

中华人民共和国卫生行业标准

WS/T 790.1—2021

区域卫生信息平台交互标准  
第1部分：总则

Regional health information platform interactive standard—  
Part 1: General specification

2021-10-27 发布

2022-04-01 实施

中华人民共和国国家卫生健康委员会 发布

## 前 言

本标准是WS/T 790《区域卫生信息平台交互标准》的第1部分。WS/T 790已经发布以下部分：

- 第1部分：总则；
- 第2部分：时间一致性服务；
- 第3部分：节点验证服务；
- 第4部分：安全审计服务；
- 第5部分：基础通知服务；
- 第6部分：居民注册服务；
- 第7部分：医疗卫生机构注册服务；
- 第8部分：医疗卫生人员注册服务；
- 第9部分：术语注册服务；
- 第10部分：健康档案存储服务；
- 第11部分：健康档案管理服务；
- 第12部分：健康档案采集服务；
- 第13部分：健康档案调阅服务；
- 第14部分：文档订阅发布服务；
- 第15部分：预约挂号服务；
- 第16部分：双向转诊服务；
- 第17部分：签约服务；
- 第18部分：提醒服务。

本标准由国家卫生健康标准委员会卫生健康信息标准专业委员会负责技术审查和技术咨询，由国家卫生健康委统计信息中心负责协调性和格式审查，由国家卫生健康委规划发展与信息化司负责业务管理、法规司负责统筹管理。

本标准起草单位：国家卫生健康委统计信息中心、上海市卫生健康信息中心、湖南省卫生计生委信息统计中心、中国人民解放军空军军医大学、华中科技大学同济医学院、国家卫生信息共享技术及应用工程技术研究中心、国家电子计算机质量监督检验中心、中国软件评测中心。

本标准主要起草人：胡建平、李岳峰、董方杰、谢桦、雷永贵、何炜、徐勇勇、沈丽宁、边红丽、黄江平、许德俊、叶彦波、冯东雷、郑良、孟晓。

# 区域卫生信息平台交互标准

## 第1部分：总则

### 1 范围

本标准规定了区域卫生信息平台交互服务编码和消息结构的编制说明、消息与服务定义、数据类型与通用元素、通用服务、通用服务处理等一系列约束。

本标准适用于基于居民健康档案的区域卫生信息平台与接入平台的医疗卫生应用系统间、平台与其它接入平台间以及平台内部服务组件间的信息交互与共享。

### 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本标准必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本标准；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本标准。

- GB/T 26231 信息技术 开放系统互连 OID的国家编号体系和注册规程
- WS 218 卫生机构(组织)分类与代码
- WS 363 卫生信息数据元目录
- WS 364 卫生信息数据元值域代码
- WS 365 城乡居民健康档案基本数据集
- WS 445（所有部分） 电子病历基本数据集
- WS/T 448 基于居民健康档案的区域卫生信息平台技术规范
- WS/T 483（所有部分） 健康档案共享文档规范
- WS/T 500（所有部分） 电子病历共享文档规范

### 3 术语和缩略语

#### 3.1 术语和定义

WS/T 448、WS/T 483界定的术语和定义适用于本标准。

#### 3.2 缩略语

下列缩略语适用于本标准。

- IST：交互服务交易（Interactive Service Transaction）
- W3C：万维网联盟（World Wide Web Consortium）
- WSDL：网络服务描述语言（Web Services Description Language）
- XML Schema：可扩展标记语言结构模式（Extensible Markup Language Schema）
- XPath：XML路径语言（XML PathLanguage）
- SOAP：简单对象访问协议（Simple Object Access Protocol）

## 4 编制说明

### 4.1 内容结构

除本部分及第2部分、第3部分、第4部分外，应包括如下内容：

封面
目次
名称
1 范围
2 规范性引用文件
3 术语和缩略语
3.1术语和定义
3.2缩略语
4 角色
4.1 角色定义
4.2 角色的交易关系
4.3 角色的交易可选性
5. 交易
5.1交易1
...
5.x 交易x
5.x.1 用例
5.x.2 交易流程
5.x.3 消息请求（或通知）
5.x.3.1触发事件
5.x.3.2消息结构及约束
5.x.4 消息应答
5.x.4.1触发事件
5.x.4.2消息结构与约束
6 交互服务
6.1 服务定义
6.2 服务技术要求
7 安全审计
7.1 安全约定
7.2 XXX交易消息审计
7.2.1 XXX交易请求者
7.2.2 XXX交易应答者

### 4.2 编码说明

#### 4.2.1 服务编码

除本部分外，其余部分的服务提供者角色代码见表1。

表1 服务提供者代码

服务提供者角色名称	英文名称	服务编码
时间一致性服务	Consistent Time Service	CT
节点认证服务	Node Authentication Service	NA
安全审计服务	Audit Trail Services	AT
居民注册服务	Person Registration Service	PR
医疗卫生机构注册服务	Medical Institution Registration Service	MIR
医疗卫生人员注册服务	Medical Staff Registration Service	MSR
术语注册服务	Terminology Registration Service	TR
健康档案存储服务	Documents Storage Service	DS
健康档案管理服务	Documents Management Service	DM
健康档案采集服务	Documents Collection Service	DC
健康档案调阅服务	Documents Retrieval Service	DR
文档订阅发布服务	Documents Subscribe-Publish Service	DSP
预约挂号服务	Appointment and Registration Service	ARS
双向转诊服务	TwoWay Referral service	TRS
签约服务	Contract Service	CS
远程会诊服务	Remote Consultation Service	RCS
提醒服务	Reminder Service	RS
居民健康卡服务	Health Card Service	HC

#### 4.2.2 交易编码

交易编码使用如下格式：

***IST-{服务编号}{交易序号}***

例如：**IST-PR1**表示居民注册服务提交居民信息服务交易。

#### 4.2.3 数据类型编码

简单数据类型采用Camel格式，复合数据类型采用Pascal格式。在与消息元素名称易产生混淆的场合，数据类型应后缀Type以示区分。

#### 4.2.4 消息元素编码

消息根元素及通用元素编码采用Pascal格式，其余元素采用Camel格式。

#### 4.3 消息结构表示

消息结构采用XML Schema架构模式进行表达。在文本描述中，消息节点为XML元素的采用元素代码或加注尖括号“<”“>”表示，、消息节点为XML属性的在属性代码前加注“@”前缀表示。

#### 4.4 消息节点的可选项约束表示

消息节点采用XPath表达式指向。消息元素及属性“可选项”约束采用表2代码表示。

表2 消息节点的可选性约束编码

约束名称	英文名称	描述	代码
必选	Required	必选节点，应填充节点信息。如果节点信息不可获得或不被传送，则必须指明空值原因。	R
条件必选	Required for data exists	条件必选，如果节点信息存在则该节点必选。	R2
可选	Optional	可选节点。无论节点信息是否存在，可选择提供或不提供的节点。	O
条件可选	Conditional	符合特定条件时必选的节点。条件可选节点应注明必选条件。	C
强制	Mandatory	强制要求该节点不可缺少，如果该节点没有值或者没有定义缺省值，消息将不能被合法解析。	M
已废弃	Invalid	已废弃的无效节点，不可选择。	X

## 5 消息与服务定义

### 5.1 通讯协议

#### 5.1.1 消息封装结构

区域卫生信息平台服务消息宜通过SOAP1.2版进行通讯。消息封装结构如下：

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  ...
</soap:Envelope>
```

在SOAP1.2版本中服务请求的HTTP头部信息中应设置Content-Type为application/soap+xml。

#### 5.1.2 消息编码

本规范所有部分中所定义的交互消息、文档编码应用 UTF-8。

