ICS 35.100.70
L 79

中华人民共和国国家标准

GB/T 16264.4—2008/ISO/IEC 9594-4:2005
代替 GB/T 16264.4—1996

信息技术 开放系统互连 目录 第4部分：分布式操作规程

Information technology—Open Systems Interconnection—The Directory—Part 4: Procedures for distributed operation

(ISO/IEC 9594-1:2005 Information technology—Open Systems Interconnection—The Directory:Procedures for distributed operation,IDT)

2008-08-06 发布　　2009-01-01 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

目　次

前言 …… Ⅰ
引言 …… Ⅱ
第一篇:综述 …… 1
1　范围 …… 1
2　规范性引用文件 …… 1
3　术语和定义 …… 2
4　缩略语 …… 6
5　约定 …… 6
第二篇:概述 …… 6
6　概述 …… 6
第三篇:分布式目录模型 …… 7
7　分布式目录系统模型 …… 7
8　DSA 交互模型 …… 8
第四篇:DSA 抽象服务 …… 11
9　DSA 抽象服务概述 …… 11
10　信息类型 …… 11
11　绑定和解绑定 …… 20
12　链接操作 …… 21
13　链接差错 …… 22
第五篇:分布式规程 …… 23
14　概述 …… 23
15　分布式目录行为 …… 24
16　操作调度程序 …… 29
17　请求有效性验证规程 …… 35
18　名(称)解析规程 …… 38
19　操作赋值(evaluation) …… 47
20　连续引用规程 …… 67
21　结果合并规程 …… 79
22　分布式鉴别规程 …… 81
第六篇:知识管理 …… 82
23　知识管理概述 …… 82
24　分等级操作绑定 …… 85
25　非特定分等级操作绑定 …… 93
附录 A(规范性附录)　分布式操作的 ASN.1 定义 …… 97
附录 B(资料性附录)　分布式名(称)解析的示例 …… 102
附录 C(资料性附录)　鉴别的分布式使用 …… 104
附录 D(规范性附录)　分等级和非特定分等级操作绑定类型的规范 …… 110
附录 E(资料性附录)　知识维护示例 …… 113

前　言

GB/T 16264 在《信息技术　开放系统互连　目录》总标题下，包括以下 10 个部分：

——第 1 部分：概念、模型和服务的概述；

——第 2 部分：模型；

——第 3 部分：抽象服务定义；

——第 4 部分：分布式操作规程；

——第 5 部分：协议规范；

——第 6 部分：选定的属性类型；

——第 7 部分：选定的客体类；

——第 8 部分：公钥和属性证书框架；

——第 9 部分：复制（待发布）；

——第 10 部分：公用目录管理机构的系统管理用法（待发布）。

本部分是 GB/T 16264 的第 4 部分。

本部分等同采用国际标准 ISO/IEC 9594-4:2005《信息技术　开放系统互连　目录　分布式操作规程》，仅有编辑性修改。

本部分代替 GB/T 16264.4—1996。

本部分与 GB/T 16264.4—1996 的差异在于：

——增加知识管理；

——扩展了各章条内容。

本部分的附录 A 和附录 D 是规范性附录，附录 B、附录 C 和附录 E 是资料性附录。

本部分由中华人民共和国信息产业部提出。

本部分由全国信息技术标准化技术委员会归口。

本部分起草单位：中国电子技术标准化研究所。

本部分主要起草人：徐冬梅、张翠、冯惠、胡顺、刘文治。

本部分于 1996 年首次发布，本次为第一次修订。

引　言

GB/T 16264 的本部分连同本标准其他部分是为方便信息处理系统之间的互连以提供目录服务而制定的。所有这些系统的集合，连同它们所拥有的目录信息可被视为一个整体，被称为“目录”。目录所拥有的信息，总称为目录信息库(DIB)，典型地被用于方便客体之间的通信、与客体的通信或有关客体的通信等，这些客体如应用实体、个人、终端和分布列表等。

目录在开放系统互连中扮演了重要角色，其目标是，在它们自身的互连标准之外做最少的技术约定的情况下，允许下述各种信息处理系统之间的互连：

——来自不同生产厂商；

——具有不同的管理；

——具有不同的复杂程度，以及

——有不同的年代。

本部分规定了目录各组件进行交互工作所遵循的规程，以便为用户提供一致的服务。

本部分提供了一些基础框架，在此框架基础上，其他标准化组织和业界论坛可以定义工业配置集。在这些框架中定义为可选的许多特性，可通过配置集的说明，在某种环境下作为必选特性来使用。ISO/IEC 9594 的第 5 版是原有国际标准第 4 版的修订和增强，但不是替代。在系统实现时仍可以声明为符合第 4 版。然而，在某些方面，将不再支持第 4 版(即不再消除一些报告上来的差错)。建议在系统实现时尽快符合第 5 版。

第 5 版详细定义了目录协议的第 1 版和第 2 版。

第 1 版和第 2 版仅定义了协议第 1 版。本版本(第 5 版)中定义的许多服务和协议被设计为可运行在第 1 版下。然而，一些增强的服务和协议，如署名差错，只有包含在操作中的所有的目录条目都协商支持协议第 2 版时才可运行。无论协商的是哪一版，第 5 版中所定义的服务之间的差异和协议之间的差异，除了那些特别分配给第 2 版的外，都可以使用 GB/T 16264.5—2008 中定义的扩展规则调节。

本部分使用术语“第 1 版系统”来指遵循国际标准第 1 版的所有系统，即 ISO/IEC 9594:1990 版本；本部分使用术语“第 2 版系统”来指遵循国际标准第 2 版本的所有系统，即 ISO/IEC 9594:1995 版本；本部分使用术语“第 3 版系统”来指遵循国际标准第 3 版的所有系统，即 ISO/IEC 9594:1998 版本；本部分使用术语“第 4 版系统”来指遵循国际标准第 4 版的所有系统，即 ISO/IEC 9594:2001 版本的第 1 部分到第 10 部分；本部分使用术语“第 5 版系统”来指遵循国际标准第 5 版的所有系统，即 ISO/IEC 9594:2005 版本。

GB/T 16264—1996 是参照 ISO/IEC 9594:1990 而制定的。我国没有制定与国际标准第 2 版、第 3 版、第 4 版对应的国家标准。本部分提到的版本号是指国际标准的版本号。

