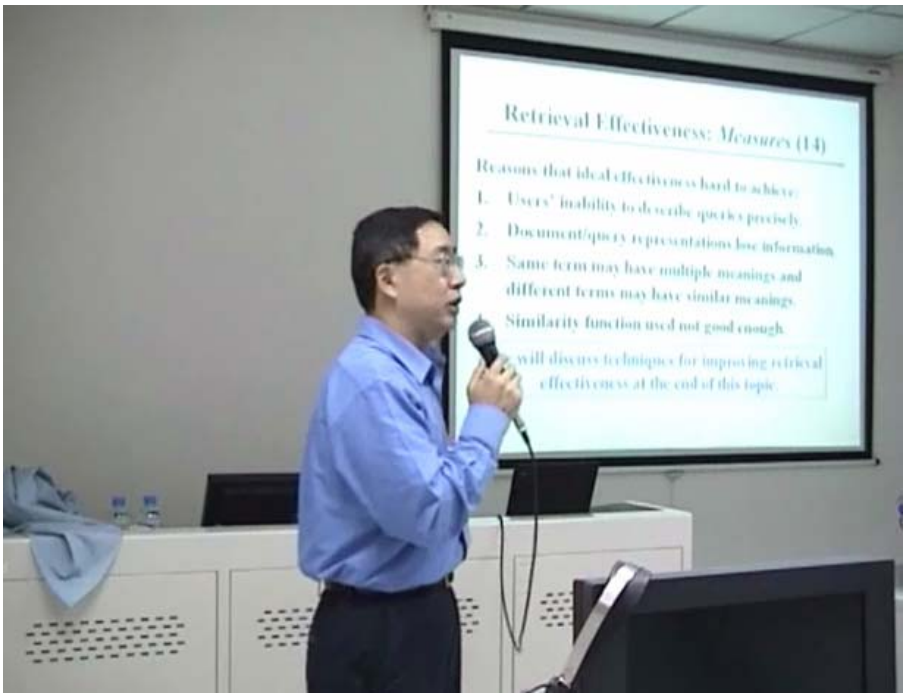
会议期间，孟小峰教授作题为“先进计算的数据管理问题”的发言

三 2006 年专家来访情况

2006.5.19-2006.5.25

孟卫一教授前来人民大学 WAMDM 实验室进行 Deep Web 数据集成方面的交流和合作。

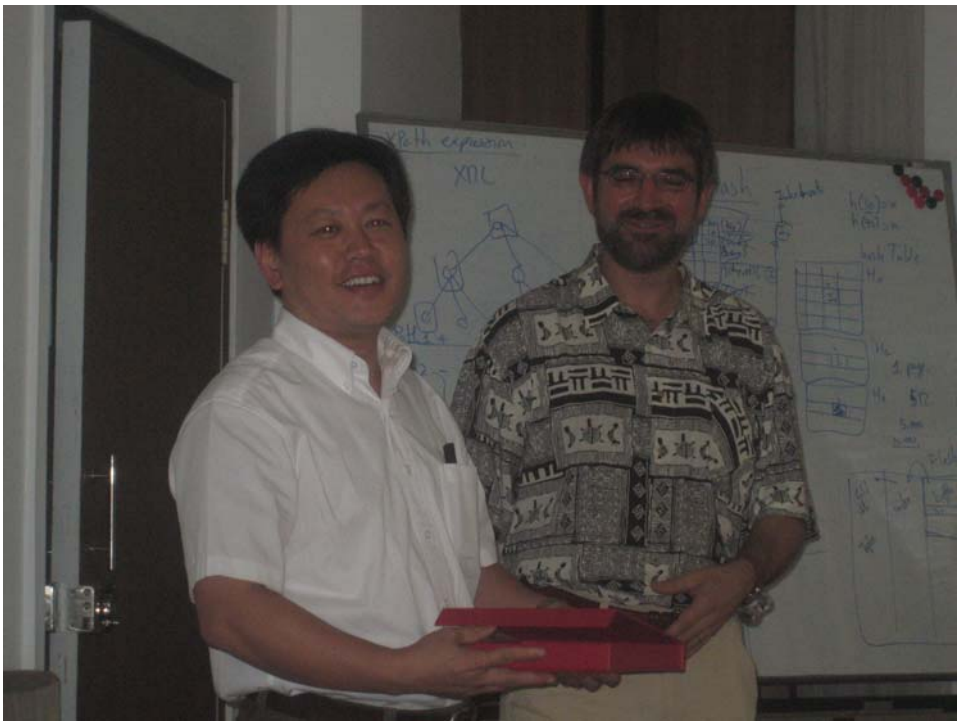
美国宾汉姆敦大学教授孟卫一受邀来本实验室进行了为期一周的学术交流。孟卫一教授多年来在理论研究方面一直与本实验室保持着密切的联系。交流期间，孟卫一教授在 Deep Web 数据集成研究领域与 Web 小组展开了深入的讨论。首先 Web 小组把近期的开展的研究工作进行了介绍，包括 Deep Web 数据集成环境下的实体识别问题、Web 数据库的选择问题、查询接口之间的查询转化问题等，孟卫一教授同时提出了许多中肯的建议。然后孟卫一教授又作了题为《Information Extraction from search Engine Returned Result Pages》的报告。



2006.9.14-2006.9.20

Philippe Pucheral 教授应邀来人民大学 WAMDM 实验室进行 flash DBMS 交流和合作。

这次交流在人民大学主办，为期一个星期。访问期间，Philippe 教授做了题为“SMIS Secured and Mobile Information Systems”的报告。并给实验室的 flash dbms 的研究工作给出了很多的指导，对研究的内容和方法都给予了大力的帮助。Philippe 教授分别和实验室的 WEB, Mobile 和 XML 组进行了交流，相互交流了各自对本领域的看法，开阔了大家的思维。



四 WAMDM 实验室研究生对外合作研究与交流情况

2006.3.1-2006.4.12

博士研究生陈继东、硕士研究生赖彩凤到香港浸会大学进行学术交流。

应香港浸会大学计算机系徐建良博士的邀请,两位研究生到港进行了为期一个半月的学术交流,主要是针对移动对象数据库领域中的关键技术,包括移动对象索引以及受限网络中移动对象聚类等技术进行深入的讨论和交流.

2006.6.15-2006.6.20

硕士研究生肖珍赴香港参加国际会议 WAIM。

会议期间，报告了 Mobile 小组的《路网上的移动对象基于组更新策略的追踪技术》的文章,与香港浸会大学徐建良博士就课题“位置服务中的隐私保护”进行讨论，初步建立了合作关系。

2006.15-2006.9.30

硕士研究生凌妍妍、李忺赴香港参加国际会议 WAIM。

参会期间，与香港城市大学就合作课题“营养食谱在线推荐与检索”进行讨论，推进课题进度，并确定了若干可以深入研究的详细议题。其中包括：食谱数据的模型简历，各类约束表示模型和管理方法，用户数据的管理和推荐策略等。

2006.9.10-2006.9.16

博士研究生刘伟、陈继东赴韩国首尔参加国际会议 VLDB2006。

博士生陈继东、刘伟与孟小峰教授一起参加了在韩国首尔举办的第三十二届超大规模数据库（VLDB 2006）国际会议，并作了“Update Efficient Indexing for Moving Objects in Road Network”和“Web Database Integration”论文报告，与参会专家进行了广泛深入的交流。

2006.11.1-2006.11..3

硕士研究生王小锋参加了第二届语义、知识与网格国际学术会议（SKG2006）。

会议期间，王小锋同学就本实验室在本体方面的研究工作和发表的论文“大规模本体数据的推理” 向与会代表进行了报告，引起了大家很大的兴趣，并提出了一些很有价值的问题。

2006.11.3-2006.11.5

硕士研究生张新参加了第三届 WEB 信息系统及其应用学术会议（WISA2006）

与会期间,张新同学向与会代表报告了本实验室发表的的三篇论文《A Framework of web Data Integrated LBS Middleware》,《OrientX: an Integrated, Schema-Based Native XML Database System》,《A Deep Web Data Integration System for Job Search》, 向同行们介绍了本实验室的研究工作与系统成果。

2006.11-9-2006.11.13

博士研究生刘伟、周军锋参加 2006 年全国数据库学术会议（NDBC2006）

NDBC 数据库学术会议是国内最高级别的数据库会议。与会期间，周军锋同学向与会代

表报告了两篇论文:“XML 数据流上的关键字查询”和“XML 数据流上的有序 XPath 查询处理”，刘伟报告了 web 数据管理方面的两篇论文：“Deep Web 数据集成中实体识别问题”和“EasyQuerier：一种基于关键词的 web 集成查询接口”。

2006.11.22-2006.12.10

博士研究生周军锋和孟小峰教授一起访问法国凡尔赛大学和希腊雅典国家技术大学。

访学期间,周军锋同学与法国凡尔赛大学和希腊雅典国家技术大学的博士生进行了学术讨论，确定了双方共同感兴趣的问题、今后的合作计划和人员互访计划。达到了对外宣传、让同行了解我们实验室和 XML 小组的目的，为下一步的研究工作提供了有益的借鉴。

