

文件编号: ORIENTX-V2.0-ARCHITECTURE

版本号: V2.0

受控状态: 受控

发放号:

# OrientX 数据管理系统

中国人民大学 IDKE 实验室

编写人: 陆世潮

<[OrientX@ruc.edu.cn](mailto:OrientX@ruc.edu.cn)>

## 目录

OrientX 数据管理系统.....	1
1. OrientX 系统概述.....	2
1.1 OrientX 系统是什么系统? .....	2
1.2 OrientX 系统的应用背景.....	2
1.3 变化.....	2
1.4 系统功能.....	2
1.5 系统特征.....	3
2. OrientX 系统的设计.....	5
2.1 总体设计 (Architecture Design) .....	5
2.2 存储管理(Storage).....	6
2.3 数据管理(Data Manager) .....	7
2.4 模式管理(Schema Manager) .....	7
2.5 索引管理(Index manager) .....	7
2.6 执行引擎(Execute Engine).....	7
3.总结.....	9

# 1. OrientX 系统概述

## 1.1 OrientX 系统是什么系统？

OrientX 是一个 Native XML 数据库管理系统 (Native XML DataBase Management System)。OrientX 是 **O**riginal **RUC** **IDKE** **N**ative **X**ML 的缩写，表示 OrientX 系统是中国人民大学 IDKE 实验室研发的 Native XML 数据管理系统。该系统的研发工作是从 2001 年至今，已经历时 3 年时间。

## 1.2 OrientX 系统的应用背景

随着因特网应用的发展，XML 逐渐成为数据描述和数据交换的标准，大量的 XML 文档出现在网络中；有效地存储 XML 数据并提供高效的 XML 数据查询，成为当前急需解决的问题。

最直接的处理办法就是，把 XML 数据存储到关系数据库中，并使用 SQL 查询作为数据存取的方法。这种方法可以利用现有成熟的关系数据库产品；它可以依据成熟完备的关系数据库理论来解决数据的存储和存取过程中的各种问题。

但是，用关系数据库来存放 XML 数据存在很多问题：将树状结构的 XML 数据转换成关系数据库的二维关系表形式时面临语义信息丢失的问题；XML 查询（例如 XPath 和 XQuery）等不能直接在关系数据库上执行，需要转换成 SQL 查询；而且其关系表形式的查询结果还必须得还原成树状形式的 XML 数据，这将会导致效率的严重下降。

因此，人们开始研究新的数据存取方式。OrientX 系统正是在这样的应用背景下产生的，它以 Native 方式存储 XML 数据，保留 XML 数据的树状模型，并支持 XPath 和 XQuery 等 XML 查询以读取数据。所谓的 XML 的 Native 存储方式，就是存储时保留数据的树形模式；根据一个结点可以直接找到其孩子结点、左右兄弟结点或父亲节点等。以 Native 方式存取 XML 数据，就无需进行数据模式的转换，也不需要查询语言的转换。

## 1.3 变化

OrientX 2.0 版本在 1.5 版本上作了如下变化：

- 增加支持 XQuery 的导航式处理
- 扩展 XQuery1.0 以支持更新功能。
- 局部程序结构的调整和算法的优化。

## 1.4 系统功能

OrientX 是 Native XML 数据库管理系统，具有如下的**系统功能**：

### 1. 数据库的建立和维护

OrientX 系统的把具有相同的或相似的模式 (schema) 的 XML 文档集合存放在同一个数据库中（在 OrientX 中称为数据集 DataSet）。此功能包括 Schema 的导入和导出，

数据库的建立和删除等。

## 2. 数据操纵.

此功能包括对数据库数据的检索、插入、修改和删除等基本操作。

当前版本的 OrientX 系统的数据检索支持 XQuery1.0, 以抽取 XML 文档片断, 重新组合、构造新的 XML 文档。(以前版本只支持 XPath, 它只能抽取 XML 文档片断, 却无法构造新的结果)。