```
<?xml version="1.0" encoding="UTF-8"?>
```

#### 5.1.3 命名空间

SOAP消息头使用指定的命名空间："http://www.chiss.org.cn/rhin/is/2015"，其中rhin指代区域卫生信息网络，is指代交互规范，2015指代本规范的版本。

在命名空间不冲突的情况下，使用名称rhin来指代该命名空间。

### 5.2 消息定义

#### 5.2.1 消息模式定义

消息模式应采用XML Schema进行定义，并遵循W3C XML Schema规范。

#### 5.2.2 消息节点编码与数据元的对应

所有部分与业务相关的消息节点在可能情况下，应引用以下标准的数据元代码，并建立节点代码与数据元代码间的关系：

- WS 218;
- WS 363;
- WS 364;

- WS 365;
- WS 371;
- WS 372;
- WS 373;
- WS 374;
- WS 375;
- WS 376;
- WS 377;
- WS 445。

### 5.2.3 消息数据类型

消息数据类型宜采用6.1所列基础数据类型。

### 5.2.4 对象 OID 标识

对象OID标识应符合GB/T 26231符定。

## 5.3 服务定义

### 5.3.1 服务描述

每个部分的交互服务均应提供用于定义服务交互接口的WSDL文件。

服务定义语言应遵循W3C WSDL 1.2规范，分别对消息、服务接口、服务绑定以及服务物理位置4个部分进行定义。

### 5.3.2 消息定义

服务定义WSDL文件通过引入已定义的消息格式文件定义消息类型。

消息元素<message>的结构模式应包括名称属性@name和一个部分元素<part>,部分元素包括名称属性@name和元素属性@element。定义要求如下:

- 消息元素<message>的名称属性@name 宜与服务操作同名;
- 部分元素<part>的名称属性@name 宜定义为“message”;
- 部分元素<part>的元素属性@element 指向消息格式文件中的相应的消息元素。

### 5.3.3 服务端口类型描述

服务端口类型元素< portType >包括名称属性@name和一系列的服务操作元素<operation>。每个操作元素包括入口消息<input>及出口消息<output>属性。定义要求如下:

- 操作元素<operation>: 应定义@name 属性, 并与消息的名称保持一致, 表示操作名称;
- 输入元素<input>: 应定义@message 属性, 并与消息格文件中请求消息元素保持一致, 表示操作的输入参数;
- 输出元素<input>: 应定义@message 属性, 并与消息格文件中应答消息元素保持一致, 表示操作的输出参数;
- 通知操作只含输出参数。

### 5.3.4 服务绑定类型

本规范中的网络服务绑定类型为"Document/literal Wrapped模式",该模式与WS-I规范兼容。选用该模式可以根据利用与函数同名的元素名进行消息的分发,可以利用Schema进行消息的验证,绑定元素定义如下:

- 绑定<binding>:应定义@name 属性,采用服务名称+“Binding”格式;应定义@type 属性,指向服务提供者对应的节点定义;
- 兼容 SOAP1.2 的绑定<wssoap12:binding>:应定义@transport 属性;应定义@styles 属性为"document";
- 操作<operation>:应定义@name 属性,表示绑定的操作;
- 输入元素<input>:应定义兼容 SOAP1.2 的<body>元素及@use 属性为“literal”,表示输入消息的格式;
- 输出元素<output>:应定义兼容 SOAP1.2 的<body>元素及@use 属性为“literal”,表示输出消息的格式。

### 5.3.5 服务物理位置

服务物理位置由以下WSDL元素约定:

- 服务<service>元素:应定义@name 属性,表示提供服务的名称;
- 端口<port>属性:应定义@name 属性,表示提供服务的端口名称;应定义@binding 属性,表示相应的绑定定义;
- 地址<address>元素:应定义@location 属性,表示实际提供服务的物理地址。

## 6 数据类型与通用元素

### 6.1 基础数据类型

本部分规定的基础数据类型适用于本标准所有部分。

基础数据类型包括简单数据类型及复合数据类型,简单数据类型符合 W3C的XML Schema1.0、1.1的数据型规定,复合类型是多种数据类型的组合型式。一般情况下,复合类型是其基类与任何类型的组合。任何类型仅当所表示的类型空值时,其属性“nullFlavor”进行进一步说明。

数据类型的及代码说明见表3,数据类型的表达示应符合A.1数据类型结构模式的约束。

表3 数据类型及代码说明

数据名称	类型代码	数据类别	备注
任何类	Any	复合	复合类型的超类,包含nullFlavor属性。
空值原因	nullFlavorPrimitive	简单	说明空值的原因,值域代码见B.1。
字符串基类	stringPrimitive	简单	字符串的基类
字符串	String	复合	字符串类型
布尔基类	booleanPrimitive	简单	布尔的基类
布尔	Boolean	复合	布尔类型
日期基类	datePrimitive	简单	日期的基类
日期	Date	复合	日期类型
时间基类	timePrimitive	简单	时间的基类
时间	Time	复合	时间类型
日期时间基类	dateTimePrimitive	简单	日期时间的基类



表 3 (续)

数据名称	类型代码	数据类别	备注
日期时间	DateTime	复合	日期时间类型
语言基类	languagePrimitive	简单	语言的基类
语言	Language	复合	语言类型
代码基类	codePrimitive	简单	代码的基类
代码	Code	复合	代码类型。
整数基类	integerPrimitive	简单	整数基类
整数	Integer	复合	整数类型
正整数基类	unsignedIntPrimitive	简单	正整数基类
正整数	UnsignedInt	复合	正整数类型
实数原始数	decimalPrimitive	复合	实数的基类
实数类型	Decimal	复合	实数类型
非限定名称	ncNamePrimitive	简单	不包含冒号(:)分隔限定前缀的名称基类
非限定名称	NCName	复合	不包含冒号(:)分隔限定前缀的名称
限定名称基类	qNamePrimitive	简单	包含冒号分隔限定前缀的名称基类
限定名称	QName	复合	包含冒号分隔限定前缀的名称
二进制基类	base64BinaryPrimitive	简单	基于64位的二进制基类
二进制	Base64Binary	复合	基于64位的二进制类型
统一资源标识符基类	uriPrimitive	简单	统一资源标识符基类
统一资源标识符	URI	复合	统一资源标识符
统一资源名称基类	urnPrimitive	简单	统一资源名称基类
统一资源名称	URN	复合	统一资源名称
标识基类	idPrimitive	简单	标识基类
标识	ID	复合	标识类型
对象标识基类	oidPrimitive	简单	对象标识基类, 格式。
对象标识	OID	复合	对象标识
统一标识基类	uuidPrimitive	简单	一标识基类
统一标识符	UUID	复合	统一标识符
取值标识基类	identivierPrimitive	简单	取值标识基类
取值标识	Identivier	复合	取值标识类型
时间段基类	periodPrimitive	简单	时间段基类
时间段	Period	复合	时间段类型
时长	Duration	复合	时间长度
时长单位基类	durationUnitPrimitive	简单	时间计量单位, 值域代码见B.2。
引用	Reference	复合	引用类型
代码概念	Coding	复合	代码概念类型
一般标识	Identifier	复合	一般标识类型
代码化的概念	CodeableConcept	复合	代码化的概念类型
数量	Quantity	复合	数量类型
人名用途列表基类	nameUsePrimitive	简单	人名用途枚举类型, 值域代码见B.3。

表3 (续)

数据名称	类型代码	数据类别	备注
人名用途类型	NameUse	复合	人名用途类型
人名	HumanName	复合	人名类型
地址用途列表	addressUsePrimitive	简单	地址用途枚举值, 值域代码见B. 4。
地址用途	AddressUse	复合	地址类
地址类别	addressTypePrimitive	简单	地址类别枚举值, 值域代码见B. 5。
地址类别	AddressType	复合	地址类别
地址	Address	复合	地址类型
资源主干元素	BackboneElement	复合	资源定义内部所有元素的基类
通讯方式	telecomModePrimitive	简单	通讯方式枚举值, 值域代码见B. 6。
通讯方式	TelecomMode	复合	通讯方式
通讯地址	Telecom	复合	通讯地址
名值对类型	NameValuePairType	复合	编码名值对类型
通用引用标识	ReferenceID	复合	注册对象的通用引用标识
值列表	ValueList	复合	多个可选的值
插槽类型	Slot	复合	插槽类型, 用于查询、过滤器、主题等插入条件。
插槽列表	SlotList	复合	多个并列的插槽
空返回类型	NullResponse	复合	返回空消息的类型

## 6.2 注册服务基础对象数据类型

与个人、机构、医疗卫生人员相关的注册服务, 其消息应采用的数据类型及代码说明见表4, 数据类型的表达示应符合A.2主数据对象数据类型结构模式的约束。

表4 注册服务基础对象数据类型

数据名称	类型代码	数据类别	备注
注册对象类型	RegObjectype	复合	通用的注册对象抽象类, 其它注册对象的父类。
血型	BooldType	复合	血型
健康史	HealthHistoryType	复合	用于过敏史、疾病史、手术史、外伤史、输血史、遗传史、家族史的数据结构。
家族史	FamilyHistoryType	复合	用于家族史描述
健康状况	PhysicalCondition	复合	包含血型、过敏史、疾病史、手术史、外伤史、输血史、遗传史、家族史。
联系人	Contact	复合	联系人信息
监护人	Guardian	复合	监护人信息
自然人信息	Person	复合	个人的基本信息
个人(患者)	Patient	复合	包含自然人信息, 作为档案记录, 包括个人档案相关信息、注册信息等外部相关信息。
机构(科室)	Organization	复合	医疗卫生机构(科室)、管理机构等, 科室进行注册时, partOf为科室所在机构。
从业人员	Practitioner	复合	医疗卫生机构从业人员, 包括医生、护士、药剂师、健康管理师、管理人员等。

表4 (续)

数据名称	类型代码	数据类别	备注
个人查询参数	PersonQueryPayload	复合	个人查询参数
机构查询参数	OrganizationQueryPayload	复合	机构查询参数
从业人员查询参数	PractitionerQueryPayload	复合	从业人员查询参数
扩展	Extension	复合	支持对象扩展
对象状态基类	objectStatusPrimitive	简单	对象状态基类。值域代码见B.7。
对象状态	ObjectStatus	复合	对象的可用状态
对象引用	ObjectRef	复合	通用注册对象引用
个人信息摘要	PersonBrief	复合	患者(个人)的信息摘要
从业人员信息摘要	PractitionerBrief	复合	从业人员信息摘要
机构(科室)信息摘要	OrganizationBrief	复合	机构(科室)信息摘要
通用查询	AdhocQuery	复合	预存的通用查询
通用查询	adhocQuery	元素	通用查询元素。本消息元素在订阅服务/Subscribe/Filter节点或查询请求中引用。

### 6.3 术语注册基础数据类型

与术语注册服务相关的消息应采用的数据类型及代码说明见表5, 数据类型的表达应符合A.3术语注册数据类型结构模式的约束。

表5 通知服务基础对象数据类型

数据名称	类型代码	数据类别	说明
通用术语注册对象	RegTermObject	复合	抽象类, 术语注册通用对象。
值集	ValueSet	复合	概念的编码集合
概念映射	ConceptMap	复合	建立源值集与目标值集间的映射关系
概念映射目标	ConceptMapTarget	复合	概念映射目标
概念映射元素	ConceptMapElement	复合	概念映射元素
内容结构体	Contains	复合	内容结构体
等值比较	EquivalenceTypeList	复合	等值比较
等值类型	EquivalenceType	复合	等值类型
值集定义	ValueSetDefine	复合	值集定义
值集展开模式	ValueSetExpansion	复合	值集展开模式
值集构成	ValueSetCompose	复合	值集构成
值集包含	ValueSetInclude	复合	值集包含
等值比较基类	equivalencePrimitive	简单	等值比较基类
等值比较类	EquivalenceType	复合	等值比较类
值集定义	ValueSetDefine	复合	值集定义
值集展开模式	ValueSetExpansion	复合	值集展开模式
值集构成	ValueSetCompose	复合	值集构成
值集包含	ValueSetInclude	复合	值集包含

### 6.4 文档共享服务数据类型

与文档共享相关服务的数据类型及通用元素，包括文档存储服务、文档管理服务、文档采集服务、文档调阅服务、文档订阅-发布服务以及其它应用服务。

数据类型的及代码说明见表6，数据类型的表达示应符合A.4文档共享服务通用元素及数据类型的结构模式的约束。文档类型未加说明时应符合WS/T 483(所有部分)及WS/T 500(所有部分)规定。

表6 文档共享服务通用元素及数据类型

数据名称	类型代码	数据类别	备注
提交集	SubmissionSet	复合	提交集类型
文件夹	Folder	复合	文件夹元素数
文档条目	DocumentEntry	复合	文档条目
文档条目关联	Association	复合	文档条目与文档条目之间的关联关系：替换、转换、签名等。
文档索引	DocumentReference	复合	文档索引
已存的文档	StoredDocument	复合	文档,用于获取文档时, id为OID。
文档	Document	复合	主要指待存的文档,用于提交文档时。用于提交文档时, id为URI,可以为暂代符或者UUID与DocumentEntry中的id一致。
作者	Author	复合	作者
关联对象	AssociationObject	复合	关联的对象
关联状态基类	associationStatusPrimitive	复合	关联状态基类, 值域见B. 8。
关联状态	AssociationStatus	复合	关联状态
关联类型基类	associationTypePrimitive	简单	关联的类型枚举, 值域见B. 9。
关联类型	AssociationType	复合	关联类型
对象类型基类	objectType	简单	对象类型枚举, 值域见B. 10。
对象类型	ObjectType	复合	对象类型

## 7 通用查询

### 7.1 应用范围

通用查询应用于个人、医疗卫生机构、医疗卫生人员及文档的一般预存储查询、通知的订阅过滤条件设定。

### 7.2 消息结构及约束

通用查询消息结构应符合A.2主数据对象数据类型及通用元素消息格式adhocQuery元素构件要求,消息补充约束见表7。

表7 通用查询 adhocQuery 的消息补充约束

节点	基数	可选项	节点说明	对应数据元标识符
adhocQuery	0..1	0	预存查询条件。	

表7 (续)

节点	基数	可选项	节点说明	对应数据元标识符
adhocQuery/id	1..1	R	预存查询标识。URI类型,不同的预存查询定义唯一的查询唯一编码,具体见各部分定义。	
adhocQuery/slot	1..*	R	预存查询参数列表,各参数之间为AND关系。	
adhocQuery/slot/name/@value	1..1	R	过滤条件名称。以"\$"符引导,表示查询过滤条件的名称,对应查询对象不同的消息节点,具体见各部分定义。	
adhocQuery/slot/valueList	1..1	R	取值列表,多值之间为OR关系	
adhocQuery/slot/valueList/value/@value	1..*	R	过滤条件值。当值@value与过滤条件所指定的节点匹配时,过滤条件成立,具体由预存查询时给定。	

## 8 通用服务处理

### 8.1 异步处理

交互服务中如需要提供异步支持,应在交互规范中编写交互服务的异步处理流程。

### 8.2 错误处理

在处理服务的网络请求出现错误时,应通过SOAP消息Fault元素将错误信息反馈给服务调用者。Fault元素应位于SOAP消息体内,在一条SOAP消息中,Fault元素只能出现一次。具体细节请参考W3C SOAP Version 1.2 Part 1: Messaging Framework规范。

Fault元素的子元素说明及约束见表8:

表8 Fault 元素消息语法约束

元素路径	元素描述	其它约束
Fault	错误	错误
Fault/Code	错误代码	由W3C SOAP 1.2规范约定
Fault/Code/Value	错误代码的取值	值域代码见B.11
Fault/Code/Subcode	子级错误代码	
Fault/Code/Subcode/Value	子级错误代码的取值	由各具体服务交互规范约定
Fault/Reason	错误原因	
Fault/Reason/Text	可供人阅读的错误详细说明	

故障消息示例见C.1。

### 8.3 审计要求

在每部分具体交互规范中应编写所有交互服务的审计消息。  
应按交互服务的角色分别对交互服务设定审计消息所需内容。

### 8.4 安全要求

在SOAP消息头中，应采用符合W3C WS-Security规范的元素进行服务交互双方的身份认证，以及确保服务交互中信息的完整性、不可篡改性及不可抵赖性。

#### 8.4.1 服务寻址

在SOAP消息中，应采用符合W3C WS-Addressing规范的元素进行服务寻址，以标识服务发送者、服务接收者和服务交易。规定如下：

- 应提供 From 节点用以标识服务的发送者；
  - 应在 Address 子节点中记录区域平台分配给服务发送者的节点标识符，该节点标识符为 OID 类型，在记录时应添加"urn:oid:"前缀；
  - 应提供 To 节点用以标识服务的接收者，取值为服务的可访问网络地址；
  - 应提供 Action 节点用以标识服务交易，根据 WS-Addressing 规范 Action 节点命名规则为：  
<目标命名空间><分隔符><端口类型><分隔符><交易(输入|输出)名称>
- 示例见C.2。

#### 8.4.2 服务访问安全

应遵循W3C WS-Security规范，在SOAP消息头中，采用WS-Security消息进行服务交互双方的身份认证，以及确保服务交互中信息的完整性、不可篡改性及不可抵赖性。Security节点规定如下：

- 应使用 BinarySecurityToken 安全性令牌，用以存储 X509v3 证书；
- 应使用 Base64Binary 对 X509v3 证书进行转码；
- 应使用数字签名确保服务传输信息的完整性、不可篡改性及不可抵赖性；

在对消息进行签名前应对被签名的消息依据W3C专用XML规范化协议1.0进行规范化，标识为"<https://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718/>"。可选用的签名算法包括：

- HMAC 算法，标识为"<http://www.w3.org/2000/09/xmldsig#hmac-sha1>";
  - DSA 算法，标识为"<http://www.w3.org/2000/09/xmldsig#dsa-sha1>";
  - PKCS1 算法，标识为"<http://www.w3.org/2000/09/xmldsig#rsa-sha1>"。
- 摘要算法应使用SHA-1算法，标识为"<http://www.w3.org/2000/09/xmldsig#sha1>"。

示例见C.3。

附 录 A  
(规范性)  
消息数据类型结构模式

### A.1 基础数据类型

基础数据类型的结构模式采用XML Schema定义如下：

<p>文件名：RHIN-Base.xsd</p> <pre> &lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;!-- Ver.0.12.3/2018-07-11 Dejun Hsu--&gt; &lt;xs:schema xmlns="http://www.chiss.org.cn/rhin/2015" xmlns:xs="http://www.w3.org/2001/XMLSchema"   targetNamespace="http://www.chiss.org.cn/rhin/2015" elementFormDefault="qualified"&gt;   &lt;xs:simpleType name="nullFlavorPrimitive"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;         NI - 无法从任何情况下获取信息(粗粒度值)         NA - 该值不适用于该上下文         NP - 在消息中该值不存在         NASK - 未询问患者         ASKU - 询问过患者,但是不知道         MSK - 因安全等原因该值被掩藏         NAV - 该值当前不可用,但在可预期的将来该值一定可用         UNK - 有值可用,但是不知道         OTH - 其它       &lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:restriction base="stringPrimitive"&gt;       &lt;xs:enumeration value="UNK"/&gt;       &lt;xs:enumeration value="NAV"/&gt;       &lt;xs:enumeration value="NASK"/&gt;       &lt;xs:enumeration value="ASKU"/&gt;       &lt;xs:enumeration value="NP"/&gt;       &lt;xs:enumeration value="NA"/&gt;       &lt;xs:enumeration value="NI"/&gt;       &lt;xs:enumeration value="MSK"/&gt;       &lt;xs:enumeration value="OTH"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; </pre>
--