2006 年发表论文列表

2006 年发表论文列表

- 谢敏，王小锋，张新，孟小峰，周军锋，XML 数据流上的有序 XPath 查询处理，计算机研究与发展，卷 43(增刊)：464－470，2006,11. (第 23 届中国数据库学术会议，广州.)
- 王小锋，张新，谢敏，孟小峰，周军锋，XML 数据流上的关键字查询. 计算机研究与发展，卷 43(增刊)：484－489，2006. (第 23 届中国数据库学术会议，广州.)
- 李先，刘伟，孟小峰，EasyQueries：一种基于关键词的 Web 集成查询接口. 计算机研究与发展, 卷 43(增刊)：54－60，2006. (第 23 届中国数据库学术会议，广州.)
- 凌妍妍，刘伟，王仲远，艾静，孟小峰：Deep Web 数据集成中的实体识别方法. 计算机研究与发展, 卷 43(增刊)：46－53，2006 . (第 23 届中国数据库学术会议，广州.)
- **X. Meng**, S.Yin and Z. Xiao: A Framework of Web Data Integrated LBS Middleware. *Wuhan University Journal of Natural Sciences*, ,11(5):1187-1191, Nov., 2006. (*The Third Web Information System and Application(WISA2006)*, Nanjing, Nov 3-5, 2006.)
- **X. Meng**, X.Wang , M. Xie and et al: OrientX: An Integrated, Schema-Based Native XML Database System. *Wuhan University Journal of Natural Sciences*,11(5):1192-1196, Nov., 2006.(*The Third Web Information System and Application(WISA2006)*, Nanjing, Nov 3-5, 2006.)
- W. Liu, X. Li, **X. Meng**, et al: A Deep Web Integration System for Job Search. *Wuhan University Journal of Natural Sciences*, 11(5):1197-1201, Nov., 2006. (*The Third Web Information System and Application(WISA2006)*, Nanjing, Nov 3-5, 2006.)
- W. Liu, C. Lin and **X. Meng**: Web Database Query Interface Annotation Based on User Collaboration. *Wuhan University Journal of Natural Sciences*, 11(5):1403-1406, Nov., 2006. (*The Third Web Information System and Application(WISA2006)*, Nanjing, Nov 3-5, 2006.)
- X. Wang, J. Ou, **X. Meng**, and Y. Chen: Abox Inference for Large Scale OWL-Lite Data. *In Proceedings of The 2th International Conference on Semantics, Knowledge, and Grids(SKG2006)*, Guilin, China, Oct. 31 - Nov. 3, 2006. (Regular paper 18%)
- L. Wang, Q. Li, Y. Li, and **X. Meng**: Dish_Master: An Intelligent and Adaptive Manager for a Web-based Recipe Database System. *In Proceedings of The 2th International Conference on Semantics, Knowledge, and Grids(SKG2006)*, Guilin, China, Oct. 31 - Nov. 3, 2006. (Regular paper 18%)
- Y. Li, **X. Meng**, Q. Li, L. Wang: Hybrid Method for Automated News Content Extraction from the Web. *In proceeding of 7th International Conference on Web Information Systems Engineering(WISE2006),pages 327-338*, Wuhan, China, October 2006
- 孟小峰, 王 宇, 王小锋, XML 查询优化研究, 软件学报, 卷 17(10):2069-2086, Oct. 2006
- W, Liu, X. Meng: Web Database Integration. *In Proceedings of the Ph.D Workshop in conjunction with VLDB 06 (VLDB-PhD2006)*, Seoul, Korea, September 11, 2006.
- J. Chen, **X. Meng**, Y. Guo, X. Zhen: Update-efficient Indexing of Moving Objects in Road Networks. *In Proceedings of the Third Workshop on Spatio-Temporal Database*

Management in conjunction with VLDB 06 (VLDB-STDBM2006), Seoul, Korea, September 11, 2006.

- Y. Chen, J. Ou, Y. Jiang, **X. Meng**: HStar-a Semantic Repository for Large Scale OWL Documents. In *Proceedings of the First Asian Semantic Web Conference (ASWC2006)*, page 415-428, Beijing, China, September 3-7, 2006. Lecture Notes in Computer Science 4185, Springer. (Full Paper 36/208=18%)
- S. Wang, X. Du, **X. Meng**, and H. Chen: Database Research: Achievements and Challenges. In *Journal of Computer Science and Technology*, Vol. 21(5):823-837, September 2006
- W. liu, **X. Meng**, W. Meng: Vision-based Web Data Records Extraction. In *Proceedings of the 9th SIGMOD International Workshop on Web and Databases (SIGMOD-WebDB2006)*, Chicago, Illinois, June 30, 2006. (12/48=25%) [PDF]
- Y. Ling, **X. Meng**, and W. Meng, Automated Extraction of Hit Numbers From Search Result Pages. *In Proceedings of the Seventh International Conference on Web-Age Information Management(WAIM2006)*, pages 73-84, Hong Kong, China,17-19 June, 2006. Lecture Notes in Computer Science 4016, Springer 2006.
- Y. Li, **X. Meng**, L. Wang, Q. Li, RecipeCrawler: Collecting Recipe Data from WWW Incrementally. *In Proceedings of the Seventh International Conference on Web-Age Information Management(WAIM2006)*, pages 263-274, Hong Kong, China, 17-19 June, 2006. Lecture Notes in Computer Science 4016, Springer 2006.
- J. Chen, **X. Meng**, B. Li, C. Lai: Tracking Network-Constrained Moving Objects with Group Updates. *In Proceedings of the Seventh International Conference on Web-Age Information Management(WAIM2006)*, page 158-169, Hong Kong, China, 17-19 June, 2006. Lecture Notes in Computer Science 4016, Springer 2006.
- J. Chen, **X. Meng**, Y. Guo, S. Grumbach, H. Sun: Modeling and Predicting Future Trajectories of Moving Objects in a Constrained Network. *In Proceedings of the 7th International Conference on Mobile Data Management (MDM 2006)*, Nara, Japan, May 9-13, 2006. IEEE Computer Society 2006: 156.
- 王宇，孟小峰，王珊，基于直方图的 XPath 含值为此路径选择性代价估计，计算机研究与发展，卷 43（2）：288-294， 2006，2
- Y. Bai, Y. Guo, **X. Meng**, T. Wan, K. Zeitouni : Efficient Dynamic Traffic Navigation with Hierarchical Aggregation Tree. In *Proceedings of the The Eighth Asia Pacific Web Conference(APWeb2006)*, pages 751-758, Harbin, China, January 16-18, 2006. LNCS 3841.
- **X. Meng**, B. Xu, Q. Liu, et al.: A Survey of Web Information Technology and Application. *Wuhan University Journal of Natural Sciences*, Vol.11(1):1-5, Jan. 2006

研究成果介绍

Web Data Management

Encountering, Keeping and Refinding

- SG-WRAP
- SG-WRAM
- Deep Web Data Integration
- Job Tong
- PIM: A NEW FOCUS

SG-WRAP

A Schema Guided Wrapper Generator

X. Meng, H. Lu, H. Wang, M. Gu

System features:

(1) Generating extraction rules with the guidance of user-defined schema;
(2) The wrapper generated based on the rules could be more accurate and better reflect the users requirements;
(3) Using different schemas, the wrapper can be easily integrated into the different data integration process.

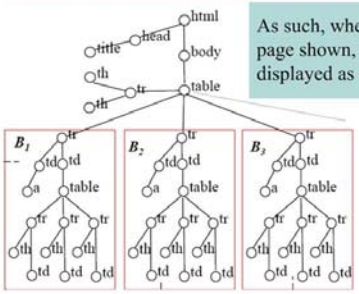
Application Examples:

- Web robots URL: www.robotstxt.org
- Quotes URL: finance.yahoo.com
- Amazon URL: www.amazon.com

Modeling Document

The fetched HTML page is parsed. Syntax errors, such as missing tags in the original HTML document are fixed during the parsing process.

Internally the HTML page is represented as a tree using the document object model (DOM) [W3C98], where a data item is a leaf node in the tree and its position in the document can be described by the path from the root to the leaf.

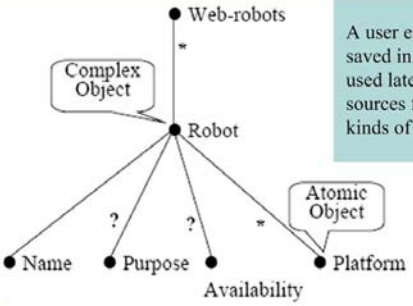


As such, when the user highlights a string in the page shown, its path can be identified, which is displayed as an HTML tree path

- Specifying the structure of ducuments
- Following the DOM
- Identifying data items with paths

Defining Schema

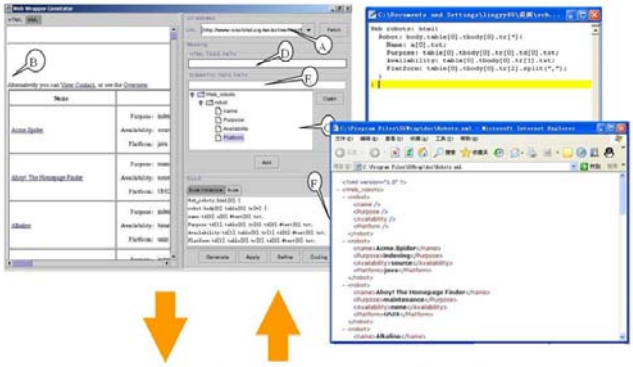
One input is the original HTML page, another input of the system is a user-defined schema for extracted data, which is obtained by the Schema Acquirer and displayed on the screen (just as the following figure shown)



A user entered schema can be saved in a Schema Base to be used later, or shared for other sources from which the same kinds of data to be extracted

- Indicating objects' relationship
- Defining the data type or data

Demonstration Interface



Getting Instances

Establishing the relationship between the HTML tree and schema tree ——

- Similarly, when the user clicks an element in the DTD, it is displayed as a semantic tree path in the schema (E).
- The Mapping Acquirer of the system captures those user clicks that associate strings in the HTML page and their corresponding elements in the DTD.
- For example, with the sample page in the left Figure , the following mapping can be captured

```
html.body.table[0].tbody[0].tr[0].a[0] <=>
    Web_robots.robot.name  "Alkaline"
```

Generating Rule

- In other words, from user interactions, the system obtains a set of instances of rules that map strings in the HTML page to elements in the schema.

```
Web_robots:html{
 Robot: body.table[0].tbody[0].tr[*] {
   Name:          a[0].txt ;
   Purpose:       table[0].tbody[0].tr[0].td[0].txt;
   Availability:  table[0].tbody[0].tr[1].td[0].txt;
   Platform:      table[0].tbody[0].tr[2].td[0].txt
   }
}
```

- With this set of instances of mapping rules, the Rule Generator generates data extraction rule by an induction algorithm, which takes the list of mapping rules instances L as input and returns a candidate rule by incorporating the similar mapping rule instances into a new extraction rule.

SG-Wrap: Architecture

Rule Inducer

Rule Refiner

Wrapper Generator

- The data extraction rules are induced from a limited set of instances. In order to guarantee that the data extraction rule is applicable for the entire HTML document, SG-WRAP includes a refining process. The Rule Refiner generates an XML document by applying the induced rule on the input page.
- This refining process continues until the user is satisfied with the data extracted. The wrapper is generated based on the final data extraction rule by the Wrapper Generator.