附录 A 是规范性附录，提供了目录分布式操作的 ASN.1 模块定义。

附录 B 是资料性附录，描述了分布式名(称)解析的一个示例。

附录 C 是资料性附录，描述了分布式操作环境中的鉴别。

附录 D 是规范性附录，提供了本目录规范中引入的 ASN.1 信息客体类的定义。

附录 E 是资料性附录，举例说明了知识的维护。

信息技术 开放系统互连 目录 第4部分：分布式操作规程

第一篇：综 述

1 范围

GB/T 16264的本部分规定涉及分布式目录应用的DSA行为。对于跨越许多DSA的广域DIB，已经设计了允许的行为以确保提供一致的服务。

虽然目录可以建立在某种通用的数据库系统之上，但它本身并不属于这样一种通用的数据库系统。目录是在假定查询操作远比更新操作频繁的情况下建立的。

2 规范性引用文件

下列文件中的条款通过GB/T 16264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 9387.1—1998 信息技术 开放系统互连 基本参考模型 第1部分：基本模型(idt ISO/IEC 7498-1:1994)

GB/T 16262.1—2006 信息技术 抽象语法记法一(ASN.1) 第1部分：基本记法规范(ISO/IEC 8824-1:2002,IDT)

GB/T 16262.2—2006 信息技术 抽象语法记法一(ASN.1) 第2部分：信息客体规范(ISO/IEC 8824-2:2002,IDT)

GB/T 16262.3—2006 信息技术 抽象语法记法一(ASN.1) 第3部分：约束规范(ISO/IEC 8824-3:2002,IDT)

GB/T 16262.4—2006 信息技术 抽象语法记法一(ASN.1) 第4部分：ASN.1规范的参数化(ISO/IEC 8824-4:2002,IDT)

GB/T 16264.1—2008 信息技术 开放系统互连 目录 第1部分：概念、模型和服务的概述(ISO/IEC 9594-1:2005,IDT)

GB/T 16264.2—2008 信息技术 开放系统互连 目录 第2部分：模型(ISO/IEC 9594-2:2005,IDT)

GB/T 16264.3—2008 信息技术 开放系统互连 目录 第3部分：抽象服务定义(ISO/IEC 9594-3:2005,IDT)

GB/T 16264.5—2008 信息技术 开放系统互连 目录 第5部分：协议规范(ISO/IEC 9594-5:2005,IDT)

GB/T 16264.6—2008 信息技术 开放系统互连 目录 第6部分：选定的属性类型(ISO/IEC 9594-6:2005,IDT)

GB/T 16264.7—2008 信息技术 开放系统互连 目录 第7部分：选定的客体类(ISO/IEC 9594-7:2005,IDT)

ISO/IEC 9594-8:2005 信息技术 开放系统互连 目录：公钥和属性证书框架

ISO/IEC 9594-9:2005 信息技术 开放系统互连 目录：复制

ISO/IEC 9594-10:2005 信息技术 开放系统互连 目录：公用目录管理机构的系统管理用法

IETF RFC 2251:1997 轻量级目录访问协议(v3)

IETF RFC 3377:2002 轻量级目录访问协议(v3):技术规范

3 术语和定义

下列术语和定义适用于 GB/T 16264 的本部分。

3.1 通信模型定义

本部分使用 GB/T 16264.5—2008 中定义的下列术语:

应用实体名称 *application-entity-title*;

3.2 基本目录定义

本部分使用 GB/T 16264.1—2008 中定义的下列术语:

a) 目录 *(the) Directory*

b) 目录信息库 *Directory Information Base*

3.3 目录模型定义

本部分使用 GB/T 16264.2—2008 中定义的下列术语:

a) 访问点 *access point*;

b) 别名 *alias*;

c) 可辨别名 *distinguished name*;

d) 目录信息树(*DIT*) *Directory Information Tree(DIT)*;

e) 目录系统代理(*DSA*) *Directory System Agent (DSA)*;

f) 目录用户代理(*DUA*) *Directory User Agent (DUA)*;

g) 相关可辨别名 *relative distinguished name*。

3.4 DSA 信息模型定义

本部分使用 GB/T 16264.2—2008 中定义的下列术语:

a) 种类 *category*;

b) 公共可用的 *commonly usable*;

c) 上下文前缀 *context prefix*;

d) 交叉引用 *cross reference*;

e) *DIB* 片段 *DIB fragment*;

f) *DSA* 信息树 *DSA information tree*;

g) *DSA* 特定条目(*DSE*) *DSA- Specific Entry (DSE)*;

h) *DSE* 类型 *DSE type*;

i) 直接上级引用 *immediate superior reference*;

j) 知识信息 *knowledge information*;

k) 知识引用种类 *knowledge reference category*;

l) 知识引用类型 *knowledge reference type*;

m) 命名上下文 *naming context*;

n) 非特定知识 *non-specific knowledge*;

o) 非特定下级引用 *non-specific subordinate reference*;

p) 操作属性 *operational attribute*;

q) 引用路径 *reference path*;

r) 特定知识 *specific knowledge*;

s) 下级引用 *subordinate reference*;