```

<xs:complexType name="Any" abstract="true">
  <xs:attribute name="nullFlavor" type="nullFlavorPrimitive" use="optional"/>
</xs:complexType>
<xs:simpleType name="stringPrimitive">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="String">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="stringPrimitive" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="booleanPrimitive">
  <xs:restriction base="xs:boolean"/>
</xs:simpleType>
<xs:complexType name="Boolean">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="booleanPrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="datePrimitive">
  <xs:restriction>
    <xs:simpleType>
      <xs:union memberTypes="xs:gYear xs:gYearMonth xs:date"/>
    </xs:simpleType>
    <xs:pattern value="-?[0-9]{4}(-(0[1-9]|1[0-2])-(0[0-9]|[1-2][0-9]|3[0-1]))?" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Date">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="datePrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="timePrimitive">
  <xs:restriction base="xs:time">
    <xs:pattern value="([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](\.[0-9]+)?" />
  </xs:restriction>
</xs:simpleType>

```



```

    </xs:restriction>
</xs:simpleType>
<xs:complexType name="Time">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="timePrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="dateTimePrimitive">
  <xs:restriction>
    <xs:simpleType>
      <xs:union memberTypes="xs:gYear xs:gYearMonth xs:date xs:dateTime"/>
    </xs:simpleType>
    <xs:pattern
      value="-?[0-9]{4}(-([0-9]{1}[0-2])|(-([0-9]{1}[1-2][0-9]{3}[0-1])(T([01][0-9]{2}[0-3]):[0-5][0-9]:[0-5][0-9](\.[0-9]+)?)(Z|(\+|-)((
0[0-9]{1}[0-3]):[0-5][0-9]{14:00))?)?)?)?"
      />
    </xs:restriction>
  </xs:simpleType>
<xs:complexType name="DateTime">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="dateTimePrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="durationUnitPrimitive">
  <xs:restriction base="stringPrimitive">
    <xs:enumeration value="yr"/>
    <xs:enumeration value="mth"/>
    <xs:enumeration value="wk"/>
    <xs:enumeration value="d"/>
    <xs:enumeration value="h"/>
    <xs:enumeration value="min"/>
    <xs:enumeration value="s"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Duration">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="unsignedIntPrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <xs:attribute name="unit" type="durationUnitPrimitive"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:simpleType name="languagePrimitive">
    <xs:restriction base="xs:language"/>
</xs:simpleType>
<xs:complexType name="Language">
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="languagePrimitive"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="codePrimitive">
    <xs:restriction base="xs:token">
        <xs:minLength value="1"/>
        <xs:pattern value="^[^s]+([\s]?[^\s]+)*"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="Code">
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="codePrimitive"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="unsignedIntPrimitive">
    <xs:restriction base="xs:nonNegativeInteger">
        <xs:pattern value="[0]([1-9][0-9]*)"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="UnsignedInt">
    <xs:annotation>
        <xs:documentation>正整数</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="unsignedIntPrimitive"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