ICDE 2002: San Jose, California, USA
pages 331-332,San Jose, CA, 26 February - 1 March 2002

SG-WRAM

Schema-Guided Wrapper Maintenance for Web-Data Extraction

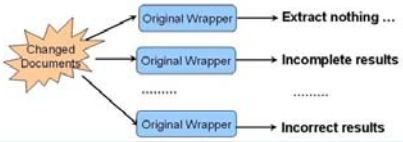
X. Meng, D. Hu, C. Li

SGWRAM-Overview

Schema Guided Wrapper Maintenance

Motivation

- The WWW is also extremely dynamic and continually evolving, which results in *frequent changes* in the structures of web documents.
- Consequently, wrappers may *stop working* when the structures of the corresponding documents are changed no matter how they generate.
- It is often necessary to constantly update or even completely rewrite existing wrappers, in order to *maintain* the desired data extraction capabilities.



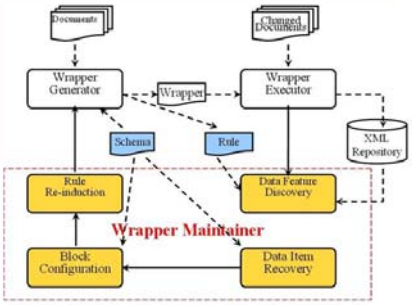
- The Web are very dynamic: contents, page structures
- Original wrappers can stop working: rely on Web page structures
- Re-generating wrappers is not easy: heavy workload to system developers

SGWRAM-Architecture

Wrapper maintainer completes the maintaining task based on four modules:

- *Data features discovery*, finds the metadata for each data item within the extraction rule and learn the content features from schema and extracted results.
- *Data item recovery*, recognizes all possible data items by traveling the HTML tree of the changed document.

- *Block Configuration*, group the recognized data items and construct candidate blocks to match candidate blocks with blocks defined in the schema.
- *Wrapper Reparation*, picks up the representative instances to re-induce the extraction.



Step 1: Data-feature discovery

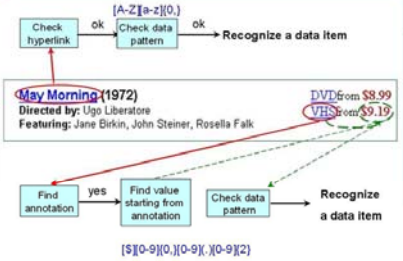
- Compute features of the data items in the original page
- Data pattern feature - A syntactic feature
  - Represented as a regular expression. E.g. $ 15.38 [$][0-9]{0,}[0-9](.)[0-9]{2}
  - Can be extracted using existing technologies

ID	DTD Element	L (hyperlink)	A (annotation)	P (data pattern)
1	Name	True	NULL	[A-Z][a-z]{0,}
2	Director	False	Directed by	[A-Z][a-z]{0,}
3	Actors	False	Featuring	[A-Z][a-z]{0,}(.)*
4	VHSPrice	False	VHS	[$][0-9]{0,}[0-9](.)[0-9]{2}
5	DVDPrice	False	DVD	[$][0-9]{0,}[0-9](.)[0-9]{2}

- **Annotations and Hyperlinks**
  - Get annotation and hyperlink information from the original page
    - Checking the XQuery based extraction rule
    - Hyperlink: step of "…/a/…" in the path
    - Annotation: function of "contains()"

Step 2: Data-Item Recovery

- Traverse the new HTML tree following the depth-first traversal order
- Use the old features to identify potential data items using 3 matching conditions: Hyperlink & Annotation & Data pattern



- A mapping list including all the recognized data items
- Each mapping contains
  - Value of the data item
  - Path to it in the HTML tree
  - Path of the corresponding DTD element

A sample mapping:
M1' (*D*: "May",
HP: …/table[0]/tr[0]/td[1]/span[0]/b[0]/a[0]/text()[0],
SP: VideoList/Video/Name)

Step 3: Block Configuration

- Observation: Data items are located in semantic blocks
- Conforms to the user-defined schema
- Data items are grouped in semantic blocks

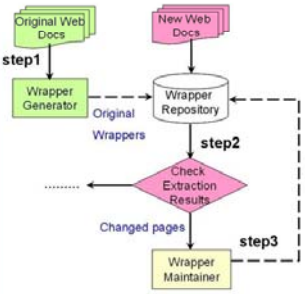
- **Results of Block Configuration**
  - A set of blocks that can fully match with the DTD
  - Each of them is represented as a list of mappings

No.	Element	PATH
1	Title	…table[1]/tr[0]/td[1]/span[0]/b[0]/a[0]/text()[0]
2	Director	…table[1]/tr[0]/td[1]/span[1]/text[contains('preceding-sibling::b[0],"Directed by")]
3	Actors	…table[1]/tr[0]/td[1]/span[2]/text()[contains('preceding-sibling::b[0],"Featuring")]
4	Title	…table[2]/tr[0]/td[1]/span[0]/b[0]/a[0]/text()[0]
5	Director	…table[2]/tr[0]/td[1]/span[1]/text[contains('preceding-sibling::b[0],"Directed by")]
6	Actors	…table[2]/tr[0]/td[1]/span[2]/text()[contains('preceding-sibling::b[0],"Featuring")]

- **Computing "Full Match" Blocks**
  - Identify the level in a top-down manner
  - Check the level by recursively considering the matches between candidate blocks and the schema

Step 4: Rule Re-Induction

- Semantic blocks contain mappings from data items in HTML to DTD elements
- Induce new extraction rule by calling the induction algorithm in wrapper generator
- Refine the rule by trying to ensure the extraction rule cover all other semantic blocks
  - Generalization is necessary



- Other approaches heavily rely on the syntactic features of the data items, and often cannot precisely recognize the data items.

SG-WRAM: a wrapper-maintenance system
- Intuition: use features that are more stable
  - Pattern
  - Hyperlink
  - Annotation

WIDM 2003: New Orleans,Louisiana,USA
Pages 1-8,New Orleans,Louisiana,USA,Novenber 7-8,2003

Deep Web Data Integration

An efficient solution to help users access Deep Web

W. Liu, X. Meng

Definition

. The contents stored in Web databases, which can be accessed through query interfaces.

•Characteristics

–Scale
- *•307,000 Deep Web sites, 450,000 Web DBs, 1,258,000 query interfaces.*

–Structure
- *•348,000 (structured) : 102,000 (text) == 3 : 1*

–Content
- *•Covering all subject domains in the real world, such as economy, sports, politics, etc.*

–Access approach
- *•Query interface: the query forms on the Web pages to access Web databases.*

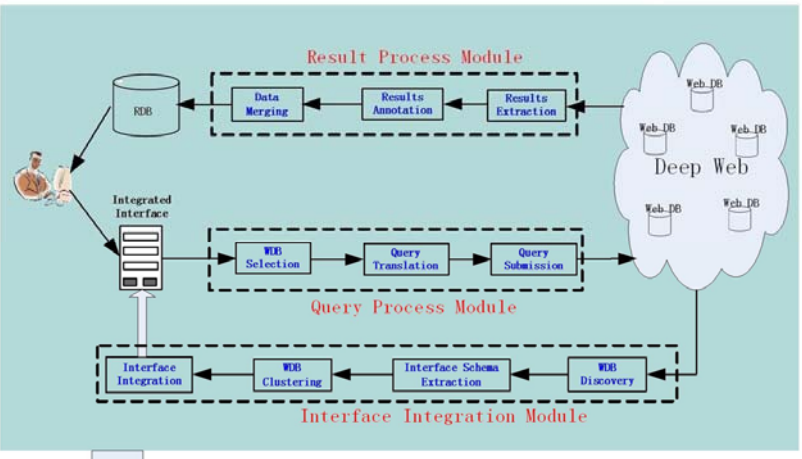
Deep Web Data Integration

Description

. Provide users a unified access to search multiple Web databases efficiently, and represent the results under a global schema.

Challenges

. How to find right Web databases?
. How to query multiple Web databases in a uniformed way?
. How to extract structured data from unstructured result pages?
. How to merge all results under a global schema forsubsequent analysis?
. .....



Interface Integration Module

. *WDB discovery:* find Web databases (or say, their query interfaces) from deep Web;
. *Interface schema extraction:* analyze and extract the schema (attribute information) from query interfaces;
. *WDB clustering:* classify Web databases by domain;
. *Interface integration:* identify matching attributes among the attributes of the query interfaces in a same domain, and generate a global integrated interface.

Query Process Module

. *WDB selection:* select the most appropriate Web databases to answer a given query on the integrated interface;
. *Query translation:* translate the query on the integrated interface into the queries on the local query interfaces;
. *Query submission:* submit the translated queries to their corresponding Web databases.

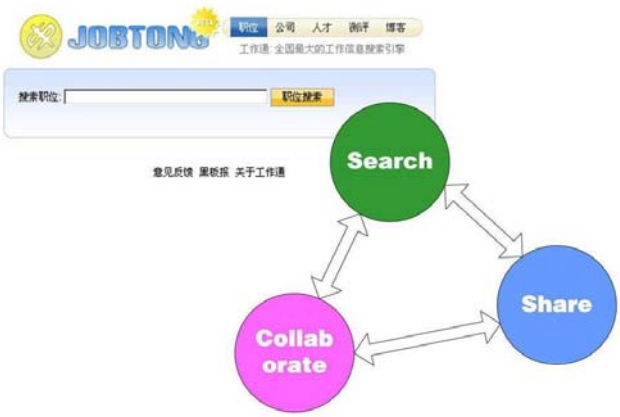
Result Process Module

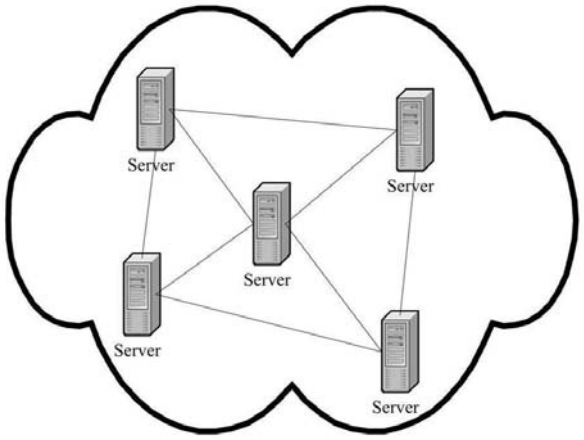
. *Results extraction:* extract structural data records and data items from result pages returned by Web databases;
. *Results annotation:* assign semantically meaningful labels for the extracted results;
. *Data merging:* merge the results from different Web databases under a global schema.