t) 上级引用 *superior reference*。

3.5 **抽象服务定义**

本部分使用 GB/T 16264.3—2008 中定义的下列术语：

a) 流结果 *streamed result*；

3.6 **目录复制定义**

本部分使用 ISO/IEC 9594-9:2005 中定义的下列术语：

a) 属性完备性 *attribute completeness*；

b) 影像操作绑定 *shadowing operational binding*；

c) 下级完备性 *subordinate completeness*；

d) 复制单元 *unit of replication*。

3.7 **分布式操作定义**

下列术语和定义适用于本部分。

3.7.1

基本客体 base object

一个客体或别名条目，是发起者所发起的某个操作的目标。

3.7.2

绑定的 DSA bound DSA

一个 DSA，发起请求的 DUA 通过与该 DSA 执行一个绑定操作而与之绑定起来。

3.7.3

绑定的 DSA 的分页结果 bound-DSA paged results

分页完全由 DUA 所绑定的 DSA 来执行。

注：这是遵循第 5 版之前的系统所支持的唯一一种分页模式。

3.7.4

链接 chaining

单链接或多链接的通用术语。

3.7.5

上下文前缀信息 context prefix information

上级 DSA 在一个 RHOB 中向下级 DSA 所提供的关于下级上下文前缀的上级 DIT 顶点的操作信息和用户信息。

3.7.6

分布式名(称)解析 distributed name resolution

在多于一个的 DSA 中执行名(称)解析的过程。

3.7.7

DSP 分页结果 DSP paged results

当执行的 DSA 与绑定的 DSA 不相同时且由初始执行者完成的分页结果，依据 DSP 协议提供。

3.7.8

差错 error

由执行者向请求者发送的信息，信息中携带了一个对之前接收到的请求的否定结果。

3.7.9

硬差错 hard error

一个明确的差错，该差错指示如果没有外部的干预，则操作目前不能被执行。

3.7.10

分等级操作绑定 hierarchical operational binding;HOB

指两个拥有命名上下文的主 DSA 之间的关系，其中一个是另一个的直接下级，在该关系中，上级 DSA 拥有一个指向下级 DSA 的下级引用。

3.7.11

初始执行者 initial performer

开始执行某个操作的第一个 DSA，即进入操作赋值阶段的第一个 DSA。

3.7.12

修改操作 modification operations

指目录修改操作，即修改条目、增加条目、移除条目和修改 DN 等。

3.7.13

多链接 multi-chaining

一种交互模式，在该模式中，自身执行某个请求的 DSA 向其他 DSA 的集合发出多个请求，或者是并列的，或者是顺序的。

3.7.14

多条目查询操作 multiple entry interrogation operations

指目录搜索操作，即列表和搜索。

3.7.15

名(称)解析 name resolution

定位某个条目的过程，该过程是通过对声称名(称)的每个 RDN 与 DIT 的顶点进行顺序匹配而完成的。

3.7.16

非特定分等级操作绑定 non-specific hierarchical operational binding；NHOB

指两个拥有命名上下文的主 DSA 之间的关系，其中一个是另一个的直接下级，在该关系中，上级 DSA 拥有一个指向下级 DSA 的非特定下级引用。

3.7.17

分解 NSSR decomposition NSSR

将非特定知识引用分解为多个子请求以便于其他 DSA 能够继续执行；这些子请求可以被执行该分解的 DSA 链接到其他 DSA；或者可以将标识了其他 DSA 的一个连续引用返回给请求者，由请求者来继续执行；或者执行分解的 DSA 可以继续执行某些子请求，而留下其他子请求交给请求者来继续执行。

3.7.18

操作进展 operation progress

一系列的值，指示了名(称)解析所完成的程度。

3.7.19

发起者 originator

发起一个特定的(分布式)操作的 DUA。

3.7.20

分页 paging

以一页或多页的分段方式返回的搜索或列表操作的结果，每页都由有限数量的条目组成。

3.7.21

执行者 *performer*

接收了某个请求的 DSA(即将执行某个操作)。

注：执行者也是初始执行者，除非对于可以包括多个 DSA 进行赋值的操作。

3.7.22

规程 procedure

关于 DSA 如何将给定的输入变元集及其 DSA 信息树映射为一个结果的一个(非正式)规范。

注：输入变元和结果可以对应为从一个被请求的操作中接收到的信息和在一个答复中发送的信息，或者它们可以表示根据一个被请求的操作而对答复进行计算的中间阶段。在 14.2，前面一种输入变元和结果类型被称为是外部的。

3.7.23

相关的分等级操作绑定 relevant hierarchical operational binding;RHOB

或者是指一个 HOB,或者是指一个 NHOB,依赖于上下文。

3.7.24

转向推荐 referral

自己不能执行某个操作的 DSA 返回的一种结果,标识了一个或多个可以执行此操作的其他 DSA。

3.7.25

答复 reply

一个结果或一个差错。

3.7.26

请求 request

由一个操作代码和相关变元所组成的信息,表示从请求者向执行者所发起的一个目录操作。

3.7.27

请求分解 request decomposition

将一个请求分解为多个子请求以便于其他 DSA 能够继续执行;这些子请求可以被执行分解的 DSA 链接到其他 DSA;或者可以将标识了其他 DSA 的连续引用返回给请求者,由请求者来继续执行;或者执行分解的 DSA 可以继续执行某些子请求,而留下其他子请求交给请求者来继续执行。

3.7.28

请求者 requester

发送一个请求以便执行(即调用)某个操作的一个 DUA 或 DSA。

3.7.29

单条目查询操作 single entry interrogation operations

指目录阅读操作,即阅读和比较操作。

3.7.30

软差错 soft error

一个差错,可以是瞬时的,也可以指示了一个局部的问题,在这种情况下,使用一个不同的知识引用或访问点可获得一个结果或一个硬差错。

3.7.31

下级 DSA subordinate DSA

共享一个 HOB 或一个 NHOB 的两个 DSA 中,其中拥有下级命名上下文的那个 DSA。

3.7.32

子请求 subrequest

通过请求分解而产生的一个请求。

3.7.33

上级 DSA superior DSA

共享一个 HOB 或一个 NHOB 的两个 DSA 中,其中拥有上级命名上下文的那个 DSA。

3.7.34

上级、下级 DSA superior, subordinate DSA

两个拥有命名上下文的主 DSA,其中一个是另一个的直接下级;这两个 DSA 之间的关系可以通过一个 HOB(或 NHOB)来显式地管理,或者依靠上级 DSA 来隐含存在,该上级 DSA 拥有一个指向下级 DSA 的下级引用(或非特定下级引用)。

3.7.35

目标客体名 target object name

一个条目的名(称),该条目或者是操作在名(称)解析的某个特定阶段所指向的客体,或者是包含在操作赋值中。

3.7.36

单链接 uni-chaining

某个自身不能直接执行操作的 DSA 可选使用的一种交互模式。DSA 的链接是通过调用另一个 DSA 的某个操作,并且将结果再转发给初始请求者来完成的。

4 缩略语

下列缩略语适用于 GB/T 16264 的本部分:

ASN.1	抽象语法标记一
DISP	目录信息影像协议
DMD	目录管理域
DOP	目录操作绑定管理协议
DSE	DSA 特定条目
HOB	分等级操作绑定
NHOB	非特定的分等级操作绑定
NSSR	非特定下级引用
RHOB	相关的分等级操作绑定

5 约定

术语"目录规范(或本目录规范)"指的是 GB/T 16264.4。术语"系列目录规范"指的是GB/T 16264(或者 ISO/IEC 9594)的所有部分。

本目录规范使用术语"第 1 版系统"来指遵循系列目录规范第 1 版的所有系统,即 GB/T 16264—1996 版本。本目录规范使用术语"第 2 版系统"来指遵循系列目录规范第 2 版本的所有系统,即 ISO/IEC 9594:1995 版本。本目录规范使用术语"第 3 版系统"来指遵循系列目录规范第 3 版的所有系统,即 ISO/IEC 9594:1998 版本。本目录规范使用术语"第 4 版系统"来指遵循系列目录规范第 4 版的所有系统,即 ISO/IEC 9594:2001 版本的第 1 部分到第 10 部分。