OrientX 所支持的 XQuery 遵循 W3C 推荐的标准草案。但是该草案的 XQuery 只支持查询, 无法支持数据更新等功能。为此, OrientX 系统扩展了 XQuery, 支持插入结点, 更新属性值和文本值, 删除结点等功能。

## 3. 数据库运行管理

**OrientX 系统目前支持简单的数据库运行管理。**

## 4. 数据组织、存储和管理功能

对 XML 数据、模式信息、存取路径等数据进行分门别类地组织、存储和管理, 确定以何种文件结构和存取方式物理地组织这些数据, 如何实现数据之间的联系, 以便提高存储空间利用率以及提高查找、增、删、改等操作的时间效率。

## 5. 安全访问控制管理

为了防止 XML 数据受到非法访问, OrientX 引入了安全访问控制管理。OrientX 为每一个数据集提供了一组结点级的访问角色, 包括读(read)、写(write)、管理(manage)等角色。通过 XPath 路径来确定用户可以访问的结点, 并赋予与该结点相应的访问权限。

## 6. 应用程序开发 API

OrientX 系统提供一套程序开发的 API 接口; 用户可以通过此接口基于 OrientX 系统开发自己的产品, 实现 XML 数据的存储、检索和管理。

值得注意的是, 与现在成熟的关系数据库管理系统产品相比, OrientX 系统还没有支持下列的功能: 例如并发控制等数据库运行管理; 数据库的转储与恢复、数据库的重组与重构、性能的监视与分析等系统维护功能。

# 1.5 系统特征

OrientX 系统作为一个数据库管理系统, 它具有如下的**系统特征**:

### 1. 它是基于模式 (Schema based) 的 Native XML 数据库管理系统。

Native XML DBMS 有是否基于模式 (schema-based) 之分。所谓基于模式, 就是创建数据库时需要指定相应的模式信息; 对数据的检索, 更新等操作也需要指定该操作对象的模式。OrientX 系统是基于模式的, 它为每个数据库保存了其相应的模式信息。因此查询数据或修改数据时只需要说明操作对象所在的数据库即可。

有了模式, 就可以判断指定的 XML 文档是否符合指定的模式; 在查询时, 可以利用模式信息加速查询处理。

### 2. 数据模型遵循 W3C 推荐的 XML Data Model 标准。

OrientX 系统处理的数据遵循 W3C 推荐的 XML 数据模型(<http://www.w3.org/XML>), 数据模式符合 W3C 推荐的 Schema 规范(<http://www.w3.org/TR/xmlschema-0/>)。因此, 只要是符合 W3C 推荐规范的数据, 都能够导入 OrientX 系统进行存储和管理。

### 3. 多样化的数据组织和存储方式

所谓数据组织和存储方式，就是如何划分 XML 数据组成一个个的记录，如何存储这些记录的关系等。OrientX 系统支持四种存储策略：Element-based(EB)、Subtree-based(SB)、Logical Clustering Storage (LC)、Clustering Element-based (CEB)。根据用户查询数据的需求和针对文档的不同特点，系统将会选择合适的存储策略。当文档比较小的时候，采用 EB 或者 CEB 方法；当文档比较大的时候，使用 LC 和 CEB 方法。对于文档性质比较强的文档，采用 EB 或者 CEB 方法；对于数据性质比较强的文档，采用 LC 或者 CEB 方法。

### 4. 数据存取路径

所谓数据存取路径，是指如何从数据库中读取数据，是顺序扫描、还是通过索引随机读取等。OrientX 系统提供最常见的类似于 DOM 接口，可以根据当前 XML 数据结点导航遍历其父亲、孩子和左右兄弟结点；此外还支持三种索引：基于元素的索引 (Elementary Index)，路径索引 (Path Index) 和值索引 (Value Index)。

所谓基于元素的索引 (Elementary Index)，是指对元素名称 (element name) 进行索引；有了此索引，就可以一次性读取到指定名称的所有元素结点在数据库中的地址 (Element node)。

所谓路径索引 (path index)，就是对一个 XPath 路径进行索引；通过该索引，可以返回匹配指定路径的元素有序对。例如对路径 A//D 建立索引，通过此索引就可以返回一系列的 (A, D) 有序对，它指示着符合路径 A//D 的嵌套关系 (父子或祖先后代关系) 的结点对在数据库中相应的地址。