VLDB2006 Ph.D. Workshop
Seoul, Korea Sept 11, 2006

JobTong (工作通)

Professional Job Information Search Engine





Introduction

- Job search engine integrate more than 100 job web sites
- Designed based on web 2. 0 principles
- Search, share, collaboration

Deep Web Crawling

- Distributed Crawling System Based On Nutch And Hadoop
- Deep Web Crawling
- Support More Than 100 Computers

Search

- One stop job search engine support more than 100 job resources
- Incremental update
- Very high search speed
- Support job ranking and duplication removal

Share, Collaborate,and more...

- User register and online resume
- Resume write once and export to different format and style
- Job recommendation based on user profile
- Blog, comment and more
- Job trend analysis based on search log and click log

Http://www.jobtong.cn

PIM: A NEW FOCUS
Personal Information Management

Introduction of PIM

PI (Personal Information)

When talking about ***Personal Information*** of PIM, we focus especially on the capacity of information to affect change in our lives and in the lives of others. So PI means the personal information which influence him or her in some sense.

PSI (Personal Space of Information)?

A ***personal space of information*** includes all the information items that are, at least nominally, under that person's control .An ***information item*** is a packaging of information. Examples of information items include: paper documents. electronic documents . email messages. web pages ,etc.

MEMEX

MEMEX is the first defination on PIM By Vannevar Bush in 1945 ,who expressed a hope that technology might be used to extend ability to handle information

What is PIM?

A branch which focuses on how to keep and store the collected personal information and on how to manage it more efficiently. And PIM can be take as a collective of various activities which are an effort to establish, use and maintain a mapping between information and need.

A framework showing PIM activities

- **Keeping activities** **Affect the input of information into a PSI.**
- **"M-level activities"** **Affect the storage of information within the PSI.**
- **Finding/re-finding activities** **Affect the output of information from a PSI.**

Research Focus of PIM

- Finding, re-finding, reminding and "re-collection" .
- Encountering, keeping, organizing & maintaining information
- From PIM to "GIM".
- Towards a unification & integration of PIM support.
- Measurement and evaluation.
- Search, filtering, auto-classification and Enhancements.
- Digital memories, ubiquitous computing, Beyond email…,etc.

Input Technology of PIM

Information Input is the first step of PIM, Which is a process that information is collected from Public Space of Information, Which includes the listed activities :

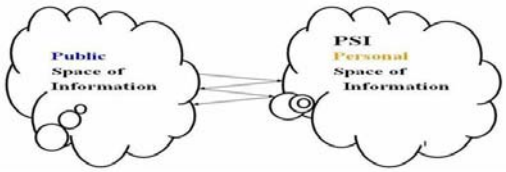
Encountering: A directed search may return an unexpected result which is potentially useful in another context. we still have many serendipitous encounters and re-encounters with information in our everyday lives.

Auto-collecting: Utilize AI tech to collect info automatically.

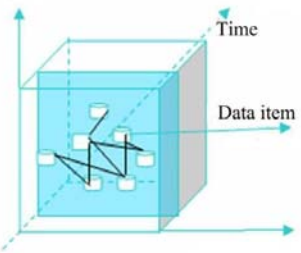
Manual input : Meta-data and other data need to be inputted manually.

Filtering and auto-classification: For so large amount of info ,it is hard to decide what info should be kept for future.

Unification & integration : Research on how to organize the info coming from various "Information islands".



Store Technology (Personal Data space)



About PDS

PDS (Personal Data Space) is the reflection of PSI in digital world, which includes all digital info stored in computer with relation to a person.

PDS has similar characters to PSI: pay as you go, open system ,revolution ,etc.

Research topics on PDS

- Data model of Personal Dataspace
- Security and privacy
- Data indexing, Backup and restore, View
- Data independence
- Meta data : DD, system table, profile
- PDSMS (Personal Dataspace Management System)

Output Technology of PIM

Information Output is the third step of PIM, Which is a process that information is token from a Personal Space of Information and tell the person in a easy way or remind him to do something.

Research topics on PI output:

Finding/Refinding: I remember I have encountered a web-sit which has selt jersey of the football team I like best. But where it is...

CHI of PIM: Let a person get info he need with the most easy way.

Mining: It is so, I have never realize it!

Reminding: Pushing relevant information on the user could serve as an important reminding function.

Key Tech: Push

The goal of research on information output is to :

Short term goal: Make advance work unnecessary for re-finding:

Long term goal: Make it so people hardly need to re-find.

Push and AI tech will play a key role to reach the goal.

Related Research Topics

- Finding/Refinding
- Reminding
- Mining
- Query Language
- CHI(Interface)

Outlook of PIM Research

challenges & issues of PIM

- **Information is fragmented; so too, is the study of PIM.**
- **How to protect the privacy and security of personal information?**
- **Who owns the information in the workplace?**
- **How do we know what is working and what isn't?**

Outlook

- PIM is a new research area of data Management.
- Great opportunity and Great challenge.
- Encourage multi-disciplinary approaches.
- Support the development of methodologies, frameworks and benchmarks for the evaluation of PIM tools and techniques.
- "Pay as you go" is an important rule in PIM research.
- From PIM to GIM, they will benefit people much.

Innovation data management!
And Innovation personal data management!

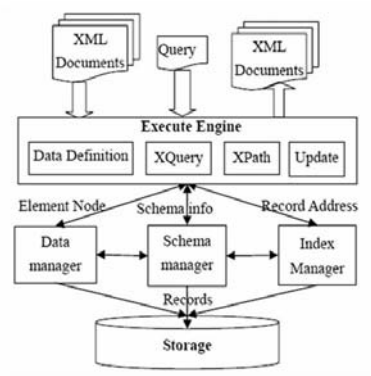
XML Data Management

Beauty & Creativity

- OrientX
- OrientStore
- XML Sequencing
- OrientX/Ontology
- Integration Auditing of Outsourced Database

OrientX
A Native XML Database System

System Architecture



Features & History

- **Features**
  - XML Schema based repository
  - Full support of XQuery1.0 and XPath2.0
  - Flexible native storage strategies
  - Novel Path index and Value index
  - Navigation and Algebra based query processing
- **History**
  - OrientX 1.0 (2002-2003)
    - OrientStore, Schema Manager, Data Manager
  - OrientX 1.5 (2003-2004)
    - XPath Processing Module, XML Labeling Module, Index Manger
  - OrientX 2.0 (2004-2005)
    - Navigation-based Query Engine
  - OrientX 2.5 (2005-now)
    - Algebra-based Query Engine

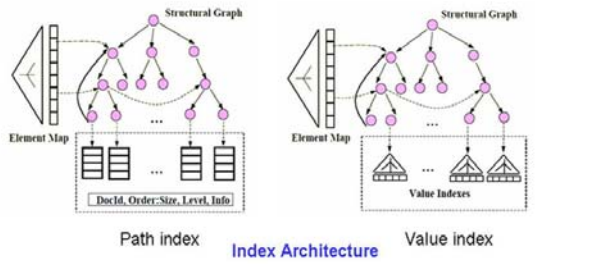
Native Storage

- **OrientX implements four kinds of storage strategies**
  - Different storage strategies can be utilized according to application requirement

Storage Strategy	Feature	Application Environment
DEB Depth-first Element Based	•One Element, One Record •Pre-Order Storage Style	Queries which are consisted of absolute paths
CEB Clustered Element Based	•One Element, One Record •Element-based Clustering Storage	Queries which contain "//" or "*" and only retrieve the data of target node
DSB Depth-first Subtree Based	•One Subtree, One Record •Pre-Order Storage Style	Queries which need to access subtrees of XML data
CSB Clustered Subtree Based	•One Subtree, One Record •Subtree-based Clustering Storage	Queries which need to access the data based on semantic blocks

Schema-guided Index

- Schema-guided Index can support twig query and predicates efficiently



Structural Graph: *the structure summary of XML data*
Element Map: *fast entries to nodes in Structural Graph*

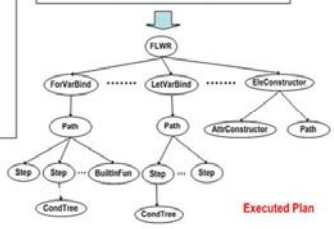
Navigation Implementation for XQuery

- According to XQuery core syntax, we abstract 13 operators to implement XQuery processing.

Operator	Describing
1. FLWR:	ForClause LetClause WhereClause ReturnClause
2. ForVarBind:	FOR $fv IN .....
3. LetVarBind:	LET $lv := .....
4. Path:	XPath in XQuery
5. Step:	Atom operator in XPath, such as"/","/".
6. CondTree:	Predicate in XQuery.
7. EleConstructor:	Element construct in XQuery
8. AttrConstructor:	Attribute construct in XQuery
9. BuiltInFun:	Function in XQuery
10.IfThenElse:	IF expression in Xquery
11.Quantify:	(Some | Every) var in XPath satisfy condition
12. SetOpt:	Union, Intersect, Except operator of node set
13. SortBy:	Sort operator

FLWR ::= ForClause LetClause WhereClause ReturnClause
ForClause ::= FOR $fv1 IN E, $fv2 IN E.....$fvn IN E
LetClause ::= LET $lv1 := E, $lv2 := E ....., $lvn := E
WhereClause ::= Where.....
ReturnClause ::= Return (E).....
E ::= FLWR | XPATH

XQuery core syntax Fragment

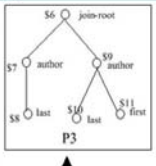


Executed Plan

Algebra-based Query Processing

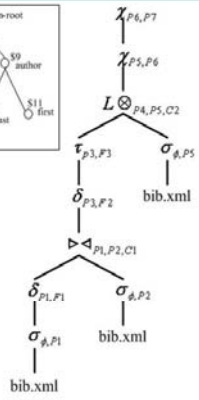
- As ***Relation Algebra***, which follows *set-at-a-time* style, XML query can be processed efficiently using ***Tree Algebra***