本目录规范使用术语"第 5 版系统"来指遵循系列目录规范第 5 版的所有系统,即 GB/T 16264—2008 版本的第 1 部分到第 7 部分以及 ISO/IEC 9594-8:2005、ISO/IEC 9594-9:2005 和 ISO/IEC 9594-10:2005。

本目录规范使用粗体字来表示 ASN.1 符号。若在常规文本中要表示 ASN.1 的类型和值时,为了区别于常规文本,使用了粗体字表示。为了表示过程的语义而引用过程名时,为了区别于常规文本,使用了粗体字表示。访问控制许可使用斜体字表示。

第二篇:概 述

6 概述

目录抽象服务允许对 DIB 中的目录信息进行查询、获取和修改。该服务按照 GB/T 16424.3 中规定的抽象目录客体来进行描述。类似的,轻量级目录访问协议(LDAP)允许对 DIB 中的目录信息进行查询、获取和修改。该协议以及它所允许的服务在 RFC 3377 中规定。

必要地,抽象目录客体的规范没有以任何方式指定目录的物理实现:尤其是它没有指定目录系统代理(DSA)的规范,这些 DSA 存储了 DIB 并对 DIB 进行管理,并且通过 DSA 提供服务。另外,它没有考

虑 DIB 是否是分布式的，即 DIB 是包含在一个单独的 DSA 内，还是分布在多个 DSA 内。因此，为了在一个分布式环境中支持抽象服务，需要 DSA 具有其他 DSA 的知识，能够导航到其他 DSA，并且与其他 DSA 进行合作等，这些需求也没有涵盖在服务描述中。

本目录规范对抽象目录客体进行了细化，这种细化通过一个或多个 DSA 客体集来表示，这些 DSA 客体共同构成了分布的目录服务。

另外，本目录规范规定了 DIB 可被分布到一个或多个 DSA 内的允许的方式。对于某种受限情况，即 DIB 包含在一个单独的 DSA 内，这种情况下目录实际上是集中式的；对于 DIB 分布在两个或多个 DSA 内的情况，则规定了知识和导航机制以确保所有的拥有组成条目的 DSA 能够潜在地访问到整个 DIB。

DIB 的一部分内容也可在多个 DSA 内被复制。本目录规范中描述的协议允许使用复制信息来提高分布式目录服务的可用性、性能和效率。复制信息的使用在某种程度上是在用户的控制之下的，通过使用服务控制选项来实现。本目录规范中描述的规程也指示了在使用复制信息时进行设计最优化的某些时机。

另外，也规定了请求处理的交互，使得特定的目录操作特性能够被它的用户所控制。尤其是当 DSA 要响应一个与其他 DSA 中拥有的信息相关的目录请求时，用户能够控制一个 DSA 是否拥有直接查询其他 DSA 的权力(即链接)，或者它是否应当在响应中提供其他能够继续处理请求的 DSA 的信息(即转向推荐)。

一般来说，一个 DSA 是进行链接还是转向推荐的决定是根据用户所设置的服务控制，以及 DSA 的自身管理、操作或技术环境来决定的。

应当意识到的是，一般来说，目录将是分布式的，因此目录请求将由任意数量的合作 DSA 来满足，这些 DSA 根据上述的条件可以是任意地进行链接或转向推荐，本目录规范规定了 DSA 在响应分布式目录请求时所采用的适当的规程。这些规程将确保分布式目录服务的用户能够感觉到规程既是用户友好的，又是协调一致的。

第三篇：分布式目录模型

7 分布式目录系统模型

目录抽象服务，如在 GB/T 16264.3 中定义的那样，将目录建模为一个客体，该客体向其用户提供了一系列的目录服务。目录用户通过一个访问点来访问它所提供的服务。目录可以拥有一个或多个访问点，且每个访问点都通过它所提供的服务以及提供这些服务的交互模式来描述其特性。

图 1 举例说明了分布式目录模型，该模型将作为规定目录分布特性的基础。它举例说明了由一个或多个 DSA 的集合所组成的目录。

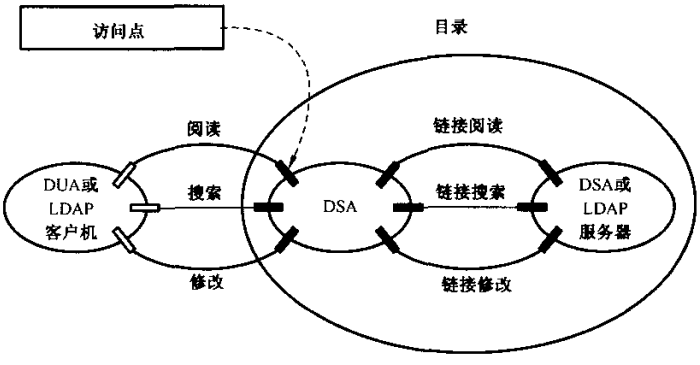


图 1 分布式目录模型中的客体

在本目录规范的后续章条中将详细规定 DSA。本章仅仅陈述一些它们的特性作为介绍性引言，同时建立本目录规范和其他目录规范之间的关系。

DSA 的定义是为了能够适应 DIB 的分布，且一系列的在物理上分布的 DSA 能够以一种预定义的、合作的方式进行交互来向目录用户(DUA 或 LDAP 客户机)提供目录服务。

图 1 举例说明了目录抽象服务和 DSA 抽象服务之间的关系。目录抽象服务在 GB/T 16264.3 中定义，是通过一系列的目录操作来提供的。为了实现该服务，组成目录的各 DSA 之间需要相互交互。

这种交互的自然特性是根据一个 DSA 可以向另一个 DSA 所提供的服务来定义的，即 DSA 抽象服务。DSA 抽象服务是通过一系列的操作来提供的，这些操作被称为链接操作，每个操作都在目录抽象服务中拥有一个对应操作。

因此，在目录抽象服务中给定的一个操作，如阅读，可要求提供服务的 DSA 通过使用链接操作来与其他的一个或多个 DSA 进行交互，如链接阅读。

注：对于作为 LDAP 请求者的 DSA 来说，进行链接操作也是可能的，例如，使用 LDAP 控制或扩展操作；然而，完成这些的规程和协议不在本目录规范的定义范围之内。

