

中华人民共和国国家标准

GB/T 21063.2—2007

政务信息资源目录体系 第2部分：技术要求

Government information resource catalog system—
Part 2: Technical requirement

2007-09-10 发布

2008-03-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

前 言

GB/T 21063《政务信息资源目录体系》目前分为以下六个部分：

- 第1部分：总体框架；
- 第2部分：技术要求；
- 第3部分：核心元数据；
- 第4部分：政务信息资源分类；
- 第5部分：政务信息资源标识符编码规则；
- 第6部分：技术管理要求。

本部分为 GB/T 21063 的第2部分。

本部分的附录 A、附录 B、附录 C、附录 D 为规范性附录，附录 E、附录 F 为资料性附录。

本部分由国务院信息化工作办公室提出。

本部分由全国信息技术标准化委员会归口。

本部分起草单位：国家信息中心、北京航空航天大学、中国电子技术标准化研究所、大唐电信科技产业集团、南大通用数据技术有限公司、清华大学、浙江建达科技有限公司。

本部分主要起草人：徐枫、马殿富、邓洁霖、吴志刚、宦茂盛、邢春晓、武晋平、于建军、鲁俊杰、高栋、吴焱、李静、焦文涛、陈文亭、张巧英、贺炜、徐俊杰。

政务信息资源目录体系

第2部分:技术要求

1 范围

GB/T 21063 的本部分规定了政务信息资源目录体系的基本技术要求和目录服务接口要求。本部分适用于规划和建立政务信息资源目录内容服务系统。

2 规范性引用文件

下列文件中的条款通过 GB/T 21063 的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

- GB/T 21062.2—2007 政务信息资源交换体系 第2部分:技术要求
- GB/T 21063.1—2007 政务信息资源目录体系 第1部分:总体框架
- GB/T 21063.3—2007 政务信息资源目录体系 第3部分:核心元数据
- GB/T 21063.4—2007 政务信息资源目录体系 第4部分:政务信息资源分类
- GB/T 21063.5—¹⁾ 政务信息资源目录体系 第5部分:政务信息资源标识符编码规则

3 术语和定义

GB/T 21063.1—2007 确立的以及下列术语和定义适用于 GB/T 21063 的本部分。

3.1

目录服务器 catalog server

按照目录服务接口的要求,提供政务信息资源发现和目录管理的计算机服务程序。

3.2

元数据库 metadata database

存储元数据的逻辑数据库。

3.3

结果集 resultset

根据目录检索请求在服务器端建立的查询结果集合。

4 技术要求

4.1 概述

目录内容服务系统由共享信息服务系统、编目系统、目录传输系统、目录管理系统、目录服务系统组成。

4.2 共享信息服务系统

共享信息服务系统由各政务部门基于统一的电子政务网络进行建设、管理和维护,其信息资源由各政务部门自身的业务信息系统提供、产生、发布并进行运行管理。共享信息服务系统应具备的基本功能要求包括:

1) 本标准即将发布。

- a) 共享信息资源生成:从各部门业务信息资源中生成本部门用于共享的信息资源;
- b) 共享信息资源发布:提供基于统一的电子政务网络的共享信息资源发布系统,发布共享信息资源;
- c) 共享信息资源访问:提供共享信息资源访问服务,用户可以浏览、查询、下载共享信息,并且基于统一的电子政务网络进行共享政务信息资源共享。

4.3 编目系统

编目系统根据部门共享信息资源的内容,提取其基本特征,按照 GB/T 21063.3—2007、GB/T 21063.4—2007和 GB/T 21063.5—2007 实现元数据赋值,形成目录内容。

编目系统应遵循以下技术要求:

- a) 编目对象是具体的共享政务信息资源,主要内容包括数据集、档案、法律法规、文件、报告、服务等,具体的形式可以是数据库、图片、文档、音频、视频、网页、服务等,有关编目的具体粒度参见附录 E;
- b) 编目系统应支持自动、机辅方式完成元数据元素的赋值;
- c) 唯一标识符管理功能:按照 GB/T 21063.5—2007,支持唯一标识符的分配和赋值,包括支持后段码的自动生成和管理;
- d) 标准符合性检查功能:支持政务信息资源元数据完整性和标准一致性检查,元数据完整性检查的主要目的是保证所有必选的元数据实体和元数据元素已经赋值,标准一致性检查的主要目标是保证已填写的元数据实体和元数据元素的取值符合 GB/T 21063.3—2007、GB/T 21063.4—2007和 GB/T 21063.5—2007 的规定;
- e) 信息资源分类:按照 GB/T 21063.4—2007,实现对共享政务信息资源的分类。

4.4 目录传输系统

目录传输系统实现目录内容在部门目录内容信息库与目录服务中心的目录内容管理信息库之间的传输。目录传输系统应遵循以下技术要求:

- a) 建立传输通道:可按照 GB/T 21062.2—2007 建立部门与目录服务中心之间的传输通道;
- b) 目录内容传送:支持各个政务部门将目录内容自动提交、存储到目录中心的目录内容管理信息库;
- c) 目录内容回退:未通过目录服务中心审核的数据将回退到相应部门的目录内容信息库。

4.5 目录管理系统

目录管理系统实现对汇聚的目录内容进行自动审核和其他相关的管理工作。目录管理系统应遵循以下技术要求:

- a) 审核功能:按照 GB/T 21063.3、GB/T 21063.4 和 GB/T 21063.5,对目录内容进行自动形式审核,符合标准的目录内容自动进入目录内容服务信息库,按照标准要求自动发布;未通过形式审核的目录内容,自动回退到相关政务部门,经形式修正后重新审核。
- b) 维护功能:对目录内容管理信息库、目录内容服务信息库中的目录内容提供维护功能。
- c) 标识符前段码管理:提供前段码分配管理功能,为各政务部门分配并维护固定的前段码。
- d) 目录服务地址管理:存储目录服务中心的目录服务地址,用于目录服务中心之间的互连互通。
- e) 监控功能:监控系统的运行。

4.6 目录服务系统

目录服务系统基于网络实现对目录内容的发布,并向用户提供目录内容的查询。目录服务系统应遵循以下技术要求:

- a) 目录内容发布:提供目录服务接口,具体的目录服务接口定义见附录 A 和附录 B,其应包含的功能如下:
 - 1) 查询请求解析;

- 2) 目录内容服务信息库查询:向目录内容信息库发出目录检索查询请求;
 - 3) 查询结果封装与返回:按照目录服务接口技术要求,封装返回的查询结果,并返回给查询请求者。
- b) 目录内容查询:为了便于人机交互,需要提供基于网络的目录内容查询功能,目录内容检索涉及的检索串的示例参见附录 F,应支持的查询检索方式如下:
- 1) 分类导航;
 - 2) 单个检索词的检索;
 - 3) 多个检索词的组合检索。

附 录 A

(规范性附录)

目录服务接口

A.1 概述

目录服务包括发现和管理两种基本功能:发现功能用来对元数据进行检索,管理功能实现元数据远程管理。

政务信息资源目录服务包含3种接口,即基础接口、发现接口和管理接口。其中,基础接口是将发现接口和管理接口中基础性的操作定义成一个公共接口。基础接口和发现接口是必选实现,管理接口为可选实现。这3类接口实现了政务信息资源的发现功能和管理功能。

- a) 基础接口:提供会话管理功能和服务自描述功能,包含有目录服务初始化接口、目录服务终止接口和服务自描述接口。
- b) 发现接口:提供元数据检索功能和元数据检索结果提取功能,包含有目录检索接口以及目录检索结果提取接口。这些接口本身并不提供资源,而是提供资源基本信息和如何去获得这些资源的元数据。
- c) 管理接口:提供元数据管理的功能,包含元数据管理接口。

政务信息资源目录服务是基于超文本传输协议(HTTP)的POST方式实现,协议消息使用XML编码。目录服务各个接口操作均是通过客户端和服务端之间传递的请求/响应消息对来实现。请求消息和响应消息是一一对应的,即对每一个请求消息有且只有一个响应消息产生。目录服务的客户端和服务端通讯建立在会话的基础上,会话通过请求消息和响应消息对来完成,每一个请求消息都有相对应的响应消息。

政务信息资源目录服务支持核心元数据及其扩展内容的查询,该元数据需要符合GB/T 21063.3—2007对于政务信息资源核心元数据的要求。目录服务支持对多个元数据库的查询,元数据库一般按照描述政务信息资源的内容分别建立。

政务信息资源目录服务可以是集中式也可以是分布式的。集中式是指元数据库和目录服务在物理上都部署在一个节点。分布式指的是元数据库物理分布、目录服务逻辑集中,分布式目录服务的具体规定见附录D。

A.2 基础接口

A.2.1 目录服务初始化

A.2.1.1 描述

目录服务初始化接口用来建立客户端和服务端之间的会话,该操作将产生唯一的标识符用来跟踪会话。

A.2.1.2 请求

目录服务初始化请求消息用来请求建立客户端和服务端之间的会话。目录服务初始化请求的参数见表A.1:

表 A.1 初始化请求参数表

参数名称	可选/必选	参数含义
protocolVersion	必选	客户端支持的协议实现版本。由客户端在请求中指出其支持的所有版本,服务器端在响应中也同样指出其支持的所有版本。会话将以双方都支持的最高相同版本实现。如果没有相同所支持的版本,则初始化被拒绝。注:版本参数由1个正整数组成,各个正整数之间由逗号分隔,其形式为“1,2”
idAuthentication	可选	认证信息,包含用户身份标识及其密码,用于服务器端对用户身份进行验证。认证信息中的用户身份标识及其密码之间用冒号分开,其形式为“userId:password”
implementationId	可选	客户端实现标识,由目录服务实现厂商自行定义
implementationName	可选	客户端实现名称,由目录服务实现厂商自行定义
implementationVersion	可选	客户端实现版本,由目录服务实现厂商自行定义
otherInfo	可选	其他信息,用于客户端和服务端传递自定义信息

目录服务初始化请求的组成结构,如图 A.1 所示:

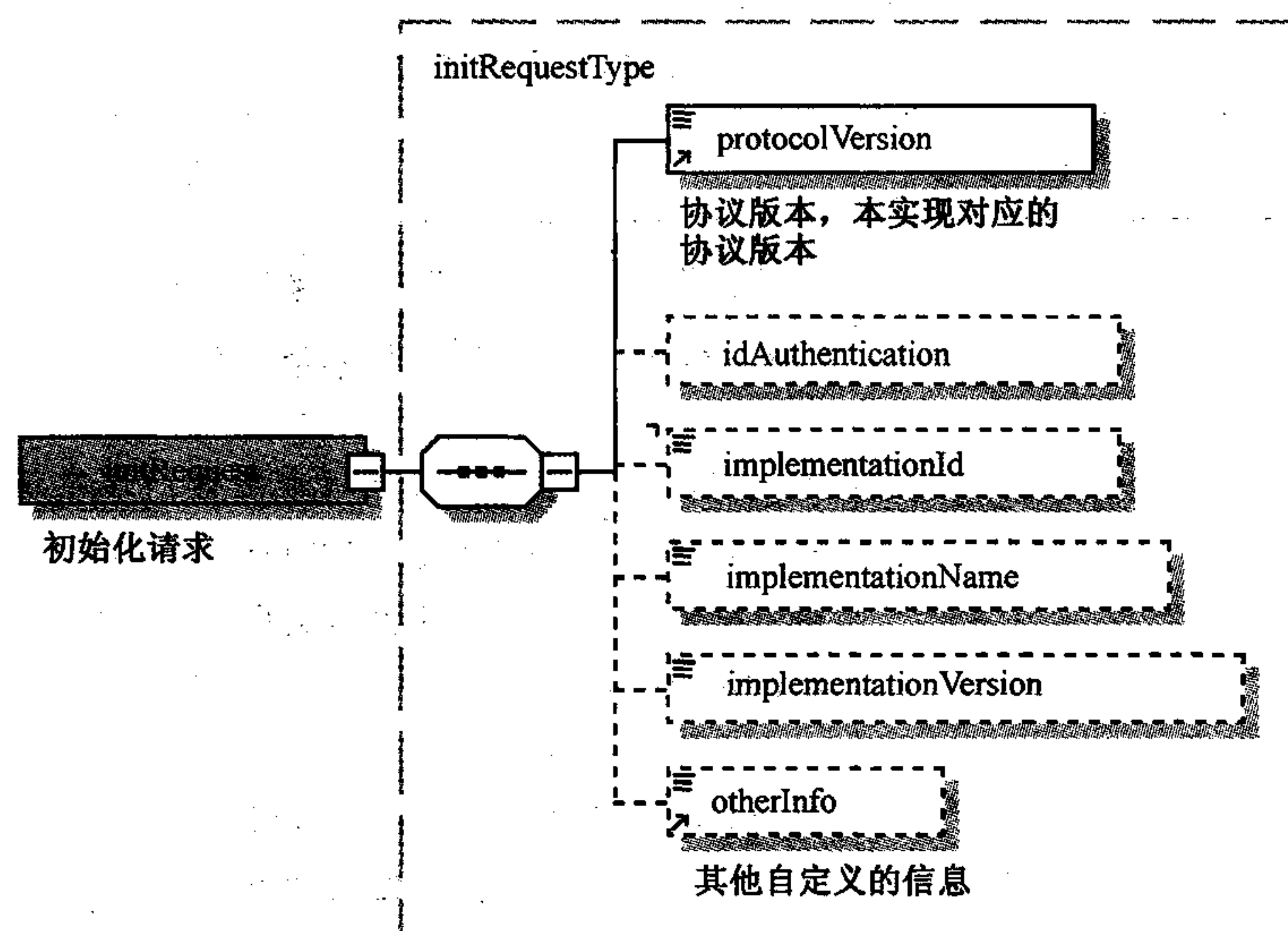


图 A.1 目录服务初始化请求的模型图

请求消息的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符 "InitRequest" 的定义 -->
<xsd:element name="initRequest" type="initRequestType">
  <xsd:annotation>
    <xsd:documentation>初始化请求。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="initRequestType">
  <xsd:annotation>
    <xsd:documentation>初始化请求的类型定义。</xsd:documentation>
  
```



```

</xsd:annotation>
<xsd:sequence>
  <xsd:element ref="protocolVersion"/>
  <xsd:element name="idAuthentication" minOccurs="0"/>
  <xsd:element name="implementationId" type="xsd:string" minOccurs="0"/>
  <xsd:element name="implementationName" type="xsd:string" minOccurs="0"/>
  <xsd:element name="implementationVersion" type="xsd:string" minOccurs="0"/>
  <xsd:element ref="otherInfo" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
<!--类型标识符 "ProtocolVersion" 的定义 -->
<xsd:element name="protocolVersion" type="xsd:string" default="1">
  <xsd:annotation>
    <xsd:documentation>协议版本,本实现对应的协议版本</xsd:documentation>
  </xsd:annotation>
</xsd:element>

```

A.2.1.3 响应

目录服务初始化响应消息用来确认在目录服务器和客户端间会话的建立,这个响应消息给出了唯一的会话标识。目录服务初始化响应的参数见表 A.2。

表 A.2 初始化响应参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识。会话标识是由 a~z、A~Z、0~9 组成,其他字符无效
protocolVersion	必选	服务器端支持的协议实现版本由客户端在请求中指出其支持的所有版本,服务器端在响应中也同样指出其支持的所有版本会话将以双方都支持的最高相同版本实现。如果没有相同所支持的版本,则初始化被拒绝
result	必选	服务器端用该参数表明是否接受建立会话的请求,该参数取值为布尔型。“true”表示建立会话成功;“false”表示建立会话失败,此时 referenceId 取为空
implementationId	可选	服务器端实现标识,由目录服务实现厂商自行定义
implementationName	可选	服务器端实现名称,由目录服务实现厂商自行定义
implementationVersion	可选	服务器端实现版本,由目录服务实现厂商自行定义
otherInfo	可选	其他信息,用于客户端和服务器端传递自定义信息。如发生异常,可使用本参数传递详细的异常信息给客户端

目录服务初始化响应的组成结构,如图 A.2 所示。

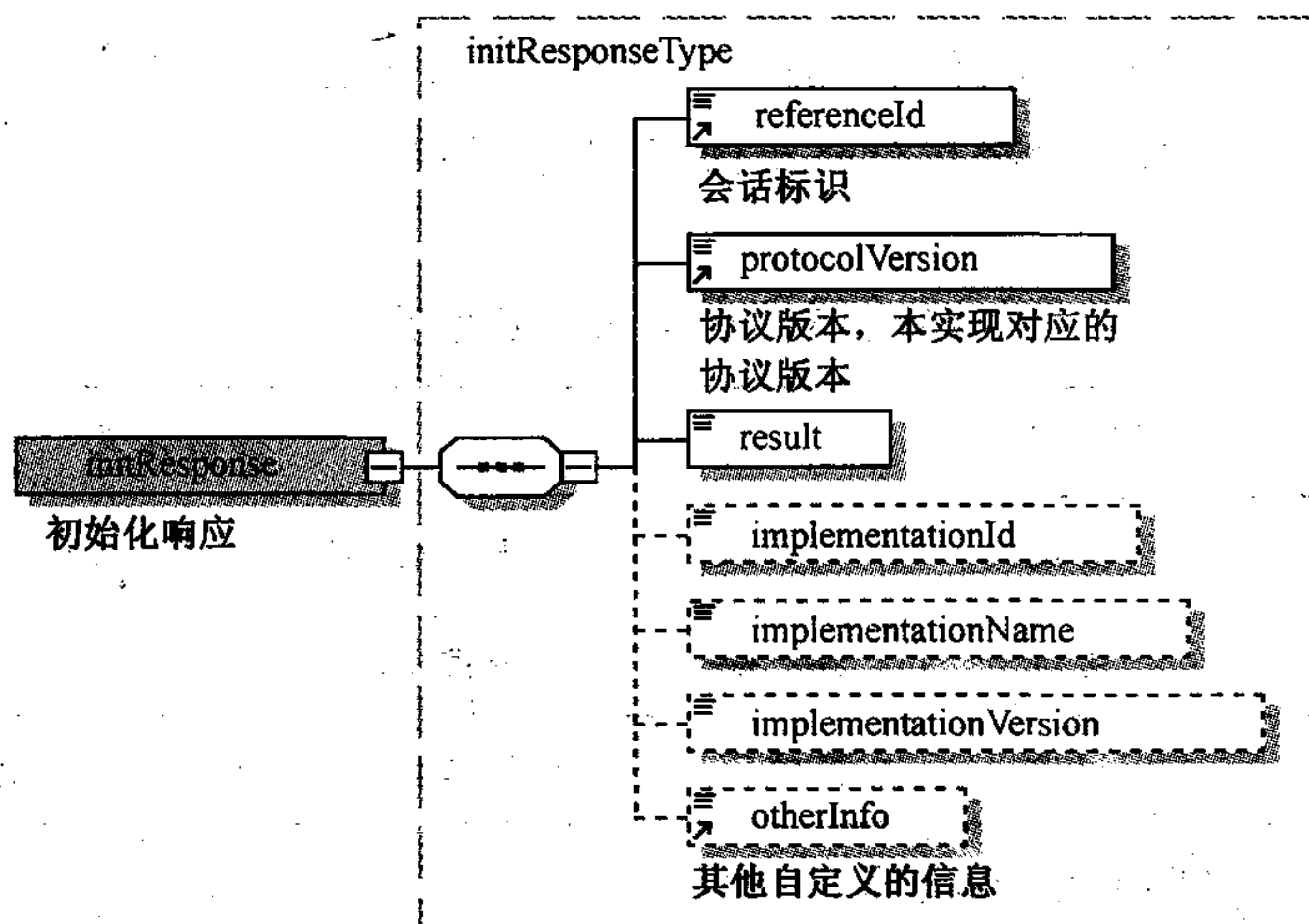


图 A.2 目录服务初始化响应的模型图

目录服务初始化响应消息的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```
<!--类型标识符 "InitResponse" 的定义 -->
```

```
<xsd:element name="initResponse" type="initResponseType">
```

```
  <xsd:annotation>
```

```
    <xsd:documentation>初始化响应。</xsd:documentation>
```

```
  </xsd:annotation>
```

```
</xsd:element>
```

```
<xsd:complexType name="initResponseType">
```

```
  <xsd:annotation>
```

```
    <xsd:documentation>初始化响应的类型定义。</xsd:documentation>
```

```
  </xsd:annotation>
```

```
  <xsd:sequence>
```

```
    <xsd:element ref="referenceId"/>
```

```
    <!--参数 referenceId 是由 a~z、A~Z、0~9 组成,其他字符无效。 -->
```

```
    <xsd:element ref="protocolVersion"/>
```

```
    <xsd:element name="result" type="xsd:boolean"/>
```