- ✓ **Select** $\sigma_{P_i,P_o,PB}(X)$
- ✓ **Sequence** $\xi_{P_i,(op_1,op_2,...op_n)}(X)$
- ✓ **Construct** $\chi_{P_i,P_o}(X)$
- ✓ **Join** $\bowtie_{P_1,P_2,C}(X_1,X_2)$
- ✓ **Group** $\gamma_{P,f}(X)$
- ✓ **Aggregate** $A_{P_i,f}(X)$
- ✓ **Duplicate Eliminate** $\delta_{P,f}(X)$
- ✓ **Sort** $\tau_{P,f}(X)$



① Pattern Tree Generating

② Algebra Plan Generating



Xiaofeng Meng,Yu Wang,Daofeng Luo,Shichao Lu,Jing An,Yan Chen,Yu Jiang,Jianbo Ou.OrientX: A Native XML Database System (Chinese). NDBC 2003

Xiaofeng Meng,Daofeng Luo,Mong Li Lee,Jing An. OrientStore: A Schema Based Native XML Storage System(demo). VLDB 2003

Jing Wang,Xiaofeng Meng,Shan Wang. SUPEX A Schema-Guided Path Index for XML Data(Poster). VLDB 2002

Shichao Lu,Xiaofeng Meng,Can Lin,Yu Wang. Navigation implementation for XQuery in OrientX(Chinese). NDBC 2004

Xiaofeng Meng,Daofeng Luo,Yu Jiang,Yu Wang. OreintXA: An effective XQuery Algebra(Chinese). Journal of Software

OrientStore
A Schema Based Native XML Storage System

Introduction

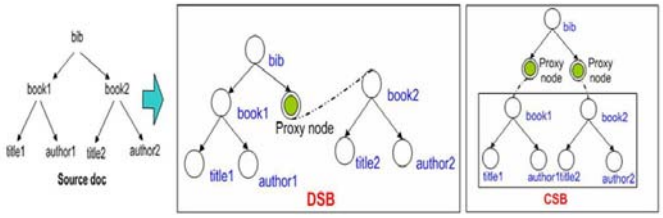
- **Native XML storage has great impact on I/O during query processing**
- **There are two issues about native XML storage:**
  - **The granularities of record:**
    - Element Based (EB)
    - Subtree Based (SB)
    - Document Based (DB)
  - **The storage configuration of record:**
    - Depth-First Order
    - Breadth-First Order
    - Clustering

Motivation

- **How to utilize XML Schema information in XML storage?**
- **How to cluster records in order to reduce I/O?**
- **How to combine the two?**

DSB & CSB

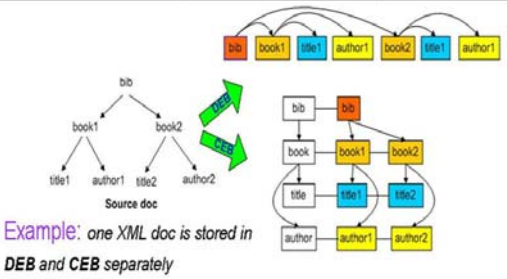
	Similarity	Difference
DSB (Depth-first Subtree Based)	Granularity of record is subtree	• Divide doc into subtrees according to the physical page size • Records are stored in depth-first order
CSB (Clustered Subtree Based)		• Partition the schema graph into semantic blocks • All instances of the same semantic block are stored together



Example: *one XML doc is stored in* ***DSB*** *and* ***CSB*** *separately*

DEB & CEB

	Similarity	Difference
DEB (Depth-first Element Based)	Granularity of storage is element	The records are stored in a pre-order fashion
CEB (Clustered Element Based)		Records of the same ***SchemaNode*** are clustered together



Example: *one XML doc is stored in* ***DEB*** *and* ***CEB*** *separately*

Why Schema?

- **For storage:**
  - Reduce the storage consumption through replacing tag name with SchemaNode ID.
  - Help to cluster records with the same *TYPE*.
- **For query processing:**
  - Help to tell possible ancestor-descendant relationship, which in turn helps avoid navigating unnecessary nodes.
  - Help to eliminate ambiguous path expression, thus avoid visiting unnecessary nodes for paths containing "//" or wildcards "*"

Conclusions

- The finer the granularity is, the more the space consumes, and also the longer the importing, exporting takes
- Clustering according to schema information dramatically reduces the IO costs than Non-Clustering strategy
- CEB needs less IO than CSB, because physical page of CEB can hold more element nodes than page of CSB
- The order of granularity is DEB = CEB < DSB < CSB, and the order of space consumptions is CEB > DEB > CSB > DSB

Xiaofeng Meng, Daofeng Luo, Mong Li Lee, Jing An. OrientStore: A Schema Based Native XML Storage System(demo). VLDB 2003.

XML Sequencing

Sequencing XML is a novel way of presenting both XML data and XML queries by structure-encoded sequences

- The sequence data representation preserves query equivalence
- Structured queries can be answered without expensive join operations
- Can develop Infrastructure that unifies indices on both the content and the structure of XML documents

Motivation

In most XML indexing solutions, tree pattern is not a first class citizen, instead, the most commonly supported query interface is the following:

Simple Paths ➔ P (Node Ids)

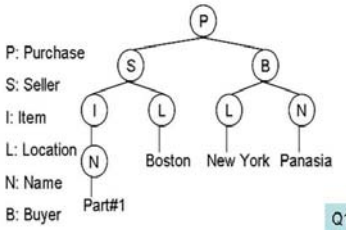
For queries with '*', '//' or branch nodes, join operation is needed.

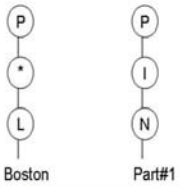
Our sequence-based XML indexing present a major departure from previous XML indexing approaches. It supports a more general query interface:

Tree Pattern ➔ P (Doc Ids)

The tree structure itself is used as the basic query unit and no join is needed

Tree Structure Sequencing





Q1:find orders with a Boston seller or Buyer Q2:find item or subitem with name part#1

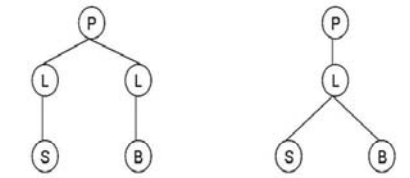
preorder sequence:
<P, PS, PSI, PSIN, PSINv1, PSL, PSLv2, PB, PBL, PBLv3, PBN, PBNv4>

Q1:<P, P * L, P *Lv2>
Q2:<P, P//I, P//IN, P//INv1>

sample document *sample query*

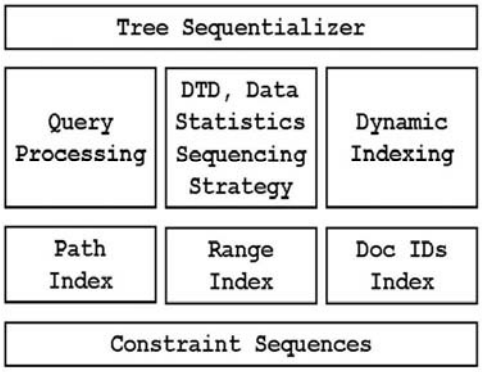
Query Equivalence

- **False Alarm:**



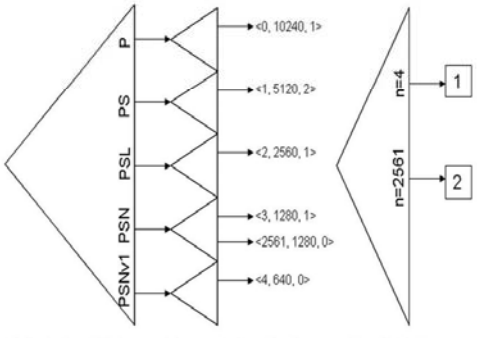
- **Solution:**
  - Introduces a class of sequences called constraint sequences, which preserve query equivalence

XSeq *Demo* Architecture

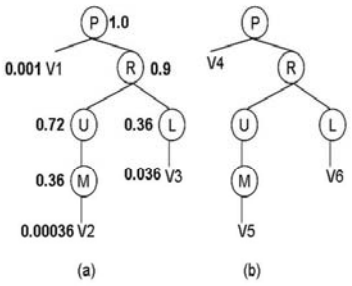


Storage Infrastructure

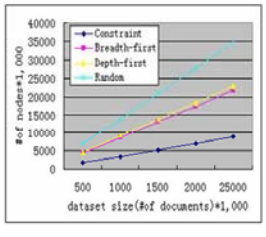
Doc1: <P, PS, PSL, PSN, PSNv1>
Doc2: <P, PS, PSN>



Performance-oriented Sequencing



PS:(a) <P, Pv1, PR, PRU, PRUM, PRUMv2, PRL, PRLv3>
(b) <P, Pv5, PR, PRU, PRUM, PRUMv6, PRL, PRLv3>
CS:(a) <P, PR, PRU, PRL, PRUM, PRLv3, Pv1, PRUMv2>
(b) <P, PR, PRU, PRL, PRUM, PRLv3, Pv5, PRUNv6>



Index size comparison

Xiaofeng Meng, Yu Jiang, Yan Chen, Haixun Wang: XSeq: An Index Infrastructure for Tree Pattern (Demo). SIGMOD 2004.

Haixun Wang, Xiaofeng Meng: On the Sequencing of Tree Structures for XML Indexing. ICDE 2005.

OrientX/Ontology
Large Scale Ontology Data Management

System Overview

- *OrientX/Ontology* supports management of Ontology encoded as OWL documents.
- *Physical storage model* is based on file system which utilizes semantic model of OWL data.
- *Inference and query* are implemented on such physical storage model.
- Currently supports characters of *OWL Lite* & *SPARQL* query

Semantic is not a fixed thing

- Semantic represents the understanding of people for some thing. It always changes with people.
- E.g. Concept "Foreigner", American treat Chinese as a foreigner and Chinese treat American as foreigner. So concept "Foreigner" has different semantic for American and Chinese.
- E.g Different teacher will have different definitions for "excellence student"

Original data to describe the right picture

1. pic-uri type singer;
2. pic-uri sing xxx;

Then people can define "FansSinger"

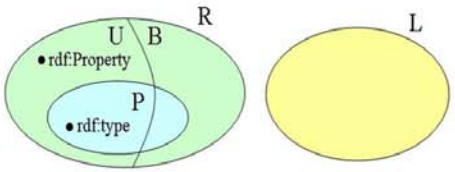
(xxx type singer) and (xxx sing yyy)=>
(xxx type FanSinger)

- People will not use same rules to express the different semantics for the same data.