8 DSA 交互模型

目录的一个基本特性是，给定一个分布式 DIB，一个用户应当潜在地能够被满足任何服务请求(在符合安全性、访问控制和管理策略等的前提下)，而不论请求所发起的访问点。为了适应此需求，必要的是任何一个与满足某个特定服务请求相关的 DSA 都应拥有被请求的信息位于何处的知识(在 GB/T 16264.2—2008 中指定)，并且，或者将这些知识返回给请求者，或者试图自己来满足此请求(请求者可以是一个 DUA、一个 LDAP 客户机或其他的 DSA；在后一种情况下，两个 DSA 都应支持 DSP)。

定义了三种 DSA 交互模式以便符合这些需求，这三种模式被称为“单链接”，“多链接”和“转向推荐”。在本目录规范的后续部分，使用通用术语“链接”在适当的上下文中来表示单链接和/或多链接。“链接”指的是 DSA 为了满足某个请求，而向另一个 DSA 发送一个或多个链接操作；“转向推荐”指的是向请求者返回知识信息，于是请求者自身可以再与知识信息中所标识的 DSA 进行交互。

单链接或转向推荐交互可源于一个单独的请求。可选的，请求可以在交互之前被分解为多个子请求。多链接或多转向推荐交互，或两者的混合，可能源自分解后的请求。定义了两种分解类型：NSSR 分解和请求分解。

8.1 一个请求的分解

8.1.1 NSSR 分解

NSSR 分解是将同样的请求准备成便于传送到(或者是顺序的，或者是并行的)多个下级 DSA 中去的过程，这是在名(称)解析的过程中遇到一个 NSSR 的结果。非特定下级引用中不包含被引用的下级命名上下文的 RDN，因此，引用的 DSA 不能够区分哪个下级 DSA 拥有哪个下级命名上下文。因此，在名(称)解析过程中，一个遇到 NSSR 的 DSA 应向每个下级 DSA(在没有影像的情况下)发送一个同样的请求。这可以是顺序执行的，也可以是并行执行的。典型地，仅有一个 DSA 能够继续执行名(称)解析；而其余 DSA 将返回一个问题为 unableToProceed 的 serviceError。在某种(很少)情况下，有可能有多个 DSA 将继续进行名(称)解析，由此导致了双重结果。

注：NSSR 不能引用 LDAP 服务器。

8.1.2 请求分解

请求分解，另一种分解请求的方式，是在 DSA 与其他一个或多个 DSA 和/或 LDAP 服务器通信之前，由 DSA 内部执行的一个过程。一个请求被分解为多个可能不同的子请求，因此每个子请求完成原始任务的一部分。请求分解能够仅被用于列表或搜索操作的操作赋值过程中。在请求分解完成后，每个子请求可以被链接到其他 DSA 和/或 LDAP 服务器以便继续任务的执行，或者可有一个局部结果(一个内嵌的转向推荐)返回给请求者。同一个子请求被产生到不同的 DSA 和/或 LDAP 服务器中的一个示例为：某个条目具有下级引用和/或 NSSR，且共同引用了多个 DSA 或 LDAP 服务器。不同的子请求被产生到相同的或不同的 DSA 和/或 LDAP 服务器中的一个示例为：两个不同的条目在某个搜索

(子树)的操作中相遇,且每个都拥有一个下级引用。

8.2 单链接

当一个 DSA 具有另一个 DSA 所拥有的命名上下文的知识时,第一个 DSA 可使用这种交互模式(图 2 中显示)将请求传递给第二个 DSA。单链接可用于与某个单独的 DSA 联系,该 DSA 是在交叉引用、下级引用、上级引用、提供者引用或主引用中被指向的 DSA。

注:在图 2 中,交互的顺序由交互线上相关联的数字来定义。

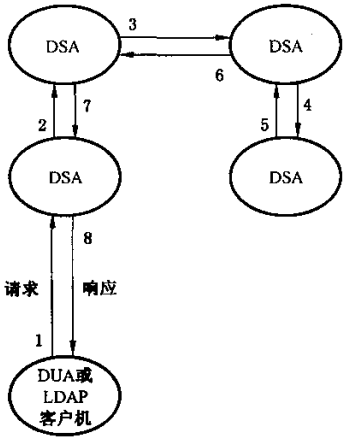


图 2 单链接模式

8.3 多链接

DSA 使用这种交互模式来传递多个出请求,这些出请求来自于同一个入请求,或者是请求分解的结果,或者是 NSSR 分解的结果。

8.3.1 并行多链接

在并行多链接情况下,DSA 将多个出请求同步进行传递(见图 3a))。并行多链接可带来性能的提高,但同时它也可以在某种环境下,如在影像存在时,导致接收到复制结果。

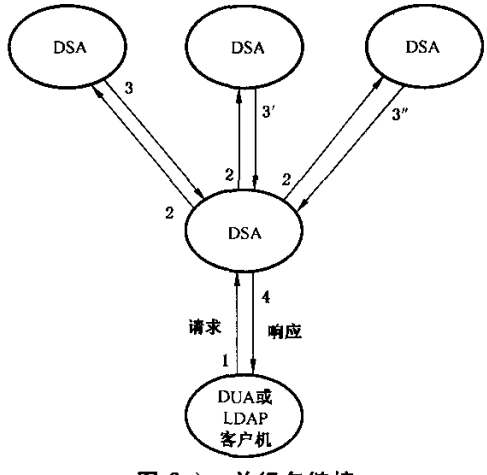
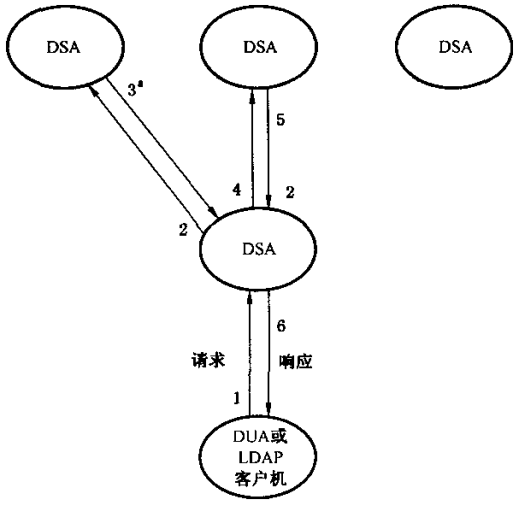


图 3a) 并行多链接

8.3.2 顺序多链接

在顺序多链接情况下,DSA 在一个时间点传递一个出请求,并且在发送下一个请求前等待此请求的结果或差错(见图 3b))。顺序多链接可以不是一种最快的交互模式,但同时它也不可以导致接收到复制结果。