<!--参数 result 取值为布尔型。“true”表示建立会话成功;“false”表示建立会话失败,此时 referenceId 取为空。 -->

```
    <xsd:element name="implementationId" type="xsd:string" minOccurs="0"/>
```

```
    <xsd:element name="implementationName" type="xsd:string" minOccurs="0"/>
```

```
    <xsd:element name="implementationVersion" type="xsd:string" minOccurs="0"/>
```

```
    <xsd:element ref="otherInfo" minOccurs="0"/>
```

<!--其他信息,用于客户端和服务端传递自定义信息。一旦发生异常,使用本参数传递详细的异常信息给客户端。 -->

```
  </xsd:sequence>
```

```
</xsd:complexType>
```

A.2.2 目录服务终止

A.2.2.1 描述

目录服务终止接口用来终止当前的会话。

A.2.2.2 请求

目录服务终止请求消息用来申请终止当前的会话,该请求消息从客户端发送到目录服务器端。服务器端一旦收到该消息,目录服务器将执行终止操作,停止该会话的所有进程。目录服务终止请求的参数见表 A.3:

表 A.3 终止请求参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识
closeReason	可选	客户端为终止服务提供原因
otherInfo	可选	其他信息,用于客户端和服务器端传递自定义信息

目录服务终止请求的组成结构,如图 A.3 所示:

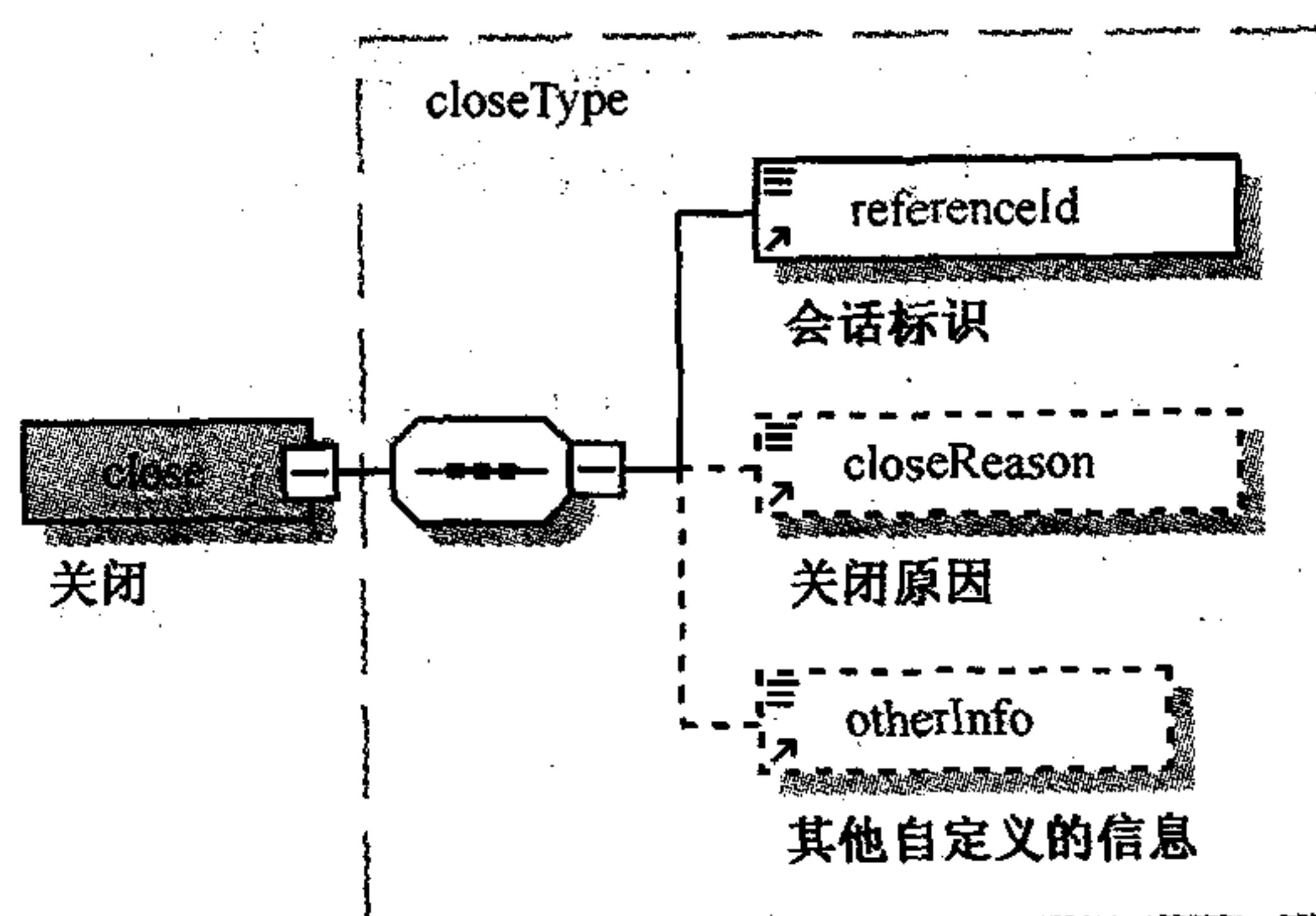


图 A.3 目录服务终止请求的模型图

请求消息的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符 "closeRequest" 的定义 -->
<xsd:element name="closeRequest" type="closeRequestType">
  <xsd:annotation>
    <xsd:documentation>关闭。 </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="closeRequestType">
  <xsd:annotation>
    <xsd:documentation>关闭的定义。 </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="closeReason" type="xsd:string" minOccurs="0"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

A.2.2.3 响应

服务器端返回目录服务终止响应,其参数的具体说明见表 A.4:

表 A.4 终止响应参数表

参数名称	可选/必选	参数含义
closeStatus	可选	服务器端终止会话的状态。该参数取值为“success”表示关闭会话成功;“failure”表示关闭会话失败

目录服务终止响应的组成结构,如图 A.4 所示:

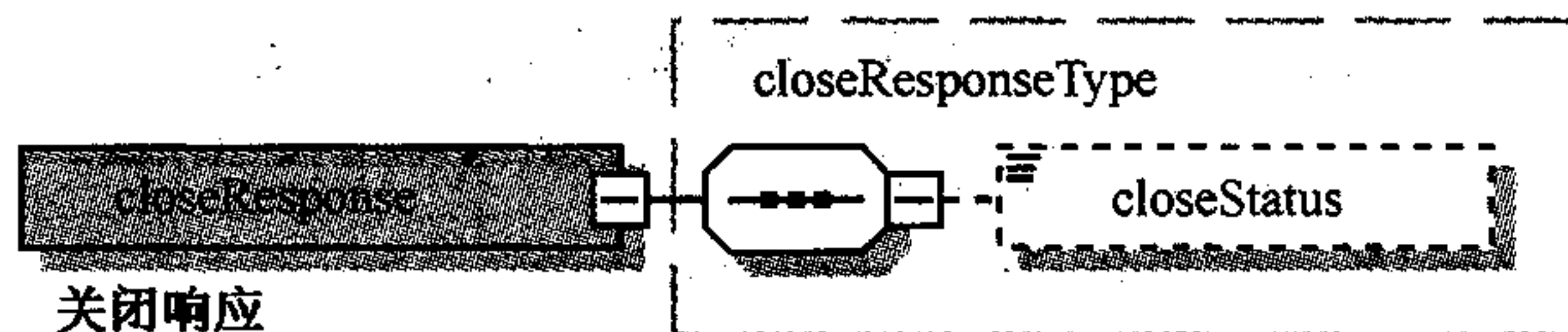


图 A.4 目录服务终止响应的模型图

请求消息的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符"closeResponse"的定义 -->
<xsd:element name="closeResponse" type="closeResponseType">
  <xsd:annotation>
    <xsd:documentation>关闭响应。 </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="closeResponseType">
  <xsd:annotation>
    <xsd:documentation>关闭响应的定义。 </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="closeStatus" type="xsd:string" minOccurs="0"/>
    <!-- 参数 closeStatus 取值为“success”表示关闭会话成功;“failure”表示关闭会话失败 -->
  </xsd:sequence>
</xsd:complexType>

```

A.2.3 服务自描述

A.2.3.1 描述

服务自描述接口通常用来提供目录服务自身的各种描述性信息。

A.2.3.2 请求

服务自描述请求消息由客户端产生并发送到目录服务器,用来请求目录服务的相关描述信息。服务自描述请求的参数定义见表 A.5:

表 A.5 服务自描述请求参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识
otherInfo	可选	其他信息,用于客户端和服务器端传递自定义信息

服务自描述请求的组成结构,如图 A.5 所示:

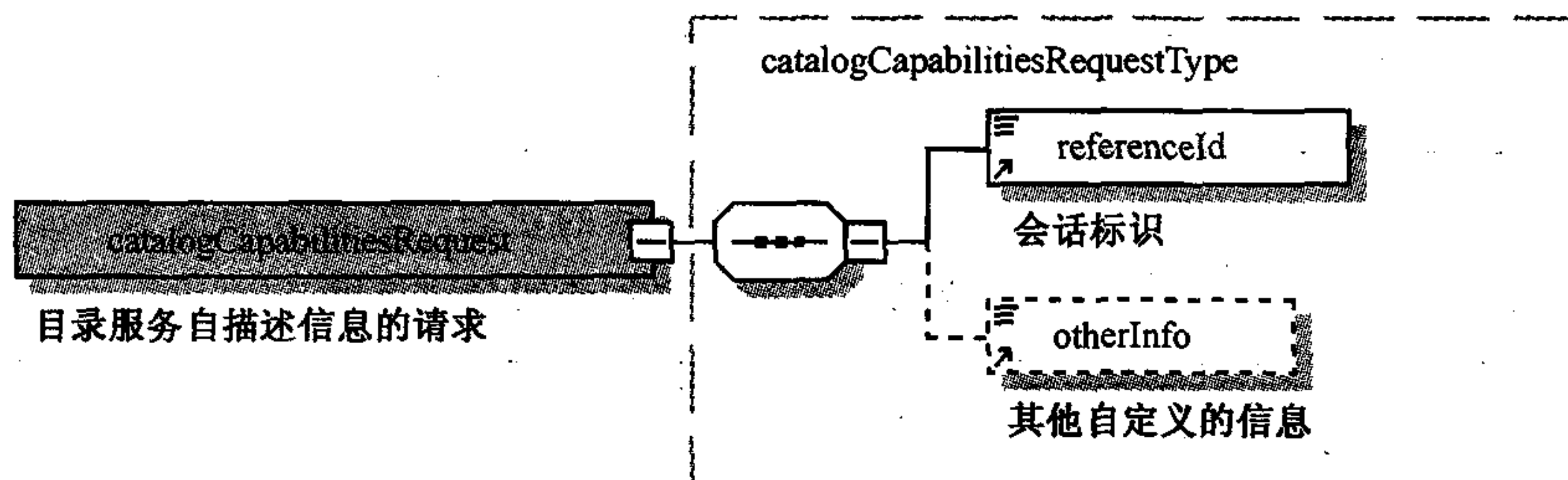


图 A.5 服务自描述请求的模型图

服务自描述请求的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```
<!-- 类型标识符"catalogCapabilitiesRequest"的定义 -->
<xsd:element name="catalogCapabilitiesRequest" type="catalogCapabilitiesRequestType">
  <xsd:annotation>
    <xsd:documentation>目录服务自描述信息的请求。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="catalogCapabilitiesRequestType">
  <xsd:annotation>
    <xsd:documentation>目录服务自描述信息请求的定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

A.2.3.3 响应

接收到客户端的服务自描述请求后,服务器端产生目录服务自描述响应消息,再通过该响应返回给客户端各目服务自描述信息。服务自描述响应的参数定义见表 A.6:

表 A.6 服务自描述响应参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识
databaseList	必选	用来返回目录服务关联的一个或多个元数据库信息列表,详细内容见表 A.8
Options	必选	功能选项,用来描述服务所支持的各项功能,包含有目录服务初始化、目录服务终止、服务自描述、目录检索、目录检索结果提取、元数据管理功能。由客户端在请求中建议目录服务需要的功能,服务器端根据自身的实现情况决定提供哪些功能。该参数的取值是枚举型,可以包含一个或多个功能参数值,诸如 init、close、capabilities、search、present、metadataManage
serviceName	必选	用于说明服务名称,由目录服务实现厂商自行定义。默认取值“政务信息资源目录服务”
serviceAbstract	必选	服务内容的简单说明,由目录服务实现厂商自行定义
serviceURL	必选	用来说明服务的网络地址,由目录服务实现厂商自行定义

表 A.6 (续)

参数名称	可选/必选	参数含义
serviceProvider	必选	用来说明服务的政务部门信息,由目录服务实现厂商继续扩展。参数是复合型,至少包含一个参数 serviceProviderName,表示提供服务的政务部门名称
serviceID	必选	服务的唯一标识,由注册系统统一分配。标识符须唯一,由系统自动随机产生使用字母(含下划线“_”)或数字组成
serviceType	必选	用于说明服务所属类型,由目录服务实现厂商自行定义,固定取值为“目录服务”
otherInfo	可选	其他信息,用于客户端和服务端传递自定义信息。若发生异常,可使用本参数传递详细的异常信息给客户端

服务自描述响应的组成结构,如图 A.6 所示:

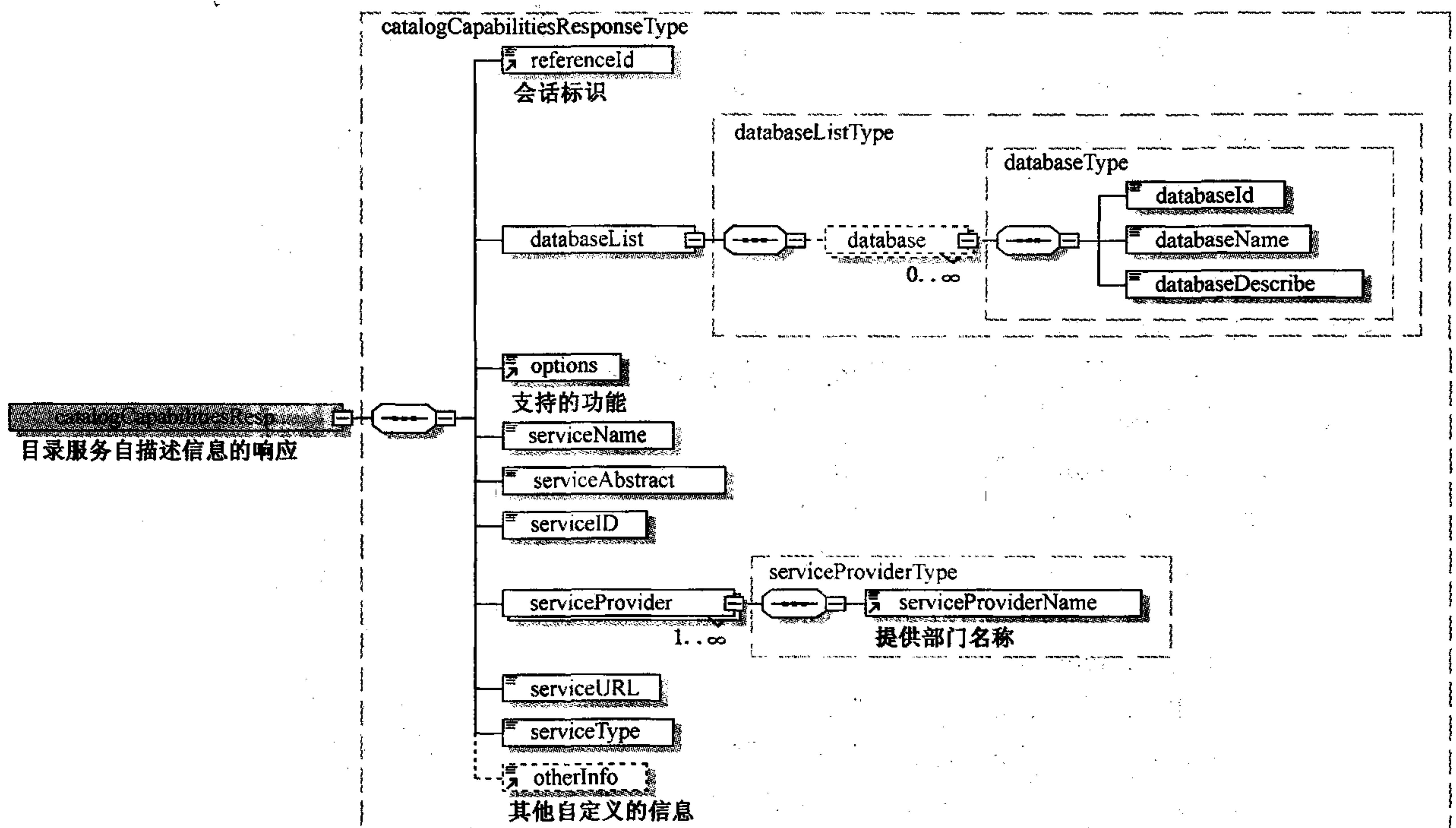


图 A.6 服务自描述响应的模型图

服务自描述响应的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符"catalogCapabilitiesResponse"的定义 -->
<xsd:element name="catalogCapabilitiesResponse" type="catalogCapabilitiesResponseType">
  <xsd:annotation>
    <xsd:documentation>目录服务自描述信息的响应。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="catalogCapabilitiesResponseType">
  <xsd:annotation>
    <xsd:documentation>目录服务自描述信息响应的定义。</xsd:documentation>
  </xsd:annotation>