RDF MODEL

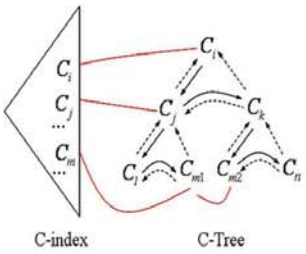
$M \subset R \times U \times (R \cup L)$

- R is the set of resources $R = U \cup B$
- U is the set of URI references *rdf:Property* $\in U$
- B is the set of blank nodes
- L is the set of literals *U, B, L are disjoint*
- P is the set of properties $P \subset R$, *rdf:type* $\in P$

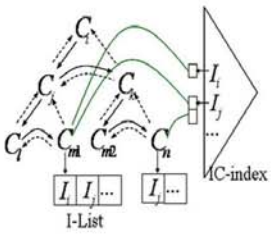


Storage Design

- Class part :
  - $C = \{URI_i \mid Ex < URI_i \ \text{rdf:type owl:Class} >\}$;
  - $R_C = \{[C_i, C_j] \mid C_i, C_j \in C, Ex < C_j \ \text{rdfs:subClassOf} \ C_i >\}$
  - $R_{CI} = \{[URI_i, C_j] \mid < URI_i \ \text{rdf:type} \ C_j >\}$;



C-Tree and C-Index for R_C Storage



I-List and IC-index for R_{CI} Storage

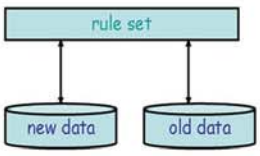
Inference Method

◆ Initial Method

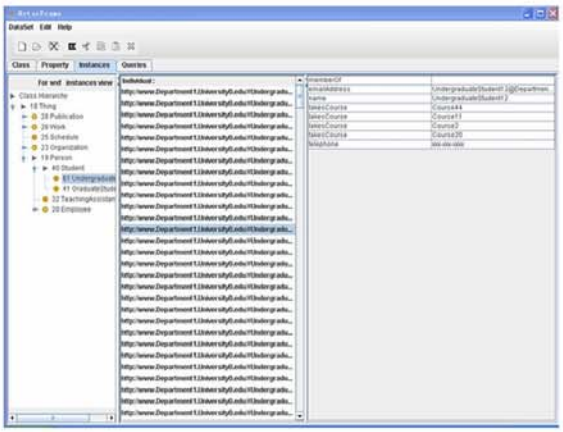
- ✓**Appropriate when Database is initialized**
- ✓**RSAB** (Rule Static Association Based)
- ✓**RDAB** (Rule Dynamic Association Based)
- ✓**RGSB** (Rule Grouped Sorted-based)

◆ Incremental Method

- ✓**Appropriate when new data and old data coexist**
- ✓**PGSB** (Pattern Group Sharing Based)



User Interface



Yan Chen, Jianbo Ou, Yu Jiang and XiaoFeng Meng, HStar-a Semantic Repository for Large Scale OWL Documents. ASWC 2006

Xiaofeng Wang, Jianbo Ou, Xiaofeng Meng and Yan Chen, Abox Inference for Large Scale OWL-Lite Data. SKG 2006

Integrity Auditing of Outsoured Database

Introduction

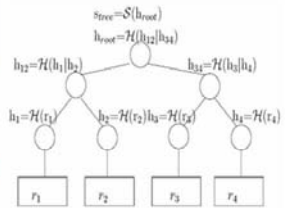
- **Providing Database As-a-Service**
  - Instead of taking care of all the database management task in house, all the user's data is sent to a service provider
- **The BENEFIT**
  - In this way the user can eliminate in-house hardware, software and expertise needs to run DBMS
- **Challenges:**
  - Communication overhead
  - Security Concern
  - ...
- **Our Focus**
  - Validation of query results (make sure that the query results returned by the service provider are both ***correct*** & ***complete***)

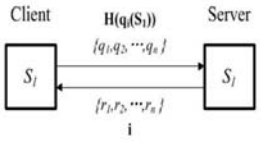
Querying Encrypted Data

- **Traditional Encryption Method**
  - **Can only handle *equivalent query* containing (=,<>)**
- **Query-Friendly Encryption Scheme**
  - **Special Index Approach (Hacigumus et.al., *SIGMOD* 2002)**
    - **Drawback : Query result is super set of the real result**
  - **Order-Preserving Encryption (Agrawal et.al, *SIGMOD 2004*)**
    - **Benefit : Enable flexible query predicates**
    - **Benefit : Guarantee the indistinguishability of encrypted data distribution from original data distribution**

Related Work

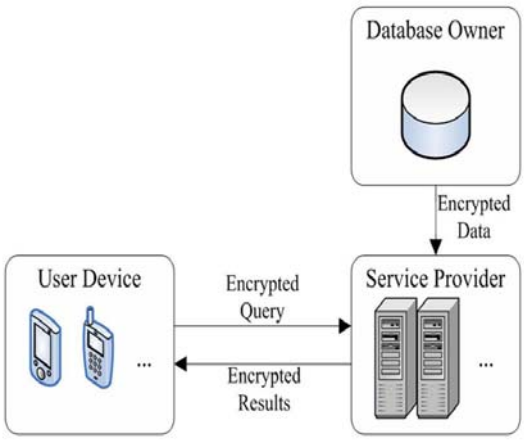
Merkle Hash Tree Based Approach
Using aggregated signature to generate a proof of *correctness* & *completeness*





Challenge Token Based Approach
An approach based on probability

Database Outsource Scenario



Auditing *Correctness*

- ***Correctness***
  - All the results must originate in the owner's data and has not been tampered with
- ***Correctness* Threats:**
  - **One may modify some fields of our data**
  - **One may add some malicious tuples into our data**
  - **One may duplicate some tuples in our data**
  - **...**

Auditing *Completeness*

- ***Completeness***
  - Result includes all records that satisfying the query
- ***Completeness* Threats:**
  - **A malicious application server may only execute the user query on part of the data or may just return part of results**
  - **One may delete some tuples from the data**
  - **One may delete some tuples from the query result**
  - **...**

- Clustering Moving Objects in Road Networks
- Indexing of Moving Objects in Road Networks
- PhoneDB
- Quality Aware Privacy Protection for LBS
- “小金灵”

Clustering Moving Objects in Road Networks

J. Chen, C. Lai, X. Meng, J. Xu, and H. Hu

Motivation

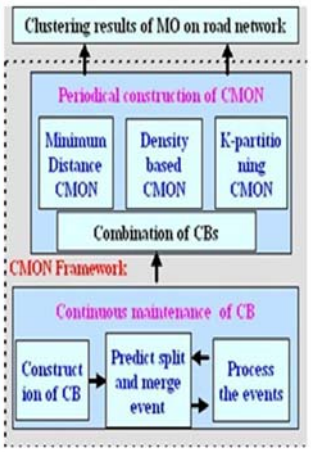
- **Application Example**
  - Real-time monitoring traffic congestion condition
- **Existing method**
  - Clustering moving objects in Euclidean spaces
  - Clustering static objects in spatial networks

- **Challenge**: Objects moving in spatial networks & Metric is network distance
- **Goal**: 1) Minimize cost of clustering and its maintenance 2) Minimize network distance computations 3) Support multiple types of cluster in a single application.

Framework: CMON

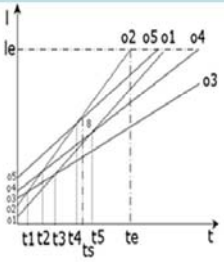
Main Idea

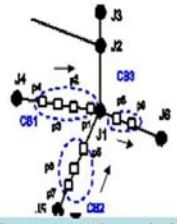
- Cluster block (CB) as underlying clustering unit
  - Easy to maintain
  - Serve as a building block of different types of clusters
- Clustering process
  - Continuous maintenance of CBs
  - Periodical construction of global clusters
- Incremental network extension for CMON construction



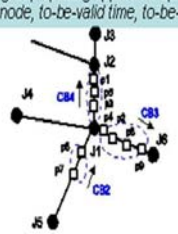
Continuous maintenance of CBs

- *Splitting event in mid of segment*
  - Predict the initial splitting time on moving along the segment
  - **Problem:** the neighborhood of objects changes over time
  - **Solution:** dynamically maintain the order of objects over time on the edge





- *Splitting event at the end of segment*
  - Use group splitting approach: split the CB by *nextnode, to-be-valid time, to-be-expired time*



Periodically Construct CMON

- *Main problems*
  - How to use CBs to construct application-level cluster with different criteria
    - Distance-based, Density-based, K-partitioning clustering
  - How to reduce network distance computation among CBs
    - Incremental network extension

Construct Clusters with Different Criteria

- **Distance-based CMON**
  - Definition: For each cluster, the minimum distance with other objects in the cluster is not longer than a user specified threshold δ ($\delta >= \varepsilon$)
  - Combination of CBs based on their network distance
- **Density-based CMON**
  - Definition: For each cluster, the average density is higher than a given threshold ρ and not any empty segment whose length is longer than E.
  - Same as the Distance-based CMON construction, but a dynamic minimum-distance constraint, related to density of candidate CB
- **K-Partitioning CMON**
  - Definition: Given a set of objects, group them into the K clusters such that the sum of distances between all adjacent objects is minimized
  - Initially select K CBs as the seeds for K clusters and assign the remaining CBs to their nearest clusters to make distance sum minimum

Conclusions and Future work

- An unifying framework for clustering moving objects in network to support different cluster criteria
- Splitting clustering costs into different granularity in conjunction with movement feature in the road network

- Future work
  - Predictive Clustering of Moving Objects
  - Movement prediction at intersections

DASFAA 2007: Bangkok,Thailand April 9-12 2007

Indexing of Moving Objects in Road Networks

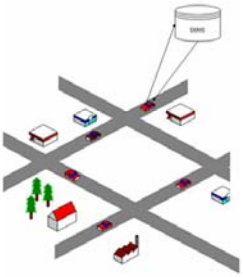
X. Meng, J. Chen, Y. Guo, Z. Xiao

Motivation

Indexing moving objects in **road network setting** with **efficient indexing update** performance

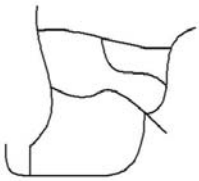
->How to exploit the network constrains in the index to support frequent location updates of moving objects?