所谓值索引 (Value Index)，类似关系数据库中的索引，它可以直接定位指定的 text 值或 attribute 值的元素结点。例如，假设在 book/@price 上建立值索引了，那么我们通过该索引就可以定位 price=100 的元素结点。

### 5. 数据检索支持 W3C 推荐的 XQuery 1.0 标准

如同关系数据库需要用 SQL 语言来查询数据一样，XML 数据库也需要有一种查询来从海量的 XML 数据检索出满足特定条件的数据片断。在诸多的 XML 查询语言中，W3C 推荐的 XQuery 脱颖而出，逐渐成为 XML 数据检索的标准。

OrientX 系统所支持的 XML 数据查询语言遵循 W3C 推荐的 XQuery 标准 (<http://www.w3.org/TR/xquery>)。

虽然 OrientX 系统能够解析所有的符合该规范的 XQuery 查询，但是它并没有处理所有的 XQuery 情形，例如 XQuery 中关于数据类型的表达式 (cast as、instance of、type switch 等)，用户自定义的函数，部分内嵌的函数 (如取系统时间，日期等)，这些情形的 XQuery，当前版本的 OrientX 暂时没有支持。OrientX 处理了 XQuery 的核心部分，例如 FLWR Expr，Construct Expr，XPath Expr 等。

### 6. XML 数据更新

关于 XML 更新，目前也没有得到业界公认的操纵语言标准。OrientX 系统扩展了 XQuery (称为 XQuery-Update)，支持插入结点，更新属性值和文本值，删除结点等功能。更新操作的目标结点，是通过原有的 XQuery 来定位；然后再指定操作类型和更新后的数值等，即可完成操作。

扩展 XQuery 以支持更新操作，有如下的两个好处：其一，它只是对 XQuery 的简

单扩充，对用户来说，这样的更新语言简单易学；其二，对于系统实现，可以利用现有的查询引擎来定位目标结点，这样能省去很多的不必要的工作。

## 2. OrientX 系统的设计

本节介绍 OrientX 系统的总体框架和各个组成模块的功能。至于各个模块的概要设计请参照其相应的概要设计说明书。

### 2.1 总体设计（Architecture Design）

描述 OrientX 系统的总体架构的最好方法是，通过检查它是如何导入一个 XML 文档并存储到数据库中去；然后在其上进行一个 XML 查询，看看它是如何实现的，如何导出文档的实现过程。

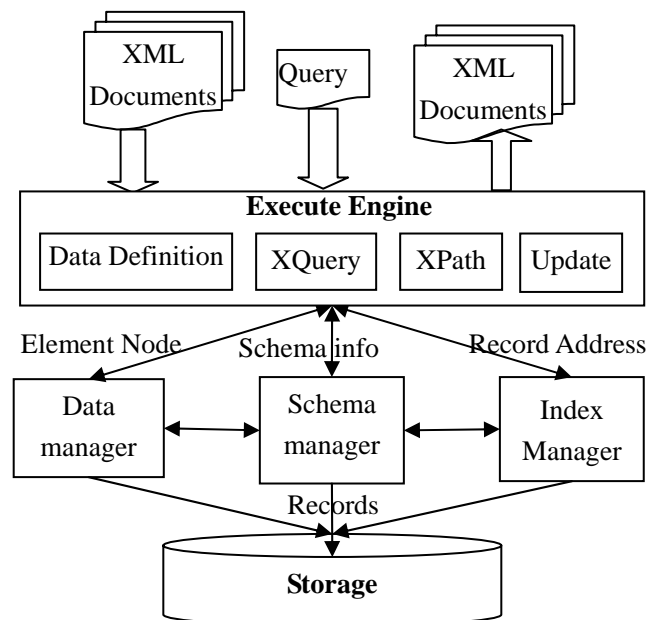


图 1 系统框架图

如图 1 所示，用户用 OrientX 系统管理 XML 数据，首先需要通过执行引擎（Execute Engine）模块建立一个数据库（前面已经说过，OrientX 系统中每一个数据库对应于具有相同的模式的数据集 dataset）。这就是数据定义（Data Definition），它确定了数据集内的所有文档的模式结构。导入文档时，执行引擎把文档传送到数据管理（Data Manager）模块；数据管理模块从逻辑上把 XML 文档划分成多个记录。然后传输到存储（Storage）模块，选择适当的文件机构进行存储。