<xs:simpleType name="integerPrimitive">
  <xs:restriction base="xs:int">
    <xs:pattern value="-?([0]|([1-9][0-9]*))"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Integer">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="integerPrimitive" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="decimalPrimitive">
  <xs:restriction base="xs:decimal">
    <xs:pattern value="-?([0]|([1-9][0-9]*))(\.[0-9]+)?"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Decimal">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="decimalPrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="ncNamePrimitive">
  <xs:restriction base="xs:NCName"/>
</xs:simpleType>
<xs:complexType name="NCName">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="ncNamePrimitive" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="qNamePrimitive">
  <xs:restriction base="xs:QName"/>
</xs:simpleType>
<xs:complexType name="QName">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="qNamePrimitive" use="optional"/>
    </xs:extension>
  </xs:complexContent>

```

```

</xs:complexType>
<xs:simpleType name="base64BinaryPrimitive">
  <xs:restriction base="xs:base64Binary"/>
</xs:simpleType>
<xs:complexType name="Base64Binary">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="base64BinaryPrimitive" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="uriPrimitive">
  <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
<xs:complexType name="URI">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="uriPrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="urnPrimitive">
  <xs:restriction base="xs:anyURI">
    <xs:pattern value="urn:[A-Za-z0-9:._\-\.]{1,}" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="URN">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="urnPrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="idPrimitive">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:pattern value="[A-Za-z0-9\-\.\.]{1,64}" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ID">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="idPrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="oidPrimitive">
    <xs:restriction base="urnPrimitive">
        <xs:minLength value="1"/>
        <xs:pattern value="urn:oid:(0|[1-9][0-9]*)\.(0|[1-9][0-9]*)*/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="OID">
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="oidPrimitive"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="uuidPrimitive">
    <xs:restriction base="urnPrimitive">
        <xs:minLength value="1"/>
        <xs:pattern
            value="urn:uuid:[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="UUID">
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="uuidPrimitive"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="instantPrimitive">
    <xs:restriction base="xs:dateTime"/>
</xs:simpleType>
<xs:complexType name="Instant">
    <xs:annotation>
        <xs:documentation>精确时间,总是包括时区.通常用于系统日志记录,该时间非用于人
    读.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="instantPrimitive" use="optional"/>
        </xs:extension>
    </xs:complexContent>

```

```

</xs:complexType>
<xs:complexType name="Identifier">
  <xs:annotation>
    <xs:documentation>标识域ID</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:sequence>
        <xs:element name="system" type="URI" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>标识域、系统</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="value" type="ID" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="Period">
  <xs:annotation>
    <xs:documentation>时间段</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:sequence>
        <xs:element name="start" type="DateTime"/>
        <xs:element name="end" type="DateTime" minOccurs="0" maxOccurs="1"/>
        <xs:element name="duration" type="Duration" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="Reference">
  <xs:annotation>
    <xs:documentation>引用</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:sequence>
        <xs:element name="reference" type="Identifier" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>引用指向.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element name="display" type="String" minOccurs="0" maxOccurs="1">
        <xs:annotation>
            <xs:documentation>显示名称.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="Coding">
    <xs:annotation>
        <xs:documentation>编码概念</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:sequence>
                <xs:element name="code" type="Code" minOccurs="0" maxOccurs="1">
                    <xs:annotation>
                        <xs:documentation>编码值</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="display" type="String" minOccurs="0" maxOccurs="1">
                    <xs:annotation>
                        <xs:documentation>编码的显示名称.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="system" type="URI" minOccurs="0" maxOccurs="1">
                    <xs:annotation>
                        <xs:documentation>所属编码系统</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="systemName" type="String">
                    <xs:annotation>
                        <xs:documentation>编码系统名称</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="version" type="String" minOccurs="0" maxOccurs="1">
                    <xs:annotation>
                        <xs:documentation>版本</xs:documentation>
                    </xs:annotation>
                </xs:annotation>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="CodeableConcept">
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:sequence>
                <xs:element name="coding" type="Coding" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="text" type="String" minOccurs="0" maxOccurs="1"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Quantity">
    <xs:annotation>
        <xs:documentation>计量值</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:sequence>
                <xs:element name="value" type="Decimal" minOccurs="0" maxOccurs="1">
                    <xs:annotation>
                        <xs:documentation>测量值</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="unit" type="String" minOccurs="0" maxOccurs="1">
                    <xs:annotation>
                        <xs:documentation>人可读的单位</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="nameUsePrimitive">
    <xs:annotation>
        <xs:documentation>姓名的用途的枚举值</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="Usual">
            <xs:annotation>

```



```

        <xs:documentation xml:lang="en">常用名</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="Official">
    <xs:annotation>
        <xs:documentation xml:lang="en">正式名字</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="Temp">
    <xs:annotation>
        <xs:documentation xml:lang="en">临时名字</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="Nickname">
    <xs:annotation>
        <xs:documentation xml:lang="en">昵称、网名、号</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="Anonymous">
    <xs:annotation>
        <xs:documentation xml:lang="en">别名（字）</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="Old">
    <xs:annotation>
        <xs:documentation xml:lang="en">曾用名</xs:documentation>
    </xs:annotation>
</xs:enumeration>
<xs:enumeration value="Maiden">
    <xs:annotation>
        <xs:documentation xml:lang="en">婚前名</xs:documentation>
    </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>
<xs:complexType name="NameUse">
    <xs:annotation>
        <xs:documentation xml:lang="en"/>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="nameUsePrimitive" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

    </xs:complexContent>
</xs:complexType>
<xs:complexType name="HumanName">
  <xs:annotation>
    <xs:documentation>人的姓名</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:sequence minOccurs="0" maxOccurs="1">
        <xs:element minOccurs="0" name="use" type="NameUse">
          <xs:annotation>
            <xs:documentation>姓名用途</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="family" type="String">
          <xs:annotation>
            <xs:documentation>姓</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="given" type="String">
          <xs:annotation>
            <xs:documentation>名字</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="prefix" type="String">
          <xs:annotation>
            <xs:documentation>前缀</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="suffix" type="String">
          <xs:annotation>
            <xs:documentation>后缀</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="period" type="Period" minOccurs="0">
          <xs:annotation>
            <xs:documentation>有效时间段</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="text" type="String">
          <xs:annotation>
            <xs:documentation>补充说明</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
    <xs:attribute name="value" type="stringPrimitive">
        <xs:annotation>
            <xs:documentation>全名</xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:simpleType name="addressUsePrimitive">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Home">
            <xs:annotation>
                <xs:documentation xml:lang="en">家庭地址</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Work">
            <xs:annotation>
                <xs:documentation xml:lang="en">工作地址</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Temp">
            <xs:annotation>
                <xs:documentation xml:lang="en">临时地址</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Old">
            <xs:annotation>
                <xs:documentation xml:lang="en">旧地址</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="AddressUse">
    <xs:annotation>
        <xs:documentation xml:lang="en">地址的用途</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="addressUsePrimitive" use="optional"/>
        </xs:extension>
    </xs:complexContent>

```

```

</xs:complexType>
<xs:simpleType name="addressTypePrimitive">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Postal">
      <xs:annotation>
        <xs:documentation xml:lang="en">投递地址</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="Physical">
      <xs:annotation>
        <xs:documentation xml:lang="en">地理位置.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="Both">
      <xs:annotation>
        <xs:documentation xml:lang="en">投递地址与地理位置通用.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="AddressType">
  <xs:annotation>
    <xs:documentation xml:lang="en">地址类别.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="addressTypePrimitive" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="Address">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:sequence>
        <xs:element minOccurs="0" name="use" type="AddressUse">
          <xs:annotation>
            <xs:documentation>地址类型</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="type" type="AddressType"/>
        <xs:element name="text" type="String" minOccurs="0">
          <xs:annotation>
            <xs:documentation>地址的纯文本写法</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="administrativeDivision" type="String">
        <xs:annotation>
            <xs:documentation>行政区划</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="country" type="String">
        <xs:annotation>
            <xs:documentation>国家</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="state" type="String">
        <xs:annotation>
            <xs:documentation>省、自治区、直辖市</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="city" type="String">
        <xs:annotation>
            <xs:documentation>地级市、区、盟</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="district" type="String">
        <xs:annotation>
            <xs:documentation>县、县级市、地区、旗</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="town" type="String">
        <xs:annotation>
            <xs:documentation>乡（镇、街道办事处）</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="street" type="String">
        <xs:annotation>
            <xs:documentation>村（街、路、弄等）</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="houseNumber" type="String">
        <xs:annotation>
            <xs:documentation>门牌号码</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="line" type="String" maxOccurs="unbounded">

```

```

        <xs:annotation>
            <xs:documentation>结构化地址的其它信息</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="postalCode" minOccurs="0" type="String">
        <xs:annotation>
            <xs:documentation>邮政编码</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="period" type="Period">
        <xs:annotation>
            <xs:documentation>地址的有效时间段</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="BackboneElement">
    <xs:annotation>
        <xs:documentation>资源定义内部所有元素的基类</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="Any"/>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="telecomModePrimitive">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Telephone">
            <xs:annotation>
                <xs:documentation xml:lang="en">投递地址</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Cellphone">
            <xs:annotation>
                <xs:documentation xml:lang="en">地理位置.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Email">
            <xs:annotation>
                <xs:documentation xml:lang="en">投递地址与地理位置通用.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="QQ"/>
        <xs:enumeration value="Wechat"/>
        <xs:enumeration value="Other"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="TelecomMode">
    <xs:annotation>
        <xs:documentation>联系方式的类型</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="telecomModePrimitive" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Telecom">
    <xs:annotation>
        <xs:documentation>通讯地址</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="use" type="TelecomMode"/>
        <xs:element name="value" type="String"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="NameValuePairType">
    <xs:annotation>
        <xs:documentation>编码名值对类型</xs:documentation>
    </xs:annotation>
    <xs:attribute name="name" type="stringPrimitive" use="required"/>
    <xs:attribute name="value" type="stringPrimitive" use="required"/>
</xs:complexType>
<xs:complexType name="ValueList">
    <xs:annotation>
        <xs:documentation>值列表，多个值之间为或的关系</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="value" type="String" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Slot">
    <xs:annotation>
        <xs:documentation>插槽类型，用于查询、过滤器、主题等插入条件。</xs:documentation>
    </xs:annotation>

```