```

```

<xsd:sequence>
  <xsd:element ref="referenceId"/>
  <xsd:element name="databaseList" type="databaseListType"/>
  <xsd:element ref="options"/>
  <xsd:element name="serviceName" type="xsd:string" default="政务信息资源目录服务"/>
  <xsd:element name="serviceAbstract" type="xsd:string"/>
  <xsd:element name="serviceID" type="xsd:string"/>
  <xsd:element name="serviceProvider" type="serviceProviderType"
maxOccurs="unbounded"/>
  <xsd:element name="serviceURL" type="xsd:string"/>
  <xsd:element name="serviceType" type="xsd:string" fixed="目录服务"/>
  <xsd:element ref="otherInfo" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
<!-- 类型标识符 "Options"的定义 -->
<xsd:element name="options" type="optionsType">
  <xsd:annotation>
    <xsd:documentation>支持的功能。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="optionsType">
  <xsd:list itemType="facilitySupportedType"/>
</xsd:simpleType>
<xsd:simpleType name="facilitySupportedType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="init"/>
    <xsd:enumeration value="close"/>
    <xsd:enumeration value="capabilities"/>
    <xsd:enumeration value="search"/>
    <xsd:enumeration value="present"/>
    <xsd:enumeration value="metadataManage"/>
  </xsd:restriction>
</xsd:simpleType>

```

复合型参数 serviceProvider 用来说明服务的政务部门信息,包含一个参数 serviceProviderName,表示提供服务的政务部门名称。serviceProvider 的具体内容见表 A.7:

表 A.7 serviceProvider 参数表

参数名称	可选/必选	参数含义
serviceProviderName	必选	表示提供服务的政务部门名称,由目录服务实现厂商自行定义

服务自描述响应中的 serviceProvider 的组成如图 A.7 所示:

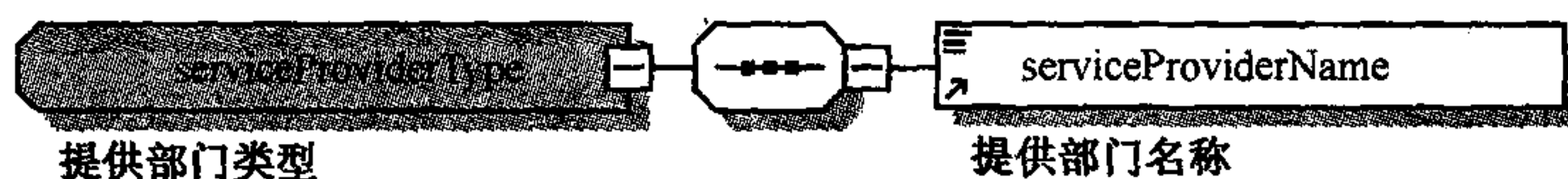


图 A.7 服务自描述响应中 serviceProvider 的模型图

服务自描述响应中的 serviceProvider 的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<xsd:complexType name="serviceProviderType">
  <xsd:annotation>
    <xsd:documentation>提供部门类型</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="serviceName"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="serviceName" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>提供部门名称</xsd:documentation>
  </xsd:annotation>
</xsd:element>

```

复合型参数 databaseList 是用于描述目录服务支持的所有元数据库信息。参数 databaseList 的具体内容见表 A. 8:

表 A. 8 databaseList 参数表

参数名称	可选/必选	参数含义
databaseId	必选	用于标识一个元数据库的标识符,由元数据库政务部门定义。该标识符是由 a~z、A~Z、0~9 组成,其他字符无效
databaseName	必选	用于说明元数据的名称,由元数据库政务部门定义
databaseDescribe	必选	用于提供元数据库的基本描述信息,由元数据库政务部门定义。诸如,元数据内容、范围、用途等

服务自描述响应中的 databaseList 的组成如图 A. 8 所示:

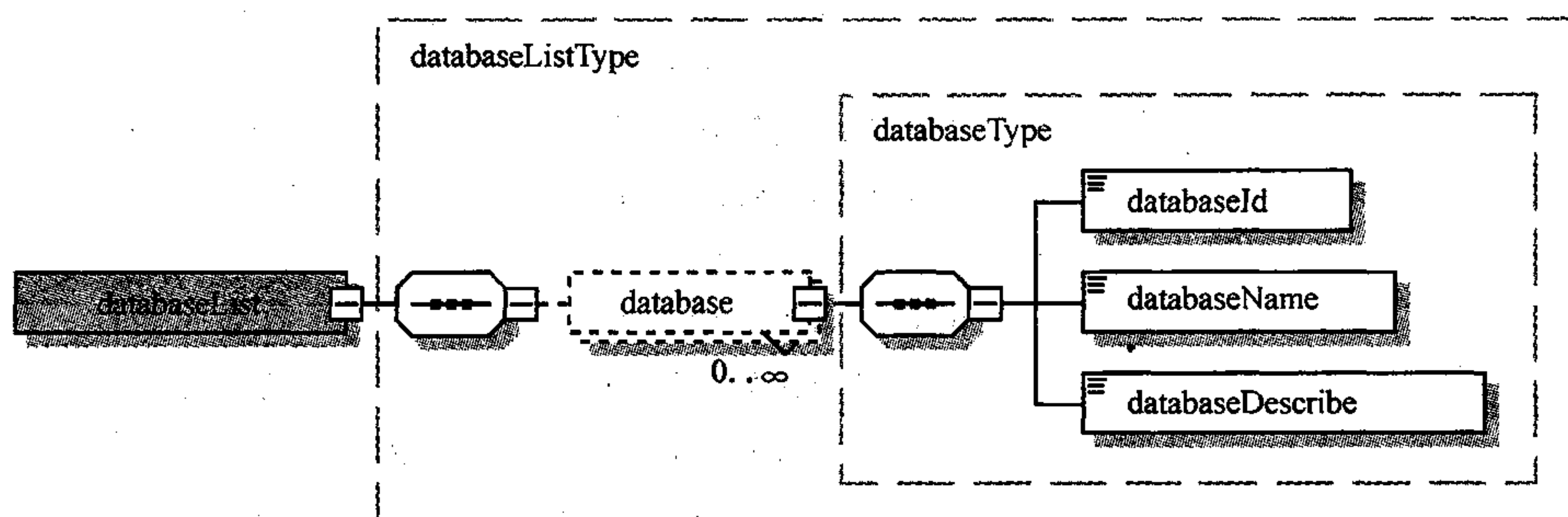


图 A. 8 服务自描述响应中“元数据库列表参数”的模型图

服务自描述响应中的 databaseList 的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<xsd:complexType name="databaseListType">
  <xsd:sequence>
    <xsd:element name="database" type="databaseType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>

```

```

</xsd:complexType>
<xsd:complexType name="databaseType">
  <xsd:sequence>
    <xsd:element name="databaseId" type="xsd:string"/>
    <xsd:element name="databaseName" type="xsd:string"/>
    <xsd:element name="databaseDescribe" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

```

A.3 发现接口

A.3.1 目录检索

A.3.1.1 描述

目录检索是目录服务器根据客户端指定的检索条件,在服务器端从元数据库中检索并标识出匹配的元数据记录,并将元数据记录保存在结果集中的过程。

每一个会话可进行多次目录检索,产生多个结果集。

A.3.1.2 请求

目录检索请求参数的详细说明见表 A.9:

表 A.9 检索请求参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识
replaceIndicator	必选	结果集覆盖标志,客户端指定的检索结果集如果已经存在,根据结果集覆盖标志,服务器端作不同的处理。该参数取值为布尔型,“true”表示允许覆盖结果集,则新产生的结果集将覆盖已经存在的结果集。“false”表示不允许覆盖已经存在的结果集。当客户端请求不是首次时,由于原结果集的存在且不允许覆盖,将导致检索操作失败
resultSetID	必选	由客户端定制检索结果集名称。当命名的结果集名称重复时,查看 replaceIndicator 参数决定是否覆盖结果集
databaseNames	必选	由客户端指定待检索的一个或多个元数据库名称。目录服务可以关联多个元数据库(元数据库的信息均通过服务自描述接口直接反馈给客户端),因此需要客户端指定检索的元数据库。该参数是复合型,包含一个或多个 databaseName 参数,用于封装一个或多个元数据库名称
Query	必选	由客户端定制的检索语句。检索串符合 Type-1 型,使用逆波兰表达式(RPN)作为检索语法
preferredRecordSyntax	可选	客户端指定检索结果的编码方式。该参数默认取值为“XML”,表示客户端指定检索结果集的编码方式采用 XML
otherInfo	可选	其他信息,用于客户端和服务器端传递自定义信息

目录检索请求的组成结构,如图 A.9 所示:

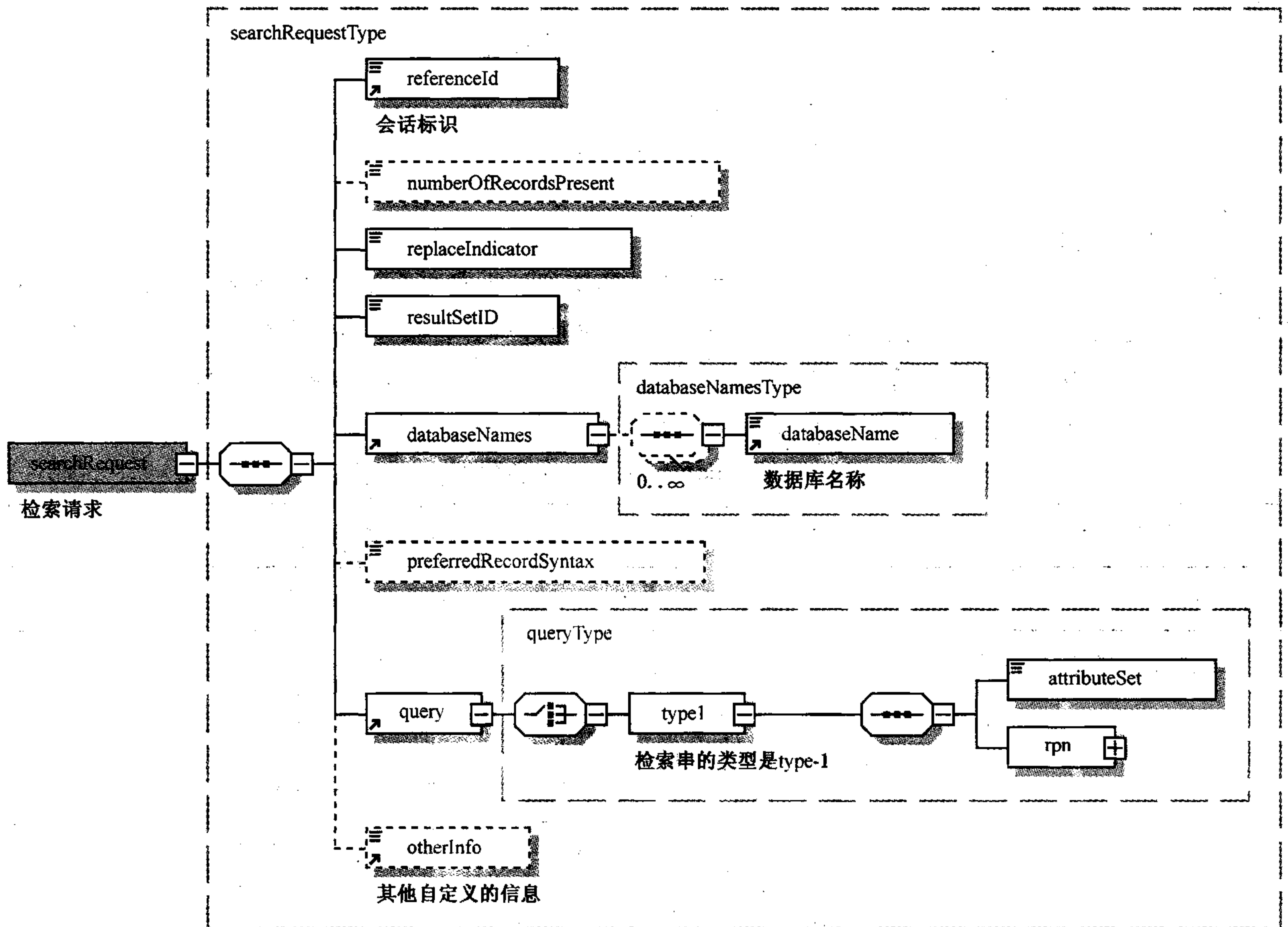


图 A.9 目录检索请求的模型图

目录检索请求的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```
<!-- 类型标识符"SearchRequest"的定义 -->
<xsd:element name="searchRequest" type="searchRequestType">
  <xsd:annotation>
    <xsd:documentation>检索请求。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="searchRequestType">
  <xsd:annotation>
    <xsd:documentation>检索请求的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="replaceIndicator" type="xsd:boolean"/>
    <xsd:element name="resultSetID" type="xsd:string"/>
    <xsd:element ref="databaseNames"/>
```

```

        <xsd:element name="preferredRecordSyntax" type="xsd:string" default="XML"
minOccurs="0"/>
        <!-- preferredRecordSyntax 表示查询结果的编码方式。默认采用 XML 方式 -->
        <xsd:element ref="query"/>
        <xsd:element ref="otherInfo" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

```

复合型参数 databaseNames 表示由客户端指定待检索的一个或多个元数据库名称。因目录服务可以关联多个元数据库(所有目录服务支持的数据库信息均通过服务自描述接口直接获取到客户端),故需要客户端指定检索的元数据库。该参数包含一个或多个 databaseName 参数,用于封装一个或多个元数据库名称。databaseNames 的具体内容见表 A. 10:

表 A. 10 databaseNames 参数表

参数名称	可选/必选	参数含义
databaseName	必选	表示由客户端指定待检索的一个或多个元数据库名称

目录检索请求中的 databaseNames 的组成如图 A. 10 所示:

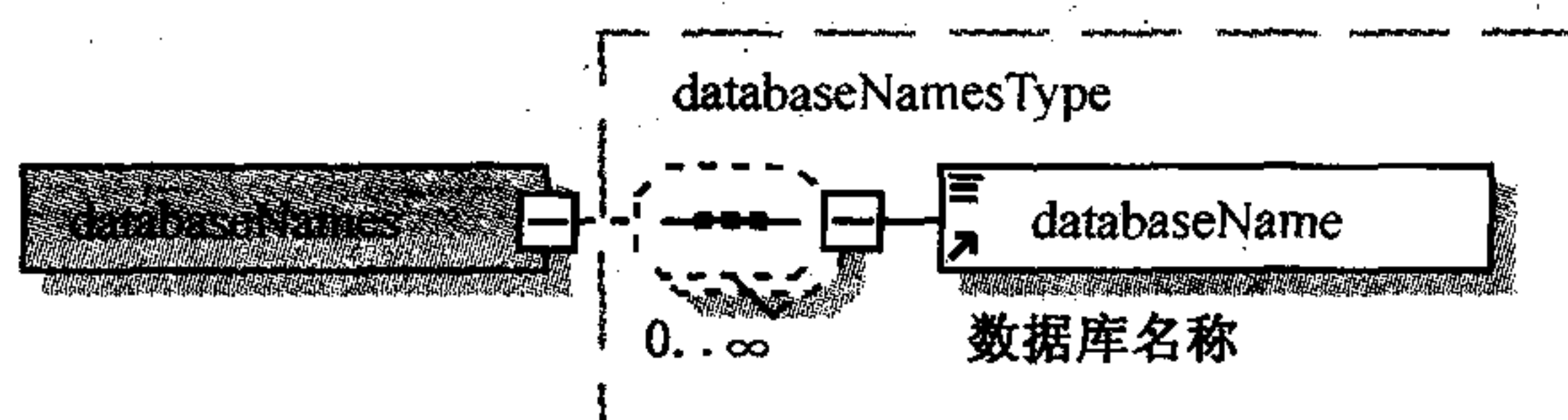


图 A. 10 目录检索请求中 DatabaseNames 的模型图

目录检索请求中 databaseNames 的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符"databaseNames"的定义 -->
<xsd:element name="databaseNames" type="databaseNamesType"/>
<xsd:complexType name="databaseNamesType">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="databaseName"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="databaseName" type="databaseNameType">
    <xsd:annotation>
        <xsd:documentation>元数据库名称。</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:simpleType name="databaseNameType">
    <xsd:annotation>
        <xsd:documentation>元数据库名称的类型定义。</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string"/>