当需要对数据进行查询检索时，一个 XML 查询（XPath 或者 XQuery 查询）以文本的形式传送到查询执行引擎（XQuery 和 XPath 执行引擎的处理策略有很大不同）；在查询执行引擎中，XML 查询将被分析(parse)成一个查询执行计划，此过程中从模式管理模块（Schema manager）读取相关信息，判断该查询是否存在语义错误，例如目标文档或数据库是否存在，XPath 路径中的结点在对应的模式中是否存在等问题；如果存在这样的错误，则系统就报告错误，查询不再往下执行。查询执行引擎还可以对查询计划进行优化；如果存在

合适的索引可以优化查询执行效率, 查询执行引擎就可以通过索引管理模块(index manager) 直接访问数据库, 而不需要通过数据管理模块(Data Manager) 导航式地访问数据库。

## 2.2 存储管理(Storage)

存储管理是 OrientX 系统的最底层的模块, 它负责 XML 文档、Schema 文档、索引文件、以及系统临时文档的存储, 其具体功能包括: 划分记录块, 确定物理文件结构和物理页内的记录顺序, 缓冲系统最近访问的物理页, 为上层的模块存取数据提供统一的接口。

存储管理包括如下几个子模块:

- **文件管理模块 (File manager)。**

OrientX 系统仍然是建立在操作系统之上, 它是以文件的方式对磁盘的访问的。例如, OrientX 系统为每个数据库(数据集) 向操作系统申请了几个大的文件的, 该数据库中的数据(包括模式信息等) 都是存放在这些文件上的。

文件管理模块为整个系统负责管理所有的涉及磁盘文件创建、打开、删除和读写的操作, 实现文件读写指针的共享, 为上层存储模块提供了独立于操作系统的接口; 这样, 上层模块对磁盘的访问, 就可以直接对“磁盘页面”进行读写删除等访问, 而不需要知道这些磁盘块如何组织在文件中的。

这模块暂缺概要设计文档。

- **存储管理(Storage Manager)**

本模块的主要功能是, 提供统一的接口, 使上层尽量独立于本部分的具体实现方式。

本模块提供的基本操作集是: 对 DataSet 的建立和删除; 对 DataSet 内 XML 文档的建立和删除; 向上层模块提供一个物理页面为单位进行页面申请和读写操作的接口, 管理系统物理存储空间的分配与回收。

此模块的概要设计请参照文档“Storager 概要设计”

- **缓冲区管理(Buffer Manager)**

OrientX 系统为了减少 I/O 次数, 缓冲了最近访问过的数据块。

上层的模块对缓冲管理模块的基本操作有: 读一块物理页面到缓冲区、释放一块缓冲区、申请一块新的物理页面、释放一块物理页面(注意, 所有对数据库的操作都通过缓冲管理模块, 所以申请和释放物理页面也是通过缓冲模块, 再由缓冲模块调用底层的存储模块的方法)。

页面淘汰算法将使用最近最少使用算法。

关于此模块的概要设计请参照文档“BufferManager 概要设计”和“缓冲管理模块设计”。

- **存取管理(Access Manager)**

此模块的功能主要是, 对存储模块(Storage Manager)和缓冲区模块(Buffer Manager) 进行包装, 向上层数据管理模块提供统一的接口, 使上层直接通过这个接口来存取数据。关于此模块的概要设计请参照文档“数据存取模块概要设计说明书”

## 2.3 数据管理(Data Manager)

DataManager 将 Parser 后的 Dom 树转化为实际存储的物理纪录（导入文档），同时提供接口，使得上层用户能够根据需要在已存储的 XML 文档节点间遍历（导航式访问数据），并能够对已存在的 XML 文档节点进行增加、删除和修改；并能都重新生成 XML 文档文件（导出数据）。