```

<xs:sequence>
  <xs:element name="name" type="String">
    <xs:annotation>
      <xs:documentation>关键字</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element maxOccurs="1" name="valueList" type="ValueList">
    <xs:annotation>
      <xs:documentation>取值列表,多值之间为OR关系</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="SlotListType">
  <xs:annotation>
    <xs:documentation>插槽列表类型,多个插槽间是并的关系。</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="slot" type="Slot"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="NullResponse"/>
</xs:schema>

```

## A.2 主数据对象数据类型及通过元素

主数据对象相关数据类型定义如下:

```

文件名: RHIN-BaseObject.xsd
<?xml version="1.0" encoding="utf-8" ?>
<!-- Ver.0.12.3/2018-07-12 Dejun Hsu-->
<xs:schema xmlns="http://www.chiss.org.cn/rhin/2015" elementFormDefault="qualified"
  targetNamespace="http://www.chiss.org.cn/rhin/2015" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="RHIN-Base.xsd"/>
  <xs:complexType name="RegObject" abstract="true">
    <xs:sequence>
      <xs:element minOccurs="0" name="id" type="MainID">
        <xs:annotation>
          <xs:documentation>注册对象的唯一标识号,由注册对象管理方分配</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" name="statusCode" type="ObjectStatus">
        <xs:annotation>

```



```

        <xs:documentation>注册对象的状态（新建\可用\废弃）</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="classCode" type="CodeableConcept">
    <xs:annotation>
        <xs:documentation>注册对象的分类（最粗粒度的分类）
如果需要细粒度的分类,在具体对象添加</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="extension" type="Extension">
    <xs:annotation>
        <xs:documentation>扩展机制.
当注册对象未含有期望的元素,可以通过该方法进行扩展.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="BloodType">
    <xs:sequence>
        <xs:element name="RH" type="Boolean"/>
        <xs:element name="ABO" type="CodeableConcept"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="HealthHistoryType">
    <xs:annotation>
        <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="name" type="CodeableConcept"/>
        <xs:element minOccurs="0" name="confirmedTime" type="DateTime"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="FamilyHistoryType">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="1" name="relatives">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="relativeType" type="Coding"/>
                    <xs:element name="diseaseHistory" type="HealthHistoryType"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:sequence>

```

```

</xs:complexType>
<xs:complexType name="PhysicalCondition">
  <xs:sequence>
    <xs:element minOccurs="0" name="bloodType" type="BloodType"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="allergicHistory"
      type="HealthHistoryType"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="diseaseHistory"
      type="HealthHistoryType"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="operationHistory"
      type="HealthHistoryType"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="traumaHistory"
      type="HealthHistoryType"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="transfusionHistory"
      type="HealthHistoryType"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="geneticHistory"
      type="HealthHistoryType"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="familyHistory"
      type="FamilyHistoryType"> </xs:element>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="disabilitySituation"
      type="CodeableConcept"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Contact">
  <xs:annotation>
    <xs:documentation>联系人类型</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="identifier" type="Identifier">
      <xs:annotation>
        <xs:documentation>标识符</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="code" type="CodeableConcept">
      <xs:annotation>
        <xs:documentation>编码</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="address" type="Address">
      <xs:annotation>
        <xs:documentation>地址</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="telecom" type="Telecom">

```

```

        <xs:annotation>
            <xs:documentation>联系电话</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="assignedPerson" type="Person"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="Guardian">
    <xs:annotation>
        <xs:documentation/>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="identifier" type="Identifier" maxOccurs="unbounded"/>
        <xs:element minOccurs="0" name="code" type="CodeableConcept">
            <xs:annotation>
                <xs:documentation>编码</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="address" type="Address"/>
        <xs:element name="telecom" type="Telecom" minOccurs="0"/>
        <xs:element minOccurs="0" name="effectiveTime" type="Period"/>
        <xs:choice>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="guardianPerson" type="Person">
</xs:element>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="guardianOrganization"
                type="Organization"> </xs:element>
        </xs:choice>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Person">
    <xs:annotation>
        <xs:documentation>居民实体类</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="RegObject">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="identifier" type="Identifier"
                    minOccurs="0"/>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="name" type="HumanName"/>
                <xs:element name="birthTime" type="Date" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>出生日期</xs:documentation>
                    </xs:annotation>

```

```

</xs:element>
<xs:element minOccurs="0" name="gender" type="Code">
  <xs:annotation>
    <xs:documentation>性别</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="address" type="Address">
  <xs:annotation>
    <xs:documentation>地址</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="telecom" type="Telecom">
  <xs:annotation>
    <xs:documentation>联系方式</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="languageCommunication" type="Language"
  maxOccurs="unbounded"/>
<xs:element minOccurs="0" name="birthPlace" type="Address">
  <xs:annotation>
    <xs:documentation>出生地</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="maritalStatusCode" type="Code">
  <xs:annotation>
    <xs:documentation>婚姻状态</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:choice minOccurs="0">
  <xs:element name="deceasedInd" type="Boolean">
    <xs:annotation>
      <xs:documentation>死亡标记</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="deceasedTime" type="DateTime">
    <xs:annotation>
      <xs:documentation>死亡时间</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:choice>
<xs:choice minOccurs="0">
  <xs:element name="multipleBirthInd" type="Boolean">
    <xs:annotation>

```

```

        <xs:documentation>多胎标记</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="multipleBirthOrderNumber" type="Integer">
    <xs:annotation>
        <xs:documentation>多胎次序</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:choice>

<xs:element minOccurs="0" name="photo" type="Base64Binary">
    <xs:annotation>
        <xs:documentation>头像图片</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Patient">
    <xs:complexContent>
        <xs:extension base="RegObject">
            <xs:sequence>
                <xs:element minOccurs="0" name="identifier" type="Identifier"
                    maxOccurs="unbounded">
                    <xs:annotation>
                        <xs:documentation>患者角色编码、如就诊号、健康卡号、住院号、门诊号等。
                    </xs:documentation>
                </xs:annotation>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:documentation>

    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="address" type="Address">
    <xs:annotation>
        <xs:documentation>地址</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="telecom" type="Telecom">
    <xs:annotation>
        <xs:documentation>联系电话
    </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="effectiveTime" type="DateTime"/>
<xs:element name="assignedPerson" type="Person" minOccurs="0"/>

```

```

        <xs:element minOccurs="0" name="assignedOrganization" type="Organization">
            <xs:annotation>
                <xs:documentation>居委会</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="contact" type="Contact" maxOccurs="unbounded"/>
        <xs:element minOccurs="0" name="guardian" type="Guardian" maxOccurs="unbounded" />
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Organization">
    <xs:annotation>
        <xs:documentation>科室、机构</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="RegObject">
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="identifier"
                    type="Identifier">
                    <xs:annotation>
                        <xs:documentation>机构编码</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="name" type="String">
                    <xs:annotation>
                        <xs:documentation>机构名称</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="healthcareFacilityTypeCode"
                    type="CodeableConcept">
                    <xs:annotation>
                        <xs:documentation>机构的类型</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="practiceSettingCode" type="CodeableConcept">
                    <xs:annotation>
                        <xs:documentation>机构的专业分类</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="address" type="Address">
                    <xs:annotation>
                        <xs:documentation>地址</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="telecom" type="Telecom">
        <xs:annotation>
            <xs:documentation>联系方式</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="effectiveTime" type="DateTime">
        <xs:annotation>
            <xs:documentation>机构创建时间</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="partOf" type="Organization" minOccurs="0">
        <xs:annotation>
            <xs:documentation>指向上一级机构</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="MainID">
    <xs:complexContent>
        <xs:extension base="ID">
            <xs:sequence>
                <xs:element name="system" type="uriPrimitive"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Practitioner">
    <xs:annotation>
        <xs:documentation>医护人员类型</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="RegObject">
            <xs:sequence>
                <xs:element minOccurs="0" name="identifier" type="Identifier"
                    maxOccurs="unbounded">
                    <xs:annotation>
                        <xs:documentation>医疗卫生人员的编码，如工号、统一编码等。</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

<xs:element minOccurs="0" name="duty" type="Coding" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>职务</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="professionalTitle" type="Coding"
  maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>职称</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="address" type="AddressType" minOccurs="0"
  maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>联系地址</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="telecom" minOccurs="0" maxOccurs="unbounded" type="Telecom">
  <xs:annotation>
    <xs:documentation>联系电话</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="assignedPerson" type="Person"/>
<xs:element name="practiceOrganization" type="Organization" minOccurs="0">
  <xs:annotation>
    <xs:documentation>科室及执业机构</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="PersonBrief">
  <xs:annotation>
    <xs:documentation>居民信息查询参数</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="identifier" type="Identifier">
      <xs:annotation>
        <xs:documentation>标识符</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="name" type="HumanName">

```



```

        <xs:annotation>
            <xs:documentation>名称</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="birthTime" type="Date">
        <xs:annotation>
            <xs:documentation>出生日期</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="gender" type="Code">
        <xs:annotation>
            <xs:documentation>性别</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="addr" type="Address">
        <xs:annotation>
            <xs:documentation>地址</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="telecom" type="Telecom">
        <xs:annotation>
            <xs:documentation>联系方式</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="OrganizationBrief">
    <xs:annotation>
        <xs:documentation>机构查询参数</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="identifier" type="Identifier">
            <xs:annotation>
                <xs:documentation>标识符</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="classCode" type="Coding" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>机构（科室）类别</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="name" type="String">
            <xs:annotation>

```