```


</xsd:simpleType>

复合型参数 query 在检索请求中使用,以表达客户端的检索条件。具体示例参见附录 F。

query 可使用多种类型的查询语法,如 XQuery、SQL 等,本部分采用 type-1 型查询语法,采用逆波兰表达式结构(Reverse Polish Notation)。query 的组成如图 A.11 所示:

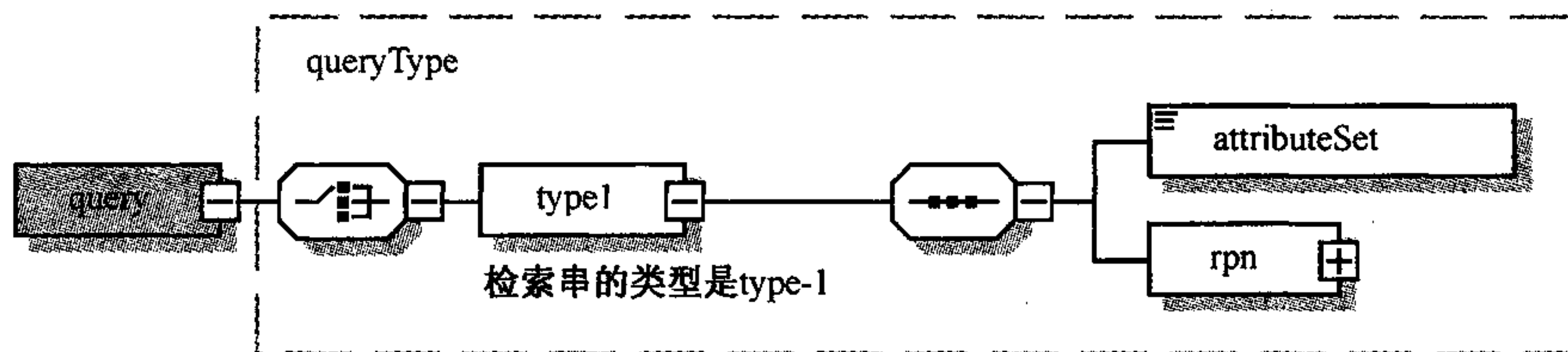


图 A.11 目录检索请求中 query 的模型图

目录检索请求中 query 的主要参数说明如下:

- 参数 rpn:表示一个采用逆波兰表达式结构”的 type-1 型检索串,包含检索操作数和检索操作符等内容;
- 参数 attributeSet:表示检索串采用的属性列表字符集名称,固定取值为“政务信息资源目录服务属性集”。

目录检索请求中 query 的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```
<!-- 类型标识符"Query"的定义 -->
<xsd:element name="query" type="queryType"/>
<xsd:complexType name="queryType">
  <xsd:choice>
    <xsd:element name="type1">
      <xsd:annotation>
        <xsd:documentation>检索串的类型是 type-1</xsd:documentation>
      </xsd:annotation>
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="attributeSet" type="xsd:string" fixed="政务信息资源目录服务属性集"/>
          <xsd:element name="rpn" type="RPNQueryType"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
```

参数 rpn 包含有一组或多组的检索操作数 operand 和检索操作符 operator。这种逆波兰表达式结构的 query 将检索操作符独立于检索操作数之外进行表示,简化了对运算表达式的解析操作。query 中的 rpn 的组成如图 A.12 所示:

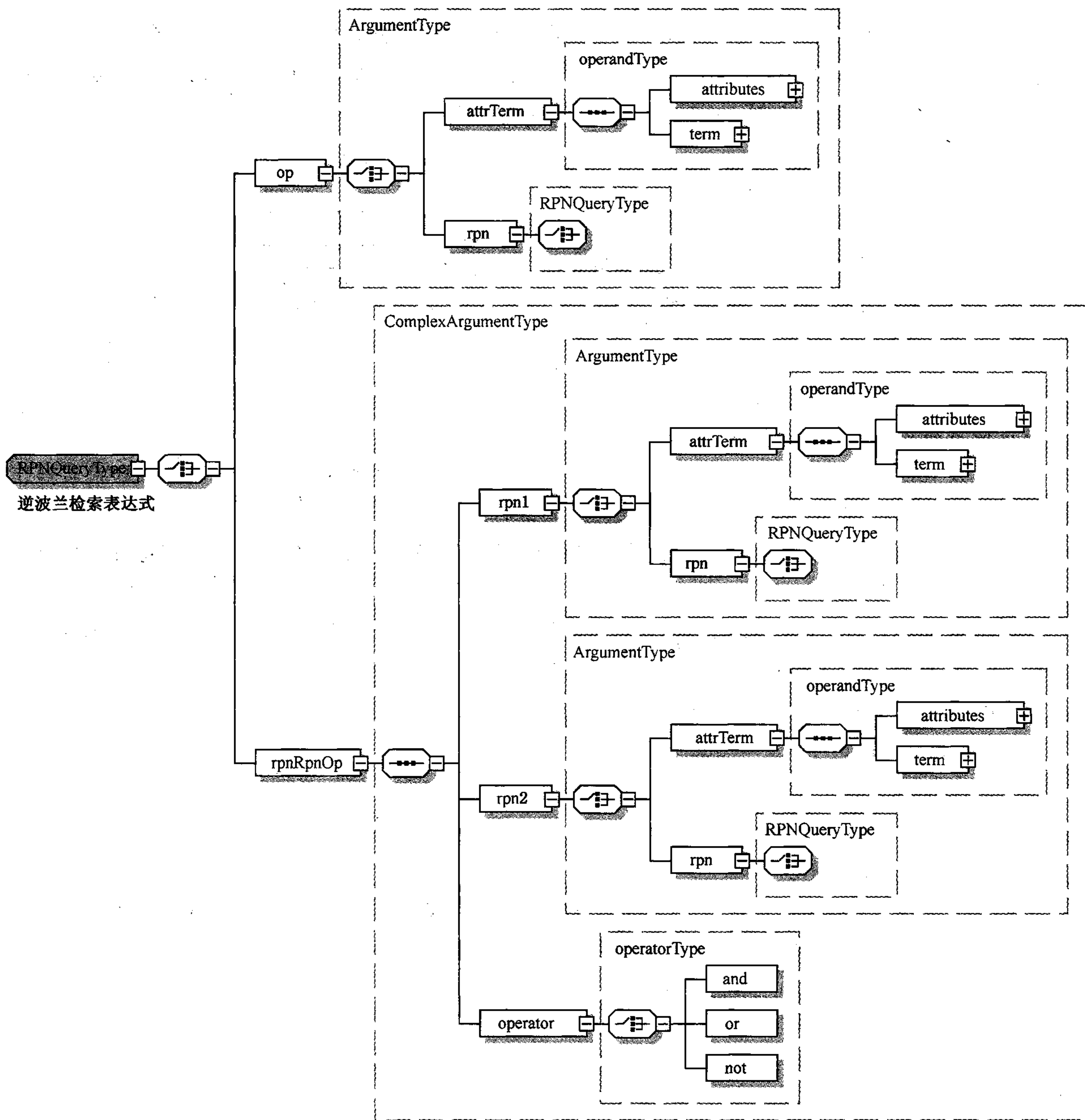


图 A. 12 query 中 rpn 的模型图

query 的 rpn 的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符 "RPNQueryType" 的定义 -->
<xsd:complexType name="RPNQueryType">
  <xsd:annotation>
    <xsd:documentation>逆波兰检索表达式。</xsd:documentation>
  </xsd:annotation>
  <xsd:choice>
    <xsd:element name="op" type="ArgumentType"/>
    <xsd:element name="rpnRpnOp" type="ComplexArugmentType"/>
  </xsd:choice>
</xsd:complexType>

```



```

    </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ArgumentType">
  <xsd:choice>
    <xsd:element name="attrTerm" type="operandType"/>
    <xsd:element name="rpn" type="RPNQueryType"/>
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ComplexArugmentType">
  <xsd:sequence>
    <xsd:element name="rpn1" type="ArgumentType"/>
    <xsd:element name="rpn2" type="ArgumentType"/>
    <xsd:element name="operator" type="operatorType"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"RPNStructure"的定义 -->
<xsd:element name="RPNStructure" type="RPNStructureType"/>
<xsd:complexType name="RPNStructureType">
  <xsd:choice>
    <xsd:element name="operand" type="operandType"/>
    <xsd:element name="rpnRpnOp">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="operator" type="operatorType"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

```

参数 operand 用来表达一个单一的简单检索语句。单一的简单检索操作数包含有 term 及其对应的 attributes, 该检索操作数将应用检索项与元数据库中的各个属性进行匹配。在匹配过程中, 需要施加属性集所提供的约束条件。

其中, 主要参数的说明:

- a) attributes 是属性列表参数, 用来提供具体的约束条件, 指明数据库中待匹配的属性字段及其关系运算符。该参数属于复合类型 attributeList, 其结构和内容见下面的“检索串中的属性列表 attributeList”说明。
- b) term 是检索项参数, 表示待匹配的各种属性值, 如待匹配的字符串 characterString、待匹配的数值 numeric、待匹配的元数据标识符 recordId、待匹配的时间值 dateTime、待匹配的字节流 general。

query 中 operand 的结构如图 A. 13 所示:

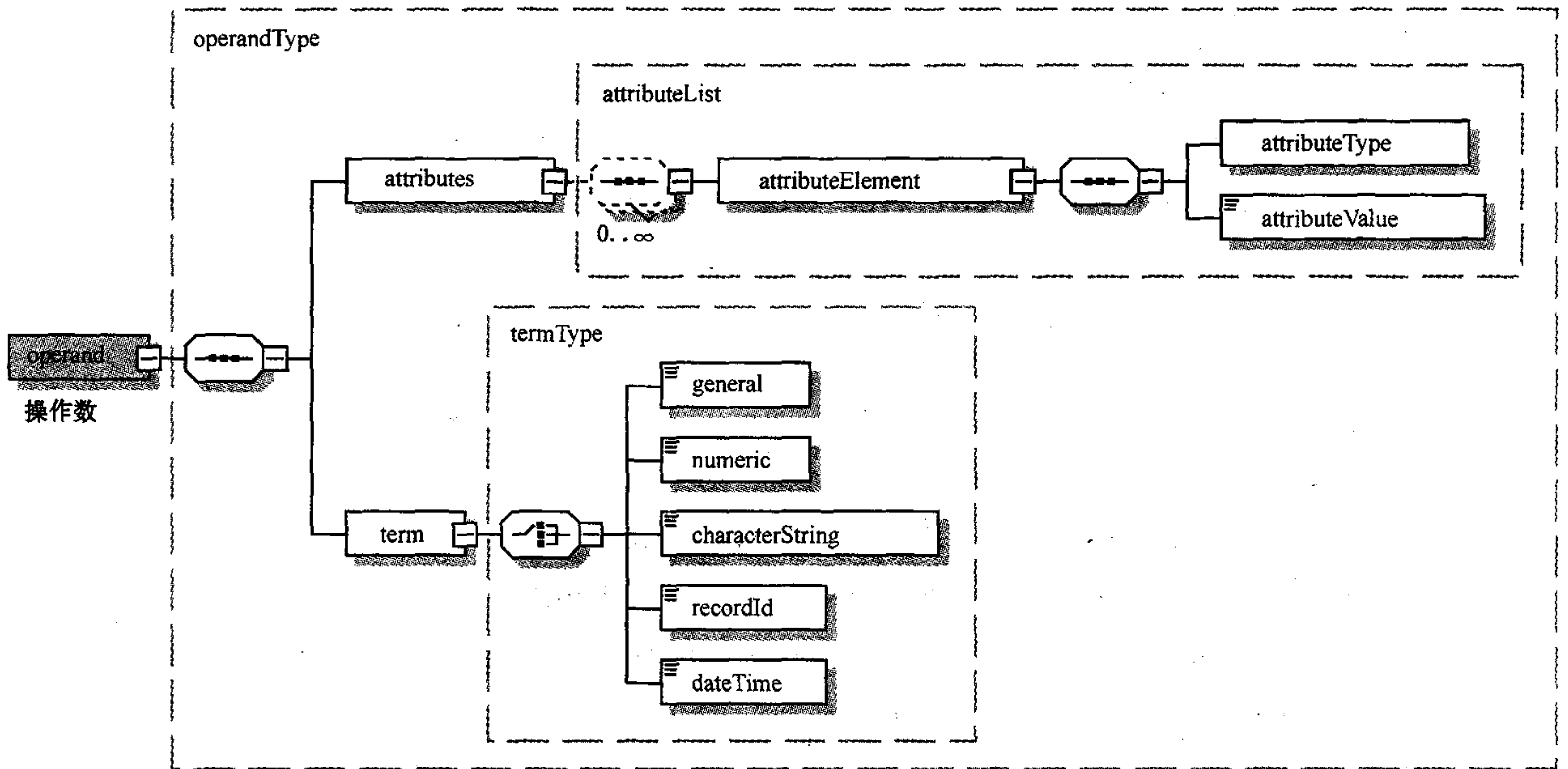


图 A.13 query 中 operand 的模型图

query 中 operand 的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符"Operand"的定义 -->
<xsd:element name="operand" type="operandType">
  <xsd:annotation>
    <xsd:documentation>操作数。 </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="operandType">
  <xsd:annotation>
    <xsd:documentation>操作数的类型定义。 </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="attributes" type="attributeList"/>
    <xsd:element name="term" type="termType"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"attributeList"的定义 -->
<xsd:complexType name="attributeList">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="attributeElement">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="attributeType"/>
          <xsd:element name="attributeValue" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```



```

    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"Term"的定义 -->
<xsd:complexType name="termType">
  <xsd:choice>
    <xsd:element name="general" type="xsd:hexBinary"/>
    <xsd:element name="numeric" type="xsd:integer"/>
    <xsd:element name="characterString" type="xsd:string"/>
    <xsd:element name="recordId" type="xsd:string"/>
    <xsd:element name="dateTime" type="generalizedTimeType"/>
  </xsd:choice>
</xsd:complexType>

```

attributeList 用于限定检索项 term, 并与其共同组成检索操作数, 形成一个完整的检索语句。

attributeList 是由一个或多个 attributeElement 组成, 每个 attributeElement 均包含 attributeType 和 attributeValue。

当 attributeType 取值为“1”, 表示元数据的属性, 用于指定元数据库中相应的属性字段, 其 attributeValue 的取值应符合附录 C 中表 C.1 的编码。

当 attributeType 取值为“2”, 表示采用的关系运算符, 用于指定该检索串的关系运算类型, 其 attributeValue 应符合附录 C 中表 C.2 的编码。

当 attributeType 取值为“3”, 表示采用的位置运算符, 用于指定该检索串的位置运算类型, 其 attributeValue 应符合附录 C 中表 C.3 的编码。

检索串中“属性列表”参数的组成结构, 如图 A.14 所示:

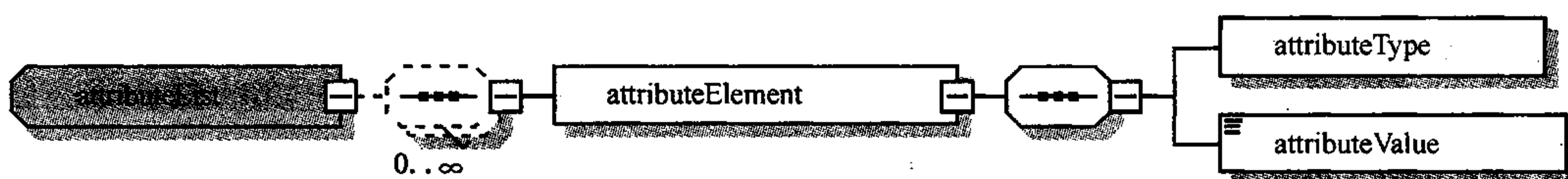


图 A.14 query 中 attributeList 的模型图

query 中 attributeList 的 XML Schema 如下(完整的 XML Schema 定义见附录 B):

```

<xsd:complexType name="attributeListType">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element ref="AttributeElement"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="AttributeElement" type="attributeElementType"/>
<xsd:complexType name="attributeElementType">
  <xsd:sequence>
    <xsd:element name="attributeType" type="xsd:integer"/>
    <xsd:element name="attributeValue" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