向上层查询模块提供接口的是 DataManager 类。它通过调用 Schema 类实现模式文档的解析和数据集的建立，通过调用 ImportHandler 类实现 XML 格式的文档的导入，上层模块利用它的导航接口可以提取想要的对象。

此模块的概要设计请参照“数据管理模块概要设计”和“DataManager 概要设计”，后者是于 2002 年 5 月写的；后者是于 2004 年 3 月写的；该模块的设计在这期间作了一些调整。

## 2.4 模式管理(Schema Manager)

SchemaManager管理的对象是系统的元信息，主要是模式（Schma）信息。Schema类似于与关系数据库中的数据字典。数据字典说明了数据库中的表结构、索引结构等信息。在 OrientX 中，一个数据集对应于一个 Schema，每一个数据集由多个符合同样 Schema 定义的文档组成。Schema 有两大作用：首先，Schema 类似于 DTD 或者 XML Schema 的作用，说明了文档的结构特征。其次，Schema 还记录了关于数据集的索引、引用和被引用关系等。

Schema Manager 在查询中的作用是：当一个查询提交时，首先用 Schema 检查该查询是否合法，比如路径是否存在，数据类型是否匹配等；如果合法，则根据 Schema 把查询中的 TagName 转换成结点名内部表示 EID。

关于此模块的概要设计，请参照文档“MetaDataManager 设计概要”。

## 2.5 索引管理(Index manager)

与关系数据库一样，为了加速数据访问的速度，Native XML 数据库管理系统也可以提供索引。OrientX 系统支持的索引有：基于元素的索引（Elementary Index），路径索引（Path Index）和值索引（Value Index）。

根据基于元素的索引（Elementary Index），就可以一下子读取到指定名称的所有元素结点在数据库中的地址（Element node）。此索引通常用于嵌套连接（containment join）。通过路径索引（Path Index），可以返回匹配指定路径的元素有序对。例如对路径 A//D 建立索引，通过此索引就可以返回一系列的 (A, D) 有序对，它指示着符合路径 A//D 的嵌套关系（父子或祖先后代关系）的结点对在数据库中相应的地址。值索引（Value Index）可以直接定位指定的 text 值或 attribute 值的元素结点。例如，假设在 book/@price 上建立值索引了，那么我们通过该索引就可以定位 price=100 的元素结点。

关于索引管理模块，暂缺设计文档。

## 2.6 执行引擎(Execute Engine)

执行引擎主要包括 XPath 查询引擎，XQuery 查询引擎，Update 引擎。而数据定义(Data

Definition)则是对数据管理模块所提供的建立/删除数据库，导入/导出/删除文档等功能进行封装。

执行引擎对系统的主要功能进行封装，向用户提供简洁明了的调用接口。

前面已经提及，OrientX 的更新功能是基于对 XQuery 的扩展的；而 XPath 则是 XQuery 的一个子集。因此，XPath 查询引擎，XQuery 查询引擎，Update 引擎这三个模块共享了一个查询分析程序，来对它们的输入（文本形式的查询表达式）进行分析，生成相应的执行计划，再由相应的执行引擎去执行。

### 2.6.1 查询分析(Query parser)

XML 查询是通过交互式或文件批处理方式提交的；无论是哪一种的提交方式，在查询执行引擎看来，所接收到的查询语句都看作是**字符流**。显然地，这是不能够直接让查询执行引擎进行处理；必须把字符流形式的查询语句转换成查询引擎可处理的内部结构，同时检查其是否符合语法，满足一定的语义要求。

经过此模块的分析，XML 查询被转换成内部结构，传递到查询执行引擎中去。

对于 XPath 查询，先是用 XQuery 的语法规则（请参照文档“XQuery 语法规则的改写”）分析，生成相应的语法树（文档“XQuery Syntax Parse 概要设计”），然后再对语法树进行分析，生成 XPath 查询引擎所能识别的内部执行计划（请参照文档“XPath 查询分析”）。