注:一个 DSA 可以联合使用并行多链接和顺序多链接。



[a] 不能继续。

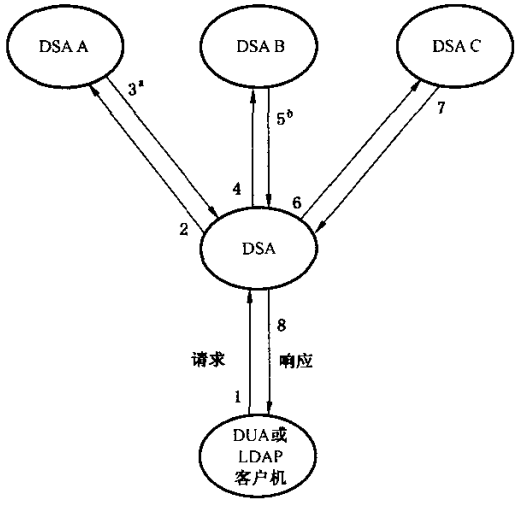
图 3b) 顺序多链接
(NSSR 分解的结果)

8.4 转向推荐

DSA 可以在响应从某个 DUA、LDAP 客户机或其他 DSA 发来的请求时,返回一个转向推荐(在图 4a)和图 4b)中描述)。转向推荐可构成整个响应(在这种情况下,它被分类为一种差错)或者仅仅是响应的一部分。转向推荐包含了知识引用,该知识引用可以是一个上级引用、下级引用、交叉引用、非特定下级引用,提供者引用或主引用等。

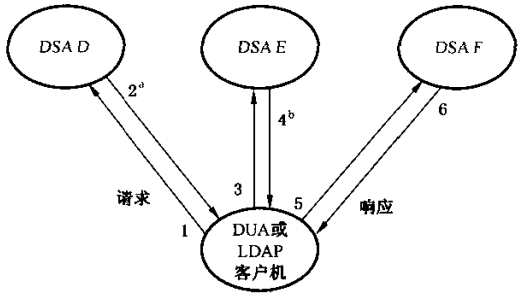
接收到转向推荐的 DSA(见图 4a))可使用包含在转向推荐中的知识引用来进行随后的链接或多播(依赖于引用的类型),将一个原始请求链接或多播到其他的 DSA。可选的,接收到转向推荐的 DSA,可在其响应中依次传递转向推荐。接收到转向推荐的一个 DUA 或 LDAP 客户机(见图 4b))可使用该转向推荐来与一个或多个其他 DSA 联系以便继续执行请求。

注:在图 4a)和图 4b)中,交互的顺序由交互线上相关联的数字来定义。



[a] 转向推荐到 B。
[b] 转向推荐到 C。

图 4a) 转向推荐模式(DSA 根据转向推荐执行动作)



[a] 转向推荐到 E。

[b] 转向推荐到 F。

图 4b) 转向推荐模式(DUA 根据转向推荐执行动作)

8.5 模式的决定

如果一个 DSA 自身不能完全地解决一个请求,则它应将该请求(或通过分解原始请求而形成的一个请求)链接到另一个 DSA,除非:

a) 用户通过服务控制禁止链接,在这种情况下,DSA 应返回一个转向推荐或一个问题为chaining Required 的serviceError;或者

b) DSA 由于管理、操作或技术等方面的原因倾向于不进行链接,在这种情况下,DSA 应返回一个转向推荐。

注 1:不进行链接的一个"技术原因"是在知识引用中标识的 DSA 不支持 DSP。

注 2:如果服务控制localScope 被设置,则 DSA(或 DMD)应或者解决该请求,或者返回一个差错。

注 3:如果用户更希望转向推荐,则用户应设置chainingProhibited。

第四篇:DSA 抽象服务

9 DSA 抽象服务概述

目录服务在 GB/T 16264.3—2008 中有全面描述。当这样的一个服务在分布式环境中提供时,如第 7 章中的建模,它可以被认为是通过一系列的 DSA 来提供的。如图 1 中所示。

对于目录服务中定义的每个操作,在 DSA 抽象服务中都定义了一个相应的"链接"操作,在完成目录服务操作中合作的 DSA 之间可以使用这些操作。这样,一个从 DUA 处接收到阅读操作的 DSA 可请求另一个 DSA 的协助(例如,另一个 DSA 中拥有目标条目或其拷贝)来满足此操作,因此会向此 DSA 发送一个链接阅读操作。

在 DSA 抽象服务中交互的信息类型在第 10 章定义。DSA 抽象服务的操作和差错在第 11 章到第 13 章定义。

10 信息类型

10.1 引言

本章标识,同时在某些情况下定义,在后续的 DSA 抽象服务的各种操作定义中所使用的一系列信息类型。

这些涉及到的信息类型包括:对于多个操作而言是通用的信息类型,或者将来可以是通用的信息类型,或者是足够复杂或自包含的信息,需要与使用它们的操作分别定义。

在 DSA 抽象服务的定义中使用的多个信息类型实际上是在别处定义的。10.2 标识了这些类型,并且指示了它们的定义来源。10.3 到 10.10 分别标识并定义了一个信息类型。