```

operator 用来限定检索操作数所产生的结果, 包含有 3 种逻辑运算符: “AND”运算、“OR”运算和“NOT”运算。query 中 operator 的组成结构, 如图 A.15 所示:

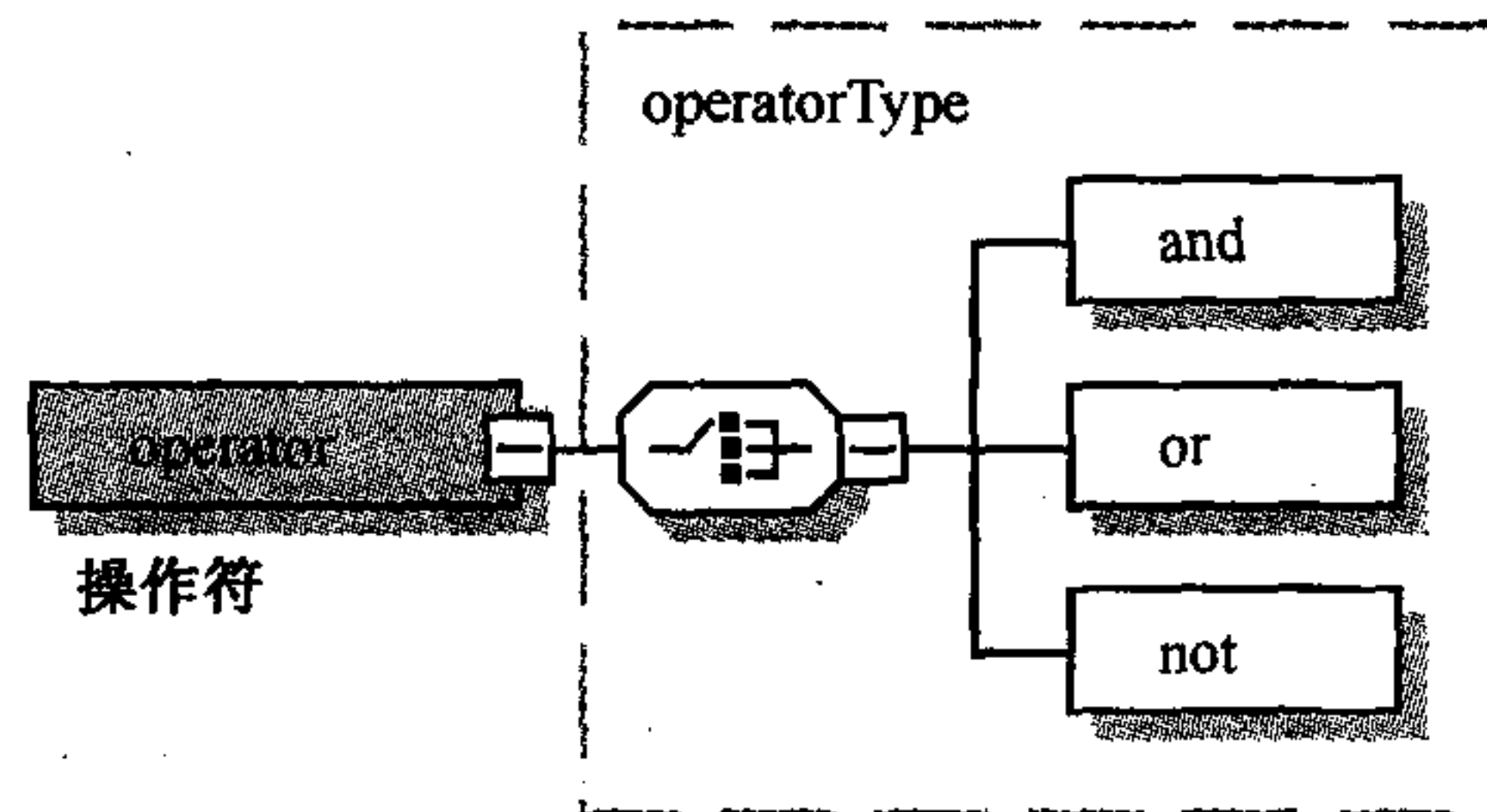


图 A.15 query 中 operator 的模型图

query 中 operator 的 XML Schema 如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符"Operator"的定义 -->
<xsd:element name="operator" type="operatorType">
  <xsd:annotation>
    <xsd:documentation>操作符。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="operatorType">
  <xsd:annotation>
    <xsd:documentation>操作符的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:choice>
    <xsd:element name="and" type="NullType"/>
    <xsd:element name="or" type="NullType"/>
    <xsd:element name="not" type="NullType"/>
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="NullType">
  <xsd:annotation>
    <xsd:documentation>抽象类型。表示元素出现即可,没有取值。</xsd:documentation>
  </xsd:annotation>
</xsd:complexType>
    
```

A.3.1.3 响应

接收到客户端的检索请求后,目录检索响应消息被服务器用作反馈目录检索结果及操作状态给客户端。目录检索响应的参数见表 A.11:

表 A.11 检索响应参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识
resultCount	必选	服务器端执行检索操作得到的命中记录数
searchStatus	必选	服务器端返回的检索状态信息。该参数取值为“success”表示检索成功;“failure”表示检索失败
otherInfo	可选	其他信息,用于客户端和服务器端传递自定义信息。如发生异常,可使用本参数传递详细的异常信息给客户端

目录检索响应的组成结构,如图 A.16 所示:

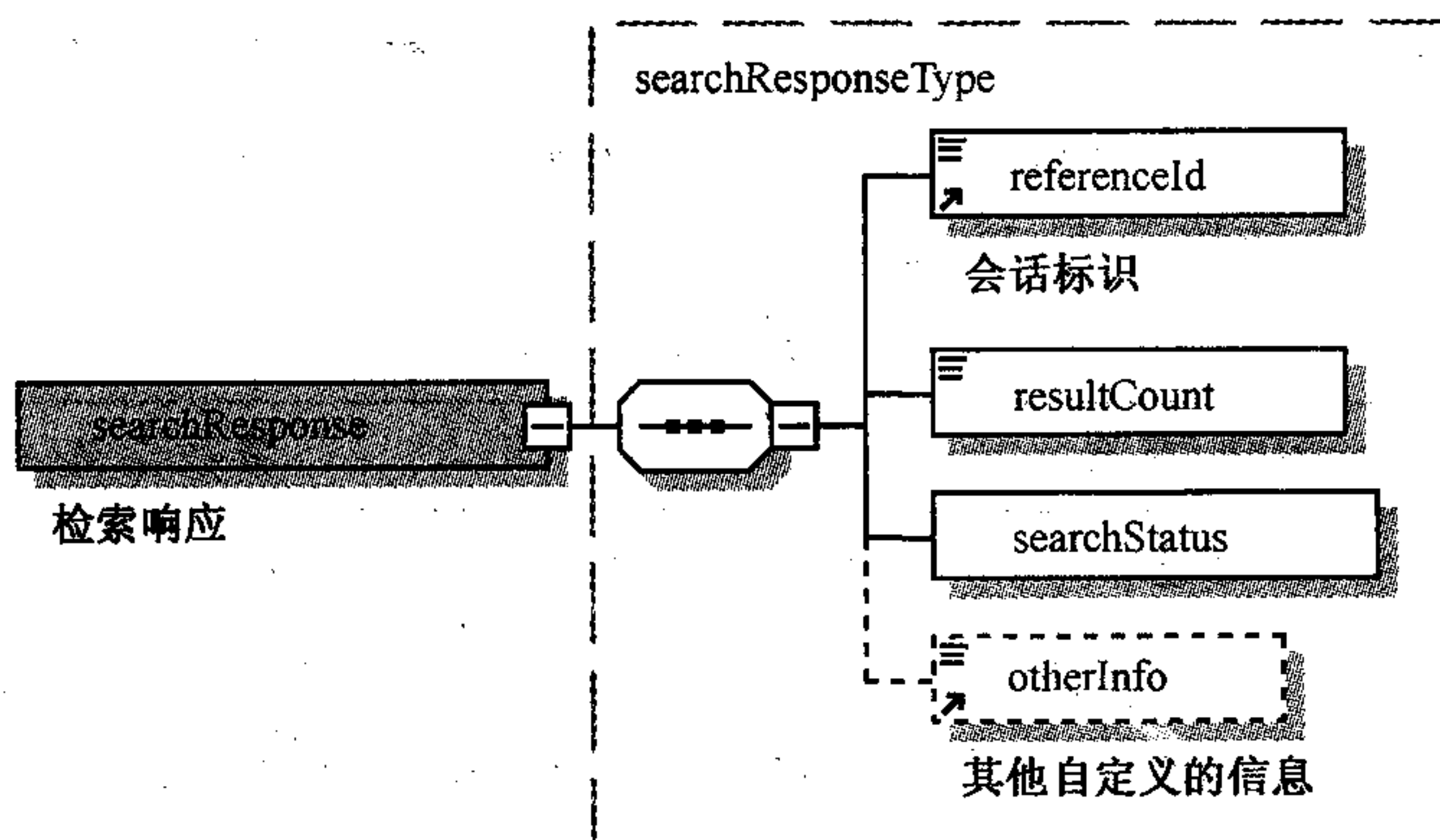


图 A.16 目录检索响应的模型图

检索响应消息的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```
<!-- 类型标识符"SearchResponse"的定义 -->
```

```
<xsd:element name="searchResponse" type="searchResponseType">
```

```
  <xsd:annotation>
```

```
    <xsd:documentation>检索响应。</xsd:documentation>
```

```
  </xsd:annotation>
```

```
</xsd:element>
```

```
<xsd:complexType name="searchResponseType">
```

```
  <xsd:annotation>
```

```
    <xsd:documentation>检索响应的类型定义。</xsd:documentation>
```

```
  </xsd:annotation>
```

```
  <xsd:sequence>
```

```
    <xsd:element ref="referenceId"/>
```

```
    <xsd:element name="resultCount" type="xsd:integer"/>
```

```
    <xsd:element name="searchStatus"/>
```

```
    <!-- 参数 searchStatus 取值为“success”表示检索成功;“failure”表示检索失败 -->
```

```
    <xsd:element ref="otherInfo" minOccurs="0"/>
```

```
    <!-- 其他信息,用于客户端和服务端传递自定义信息。一旦发生异常,使用本参数传递详细的异常信息给客户端。 -->
```

```
  </xsd:sequence>
```

```
</xsd:complexType>
```

A.3.2 目录检索结果提取

A.3.2.1 描述

目录检索仅标识符合检索条件的元数据记录,目录检索结果则由服务器端从结果集中提取并返回若干元数据记录给客户端。

A.3.2.2 请求

目录检索结果提取时,有 3 种属性提取范围:

第一种是政务信息资源核心元数据的提取。通过设置参数 elementSetName 中 genericElementSetName 参数的取值为“Brief”,提取政务信息资源核心元数据。

第二种是政务信息资源全部元数据内容的提取。通过设置参数 elementSetName 中 genericEle-

mentSetName 参数的取值为“Full”或空值来表示提取全集,提取全部的政务信息资源元数据。

第三种是由客户端定制提取的元数据属性字段范围,通过设置参数 elementSetName 中的 elementSet 参数。该 elementSet 参数包含一个或多个 element 参数,用来封装“待提取的元数据属性字段标识符”。该元数据属性字段标识符要求符合表 C.1 的编码。

目录检索结果提取请求的参数见表 A.12:

表 A.12 提取请求参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识
resultSetID	必选	客户端指定提取元数据所属的结果集。因可以进行多次检索操作后再提取操作,故需要客户端指定从哪个检索中间结果集中进行提取
resultSetStartPoint	必选	客户端要求提取元数据位于结果集的位置。结果集中元数据记录的标识符是从“1”开始编号
numberOfRecordsRequested	必选	客户端指定提取的元数据记录的数目
elementSetName	必选	由客户端指定提取结果集的属性范围
preferredRecordSyntax	可选	客户端指定提取结果的编码方式。该参数默认取值为“XML”,表示客户端指定提取结果集的编码方式采用 XML
otherInfo	可选	其他信息,用于客户端和服务端传递自定义信息

目录检索结果提取请求的组成结构,如图 A.17 所示:

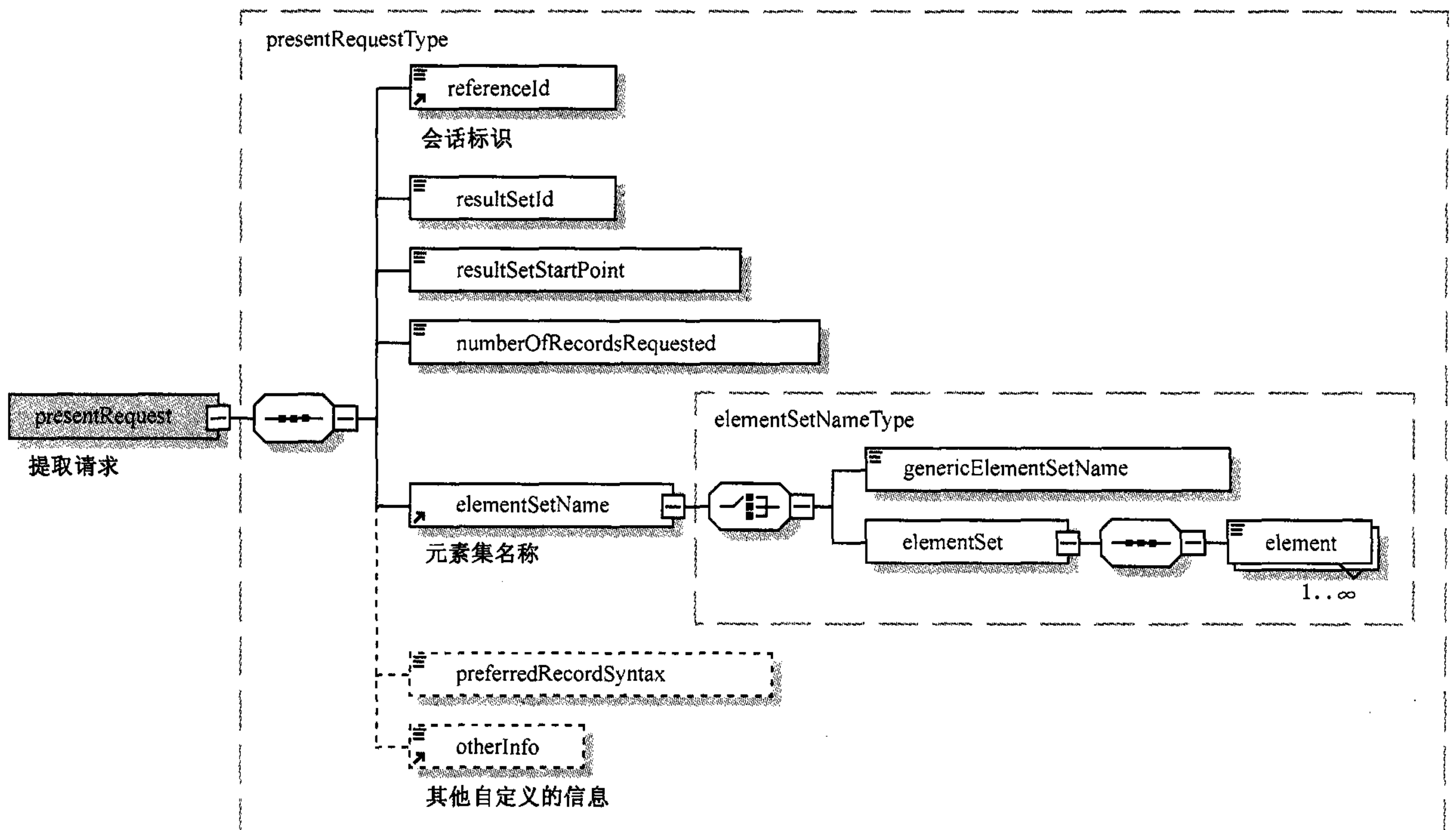


图 A.17 目录检索结果提取请求的模型图

目录检索结果提取请求消息的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

<!-- 类型标识符“PresentRequest”的定义 -->

<xsd:element name="presentRequest" type="presentRequestType">


```

    <xsd:annotation>
      <xsd:documentation>提取请求。</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
<xsd:complexType name="presentRequestType">
  <xsd:annotation>
    <xsd:documentation>提取请求的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="resultSetId" type="resultSetIdType"/>
    <xsd:element name="resultSetStartPoint" type="xsd:integer"/>
    <xsd:element name="numberOfRecordsRequested" type="xsd:integer"/>
    <xsd:element ref="elementSetName"/>
    <xsd:element name="preferredRecordSyntax" type="xsd:string" default="XML"
minOccurs="0"/>
    <!-- preferredRecordSyntax 表示提取结果的编码方式。默认采用 XML 方式 -->
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"ElementSetName"的定义 -->
<xsd:element name="elementSetName" type="elementSetNameType">
  <xsd:annotation>
    <xsd:documentation>元素集名称。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="elementSetNameType">
  <xsd:annotation>
    <xsd:documentation>元素集名称的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:choice>
    <xsd:element name="genericElementSetName" type="xsd:string"/>
    <xsd:element name="elementSet">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="element" type="xsd:string" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

```

A.3.2.3 响应

目录检索结果提取响应的参数见表 A.13:

表 A. 13 提取响应参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识
numberOfRecordsReturned	必选	服务器端返回的元数据数目
presentStatus	必选	服务器端返回的提取状态。该参数取值为“success”表示提取成功;“failure”表示提取失败
records	必选	服务器端返回的结果集。该参数是复合型,详细说明见表 A. 14
otherInfo	可选	其他信息,用于客户端和服务端传递自定义信息。如发生异常,可使用本参数传递详细的异常信息给客户端

目录检索结果提取响应的组成结构,如图 A. 18 所示:

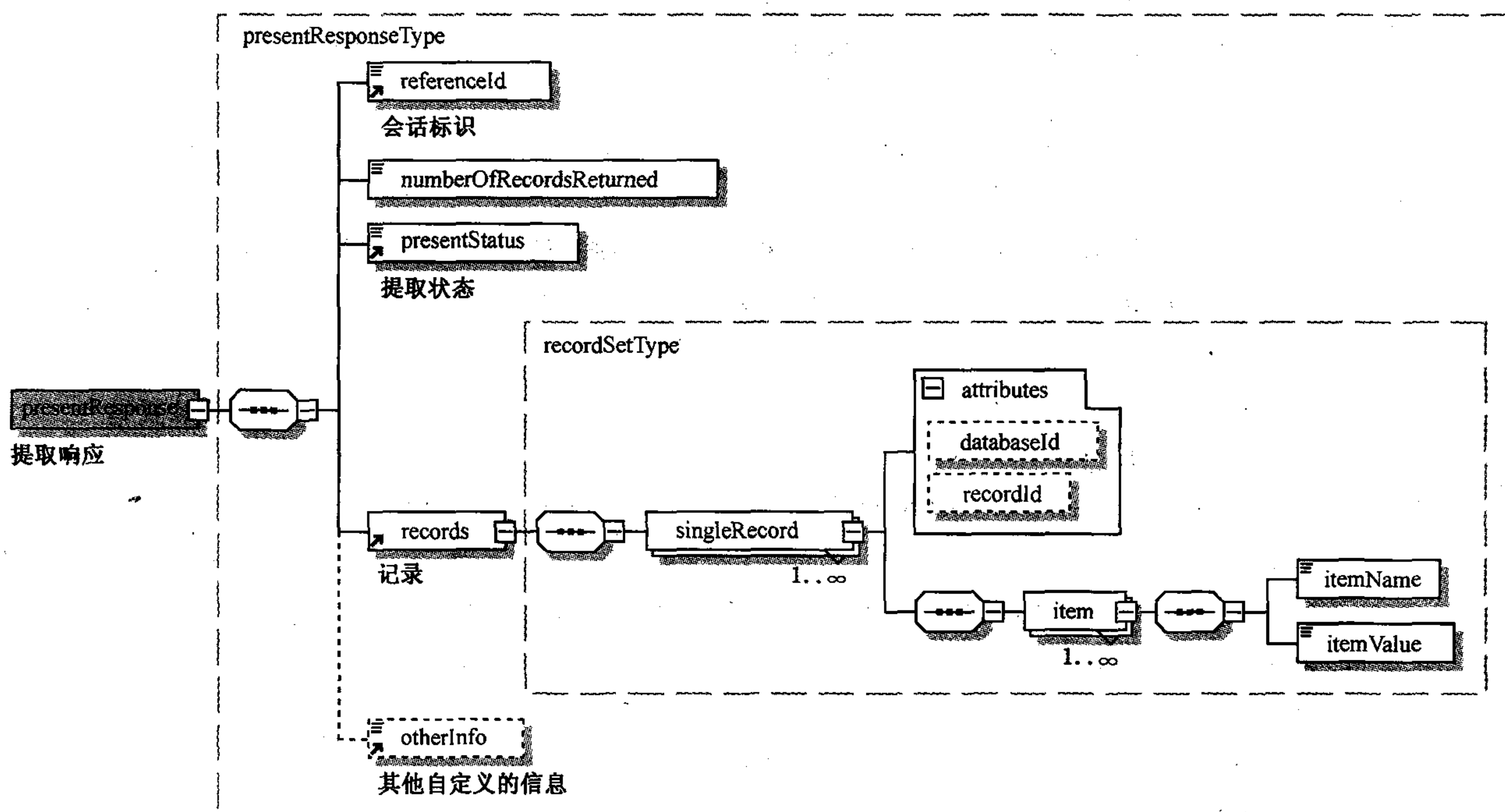


图 A. 18 目录检索结果提取响应的模型图

目录检索结果提取响应的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```

<!-- 类型标识符 "PresentResponse" 的定义 -->
<xsd:element name="presentResponse" type="presentResponseType">
  <xsd:annotation>
    <xsd:documentation>提取响应。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="presentResponseType">
  <xsd:annotation>
    <xsd:documentation>提取响应的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="numberOfRecordsReturned" type="xsd:integer"/>

```


<xsd:element ref="presentStatus"/>

<!--参数 presentStatus 取值为“success”表示提取成功;“failure”表示提取失败。-->

<xsd:element ref="records"/>

<xsd:element ref="otherInfo" minOccurs="0"/>

<!-- 其他信息,用于客户端和服务端传递自定义信息。一旦发生异常,使用本参数传递详细的异常信息给客户端。 -->

</xsd:sequence>

</xsd:complexType>

复合型参数 records 包含了一组或多组 singleRecord 参数,封装了元数据内容。目录检索结果提取响应中 singlerecords 的详细说明如表 A. 14 所示:

表 A. 14 singleRecord 参数表

参数名称	可选/必选	参数含义
Item	必选	结果集中的元数据记录,包含有一组或多组元数据属性参数 itemName 和元数据属性值参数 itemValue。提取的结果集是核心元数据时,参数 itemName 取值要求符合本规范中表 C. 1 的编码
databaseId	可选	表明目录检索结果提取的目标元数据库名称或标识,该参数用于核对由客户端指定检索的元数据库名称
RecordId	可选	服务器端返回的元数据标识符

目录检索结果提取响应中 records 的组成结构,如图 A. 19 所示:

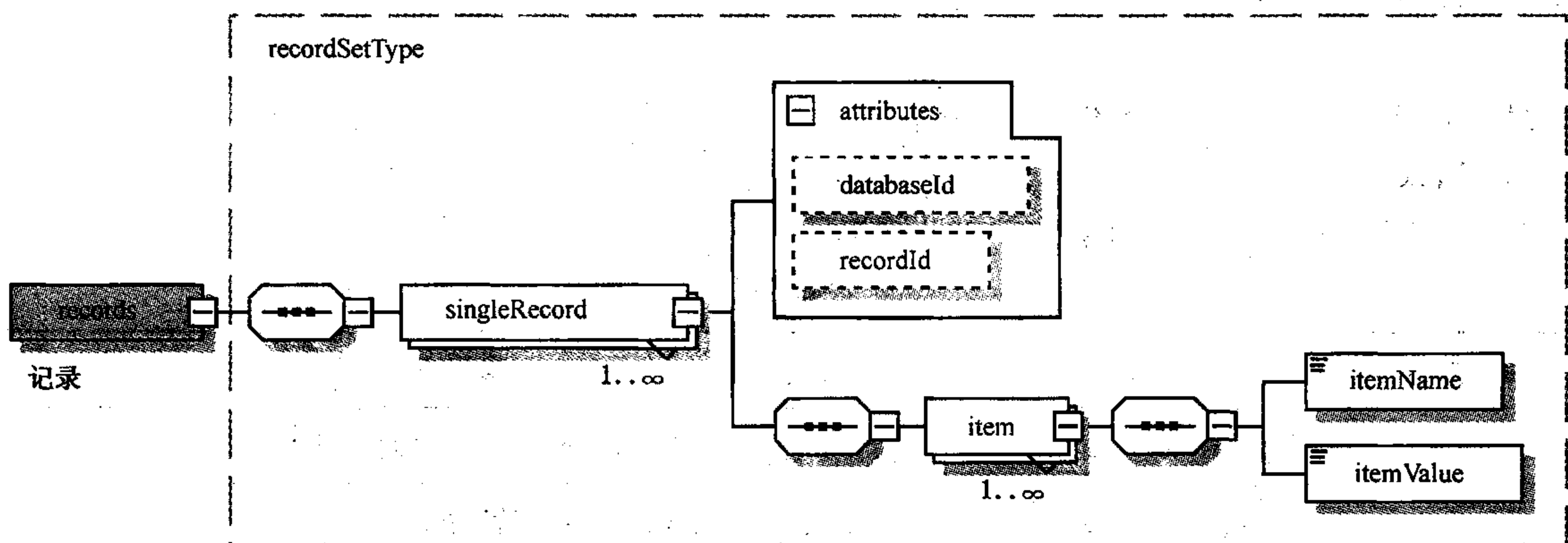


图 A. 19 目录检索结果提取响应中 records 的模型图

目录检索结果提取响应中 records 的 XML Schema 定义如下:

<xsd:element name="recordSet" type="recordSetType">

<xsd:annotation>

<xsd:documentation>记录集。</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:complexType name="recordSetType">

<xsd:annotation>

<xsd:documentation>记录集的类型定义。</xsd:documentation>

</xsd:annotation>

<xsd:sequence>

```

<xsd:element name="singleRecord" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="item" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="itemName" type="xsd:string"/>
            <xsd:element name="itemValue" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="databaseId" type="xsd:string" use="optional"/>
    <xsd:attribute name="recordId" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

```

A.4 管理接口

A.4.1 元数据管理

A.4.1.1 描述

管理接口实现元数据的各种创建、删除和更新操作。该管理接口为可选实现。

A.4.1.2 请求

元数据管理请求参数的详细说明见表 A.15:

表 A.15 元数据管理请求参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识,即会话标识
Type	必选	用于指定进行元数据的各种管理操作,详细内容见表 A.16
databaseName	必选	由客户端指定执行元数据管理操作的目标元数据库名称
recordId	可选	待删除或待更新的元数据标识号。该元数据标识号用在元数据库中指定唯一的元数据。可以通过目录检索结果提取接口获取其响应消息,该响应中的结果集包含有唯一的元数据标识号。 注:当 type 值为 1 时,进行元数据插入操作,本参数不出现;当 type 值为 2、3 时,进行元数据删除或更新操作,本参数为必选参数
Data	可选	待创建或待更新的元数据内容,符合 XML 格式。 当 type 值为 1、3 时,进行元数据插入或更新操作,本参数为必选参数;当 type 值为 2 时,进行元数据删除操作,本参数不出现
otherInfo	可选	其他信息,用于客户端和服务端传递自定义信息

元数据管理请求中,操作类型参数 type 的取值,如表 A.16 所示:

表 A.16 操作类型参数 type 取值列表

名称	取值	含义
创建元数据	1	用于表示该次请求是进行元数据导入操作
删除元数据	2	用于表示该次请求是进行元数据删除操作
更新元数据	3	用于表示该次请求是进行元数据更新操作

元数据管理请求元数据的创建、删除和更新时,使用到的参数有所不同:

- 元数据创建操作仅使用到 referenceId、type、databaseName、data 参数;
- 元数据删除操作仅使用到 referenceId、type、databaseName、recordId 参数;
- 元数据更新操作仅使用到 referenceId、type、databaseName、recordId、data 参数。

元数据管理请求的组成结构,如图 A.20 所示:

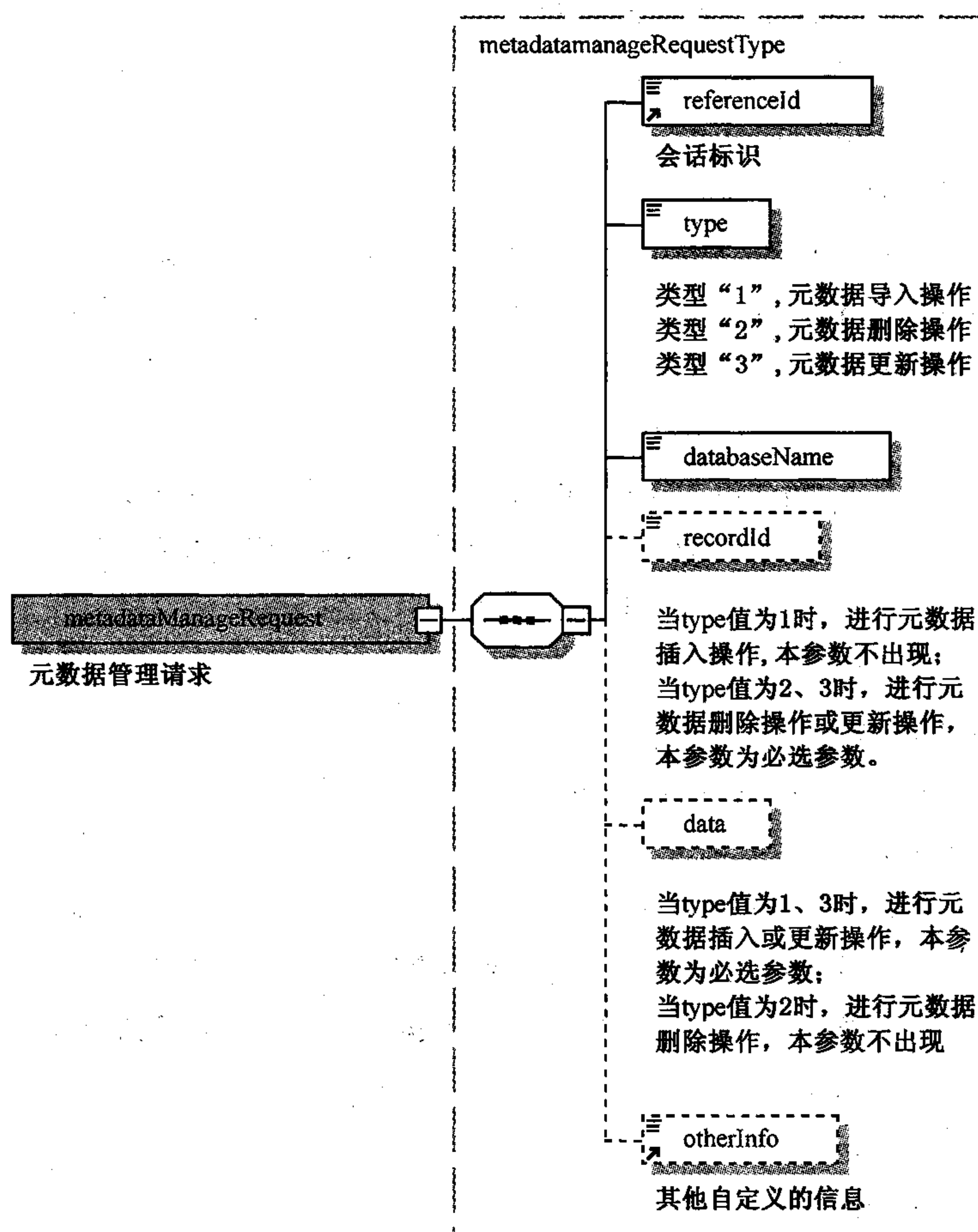


图 A.20 元数据管理请求的模型图

元数据管理请求的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B):

```
<!-- 类型标识符"metadataManageRequest"的定义 -->
<xsd:element name="metadataManageRequest" type="metadataManageRequestType">
  <xsd:annotation>
    <xsd:documentation>元数据管理请求。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

```

<xsd:complexType name="metadataManageRequestType">
  <xsd:annotation>
    <xsd:documentation>元数据管理请求的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="type" type="xsd:int">
      <xsd:annotation>
        <xsd:documentation>
          类型“1”，元数据导入操作
          类型“2”，元数据删除操作
          类型“3”，元数据更新操作
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="databaseName" type="xsd:string"/>
    <xsd:element name="recordId" type="xsd:int" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          当 type 值为 1 时，进行元数据插入操作，本参数不出现；
          当 type 值为 2、3 时，进行元数据删除或更新操作，本参数为必选参数。
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="data" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          当 type 值为 1、3 时，进行元数据插入或更新操作，本参数为必选参数；
          当 type 值为 2 时，进行元数据删除操作，本参数不出现。
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

A.4.1.3 响应

服务器返回该元数据管理响应消息来反馈执行元数据创建、删除、更新等操作的状态信息，将元数据的创建、删除或更新操作执行结果返回给客户端。元数据管理响应参数见表 A.17：

表 A.17 元数据管理响应参数表

参数名称	可选/必选	参数含义
referenceId	必选	用于识别一个请求所启动的操作的标识
operationState	必选	由服务器端返回的元数据管理操作执行结果状态。取值为“success”，表示执行元数据管理操作成功；取值为“failure”，表示执行元数据管理操作失败
operationType	可选	用来表示服务器端执行的元数据管理操作类型
otherInfo	可选	其他信息，用于客户端和服务端传递自定义信息。如发生异常，可使用本参数传递详细的异常信息给客户端

元数据管理响应的组成结构，如图 A.21 所示：

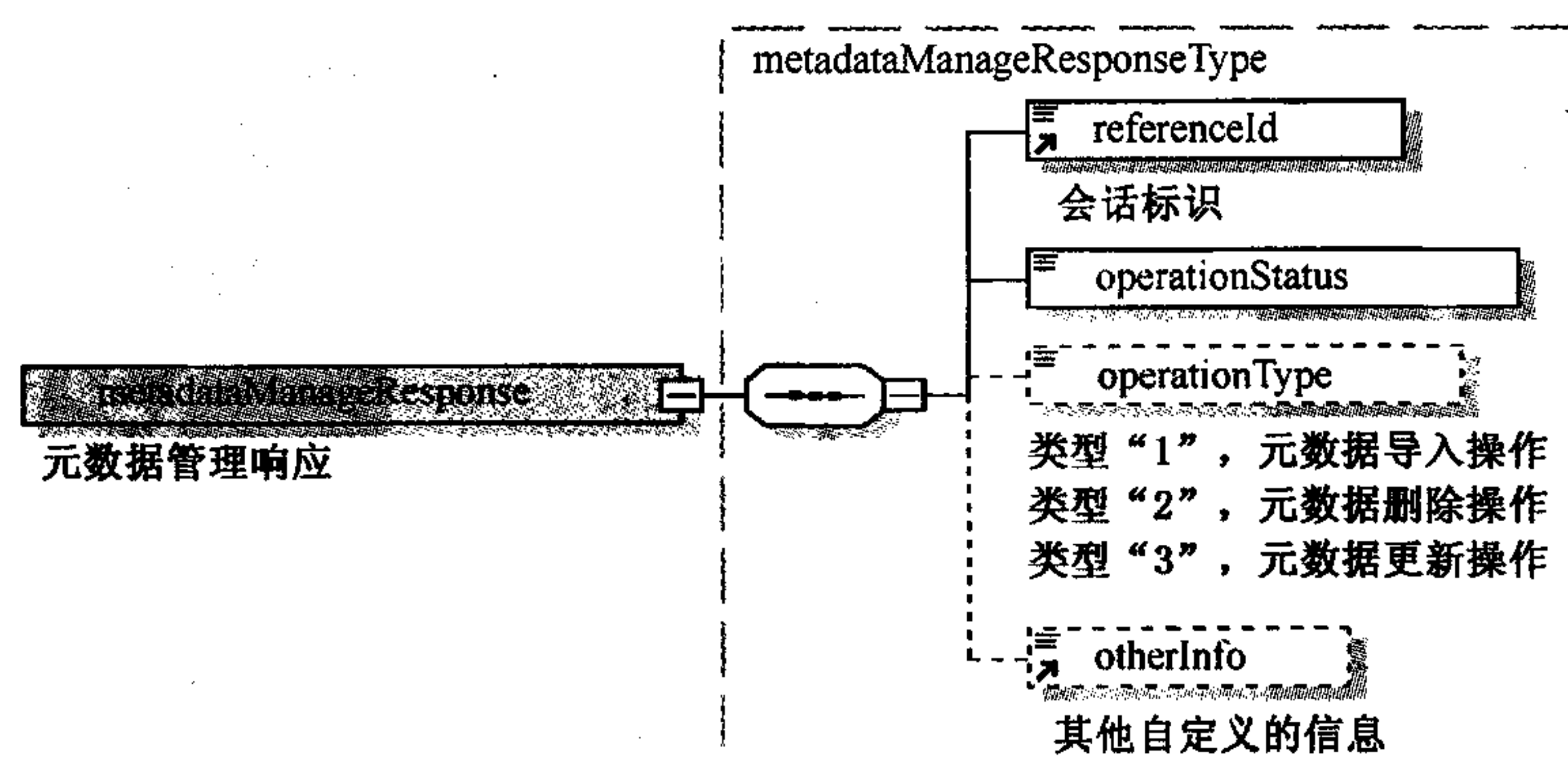


图 A.21 元数据管理响应的模型图

元数据管理响应的 XML Schema 定义片段如下(完整的 XML Schema 定义见附录 B)：