对于 XQuery 和 XQuery-Update，也是用同样的语法分析程序进行语法解析，但与 XPath 查询不一样的是，它只有逻辑上的语法树，没有生成物理的语法树；进行语法分析同时，生成相应的执行计划。

为了实现 XQuery 的导航式处理，我们定义了 13 个操作符，用于明确定义 XQuery 的语义，在查询引擎内部表示 XQuery 的执行计划，操纵 XML 数据等。这些操作符都对应着 XQuery 表面语法中的一个或多个产生式；XQuery 表面语法中的每一个产生式可以用 XAlgebra 中的一个或多个操作符迭代表示出来。这些对应关系是**有限确定的**（可以通过穷尽法证明）。

因此，从 XQuery 语法生成相应查询执行计划的算法，是基于 XQuery 语法产生式的。也就是，在对 XQuery 查询进行语法分析时，每当按照某一个语法产生式进行规约，如果该语法产生式对应于某个操作符，就构建新的操作；而产生式右边的子表达式的相应的子操作，作为新的操作结点的子操作。当查询的语法分析完毕，与语法树的根节点所对应的操作树，就是查询的执行计划

### 2.6.2 XPath 查询引擎

Optimizer 是 XPath 处理流程中的第二步，采用逻辑优化的方法，制定一系列优化规则，把 Parse 得到的内部路径根据规则转化成优化的查询计划。同时吸收了物理优化的思想，在优化规则中利用模式、索引等信息进行查询优化。

其关键的思想是：先利用模式信息对查询路径进行预处理，再用最小分解的方法处理复杂路径，对每一个子路径，再利用索引信息进一步分解，以获得最优的执行方案。

查询执行，就是按照一定次序去完成执行计划上的操作。执行的结果是一棵 XML 子树，可供进一步操作，也可以打印成一个 XML 文档。

关于此模块的概要设计请参照文档“XPath 处理模块概要设计”。

### 2.6.3 XQuery 查询引擎



XQuery 的处理，根据其读取数据的方式，可以分为导航式处理和代数式处理。OrientX 系统在当前版本只支持导航式处理。

导航式处理 XQuery，其主要思想是，从文档的根节点开始遍历 XML 文档，每次一结点地读取数据，构建新的结点。而查询的执行计划中的每一个操作，也是一次一结点式地操纵数据。关于导航式处理的具体构思，请看论文“OrientX 系统中的 XQuery 导航式实现”。

关于此模块的概要设计，请参照文档“导航式处理 XQuery 的概要设计”。

#### 2.6.4 Update 引擎

前面已经提及，OrientX 系统的更新语言是扩展 XQuery 语言。因此，XQuery 查询引擎负责定位更新操作的目标结点；而 Update 引擎的主要功能侧重于如何更新操作本身，以及如何检验该操作是否符合该文档的模式要求，只有符合模式约束的操作才会被执行等。

关于此模块的概要设计，请看文档“XQuery-Update 模块概要设计”。

## 3. 总结

OrientX 系统是一个 Native XML 数据库管理系统，可供管理 XML 数据，具体包括导入模式（schema）文档创建数据库、删除数据库、导入存储 XML 文档、导出数据库或 XML 文档、XPath 和 XQuery 查询，XML 数据更新等功能

OrientX 系统为 XML 数据管理相关研究工作提供了一个实验平台的。

OrientX 系统 version2.0 提供了 Client-Server 的应用体系结构；用户可以在客户端通过命令进行数据管理（建立删除数据库、导入导出文档、数据查询等操作）

OrientX 系统提供一套 API 接口，支持 XML 数据管理的应用开发。

关于 OrientX 系统的功能演示，请登录我们的主页：<http://idke.ruc.edu.cn/OrientX>

如果用户需要 OrientX 系统相关的程序包、源代码、说明文档，请向我们索取：[OrientX@ruc.edu.cn](mailto:OrientX@ruc.edu.cn) 或登录我们的主页：<http://idke.ruc.edu.cn/OrientX>