```

        <xs:documentation>机构名称</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="address" type="Address">
      <xs:annotation>
        <xs:documentation>地址</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="telecom" type="Telecom">
      <xs:annotation>
        <xs:documentation>联系方式</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="partOf" type="OrganizationBrief"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PractitionerBrief">
  <xs:annotation>
    <xs:documentation>医护人员查询参数</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="identifier" type="Identifier">
      <xs:annotation>
        <xs:documentation>标识符</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="name" type="HumanName">
      <xs:annotation>
        <xs:documentation>医务人员姓名</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="organization" type="OrganizationBrief"/>
    <xs:element minOccurs="0" name="duty" type="Coding" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>职务</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="professionalTitle" type="Coding" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>职称</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>

```

```

</xs:complexType>
<xs:complexType name="Extension">
  <xs:annotation>
    <xs:documentation>扩展类型,支持扩展机制。</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="name" type="URI">
      <xs:annotation>
        <xs:documentation>关键字</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="unbounded" name="value" type="Any">
      <xs:annotation>
        <xs:documentation>取值,可以为Any的任何继承类</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="objectStatusPrimitive">
  <xs:annotation>
    <xs:documentation>
      New      - 新建
      Active   - 可用
      Nullified - 废弃
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="stringPrimitive">
    <xs:enumeration value="New"/>
    <xs:enumeration value="Active"/>
    <xs:enumeration value="Nullified"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ObjectStatus">
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="objectStatusPrimitive"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AdhocQuery">
  <xs:annotation>
    <xs:documentation>预存查询类型</xs:documentation>
  </xs:annotation>

```

```

<xs:sequence>
  <xs:element name="id" type="URN">
    <xs:annotation>
      <xs:documentation>预存储查询的唯一编号</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element maxOccurs="unbounded" name="slot" type="Slot">
    <xs:annotation>
      <xs:documentation>预存储查询参数列表,各参数之间为AND关系</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:element name="adHocQuery" type="AdHocQuery">
  <xs:annotation>
    <xs:documentation>通用查询元素
    本消息元素在订阅服务/Subscribe/Filter节点或查询中引用。</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="ObjectRef">
  <xs:annotation>
    <xs:documentation>对象索引</xs:documentation>
  </xs:annotation>
  <xs:sequence minOccurs="0">
    <xs:element name="id" type="MainID" maxOccurs="1" minOccurs="0">
      <xs:annotation>
        <xs:documentation>主域标识（注册码、主索引号）</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

### A.3 术语注册数据类型

术语相关数据类型定义如下：

```

文件名：RHIN-BaseTerminology.xsd
<?xml version="1.0" encoding="utf-8" ?>
<!-- Ver.0.11.2/2018-05-08 Dejun Hsu-->
<xs:schema xmlns="http://www.chiss.org.cn/rhin/2015" elementFormDefault="qualified"
  targetNamespace="http://www.chiss.org.cn/rhin/2015" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="RHIN-BaseObject.xsd"/>

```

```

<xs:complexType abstract="true" name="RegTermObject">
  <xs:annotation>
    <xs:documentation>术语通用注册对象</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RegObject">
      <xs:sequence>
        <xs:element name="name" type="String" minOccurs="0">
          <xs:annotation>
            <xs:documentation>术语注册对象名称，如值集、值集映射等名称。</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="publisher" type="String">
          <xs:annotation>
            <xs:documentation>发布者</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="description" type="String">
          <xs:annotation>
            <xs:documentation>描述</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="effectiveTime" type="Period">
          <xs:annotation>
            <xs:documentation>有效时间</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="group" type="Reference">
          <xs:annotation>
            <xs:documentation>使用组织名称</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ValueSet">
  <xs:annotation>
    <xs:documentation>值集</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RegTermObject">
      <xs:sequence>

```

```

        <xs:element minOccurs="0" name="define" type="ValueSetDefine">
            <xs:annotation>
                <xs:documentation>自定义值集编码，不从编码系统引入，形成隐式编码系统
            </xs:documentation>
        </xs:element>
        <xs:element minOccurs="0" name="compose" type="ValueSetCompose">
            <xs:annotation>
                <xs:documentation>值集的组合定义，声明值集由哪些编码系统的哪些编码组成
            </xs:documentation>
        </xs:element>
        <xs:element minOccurs="0" name="expansion" type="ValueSetExpansion">
            <xs:annotation>
                <xs:documentation>展开模式。相当于把所有编码全部列出</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexType>
<xs:complexType name="ConceptMap">
    <xs:annotation>
        <xs:documentation>概念映射</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
        <xs:extension base="RegTermObject">
            <xs:sequence>
                <xs:element name="source" type="Reference">
                    <xs:annotation>
                        <xs:documentation>编码映射的源值集</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="target" type="Reference">
                    <xs:annotation>
                        <xs:documentation>编码映射的目标值集</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element minOccurs="0" name="element" type="ConceptMapElement">
                    <xs:annotation>
                        <xs:documentation>映射的编码 </xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ConceptMapTarget">
    <xs:annotation>
        <xs:documentation>概念映射目标</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="code" type="String" minOccurs="0">
            <xs:annotation>
                <xs:documentation>目标编码值</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="codeSystem" type="OID" minOccurs="0">
            <xs:annotation>
                <xs:documentation>目标编码系统</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="equivalence" type="EquivalenceType">
            <xs:annotation>
                <xs:documentation>编码映射关联</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ConceptMapElement">
    <xs:annotation>
        <xs:documentation>概念映射元素</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="code" type="Code">
            <xs:annotation>
                <xs:documentation>源编码值 </xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="codeSystem" type="OID">
            <xs:annotation>
                <xs:documentation>源编码系统</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="target" type="ConceptMapTarget">

```

```

    <xs:annotation>
      <xs:documentation>目标编码</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="Contains">
  <xs:sequence>
    <xs:element name="code" type="Coding" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>编码值</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="contains" type="Contains"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="equivalencePrimitive">
  <xs:annotation>
    <xs:documentation>Equivalent - 相等, 等价的 Equal - 相同的 Wider - 目标编码的含义比源编码宽泛
Subsumes - 源编码
is-a目标编码 Narrower - 目标编码的含义比源编码狭窄 Spechializes - 目标编码 is-a 源编码 Inexact -
不精确 UnMatched - 不匹配
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="Equivalent"/>
    <xs:enumeration value="Equal"/>
    <xs:enumeration value="Wider"/>
    <xs:enumeration value="Subsumes"/>
    <xs:enumeration value="Narrower"/>
    <xs:enumeration value="Spechializes"/>
    <xs:enumeration value="Inexact"/>
    <xs:enumeration value="UnMatched"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="EquivalenceType">
  <xs:annotation>
    <xs:documentation>等值类型</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="Any">
      <xs:attribute name="value" type="equivalencePrimitive" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```



```

    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ValueSetDefine">
    <xs:annotation>
      <xs:documentation>值集定义</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="code" type="Code" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ValueSetExpansion">
    <xs:annotation>
      <xs:documentation>值集展开模式</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="version" type="String">
        <xs:annotation>
          <xs:documentation>展开模式对应的版本信息。如果与值集的版本信息不符合，则需要重新获取展
          开模式编码列表</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="effectiveTime" type="DateTime">
        <xs:annotation>
          <xs:documentation>展开模式生成的时间</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="contains" type="Contains">
        <xs:annotation>
          <xs:documentation>编码列表</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ValueSetCompose">
    <xs:annotation>
      <xs:documentation>值集购成</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="import" type="OID" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>整个编码系统的导入</xs:documentation>
        </xs:annotation>
      </xs:element>

```

```

    <xs:element name="include" minOccurs="0" maxOccurs="unbounded" type="ValueSetInclude">
      <xs:annotation>
        <xs:documentation>仅包括指定编码系统中的一部分编码值</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="exclude" minOccurs="0" maxOccurs="unbounded" type="ValueSetInclude">
      <xs:annotation>
        <xs:documentation>包含一个编码系统，但是指定的编码值排除在外</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ValueSetInclude">
  <xs:annotation>
    <xs:documentation>值集包含</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="system" type="OID">
      <xs:annotation>
        <xs:documentation>指定的编码系统 </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="code" type="Code" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>仅有这些编码加入值集</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

#### A.4 文档共享数据类型

文档共享相关数据类型定义如下：

文件名：RHIN-BaseDocument.xsd
<pre> &lt;?xml version="1.0" encoding="utf-8" ?&gt; &lt;!-- Ver.0.12.0/2018-06-12 Dejun Hsu--&gt; &lt;xs:schema xmlns="http://www.chiss.org.cn/rhin/2015" elementFormDefault="qualified"   targetNamespace="http://www.chiss.org.cn/rhin/2015" xmlns:xs="http://www.w3.org/2001/XMLSchema"&gt;   &lt;xs:include schemaLocation="RHIN-BaseObject.xsd"/&gt;   &lt;xs:complexType name="SubmissionSet"&gt;     &lt;xs:annotation&gt; </pre>

```

    <xs:documentation>提交集元数据消息体</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RegObject">
      <xs:sequence>
        <xs:element name="uniqueId" type="OID">
          <xs:annotation>
            <xs:documentation>提交集的全球唯一标识符,该标识符由文档源提供
</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="submissionTime" type="DateTime">
          <xs:annotation>
            <xs:documentation>文档集提交时间</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="title" type="String">
          <xs:annotation>
            <xs:documentation>文档集标题</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="patientId" type="Identifier">
          <xs:annotation>
            <xs:documentation>患者本区域的唯一标识符</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="unbounded" name="author" type="Author" minOccurs="0">
          <xs:annotation>
            <xs:documentation>作者信息,可以是人也可以是机器</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="intendedRecipient"
          type="Reference">
          <xs:annotation>
            <xs:documentation>预期接收者,对该提交集接收者进行的约束限制
</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="contentTypeCode" type="CodeableConcept"/>
        <xs:element minOccurs="0" name="sourceId" type="OID">
          <xs:annotation>
            <xs:documentation>文档集提交源</xs:documentation>
          </xs:annotation>