- The spatial property of objects movement is captured by the network and indexed by R-tree
- The R-tree remains fixed since the road network seldom change
- Index the objects on each road segment by a dynamic structure

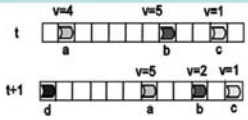


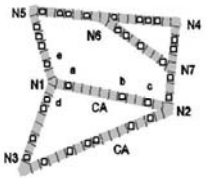
A graph of cellular automata (GCA) Model

GCA - integrate traffic movement features into model of **moving objects** and the **road network**



- Structure of GCA
  - each edge in the GCA consists of a cellular automaton (CA)
- Instance of a GCA
  - a mapping from the cells of the GCA to moving objects
- Transition of GCA



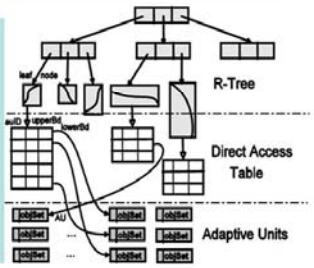


Index Structure

- Adaptive Unit (AU) - a dynamic data structure
  - Group moving objects with the same **direction**, similar **velocity** and location

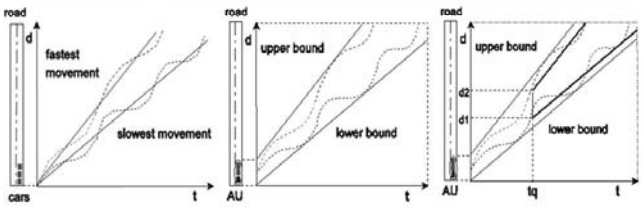
  AU = (auID, objSet, upperBound, lowerBound, edgeID,...)

- R-tree for the road segments + Adaptive Units
  - Dimension reduction by the road network
  - One-dimensional AU structure
  - Predicted Trajectory bounds



Trajectory Prediction

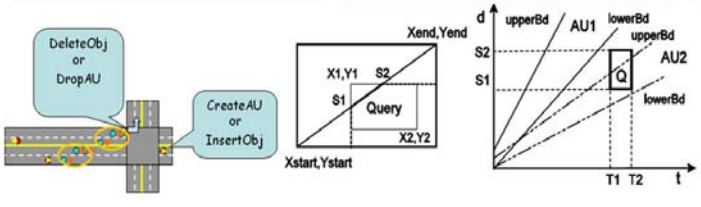
- Simulation based Prediction - SP method
  - Simulate object's future trajectory
    - Two simulated trajectories based on different assumptions on the traffic conditions
  - Linearization of the discrete points to form trajectory bounds
  - Adaptation of trajectory bounds
    - Assume the same trends (slope) of bounds and adjust only initial locations



Index Update and Query Algorithms

- Update Algorithms (the update of AUs)
  - Creating an AU
  - Dropping an AU
  - Adding objects to an AU
  - Removing objects from an AU

- Query Algorithm (Window Queries)
  - Spatial search in the R-Tree
  - Transform the 2D search to 1D search
    - (X1, Y1, X2, Y2)-> (S1, S2)
  - Find the intersected AUs



Conclusions and Future Work

AU index scheme achieves high update performance
- Few update frequency: an accurate prediction method
- Low update and query cost: one-dimensional AU structure

More works
- Predictive query algorithm (processing at the intersections)
- KNN, CKNN processing in road networks

VLDB-STDBM 2006: Seoul, Korea pages 9-16,September 11, 2006

PhoneDB

A Small Footprint DBMS for Mobile Phones

J. Chen, S. Yin, C. Lai and X. Meng

PhoneDB

- Scalable
- High availability
- High-performance
- Transaction-protected data management
- Ease development and maintenance of cellphone applications

Local data management service on mobile phones

Platform

- Lenovo Smartphone

Hardware

- ARM7 39M
- 1.5M RAM Memory
- 8M FLASH NAND
- Nucleus OS & TI FFS

Software

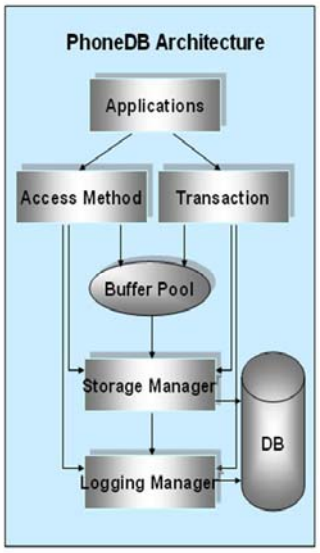
- Nucleus Plus/ROSE33 OS
- TI Flash File System
- TI MMI Framework

Implementation

- Develop with ANSI C
- PhoneDB LIB of 100K size
- Use 100K RAM at runtime

Resource limitations

- Low computation
- Very limited RAM memory
- Slow write and erase of Flash memory
- Low network bandwidth
- Limited power of battery



Architecture

- Access Methods
- Buffer Pool
- Storage Manager
- Transaction
- Logging Manager

Design Rules

- Compression for data structure and code
- Reduce the usage of RAM
- Minimize write operation
- Make full use of fast-read operation
- Minimize updates to reduce erase operation

Key Techniques

- **Key-based data model**

  (key, value) representation like BerkeleyDB

  Application-defined scheme

- **Write Buffer**

  Data are read directly from flash memory

  Records in buffer must be dirty

  Defer and reduce write and erase operation

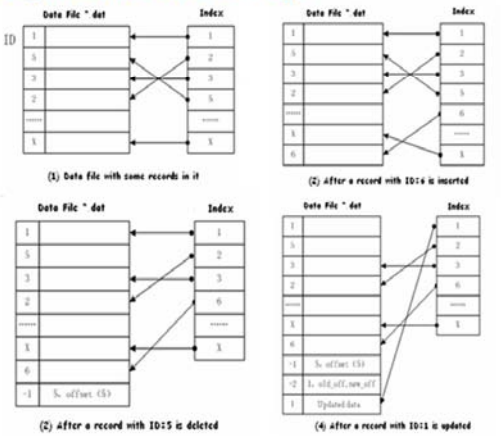
  Improved LRU Replacement, combined with Update Frequency Based (UFB)

- **Log-based recovery**

"Immediate write"	Log-based recovery
Write log record to data file	Delete log record and update log record
Write data record to data file	Timestamps of index and data
Update index in RAM	Keep the consistence of index and data

Key Techniques

- **Log-based record storage**



Key Techniques

- **Log-based record storage**

- Append-only data storage
- Add the log record to the data file
  - Data records
  - Log records (delete log and update log)
- Separate the index file from data file
  - B+ tree with append write
  - Splitting Prediction for ordered data insertion
- Space recycle
  - Garbage collection when few valid data
- Application shared pointer

Quality Aware Privacy Protection for Location-based Services

Z. Xiao, X. Meng, J. Xu (HKBU)

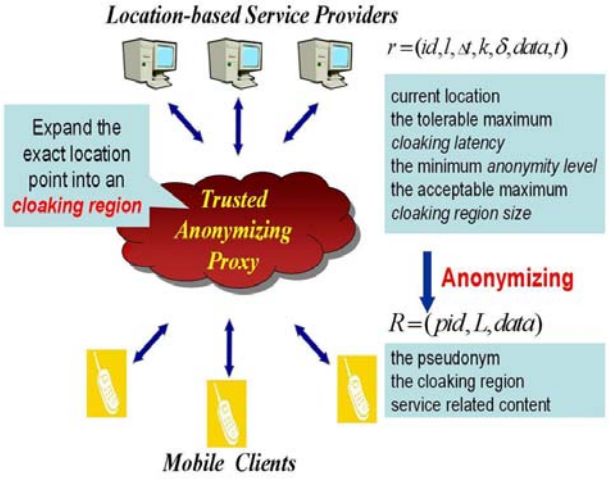
Location-Based Services

- Mobile yellow page
- Buddy trackers
- Electronic tour guides
- Traffic navigation
- Electronic coupons
- Emergency support service

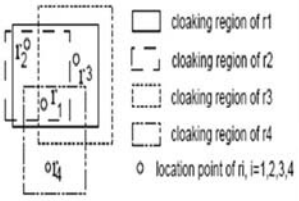
Privacy threat and Requirements

- Unique Identifier and Location Information cause identification of special individual
- Location Anonymity
  - protect the user's sensitive location (e.g., in a clinic or nightclub).
  - k-anonymity model: cloak the user's location by an extended region large enough such that it contains at least *k*-1 other users (location anonymity set).
- Identifier Anonymity
  - hide the user's identifier with sensitive message (e.g., political or financial data).
  - k-anonymity model: the requests' locations are cloaked such that any location l is covered by at least k-1 other requests so that a request is not distinguishable from the other k-1 requests (identifier anonymity set).

System Model



Quality Aware Location K-Anonymity Model



location anonymity set of r_1: $U.out=\{r_1,r_2,r_3\}$

identifier anonymity set of r_1: $U.in=\{r_1,r_2,r_3,r_4\}$

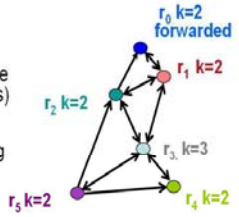
Location Privacy
to expand the user location into a cloaking region such that the *k*-anonymity model is satisfied.

Temporal QoS
the request must be anonymized by the predefined maximum cloaking delay $t+\Delta t$

Spatial QoS
the cloaking region must be inside a circle $L\subseteq\Omega(l,\delta)$

Cloaking Algorithm

- Directed Graph $G_d=(V,E_d)$
  - V:the set of requests
  - *E:the* neighborship between the corresponding nodes (requests)
- Min Heap
  - to order the requests according to their cloaking deadlines
- Spatial Index
  - over the location points of all requests
  - window query to quickly find the neighbors of a request.



r_1. Uout = Uin = {r_0; r_1; r_2}
k_o=k_i=2>k-1=1,success
r_1.L=MBR{r_0; r_1; r_2}

Maintenance
For new incoming request r
Insert into the spatial index and the heap
Insert into the directed graph and construct the neighbor edges

Cloaking
For the first r approaching its deadline.
compute the number of r's neighbors k_o and k_i that have been anonymized successfully in U.out and U.in
If both k_o and k_i >=k-1, success

Improvement with dummy

- Motivation
  - When cloaking fails, generate dummy requests can guarantee a 100% success rate.
  - Only need to maintain the in-degree $r.k_i$ and out-degree $r.k_o$ of each node *r*.
- Requirements
  - Dummies should be both in-degree neighbors and out-degree neighbors of r, thus the privacy level will be higher.
  - Dummies must be indistinguishable from actual requests.
  - Dummies should satisfy the spatial QoS requirement of r.