10.2 其他地方定义的信息类型

下列信息类型在 GB/T 16264.2—2008 中定义：

——aliasedEntryName；

——DistinguishedName；

——Name；

——RelativeDistinguishedName。

下列信息类型在 GB/T 16264.3—2008 中定义：

（绑定）

——DirectoryBind。

（操作）

——Abandon。

（差错）

——abandoned；

——attributeError；

——nameError；

——securityError；

——serviceError；

——updateError。

（信息客体类）

——OPTIONALLY-PROTECTED。

（数据类型）

——SecurityParameters。

下列信息类型在 GB/T 16264.6—2008 中定义：

——PresentationAddress。

10.3 链接变元

链接变元ChainingArguments 出现在每个链接操作中，向某个 DSA 传递信息，这些信息是在该 DSA 成功执行全部任务中它需要完成的部分时所需的信息：

```
ChainingArguments ::= SET {
        originator              [0]     DistinguishedName OPTIONAL,
        targetObject            [1]     DistinguishedName OPTIONAL,
        operationProgress       [2]     OperationProgress
                                        DEFAULT { nameResolutionPhase notStarted },
        traceInformation        [3]     TraceInformation,
        aliasDereferenced       [4]     BOOLEAN DEFAULT FALSE,
        aliasedRDNs             [5]     INTEGER OPTIONAL,
                                ——仅出现在第 1 版本系统中
        returnCrossRefs         [6]     BOOLEAN DEFAULT FALSE,
        referenceType           [7]     ReferenceType DEFAULT superior,
        info                    [8]     DomainInfo OPTIONAL,
        timeLimit               [9]     Time OPTIONAL,
        securityParameters      [10]    SecurityParameters DEFAULT { },
        entryOnly               [11]    BOOLEAN DEFAULT FALSE,
        uniqueIdentifier        [12]    UniqueIdentifier OPTIONAL,
```

```
        authenticationLevel         [13]    AuthenticationLevel OPTIONAL,
        exclusions                  [14]    Exclusions OPTIONAL,
        excludeShadows              [15]    BOOLEAN DEFAULT FALSE,
        nameResolveOnMaster         [16]    BOOLEAN DEFAULT FALSE,
        operationIdentifier         [17]    INTEGER OPTIONAL,
        searchRuleId                [18]    SearchRuleId OPTIONAL,
        chainedRelaxation           [19]    MRMapping OPTIONAL,
        relatedEntry                [20]    INTEGER OPTIONAL,
        dspPaging                   [21]    BOOLEAN DEFAULT FALSE,
        nonDapPdu                   [22]    ENUMERATED { ldap (0) } OPTIONAL,
        streamedResults             [23]    INTEGER OPTIONAL
        excludeWriteableCopies      [24]    BOOLEAN DEFAULT FALSE }
Time ::= CHOICE {
        utcTime             UTCTime,
        generalizedTime     GeneralizedTime }
```

不同组件的含义定义如下：

a) 组件originator 携带了请求的最初发起者的名(称)，除非已经在安全参数中指定。如果在CommonArguments 中出现了requester，则该变元可以被忽略。

注 1：当发起者具有通过上下文所区分的可替代名(称)时，则用于originator 值的名(称)应是主可辨别名，如果已知的话。否则，基于originator 值的鉴别和访问控制可以不按照期望的方式工作。

b) 组件targetObject 携带了其目录条目是要被指向的客体名。该客体的角色依赖于相关的特定操作：它可以是其条目要被操作的客体，或者是某个包含多个客体的请求或子请求的基本客体(例如chainedList 或chainedSearch)。当且仅当本组件的取值与链接操作中的客体或基本客体参数的取值相同时，本组件才能够被忽略，在这种情况下，它的隐含值即为该参数值。

当targetObject 所包括的 RDN 包含了属性类型和值对，且有多个通过上下文所区分的可辨别值时，被解析的 RDN 应是主 RDN。

c) 组件operationProgress 用于向 DSA 通报操作的执行进展，因此也通报了该 DSA 在整个执行过程中所期望承担的角色。本组件中携带的信息在 10.5 指定。

d) 组件traceInformation 用于当操作中有链接时，防止在 DSA 间出现环回。一个 DSA 在向另一个 DSA 链接一个操作之前，将在踪迹信息中增加一个新的元素。当一个 DSA 被请求执行某个操作时，该 DSA 将通过检查踪迹信息来证实该操作没有形成一个环回。本组件中携带的信息在 10.6 指定。

e) 组件aliasDereferenced 是一个布尔值，用于指示在分布式名(称)解析过程中，到目前为止所遇到的一个或多个别名条目是否被解除引用。缺省值FALSE 指示没有别名条目被解除引用。

f) 组件aliasedRDNs 指示了已经有多少个targetObject 的 Name 中的 RDN 从一个(或多个)别名条目的aliasedEntryName 属性中产生出来。当遇到一个别名条目并被解除引用后，该整数值被设置。当且仅当组件aliasDereferenced 为TRUE 时，本组件应存在。

注 2：本组件是为了与目录第 1 版本的实现相兼容而提供的。根据目录规范的后续版本实现的 DUA(和 DSA)应当总是忽略后续请求中CommonArguments 中的此参数。在这种方式下，如果别名解除引用到其他的别名时，目录不会发出差错信号。

g) 组件returnCrossRefs 是一个布尔值，指示了在执行一个分布式操作的过程中所使用的知识引用是否被要求作为交叉引用，伴随一个结果或转向推荐而传递回初始的 DSA。缺省值为 FALSE 指示了这样的知识引用将不被返回。

h) 组件referenceType 向被要求执行操作的 DSA 指示，使用了什么类型的知识将请求路由到该 DSA。因此，该 DSA 能够在调用者所拥有的知识中检测到差错。如果有这样的差错被检测出，则它应在serviceError 中指示，且携带的问题为invalidReference。ReferenceType 在 10.7 中完整描述。

注 3：如果referenceType 缺失，则假设值为superior。

i) 组件info 用于在多个 DSA 之间传递 DMD 特定的信息，这些 DSA 包含在某个公共请求的处理中。本组件的类型为DomainInfo，是一个无限制的类型：

DomainInfo ::= ABSTRACT-SYNTAX. &Type

j) 组件timeLimit，如果存在，则指示了操作应当完成的时间限制（见 16.1.4.1）。在Time 的值用于任何一个比较操作之前，且Time 的语法选择为UTCTime 类型时，两位数字的年字段值应被合理化为四位数字的年字段值，如下所述：

——如果两位数字的值是 00 到 49（含），则最后的值应当加 2000。

——如果两位数字的值是 50 到 99（含），则最后的值应当加 1900。

注 4：使用GeneralizedTime 可阻止与某些实现的互操作，这些实现不关注选择UTCTime 或GeneralizedTime 的可能性。谁规定了本目录规范将应用的域，例如轮廓分组，则应当由谁来负责什么时候可使用GeneralizedTime。在任何情况下，UTCTime 都不能被用于表示超出 2049 年以后的日期。

k) 组件SecurityParameters 在 GB/T 16264.3—2008 中指定。它的缺失被认为与安全参数为空集是等价的。

l) 如果初始的操作是一个搜索操作，其中变元subset 被设置为oneLevel，且遇到一个别名条目是baseObject 的直接下级，在这种情况下，组件entryOnly 被设置为TRUE。成功地对targetObject 的名（称）执行了名（称）解析的 DSA 应对这个唯一的命名条目执行客体赋值。

m) 组件uniqueIdentifier 是可选提供的，当它被要求证实发起者的名（称）时才提供。其数据类型 UniqueIdentifier 在 GB/T 16264.2—2008 中描述。

n) 组件authenticationLevel 是可选提供的，当它被要求指示鉴别执行的方式时才提供。其数据类型AuthenticationLevel 在 GB/T 16264.2—2008 中描述。

o) 组件exclusions 仅对于搜索操作有意义；它如果存在的话，指示了targetObject 的哪个下级条目子树应从搜索操作的结果中排除出去（见 10.9）。