```

```

</xs:element>
<xs:element minOccurs="0" name="homeCommunityId" type="URI">
  <xs:annotation>
    <xs:documentation>区域的唯一标识符</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="comments" type="String">
  <xs:annotation>
    <xs:documentation>注释</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:choice maxOccurs="unbounded" minOccurs="0">
  <xs:element minOccurs="0" name="folder" type="Folder">
    <xs:annotation>
      <xs:documentation>提交集中文件夹:
可以指定哪些文档需要放置在文档库中,或者
单纯的在文档库中创建空文件夹,以备以后使用</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="0" name="documentEntry" type="DocumentEntry">
    <xs:annotation>
      <xs:documentation>提交集中的文档</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="0" name="association" type="Association">
    <xs:annotation>
      <xs:documentation>记录提交集中文档与文档\文档与文件夹之间的关联关系:
包括本提交集中的文档 (Original)
或者文档库已存在文档 (Reference) </xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:choice>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Folder">
  <xs:annotation>
    <xs:documentation>文件夹元数据消息体</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RegObject">
      <xs:sequence>

```

```

    <xs:element name="uniqueId" type="OID">
      <xs:annotation>
        <xs:documentation>文件夹的唯一标识符,该标识符由文档源提供</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="title" type="String">
      <xs:annotation>
        <xs:documentation>文件夹的标题</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="lastUpdateTime" type="DateTime">
      <xs:annotation>
        <xs:documentation>文件夹的最后更新时间</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="patientId" type="Identifier">
      <xs:annotation>
        <xs:documentation>患者本区域唯一标识符
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="typeCode"
      type="CodeableConcept"/>
    <xs:element minOccurs="0" name="homeCommunityId" type="OID">
      <xs:annotation>
        <xs:documentation>区域的唯一标识符</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="comments" type="String">
      <xs:annotation>
        <xs:documentation>注释</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="DocumentEntry">
  <xs:annotation>
    <xs:documentation>文档元数据.消息体</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RegObject">

```

```

<xs:sequence>
  <xs:element minOccurs="0" name="languageCode" type="Code">
    <xs:annotation>
      <xs:documentation>语言</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="0" name="title" type="String">
    <xs:annotation>
      <xs:documentation>文档标题</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="0" name="confidentialityCode" type="Coding">
    <xs:annotation>
      <xs:documentation>隐私性</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="creationTime" type="DateTime">
    <xs:annotation>
      <xs:documentation>文档的创建时间</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="0" name="mimeType" type="String">
    <xs:annotation>
      <xs:documentation>文件的MIME格式</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="0" name="formatCode" type="Code">
    <xs:annotation>
      <xs:documentation>文件格式</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="0" name="objectType" type="ObjectType">
    <xs:annotation>
      <xs:documentation>文档元数据的对象类型:
        静态的-元数据指向的文档是存在的
        按需的-元数据指向的文档当前在文档库不存在,但是能够按需动态生成出来</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element minOccurs="0" name="size" type="Integer">
    <xs:annotation>
      <xs:documentation>文件的大小</xs:documentation>
    </xs:annotation>
  </xs:element>

```

```

<xs:element minOccurs="0" name="hash" type="String">
  <xs:annotation>
    <xs:documentation>文件的哈希值</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="serviceStartTime" type="DateTime">
  <xs:annotation>
    <xs:documentation>文档指代的就诊活动的开始时间</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="serviceStopTime" type="DateTime">
  <xs:annotation>
    <xs:documentation>文档指代的就诊活动的结束时间</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="patientId" type="Identifier">
  <xs:annotation>
    <xs:documentation>患者在本区域的唯一标识符</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="sourcePatientInfo" type="Patient">
  <xs:annotation>
    <xs:documentation>患者在文档传输源系统中的信息</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="author" type="Author">
  <xs:annotation>
    <xs:documentation>作者</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="legalAuthenticator" type="Practitioner">
  <xs:annotation>
    <xs:documentation>审核者</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="healthcareFacilityTypeCode"
  type="CodeableConcept">
  <xs:annotation>
    <xs:documentation>文档指代就诊活动所在机构的类型</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="practiceSettingCode" type="CodeableConcept">
  <xs:annotation>

```

```

        <xs:documentation>文档指代就诊活动所在机构的专业分类</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="typeCode" type="CodeableConcept">
    <xs:annotation>
        <xs:documentation>就诊活动的分类（细粒度）</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="eventCode"
    type="CodeableConcept">
    <xs:annotation>
        <xs:documentation>患者就诊活动中事件编码列表</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="uniqueId" type="OID">
    <xs:annotation>
        <xs:documentation>文档的唯一标识符</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="repositoryUniqueId" type="OID">
    <xs:annotation>
        <xs:documentation>文档存储库的唯一标识符</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="homeCommunityId" type="OID">
    <xs:annotation>
        <xs:documentation>区域的唯一标识符</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="comments" type="String">
    <xs:annotation>
        <xs:documentation>注释</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Association">
    <xs:annotation>
        <xs:documentation>关联关系</xs:documentation>
    </xs:annotation>
</xs:complexType>

```



```

<xs:extension base="RegObject">
  <xs:sequence>
    <xs:element name="typeCode" type="AssociationType">
      <xs:annotation>
        <xs:documentation>关联类型</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="sourceObject" type="AssociationObject">
      <xs:annotation>
        <xs:documentation>关联源 </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="targetObject" type="AssociationObject">
      <xs:annotation>
        <xs:documentation>关联目标 </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="DocumentReference">
  <xs:sequence>
    <xs:element name="documentUniqueId" type="OID">
      <xs:annotation>
        <xs:documentation>文档唯一标识符</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="repositoryUniqueId" type="OID">
      <xs:annotation>
        <xs:documentation>文档存储库唯一标识符</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element minOccurs="0" name="homeCommunityId" type="OID">
      <xs:annotation>
        <xs:documentation>区域唯一标识符</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="StoredDocument">
  <xs:annotation>
    <xs:documentation>已存的文档,id为OID</xs:documentation>
  </xs:annotation>

```

```

</xs:annotation>
<xs:complexContent>
  <xs:extension base="DocumentReference">
    <xs:sequence>
      <xs:element name="mimeType" type="String">
        <xs:annotation>
          <xs:documentation>内容的类型</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="content" type="base64BinaryPrimitive">
        <xs:annotation>
          <xs:documentation>文档内容</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Document">
  <xs:annotation>
    <xs:documentation>用于提交文档时,id为URI,可以为暂代符或者UUID
与DocumentEntry中的id一致</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="id" type="URI">
      <xs:annotation>
        <xs:documentation>内容的类型</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="content" type="base64BinaryPrimitive">
      <xs:annotation>
        <xs:documentation>文档内容</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Author">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="authorRole" type="CodeableConcept">
      <xs:annotation>
        <xs:documentation>作者角色</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

<xs:element maxOccurs="unbounded" minOccurs="0" name="authorSpecialty"
  type="CodeableConcept">
  <xs:annotation>
    <xs:documentation>作者专业</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="authorPerson" type="Reference">
  <xs:annotation>
    <xs:documentation>作者</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="authorTelecommunication"
  type="Telecom">
  <xs:annotation>
    <xs:documentation>作者联系方式</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element maxOccurs="unbounded" minOccurs="0" name="authorInstitution"
  type="Reference">
  <xs:annotation>
    <xs:documentation>作者所在机构</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="AssociationObject">
  <xs:annotation>
    <xs:documentation>标识关联的对象</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="id" type="URI"/>
    <xs:element name="status" type="AssociationStatus"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="associationStatusPrimitive">
  <xs:annotation>
    <xs:documentation>
      Original      - 原始
      Reference     - 引用
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="stringPrimitive">
    <xs:enumeration value="original"/>
  </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="reference"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="AssociationStatus">
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="associationStatusPrimitive"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="associationTypePrimitive">
    <xs:annotation>
        <xs:documentation>
            HasMember    - 拥有
            RPLC         - 替换
            APND         - 新增
            XFRM         - 转换
            XFRM_RPLC    - 转换并替换
            Signs        - 签名
            IsSnapshotOf - 快照
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="urnPrimitive">
        <xs:enumeration value="urn:rhin:AssociationType:HasMember"/>
        <xs:enumeration value="urn:rhin:AssociationType:RPLC"/>
        <xs:enumeration value="urn:rhin:AssociationType:XFRM"/>
        <xs:enumeration value="urn:rhin:AssociationType:APND"/>
        <xs:enumeration value="urn:rhin:AssociationType:XFRM_RPLC"/>
        <xs:enumeration value="urn:rhin:AssociationType:signs"/>
        <xs:enumeration value="urn:rhin:AssociationType:IsSnapshotOf"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="AssociationType">
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="associationTypePrimitive"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:simpleType name="objectType">
    <xs:annotation>
        <xs:documentation>
            Stable      - 静态

```

```
        OnDemand - 按需
    </xs:documentation>
</xs:annotation>
<xs:restriction base="urnPrimitive">
    <xs:enumeration value="urn:rhin:ObjectType:Stable"/>
    <xs:enumeration value="urn:rhin:ObjectType:OnDemand"/>
</xs:restriction>
</xs:simpleType>
<xs:complexType name="ObjectType">
    <xs:complexContent>
        <xs:extension base="Any">
            <xs:attribute name="value" type="objectType"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:schema>
```

附 录 B  
(规范性)  
基本数据类型值域代码表

### B.1 空值原因代码

空值原因代码见表B.1。

表 B.1 空值原因代码 (CV\_NullFlavor)

代码	描述	英文描述
UNK	该值未知	Unknown
NAV	该值当前不可用	
NASK	未询问	Did not ask
ASKU	已询问, 但未获值。	Asked, but not valued
NP	该值不填充	Not Populated
NA	该值不适用于消息语境	Not Applicable
NI	无法取值	
MSK	隐藏值	
OTH	其它	Other

### B.2 时长计量单位

时间长度计量单位代码见表B.2。

表 B.2 时长计量单位代码 (CV\_DurationUnit)

代码	描述	英文描述
yr	年	Year
nth	月	Month
wk	周	Week
d	日	Day
h	时	hour
min	分	minute
s	秒	second