DASFAA 2007: Bangkok, Thailand, April 9-12, 2007

“小金灵”嵌入式移动数据库系统

系统概述

“小金灵”是面向掌上电脑、PDA、手机等移动设备，进行数据存储管理的嵌入式移动数据库产品

应用需求：

- 公共信息发布
  - 股票行情、天气和交通信息
- 实时数据采集
  - 保险业保单数据采集
- 移动商务
  - 移动销售管理，移动股票证券交易等
- 多媒体应用
  - MP3，动画，图像，TV等
- 个人信息管理和位置服务

设计要求：

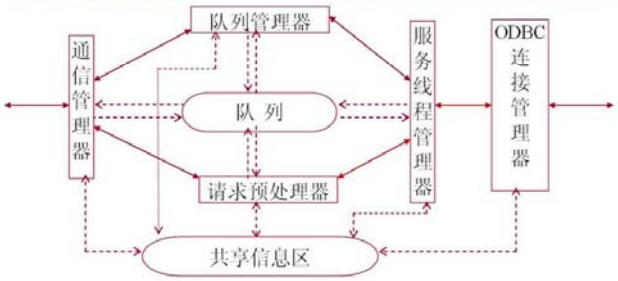
- 支持多种嵌入式操作系统
- 微小型内核结构
- 完善的数据同步机制
- 高效的事务管理功能
- 高效的查询处理
- 多种安全保护机制
- 支持多种连接协议，接口简明实用
- 自动管理功能

体系结构

- 三层体系结构
  - “小金灵”嵌入式数据库应用：移动客户端应用
  - “小金灵”同步服务器：移动客户端与远程主数据库间的数据交换
  - 远程主数据库服务器

小金灵同步服务器

- 同步服务器技术特点和同步机制
  - 灵活的同步方式（上载、下载、混合）
  - 表级异步多主本复制技术
  - 支持数据同步对象
  - 灵活的数据应用模式
  - 增量数据上载
  - 支持多种通讯协议
  - 支持多种数据源
  - 支持事务性数据同步
  - 必要的冲突解决方案
  - 必要的应用模式设计工具



技术特性及性能指标

1. 平台适应性

支持WinCE, PalmOS, Hopen, Linux等

2. 微小内核，所需存储空间小

占用内存<100KB，执行代码<300K

3. 支持基本的SQL功能

ANSI SQL子集：建表、数据收集、更新、浏览查询

4. 支持多种数据类型

Int, float, decimal, char, date, long等

5. 支持内部滚动游标

6. 可伸缩性

组件化和应用可定制功能

7. 开发效率高

简单的API和构件库

8. 支持多种数据源

Oracle, DB2, Sybase, Microsoft SQL Server, Kingbase等

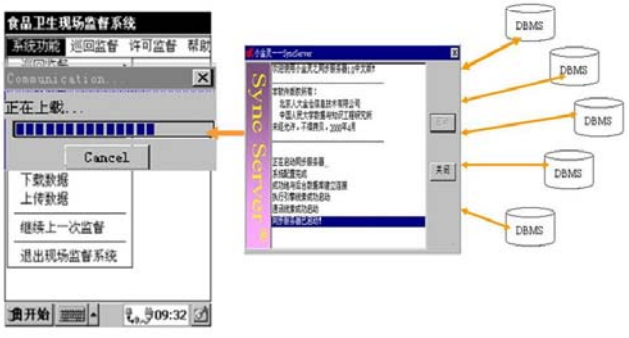
9. 与远程数据库数据交换

同步服务器提供双向复制

10. 通用的数据库用户界面

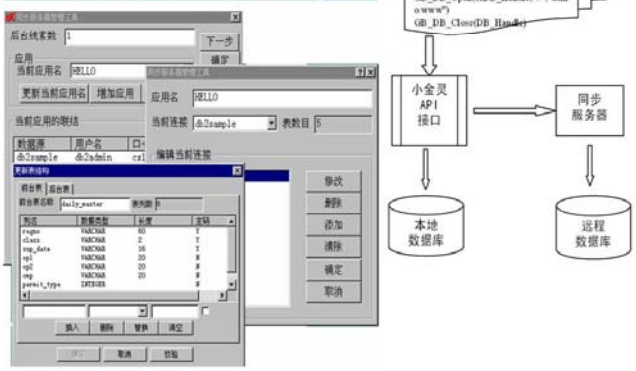
典型应用

- 面向数据采集：食品卫生监督检测系统
- 面向无线通讯：军用后勤指挥掌上电脑系统
- 面向信息家电类：手机、机顶盒、车载计算等



应用开发与系统管理工具

- 应用开发接口API和构件库
- DB CENTER管理工具
- 同步应用管理器 （Sync Manager）



欢迎访问实验室网站

WAMDM 网络与移动数据管理实验室

Innovative Data Systems Research

Lab of Web and Mobile Data Management

Home | Seminars | News | Projects | Publications | People | Systems | Activities | Reports

Introductions

WAMDM means " Web And Mobile Data Management", Which is Professor Xiaofeng Meng's research lab and is affiliated with the Key Laboratory for Data Engineering and Knowledge Engineering MOE and the Department of Computer Science, School of Information, Renmin University of China.

The research vision in WAMDM is how database techniques would fit into the Web and Mobile computing environments. The research style in our lab is having two tracks – research and system – in order to ensure that the research is actually applied. Innovative data systems research is our goal.

WAMDM Lab has been conducting database related research for many years, and is considered one of the best database groups in the country. It's projects range from the Web Data Management, XML Data Management to Mobile Data Management, focusing on Web data extraction, Deep Web data integration, dataspace for PIM, native XML Database, ontology data management, road network moving objects management, smart DBMS, RIFD data management, data management in Context-aware computing, Location privacy, outsourced databases security, etc..

The site contains information on the projects that are currently in progress and the people in the group. You can also find information on the weekly seminar and annual report. In addition, this site hosts the following webpages:

Database Society of China Computer Federation

MDM2008

WISA2007

WAMDM Lab locates at the First floor, Computer Building, Renmin University.

WAMDM Annual Report 2006 hot

Annual Report 2006 of WAMDM! Go to see more detail !

Conference

- **Host Conference**
  - WISA2007: **4th Web Information System and Application**
  - DASFAA 2007 Workshop: **DASFAA 2007 International Workshop on Scalable Web Information Integration and Service**
  - MDM 2008: **The 9th International Conference on Mobile Data Management**
- **Call For Papers**
  - 2007 IEEE 23rd International Conference on Data Engineering (ICDE 2007) : Paper submission Deadline: July 12, 2006, 11:55 PM Pacific Standard Time; The Marmara Hotel, Istanbul, Turkey
  - 33rd International Conference on Very Large Data Bases: Abstract submission deadline: March 14, 2007 (5:00pm PST); University of Vienna, Austria
  - JOS Special Issue on Deep Web Data Integration

News

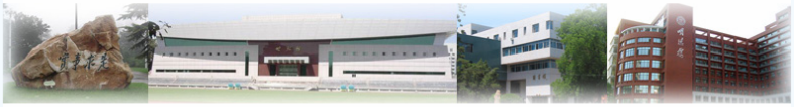
- JobTong (A Deep Web data integration system) is released
- Prof. Xiaofeng Meng and Junfeng Zhou, visited University of Versailles Saint-Quentin en Yvelines and National Technical University of Athens from 2006.11.22~2006.12.10
- Prof. Xiaofeng Meng attended Dagstuhl Seminars from 2006.11.20~2006.11.22

[more]

Recent and Selected Publications

- J. Chen, C. Lai, **X. Meng**, J. Xu: Clustering Moving Objects in Spatial Networks. To appear in *Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007)*, Bangkok, Thailand, April 9-12, 2007. (Full paper: 70/375=18.7%)
- Z. Xiao, **X. Meng**, J. Xu: Quality Aware Privacy Protection for Location-based Services. To appear in *Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007)*, Bangkok, Thailand, April 9-12, 2007. (Full paper: 70/375=18.7%)
- X. Li, W. Meng, **X. Meng**: EasyQuerier: A Keyword Query Interface For Web Database Integration System. To appear in *Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007)*, Bangkok, Thailand, April 9-12, 2007. (Short paper: 70+24/375=25.1%)

- [more]



Maintained by Zhongyuan Wang(zhywangchina@163.com)

http://idke.ruc.edu.cn/wamdm/

实验室人员

Professor

Dr. Xiaofeng Meng 孟小峰

Professor, Vice Dean of School of Information
Head of Dept of Computer Science,
School of Information , Renmin University of China

Ph.D. Candidate


Wei Liu
刘伟


Fangjiao Jiang
姜芳艽


Junfeng Zhou
周军锋


Jidong Chen
陈继东


Ling Wang
王玲


Yukun Li
李玉坤


Da Zhou
周大


Xiao Pan
潘晓

M.Sc. Student


Can Lin
林灿


Xian Li
李忺


Xiaofeng Wang
王小锋


Shaoyi Yin
尹少宜


Caifeng Lai
赖彩凤


Yanyan Ling
凌妍妍


Min Xie
谢敏


Xin Zhang
张新

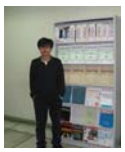

Zhen Xiao
肖珍


Linlin Jia
贾琳琳


Jing Huang
黄静


JinQing Zhu
朱金清


Li Xiang
向锂


Wei Wang
王伟


Ruilong Huo
霍瑞龙

Undergraduate


Zhongyuan Wang
王仲远


Jing Ai
艾静


Xiangyu Zhang
张相於


Junjin Xu
徐俊劲


Xing Hao
郝兴