```

<!-- 类型标识符"metadataManageResponse"的定义 -->
<xsd:element name="metadataManageResponse" type="metadataManageResponseType">
  <xsd:annotation>
    <xsd:documentation>元数据管理响应。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="metadataManageResponseType">
  <xsd:annotation>
    <xsd:documentation>元数据管理响应的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="operationStatus">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="success"/>
          <xsd:enumeration value="failure"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```
<xsd:element name="operationType" type="xsd:int" minOccurs="0" >
  <xsd:annotation>
    <xsd:documentation>
      类型“1”,元数据导入操作
      类型“2”,元数据删除操作
      类型“3”,元数据更新操作
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element ref="otherInfo" minOccurs="0"/>
<!-- 其他信息,用于客户端和服务端传递自定义信息。一旦发生异常,使用本参数
传递详细的异常信息给客户端。 -->
</xsd:sequence>
</xsd:complexType>
```


附录 B

(规范性附录)

目录服务接口 XML Schema 描述

```

<? xml version="1.0" encoding="GB2312"? >
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <! -- 类型标识符"PDU" -->
  <xsd:element name="PDU" type="PDUType">
    <xsd:annotation>
      <xsd:documentation>协议数据单元。</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name="PDUType">
    <xsd:annotation>
      <xsd:documentation>协议数据单元的定义。</xsd:documentation>
    </xsd:annotation>
    <xsd:choice>
      <xsd:element ref="initRequest"/>
      <xsd:element ref="initResponse"/>
      <xsd:element ref="catalogCapabilitiesRequest"/>
      <xsd:element ref="catalogCapabilitiesResponse"/>
      <xsd:element ref="searchRequest"/>
      <xsd:element ref="searchResponse"/>
      <xsd:element ref="presentRequest"/>
      <xsd:element ref="presentResponse"/>
      <xsd:element ref="metadataManageRequest"/>
      <xsd:element ref="metadataManageResponse"/>
      <xsd:element ref="closeRequest"/>
      <xsd:element ref="closeResponse"/>
    </xsd:choice>
  </xsd:complexType>
  <! -- 类型标识符"InitRequest"的定义 -->
  <xsd:element name="initRequest" type="initRequestType">
    <xsd:annotation>
      <xsd:documentation>初始化请求。</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name="initRequestType">
    <xsd:annotation>
      <xsd:documentation>初始化请求的类型定义。</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>

```

```

    <xsd:element ref="protocolVersion"/>
    <xsd:element name="idAuthentication" minOccurs="0"/>
    <xsd:element name="implementationId" type="xsd:string" minOccurs="0"/>
    <xsd:element name="implementationName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="implementationVersion" type="xsd:string" minOccurs="0"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符 "ProtocolVersion" 的定义 -->
<xsd:element name="protocolVersion" type="xsd:string" default="1">
  <xsd:annotation>
    <xsd:documentation>协议版本,本实现对应的协议版本</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<!-- 类型标识符 "InitResponse" 的定义 -->
<xsd:element name="initResponse" type="initResponseType">
  <xsd:annotation>
    <xsd:documentation>初始化响应。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="initResponseType">
  <xsd:annotation>
    <xsd:documentation>初始化响应的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <!-- 参数 referenceId 是由 a~z、A~Z、0~9 组成,其他字符无效。 -->
    <xsd:element ref="protocolVersion"/>
    <xsd:element name="result" type="xsd:boolean"/>
    <!-- 参数 result 取值为布尔型。“true”表示建立会话成功;“false”表示建立会话失败,
此时 referenceId 取为空。 -->
    <xsd:element name="implementationId" type="xsd:string" minOccurs="0"/>
    <xsd:element name="implementationName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="implementationVersion" type="xsd:string" minOccurs="0"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
    <!-- 其他信息,用于客户端和服务端传递自定义信息。一旦发生异常,使用本参数
传递详细的异常信息给客户端。 -->
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符 "catalogCapabilitiesRequest" 的定义 -->
<xsd:element name="catalogCapabilitiesRequest" type="catalogCapabilitiesRequestType">
  <xsd:annotation>
    <xsd:documentation>目录服务自描述信息的请求。</xsd:documentation>

```



```

    </xsd:annotation>
</xsd:element>
<xsd:complexType name="catalogCapabilitiesRequestType">
  <xsd:annotation>
    <xsd:documentation>目录服务自描述信息请求的定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"catalogCapabilitiesResponse"的定义 -->
<xsd:element name="catalogCapabilitiesResponse" type="catalogCapabilitiesResponseType">
  <xsd:annotation>
    <xsd:documentation>目录服务自描述信息的响应。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="catalogCapabilitiesResponseType">
  <xsd:annotation>
    <xsd:documentation>目录服务自描述信息响应的定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="databaseList" type="databaseListType"/>
    <xsd:element ref="options"/>
    <xsd:element name="serviceName" type="xsd:string" default="政务信息资源目录服务"/>
    <xsd:element name="serviceAbstract" type="xsd:string"/>
    <xsd:element name="serviceID" type="xsd:string"/>
    <xsd:element name="serviceProvider" type="serviceProviderType"
maxOccurs="unbounded"/>
    <xsd:element name="serviceURL" type="xsd:string"/>
    <xsd:element name="serviceType" type="xsd:string" fixed="目录服务"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="databaseListType">
  <xsd:sequence>
    <xsd:element name="database" type="databaseType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="databaseType">
  <xsd:sequence>

```

```

    <xsd:element name="databaseId" type="xsd:string"/>
    <xsd:element name="databaseName" type="xsd:string"/>
    <xsd:element name="databaseDescribe" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<! -- 类型标识符 "Options"的定义 -->
<xsd:element name="options" type="optionsType">
  <xsd:annotation>
    <xsd:documentation>支持的功能。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="optionsType">
  <xsd:list itemType="facilitySupportedType"/>
</xsd:simpleType>
<xsd:simpleType name="facilitySupportedType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="init"/>
    <xsd:enumeration value="close"/>
    <xsd:enumeration value="capabilities"/>
    <xsd:enumeration value="search"/>
    <xsd:enumeration value="present"/>
    <xsd:enumeration value="metadataManage"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="serviceProviderType">
  <xsd:annotation>
    <xsd:documentation>提供部门类型</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="serviceProviderName"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="serviceProviderName" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>提供部门名称</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<! -- 类型标识符"SearchRequest"的定义 -->
<xsd:element name="searchRequest" type="searchRequestType">
  <xsd:annotation>
    <xsd:documentation>检索请求。</xsd:documentation>
  </xsd:annotation>
</xsd:element>

```



```

<xsd:complexType name="searchRequestType">
  <xsd:annotation>
    <xsd:documentation>检索请求的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="replaceIndicator" type="xsd:boolean"/>
    <xsd:element name="resultSetID" type="xsd:string"/>
    <xsd:element ref="databaseNames"/>
    <xsd:element name="preferredRecordSyntax" type="xsd:string" default="XML"
minOccurs="0"/>
    <!-- preferredRecordSyntax 表示查询结果的编码方式。默认采用 XML 方式 -->
    <xsd:element ref="query"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"SearchResponse"的定义 -->
<xsd:element name="searchResponse" type="searchResponseType">
  <xsd:annotation>
    <xsd:documentation>检索响应。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="searchResponseType">
  <xsd:annotation>
    <xsd:documentation>检索响应的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="resultCount" type="xsd:integer"/>
    <xsd:element name="searchStatus"/>
    <!-- 参数 searchStatus 取值为“success”表示检索成功;“failure”表示检索失败 -->
    <xsd:element ref="otherInfo" minOccurs="0"/>
    <!-- 其他信息,用于客户端和服务端传递自定义信息。一旦发生异常,使用本参数
传递详细的异常信息给客户端。 -->
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"databaseNames"的定义 -->
<xsd:element name="databaseNames" type="databaseNamesType"/>
<xsd:complexType name="databaseNamesType">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element ref="databaseName"/>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:element name="databaseName" type="databaseNameType">
  <xsd:annotation>
    <xsd:documentation>元数据库名称。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="databaseNameType">
  <xsd:annotation>
    <xsd:documentation>元数据库名称的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!-- 类型标识符"Query"的定义 -->
<xsd:element name="query" type="queryType"/>
<xsd:complexType name="queryType">
  <xsd:choice>
    <xsd:element name="type1">
      <xsd:annotation>
        <xsd:documentation>检索串的类型是 type-1</xsd:documentation>
      </xsd:annotation>
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="attributeSet" type="xsd:string" fixed="政务信
息资源目录服务属性集"/>
          <xsd:element name="rpn" type="RPNQueryType"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
<!-- 类型标识符"RPNQueryType"的定义 -->
<xsd:complexType name="RPNQueryType">
  <xsd:annotation>
    <xsd:documentation>逆波兰检索表达式。</xsd:documentation>
  </xsd:annotation>
  <xsd:choice>
    <xsd:element name="op" type="ArgumentType"/>
    <xsd:element name="rpnRpnOp" type="ComplexArugmentType"/>
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ArgumentType">
  <xsd:choice>
    <xsd:element name="attrTerm" type="operandType"/>
    <xsd:element name="rpn" type="RPNQueryType"/>
  </xsd:choice>

```



```

    </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ComplexArugmentType">
  <xsd:sequence>
    <xsd:element name="rpn1" type="ArgumentType"/>
    <xsd:element name="rpn2" type="ArgumentType"/>
    <xsd:element name="operator" type="operatorType"/>
  </xsd:sequence>
</xsd:complexType>
<! -- 类型标识符"RPNStructure"的定义 -->
<xsd:element name="RPNStructure" type="RPNStructureType"/>
<xsd:complexType name="RPNStructureType">
  <xsd:choice>
    <xsd:element name="operand" type="operandType"/>
    <xsd:element name="rpnRpnOp">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="operator" type="operatorType"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
<! -- 类型标识符"Operand"的定义 -->
<xsd:element name="operand" type="operandType">
  <xsd:annotation>
    <xsd:documentation>操作数。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="operandType">
  <xsd:annotation>
    <xsd:documentation>操作数的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="attributes" type="attributeList"/>
    <xsd:element name="term" type="termType"/>
  </xsd:sequence>
</xsd:complexType>
<! -- 类型标识符"attributeList"的定义 -->
<xsd:complexType name="attributeList">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="attributeElement">
      <xsd:complexType>

```

```

        <xsd:sequence>
            <xsd:element name="attributeType"/>
            <xsd:element name="attributeValue" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"Term"的定义 -->
<xsd:complexType name="termType">
    <xsd:choice>
        <xsd:element name="general" type="xsd:hexBinary"/>
        <xsd:element name="numeric" type="xsd:integer"/>
        <xsd:element name="characterString" type="xsd:string"/>
        <xsd:element name="recordId" type="xsd:string"/>
        <xsd:element name="dateTime" type="generalizedTimeType"/>
    </xsd:choice>
</xsd:complexType>
<!-- 类型标识符"Operator"的定义 -->
<xsd:element name="operator" type="operatorType">
    <xsd:annotation>
        <xsd:documentation>操作符。</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:complexType name="operatorType">
    <xsd:annotation>
        <xsd:documentation>操作符的类型定义。</xsd:documentation>
    </xsd:annotation>
    <xsd:choice>
        <xsd:element name="and" type="NullType"/>
        <xsd:element name="or" type="NullType"/>
        <xsd:element name="not" type="NullType"/>
    </xsd:choice>
</xsd:complexType>
<xsd:complexType name="NullType">
    <xsd:annotation>
        <xsd:documentation>抽象类型。表示元素出现即可,没有取值。</xsd:documentation>
    </xsd:annotation>
</xsd:complexType>
<!-- 类型标识符"PresentRequest"的定义 -->
<xsd:element name="presentRequest" type="presentRequestType">
    <xsd:annotation>
        <xsd:documentation>提取请求。</xsd:documentation>

```



```

    </xsd:annotation>
</xsd:element>
<xsd:complexType name="presentRequestType">
  <xsd:annotation>
    <xsd:documentation>提取请求的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="resultSetId" type="resultSetIdType"/>
    <xsd:element name="resultSetStartPoint" type="xsd:integer"/>
    <xsd:element name="numberOfRecordsRequested" type="xsd:integer"/>
    <xsd:element ref="elementSetName"/>
    <xsd:element name="preferredRecordSyntax" type="xsd:string" default="XML"
minOccurs="0"/>
    <!-- preferredRecordSyntax 表示提取结果的编码方式。默认采用 XML 方式 -->
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"ElementSetName"的定义 -->
<xsd:element name="elementSetName" type="elementSetNameType">
  <xsd:annotation>
    <xsd:documentation>元素集名称。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="elementSetNameType">
  <xsd:annotation>
    <xsd:documentation>元素集名称的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:choice>
    <xsd:element name="genericElementSetName" type="xsd:string"/>
    <xsd:element name="elementSet">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="element" type="xsd:string" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
<!-- 类型标识符"PresentResponse"的定义 -->
<xsd:element name="presentResponse" type="presentResponseType">
  <xsd:annotation>
    <xsd:documentation>提取响应。</xsd:documentation>

```

```

    </xsd:annotation>
</xsd:element>
<xsd:complexType name="presentResponseType">
  <xsd:annotation>
    <xsd:documentation>提取响应的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="numberOfRecordsReturned" type="xsd:integer"/>
    <xsd:element ref="presentStatus"/>
    <!-- 参数 presentStatus 取值为“success”表示提取成功；“failure”表示提取失败。 -->
    <xsd:element ref="records"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
    <!-- 其他信息,用于客户端和服务端传递自定义信息。一旦发生异常,使用本参数
传递详细的异常信息给客户端。 -->
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"Records"的定义 -->
<xsd:element name="records" type="recordSetType">
  <xsd:annotation>
    <xsd:documentation>记录。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<!-- 类型标识符"PresentStatus"的定义 -->
<xsd:element name="presentStatus" type="presentStatusType">
  <xsd:annotation>
    <xsd:documentation>提取状态。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="presentStatusType">
  <xsd:annotation>
    <xsd:documentation>提取状态的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="success"/>
    <xsd:enumeration value="failure"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- 类型标识符"metadataManageRequest"的定义 -->
<xsd:element name="metadataManageRequest" type="metadataManageRequestType">
  <xsd:annotation>
    <xsd:documentation>元数据管理请求。</xsd:documentation>
  </xsd:annotation>

```



```

</xsd:element>
<xsd:complexType name="metadataManageRequestType">
  <xsd:annotation>
    <xsd:documentation>元数据管理请求的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>
    <xsd:element name="type" type="xsd:int">
      <xsd:annotation>
        <xsd:documentation>
          类型“1”,元数据导入操作
          类型“2”,元数据删除操作
          类型“3”,元数据更新操作
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="databaseName" type="xsd:string"/>
    <xsd:element name="recordId" type="xsd:int" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          当 type 值为 1 时,进行元数据插入操作,本参数不出现;
          当 type 值为 2、3 时,进行元数据删除操作或更新操作,本参数为必选参数。
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="data" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          当 type 值为 1、3 时,进行元数据插入或更新操作,本参数为必选参数;
          当 type 值为 2 时,进行元数据删除操作操作,本参数不出现。
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"metadataManageResponse"的定义 -->
<xsd:element name="metadataManageResponse" type="metadataManageResponseType">
  <xsd:annotation>
    <xsd:documentation>元数据管理响应。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="metadataManageResponseType">

```