### B.3 人名用途代码

人名用途代码见表B.3。

表 B.3 人名用途代码 (CV\_NameUse)

代码	描述	英文描述
Usual	常规	Usual
Official	办公	Official
Temp	临时	Temp
Nickname	昵称	Nickname
Anonymous	匿名	Anonymous
Old	曾用名	Old
Maiden	乳名	Maiden

## B.4 地址用途代码

地址用途代码见B.4。

表 B.4 地址用途代码 (CV\_AddressUse)

代码	描述	英文描述
Home	居住地址	communication address at home
Work	工作地址	An office address.
Temp	临时地址	A temporary address.
Old	旧地址	This address is no longer in use

## B.5 地址类别代码

地址类别代码见B.5。

表 B.5 地址类别代码 (CV\_AddressType)

代码	描述	英文描述
Postal	投递地址	Postal address
Physical	物理位置	Physical address
Both	投递与物理位置通用	Both postal & physical address

## B.6 通讯方式代码

通讯方式代码见表B.6。

表 B.6 通讯方式代码 (CV\_TelecomMode)

代码	描述	英文描述
Telephone	电话	Telephone
Cellphone	手机	Cellphone
Email	电子邮箱	Email
QQ	QQ	QQ

表 B.6 (续)

代码	描述	英文描述
Wechat	微信	Wechat
Other	其它	Other

## B.7 对象状态代码

对象状态代码见表B.7。

表 B.7 对象状态代码 (CV\_ObjectStatus)

代码	描述	英文描述
New	新建, 未激活状态	Newly created
Active	已激活, 有效状态	Active
Nullified	已注销, 无效状态	Nullified

## B.8 文档关联状态代码

文档关联状态代码见表B.8。

表 B.8 文档关联状态代码 (CV\_AssociationStatus)

代码	描述	英文描述
Original	原始	Original
Reference	引用	Reference

## B.9 文档关联类型代码

文档关联类型代码见表B.9。

表 B.9 文档关联类型代码 (CV\_AssociationType)

代码	描述	英文描述
urn:rhin:AssociationType:APND	新增	Apend
urn:rhin:AssociationType:RPLC	替换	Replace
urn:rhin:AssociationType:XFRM	转换	Transform
urn:rhin:AssociationType:XFRM_RPLC	转换并替换	Replace
urn:rhin:AssociationType:Sign	签名	Signature
urn:rhin:AssociationType:IsSnapshotOf	快照	Is Snapshot Of

## B.10 对象类型代码

对象类型代码见表B.10。



表 B.10 文档关联类型代码 (CV\_ObjectType)

代码	描述	英文描述
urn:rhin:ObjectType:Stable	静态	Stable
urn:rhin:ObjectType:OnDemand	按需	OnDemand

## B.11 错误代码

错误代码见表B.11。

表 B.11 错误代码 (CV\_FaultCode)

代码	描述	英文描述
VersionMismatch	SOAP Envelope 元素的无效命名空间被发现	Version Mismatch
MustUnderstand	Header 中一个标记为 mustUnderstand 的元素无法被正确处理。	Must Understand
DataEncodingUnknown	数据编码不被支持	Data Encoding Unknown
Sender	由于消息内容的不正确而造成的无法被服务端处理。	Sender
Receiver	由于消息的处理而不是消息的内容造成消息无法被服务端处理。	Receiver

## 附录 C (资料性) 示例

### C.1 故障消息示例

故障消息示例见示例1

示例1：故障消息示例

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-soapelope"
  xmlns:rhin="http://www.chiss.org.cn/rhin/is/2015">
  <soap:Header/>
  <soap:Body>
    <soap:Fault>
      <soap:Code>
        <soap:Value>soap:Receiver</soap:Value>
        <soap:SubCode>
          <soap:Value>rhin:TimeOutFault</soap:Value>
        </soap:SubCode>
      </soap:Code>
      <soap:Reason>
        <soap:Text>服务访问超时</soap:Text>
      </soap:Reason>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

### C.2 服务寻址消息示例

服务寻址消息示例见示例2

示例2：服务寻址消息示例

```
<wsa:MessageID>uuid:aaaabbbb-cccc-dddd-eeee-ffffffff</wsa:MessageID>
<wsa: From>http://app.example/client1</wsa: From>
<wsa: To S:mustUnderstand="1">mailto:joe@fabrikam123.example</wsa: To>
<wsa: Action>http://fabrikam123.example/mail/Delete</wsa: Action>
</S:Header>
```

### C.3 安全认证消息示例

安全认证消息示例见示例3

## 示例3: 安全认证消息示例

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope">
  <SOAP-ENV:Header>
    <!-- 安全认证节点 -->
    <wsse:Security SOAP-ENV:mustUnderstand="true"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <!-- 时间戳, 注意该时间为世界时, 即格林尼治时 -->
      <wsu:Timestamp wsu:Id="TS-74063B34A4513D474215258757494131">
        <wsu:Created>2018-05-09T14:22:29.239Z</wsu:Created>
        <wsu:Expires>2018-05-09T14:27:29.239Z</wsu:Expires>
      </wsu:Timestamp>
      <!-- 安全令牌, 公钥证书, X509, 二进制形式 -->
      <wsse:BinarySecurityToken
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
wsu:Id="X509-74063B34A4513D474215258757494322">
MIIDYzCCAkugAwIBAgIEFfw0TANBgqhkiG9w0BAQsFADBiMQswCQYDVQQGEwJDTjELMAkGA1UECBMCMox
CzAJBgNVBACt...
gPsAk6XA==</wsse:BinarySecurityToken>
      <!-- 数字签名 -->
      <ds:Signature Id="SIG-74063B34A4513D474215258757494516" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <!-- XML 规范化标准, 本规范系列采用 Exclusive XML Canonicalization -->
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <ec:InclusiveNamespaces PrefixList="SOAP-ENV" xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </ds:CanonicalizationMethod>
          <!-- XML 签名算法, 本示例采用的是 RSA-SHA256 -->
          <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
          <!-- 签名内容, 通过一系列的 Reference 指定要对哪些内容进行签名,
          本示例签名内容包括时间戳、正文以及数字证书 -->
          <ds:Reference URI="#TS-74063B34A4513D474215258757494131">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <ec:InclusiveNamespaces PrefixList="wsse SOAP-ENV" xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" />
              </ds:Transform>
            </ds:Transforms>
            <!-- XML 摘要算法, 本示例采用的是 SHA256 -->
            <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmenc#sha256" />
            <ds:DigestValue>W1K7ZEs4FZUuBrLIBLRDmB1v81ur2d3mR372m6ocZCw=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#id-74063B34A4513D474215258757494465">

```

```

<ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
    <ec:InclusiveNamespaces PrefixList="" xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" />
  </ds:Transform>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
<ds:DigestValue>6zqD/4ayuCV58PGsZ8s7kdMMIqVGt7aO4kIE+ckeUzA=</ds:DigestValue>
</ds:Reference>
  <ds:Reference URI="#X509-74063B34A4513D474215258757494322">
    <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
        <ec:InclusiveNamespaces PrefixList="SOAP-ENV" xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
    <ds:DigestValue>oR711NX4m2tCrRoluJ/t4IvejFPG/dQqHyJSUChIWUk=</ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
  <!-- 正文的最终签名值 -->
  <ds:Signature Value>VKwC6AjL6gSGN078sncOqV+IfZl9v6Ggli2ZJCRuHtYh0YW6DcUw7kP8HRtsYif0sFb/5hZUIUUCc6r
oFdB0IEJV7+P+f6b1fdAWttaQYfDppY111dUXbCuBVYFVBgJ9JU7v2rjOwjsVhdDnMkI4kH7qJ/eFnFAIOzHenPyvzTxwv3d
b+3SDpW5iXnxKdPPfJCwmoTELksjq/7YjbQkSuj3NDKnF4odKW/yMDXNbpYef9s+uAgrwPs6uqV6XFKN2KZR9I4Kd7+0q
atYmdiaRKcdkVgtntuK9gPWu7nkKxxmVuMY6JjUc/2ke++xFSl1Gii+n9jQg0DQUpCDsJm5g==</ds:Signature Value>
  <!-- 签名密钥，引用指向前面的 X509 数字证书 -->
  <ds:KeyInfo Id="KI-74063B34A4513D474215258757494363">
    <wsse:SecurityTokenReference wsu:Id="STR-74063B34A4513D474215258757494384">
      <wsse:Reference URI="#X509-74063B34A4513D474215258757494322"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
    </wsse:SecurityTokenReference>
  </ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</SOAP-ENV:Header>
  <!-- 消息正文 -->
  <SOAP-ENV:Body wsu:Id="id-74063B34A4513D474215258757494465"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <sayHello xmlns="http://jeffhanson.com/services/helloworld">
      <value xmlns="">Hello world!</value>
    </sayHello>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## 参 考 文 献

- [1] IHE International, Inc.IHE IT Infrastructure Technical Framework, Volume 1 (ITI TF-1): Integration Profiles,2015.
  - [2] IHE International, Inc.IHE IT Infrastructure Technical Framework, Volume 2a (ITI TF-2a): Transactions Part A,2015.
  - [3] IHE International, Inc.IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B,2015.
  - [4] IHE International, Inc.IHE Technical Framework Supplement - Document Metadata Subscription (DSUB),2010
  - [5] W3C.Web Services Description Language 1.1(WSDL 1.1)
  - [6] W3C.Simple Object Access Protocol 1.2(SOAP 1.2)
  - [7] W3C.Web Services Addressing (WS-Addressing),10 August 2004
  - [8] W3C/OASIS Standard.Web Services Base Notification 1.3(WS-Base Notification),1 October 2006
  - [9] W3C/OASIS Standard. Web Services Topics 1.3(WS-Topics),1 October 2006
  - [10] W3C.Web Services Security 1.0(WS-Security), April 5, 2002
  - [11] W3C.XML Schema Definitions 1.1(XSD)
  - [12] W3C.XML Path Language 2.0(X Path 2.0)
-