```

<xsd:annotation>
  <xsd:documentation>元数据管理响应的类型定义。</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
  <xsd:element ref="referenceId"/>
  <xsd:element name="operationStatus">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="success"/>
        <xsd:enumeration value="failure"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="operationType" type="xsd:int" minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>
        类型“1”，元数据导入操作
        类型“2”，元数据删除操作
        类型“3”，元数据更新操作
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element ref="otherInfo" minOccurs="0"/>
  <!-- 其他信息,用于客户端和服务端传递自定义信息。一旦发生异常,使用本参数
传递详细的异常信息给客户端。 -->
</xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"generalizedTimeType"的定义 -->
<xsd:simpleType name="generalizedTimeType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<!-- 类型标识符"closeRequest"的定义 -->
<xsd:element name="closeRequest" type="closeRequestType">
  <xsd:annotation>
    <xsd:documentation>关闭请求。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="closeRequestType">
  <xsd:annotation>
    <xsd:documentation>关闭请求的定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="referenceId"/>

```



```

    <xsd:element name="closeReason" type="xsd:string" minOccurs="0"/>
    <xsd:element ref="otherInfo" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"closeResponse"的定义 -->
<xsd:element name="closeResponse" type="closeResponseType">
  <xsd:annotation>
    <xsd:documentation>关闭响应。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="closeResponseType">
  <xsd:annotation>
    <xsd:documentation>关闭响应的定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="closeStatus" type="xsd:string" minOccurs="0"/>
    <!-- 参数 closeStatus 取值为“success”表示关闭会话成功；“failure”表示关闭会话失败-->
  </xsd:sequence>
</xsd:complexType>
<!-- 类型标识符"ReferenceId"的定义 -->
<xsd:element name="referenceId" type="referenceIdType">
  <xsd:annotation>
    <xsd:documentation>会话标识。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="referenceIdType">
  <xsd:annotation>
    <xsd:documentation>操作的引用 ID 的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:integer"/>
</xsd:simpleType>
<!-- 类型标识符"ResultSetId"的定义 -->
<xsd:element name="resultSet" type="resultSetIdType">
  <xsd:annotation>
    <xsd:documentation>结果集 ID。</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:simpleType name="resultSetIdType">
  <xsd:annotation>
    <xsd:documentation>结果集 ID 的类型定义。</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:element name="external">

```

```

    <xsd:annotation>
      <xsd:documentation>外部元素。</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name="externalType">
    <xsd:annotation>
      <xsd:documentation>外部类型。用于引入元数据检索和提取协议以外的类型。
    </xsd:documentation>
    </xsd:annotation>
  </xsd:complexType>
  <xsd:element name="recordSet" type="recordSetType">
    <xsd:annotation>
      <xsd:documentation>记录集。</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name="recordSetType">
    <xsd:annotation>
      <xsd:documentation>记录集的类型定义。</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="singleRecord" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="item" maxOccurs="unbounded">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="itemName" type="xsd:string"/>
                  <xsd:element name="itemValue" type="xsd:string"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
          <xsd:attribute name="databaseId" type="xsd:string" use="optional"/>
          <xsd:attribute name="recordId" type="xsd:string" use="optional"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="otherInfo" type="xsd:string">
    <xsd:annotation>
      <xsd:documentation>其他自定义的信息。</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>

```


附 录 C
(规范性附录)

政务信息资源目录服务属性集

本附录定义了政务信息资源目录服务需要的完整属性集,主要用于目录服务的目录检索操作,用来规范 query 中的 operand 的 attributeList 参数取值。attributeList 是由一个或多个 attributeElement 组成,每个 attributeElement 均包含 attributeType 和 attributeValue。

- a) attributeType 取值为“1”,表示元数据的属性,用于指定元数据库中相应的属性字段,具体内容见表 C.1;
- b) attributeType 取值为“2”,表示采用的关系运算符,用于指定该检索串的关系运算类型,具体内容见表 C.2;
- c) attributeType 取值为“3”,表示采用的位置运算符,用于指定该检索串的位置运算类型,具体内容见表 C.3。

核心元数据属性集的编码规则示意图如图 C.1 所示,其编码规则如下:

- a) 第一层代码只有一位阿拉伯数字,其取值为“1”,表示核心元数据;
- b) 第二层是核心元数据元素和元数据实体,采用三位阿拉伯数字递增顺序码的方法进行编码,例如“资源名称”的编码为“1.001”,“资源发布日期”的编码为“1.002”;
- c) 第三层是元数据实体所包含的元数据元素,采用两位阿拉伯数字递增顺序码的方法进行编码,例如“资源提供单位”的编码为“1.004.01”。

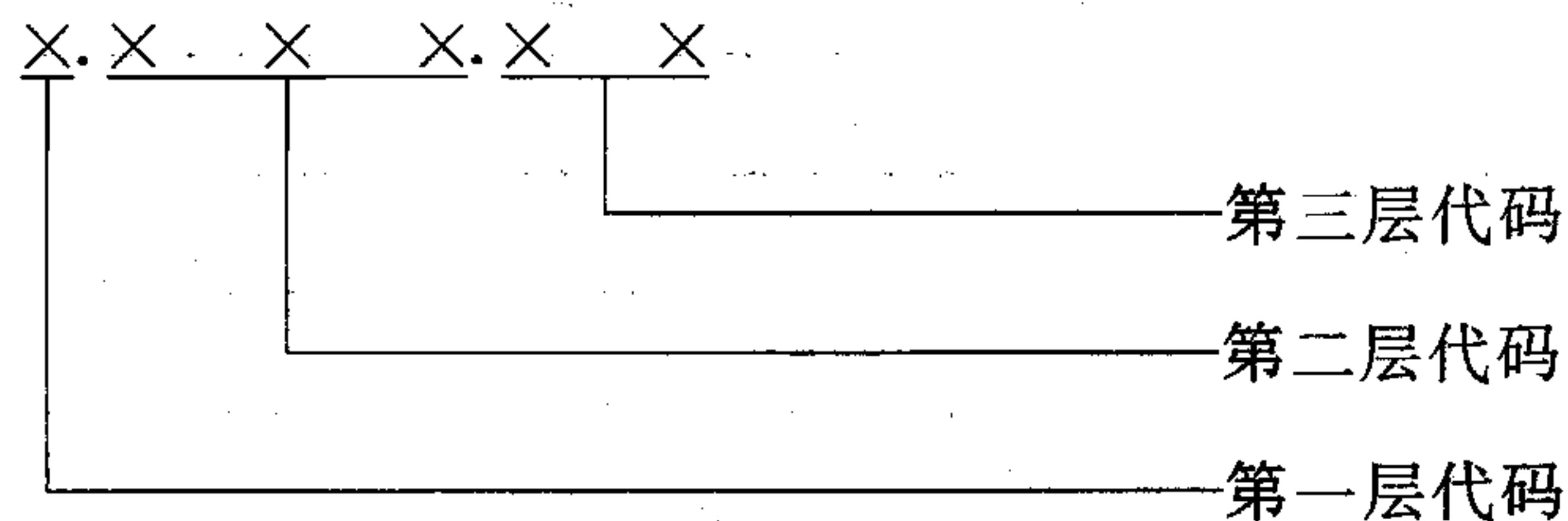


图 C.1 核心元数据属性集的编码规则示意图

核心元数据属性集见表 C.1。

表 C.1 核心元数据属性集

编 码	中文名称	短 名
1.000	资源核心元数据全文	resMetadata
1.001	资源名称	resTitle
1.002	资源发布日期	pubCate
1.003	资源摘要	abstract
1.004	资源提供方	IdPoC
1.004.01	资源提供单位	rpOrgName
1.004.02	资源提供方地址	cntAdd
1.005	关键字说明	CescKeys
1.005.01	关键字	keyword
1.005.02	词典名称	thesaName
1.06	信息资源分类	TpCat

表 C.1 (续)

编 码	中文名称	短 名
1.06.01	分类方式	cateStd
1.06.02	类目名称	cateName
1.06.03	类目编码	cateCode
1.06.04	分类标准	cateStd
1.07	在线资源链接地址	onLineSrc
1.08	资源标识符	resIC
1.09	服务信息	ServInfo
1.09.01	服务地址	servURL
1.09.02	服务类型	servType
1.10	元数据标识符	mdId
1.11	元数据维护方	MdContact
1.11.1	元数据联系单位	rpOrgName
1.11.2	元数据维护方地址	cntAdd
1.12	元数据更新日期	mdDateUpd

关系运算符属性集见表 C.2。

表 C.2 关系运算符属性集

编 码	名 称	符 号
1	小于 less than	<
2	小于等于 less than or equal	<=
3	等于 equal	=
4	大于等于 greater or equal	>=
5	大于 greater than	>
6	不等于 not equal	!=(或<>)

位置运算符属性见表 C.3。

表 C.3 位置运算符属性集

编 码	中文名称	英文名称
3	任意位置出现	any position in field

附录 D
(规范性附录)
分布式目录服务访问模式

分布式目录访问主要针对的是对物理上分布存储的目录内容进行检索,在逻辑上形成统一的目录服务。

基本的分布式目录服务是两层架构,由目录服务网关和目录服务器组成,如图 D.1 所示。

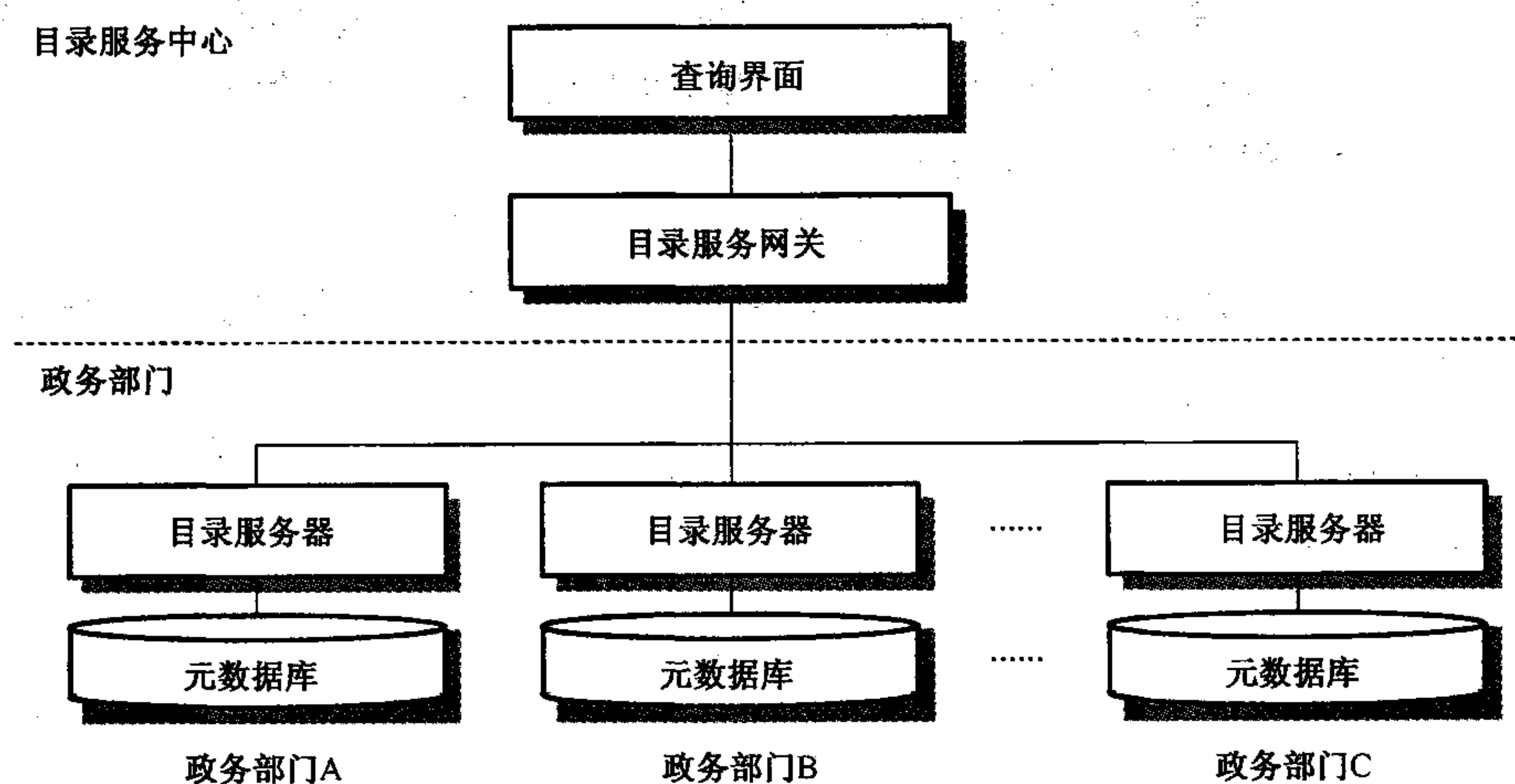


图 D.1 两级分布式目录服务结构示意图

目录服务网关代理对多个节点的访问。对于查询系统来说,多个节点的目录服务网关提供的是一个统一的目录服务。

查询系统与目录服务网关、目录服务网关与目录服务器之间的通讯遵循目录服务接口的技术规定和要求。

附录 E
(资料性附录)
资源编目粒度

共享政务信息资源的主要内容包括数据集、档案、文件、报告、服务等,具体的形式可以是数据库、图片、文档、音频、视频、服务等。

数据集是可以标识的数据集合,它能够用一个数据字典唯一描述。在数据组成上,它是由若干数据记录组成的数据集合;例如宏观经济管理中使用的国家重点企业资产数据集、国土管理中的全国1:10 000土地利用现状数据集等。对于数据库形式的数据集,以逻辑数据库为单位进行编目,逻辑数据库可以由若干张数据库表组成。对于空间地图数据,以图层为单位进行编目。

档案、文件、报告等一般以单个文件为单位编目。

图片、文档、音频、视频一般以独立数据文件为单位编目。

服务是提供应用系统使用的计算机程序接口,一般以单个服务为基本粒度进行编目。例如,提供国家重点企业资产数据查询的服务。

附 录 F
(资料性附录)
检索串示例

本附录提供了两组检索串的示例:示例 1 是只含有 1 个检索操作数的政务信息资源目录服务检索串;示例 2 是含有 3 个检索操作数的政务信息资源目录服务检索串。

示例 1:表示的检索语句是在“属性字段为资源名称(1.001)中”,查询满足条件为“等于(3)字符串(SARS)”的所有元数据。

```

<query>
  <type1>
    <attributeSet>政务信息资源目录服务属性集</attributeSet>
    <rpn>
      <op>
        <attrTerm>
          <attributes>
            <attributeElement>
              <attributeType>1</attributeType>
              <attributeValue>
                <numeric>1.001</numeric>
              </attributeValue>
            </attributeElement>
            <attributeElement>
              <attributeType>2</attributeType>
              <attributeValue>3</attributeValue>
            </attributeElement>
          </attributes>
          <term>
            <characterString>SARS</characterString>
          </term>
        </attrTerm>
      </op>
    </rpn>
  </type1>
</query>

```

示例 2:表示的检索语句是在“属性字段为资源名称(1.001)中”,查询满足条件为“等于(3)字符串(restriction:1)”或者“属性字段为资源摘要(1.003)中”,查询满足条件为“等于(3)字符串(restriction:2)”的所有元数据;再限定在“属性字段为资源出版日期(1.002)中”,查询满足条件为“不大于(2)日期值为(2002-01-01)”的元数据。

```

<query>
  <type-1>
    <attributeSet>政务信息资源目录服务属性集</attributeSet>
    <rpn>
      <rpnRpnOp>

```

```

<rpn1>
  <rpnRpnOp>
    <rpn1>
      <op>
        <attrTerm>
          <attributes>
            <attributeElement>
              <attributeType>1</attributeType>
              <attributeValue>1.001</attributeValue>
            </attributeElement>
            <attributeElement>
              <attributeType>2</attributeType>
              <attributeValue>3</attributeValue>
            </attributeElement>
          </attributes>
          <term>
            <characterString>restriction:1</characterString>
          </term>
        </attrTerm>
      </op>
    </rpn1>
  <rpn2>
    <op>
      <attrTerm>
        <attributes>
          <attributeElement>
            <attributeType>1</attributeType>
            <attributeValue>1.003</attributeValue>
          </attributeElement>
          <attributeElement>
            <attributeType>2</attributeType>
            <attributeValue>3</attributeValue>
          </attributeElement>
        </attributes>
        <term>
          <characterString>restriction:2</characterString>
        </term>
      </attrTerm>
    </op>
  </rpn2>
  <operator>
  <OR/>
  </operator>
</rpnRpnOp>

```



```

</rpn1>
<rpn2>
  <op>
    <attrTerm>
      <attributes>
        <attributeElement>
          <attributeType>1</attributeType>
          <attributeValue>1.002</attributeValue>
        </attributeElement>
        <attributeElement>
          <attributeType>2</attributeType>
          <attributeValue>2</attributeValue>
        </attributeElement>
      </attributes>
      <term>
        <dateTime>2002-01-01</dateTime>
      </term>
    </attrTerm>
  </op>
</rpn2>
< operator >
  <and/>
</ operator >
</rpnRpnOp>
</rpn>
</type-1>
</query>

```

参 考 文 献

- [1] ANSI/NISO Z39.50-1995, Information Retrieval (Z39.50): Application Service Definition and Protocol specification, <http://www.loc.gov/z3950/agency/document.html>
- [2] GB/T 18793—2002 信息技术 可扩展置标语言(XML) 1.0
-