p) 组件excludeShadows 仅对于搜索和列表操作有意义；它指示了搜索应被应用于条目，而不能应用于条目拷贝。本可选组件可以被某个 DSA 使用，用以避免接收到复制的结果（见 20.1）。

q) 组件nameResolveOnMaster 仅在名（称）解析时有意义，且仅在遇到 NSSR 时，才被设置。如果被设置为TRUE，则它指示后续的名（称）解析，即匹配从nextRDNToBeResolved 开始的剩余 RDN，不能够部署条目拷贝信息，其中包括多主实现中的可写拷贝；后续对每个剩余 RDN 的解析都应在该 RDN 所标识的条目的主 DSA 中执行（见 20.1）。

r) 组件operationIdentifier 简化了 DAP 操作与后续相关的 DSP 操作以及结果之间的相关性。它由第一个接收到 DAP 请求的 DSA 所分配，或者从要求链接的 DSP 请求的链接变元中复制。分配operationIdentifier 的 DSA 在一段足够长的时间段内，不能重新使用已分配的整数值。相关的 DAP 和 DSP 请求及结果的相关性通过这样的方式被简化，即对于每个操作和结果，DSA 都将operationIdentifier 以及分配它的 DSA（在链接请求的traceInformation 中的第一个 DSA）的名（称）记入日志。这样的相关性可能在用于日志、审计、计费和结算等目的时是有用的。

s) 组件searchRuleId 携带了某个搜索规则的唯一标识符。在下述情况下，它将包含在执行初始搜索规程（Ⅰ）的 DSA 中，即如果该规程在一个服务特定的管理区内发起，且当在 DIT 中往下执行时，或者跟随着别名时，或者跟随着分等级分组指针时，搜索操作被传递到其他 DSA。

t) 组件chainedRelaxation 可以在一个分布方式下为链接搜索操作执行放宽。如果某个 DSA 接收到一个链接搜索操作，且支持放宽策略，则该 DSA 能够使用所提供的chainedRelaxation 组件来替代任何它可以实现的其他放宽策略，因此使得放宽可以在那些潜在地返回搜索结果的 DSA 间进行协调。

u) 当接收 DSA 被要求解析相关的条目时，元素relatedEntry 应存在。若存在时，接收 DSA 应仅对SearchArgument 中joinAttributes 的值relatedEntry 所指定的特定相关条目元素进行响应。这样，值为零的一个relatedEntry 应选择SearchArgument 的joinAttributes 序列中的第一个元素。该值永远不能超过一个低于SearchArgument 中joinAttributes 组件内元素个数的一个值。在某个指定相关条目的 DSP 操作中，如果ChainingArguments 内的relatedEntry 元素缺失，则表示正在被链接的分布式操作是基本客体搜索，而不是搜索的相关条目部分。

注 5：如果一个将要被链接指向的 DSA 被要求既处理通常的搜索结果又处理相关条目结果，则通过向此 DSA 发送两个不同的 DSP 操作来实现。

当relatedEntry 元素存在，则应应用下列特殊的规则：

——在评估SearchArgument 的组件selection 的子组件infoTypes 时，infoTypes 应被认为具有值attributeTypesAndValues，而不论初始指定的值是什么；

——在JoinAttPair 的任意joinAtt 组件中指定的所有属性应被包含在选择中，不论之前是否被包含在内；

——整理相关条目结果的 DSA 应忽略值和未指定的变元，这样使得结果可以符合初始的用户请求。

在某个支持相关条目的 DSA 的后续的出ChainingArguments 中，应将变元relatedEntry 传递出去。

v) 如果绑定 DSA 不同于初始执行者(见 15.5.5)，且绑定 DSA 支持 DSP 分页结果，则它可以设置dspPaging 组件值为 TRUE 来指示初始执行者提供 DSP 分页结果。如果该组件取值为 FALSE(缺省值)，则初始执行者不应执行 DSP 分页结果。一个支持 DSP 分页结果的初始执行者不应将该组件前向到它正在发送子请求的那些 DSA。

w) 组件nonDapPdu 用于指示 PDU 在链接变元中是否被封装，该链接变元源自一个非 DAP 请求，如一个 LDAP 请求。

x) 组件streamedResults 用作一个计数器，来决定在响应该操作时，是否要链接流结果。当且仅当该计数器存在，且 DSA 理解流结果，并且 DSA 希望为其链接操作接收流结果时，每个包含在名(称)解析中的 DSA 才将该计数器加 1。因此，该计数器被最终完成名(称)解析的 DSA 使用来判断之前的每个 DSA 是否都准备要处理流结果。

y) 组件excludeWriteableCopies 仅针对搜索和列表操作有意义；它指示搜索应应用于条目的主拷贝，而不能应用于这些条目的可写拷贝。本可选组件可以被某个 DSA 使用作为一种方式来避免接收到复制结果(见 20.1)。

10.4 链接结果

链接结果ChainingResults 出现在每个操作的结果中，并向调用该操作的 DSA 提供反馈信息。

```
ChainingResults ::= SET {
        info [0] DomainInfo OPTIONAL,
        crossReferences [1] SEQUENCE SIZE (1..MAX) OF CrossReference OPTIONAL,
        securityParameters [2] SecurityParameters DEFAULT { },
        alreadySearched [3] Exclusions OPTIONAL }
```

不同组件的含义定义如下：

a) 组件info 用于在处理一个公共请求过程中所包含的多个 DSA 之间传递 DMD 特定的信息。

该组件的类型为DomainInfo，是一个无限制类型。

b) 组件crossReferences 一般不出现在ChainingResults 中，除非相应请求中的returnCrossRefs 组件取值为 TRUE。该组件由一个按序排列的CrossReference 项所组成，每个项中都包含一个contextPrefix 和一个accessPoint 描述符(见 10.8)。

```
CrossReference ::= SET {
contextPrefix [0] DistinguishedName,
accessPoint [1] AccessPointInformation }
```

当某个操作的targetObject 变元中的一部分与某个 DSA 的上下文前缀之一相匹配时，则该 DSA 可增加一个CrossReference。DSA 的管理机构可有策略规定不能返回这样的知识，在这种情况下，则不会在序列中增加相应项。

c) 数据类型SecurityParameters 在 GB/T 16264.3—2008 中指定。组件securityParameters 的缺失被认为与安全参数为空集时是等价的。

d) 组件alreadySearched，如果存在，指示了作为targetObject 下级的哪个下级 RDN 已经被作为链接搜索操作的一部分被处理过了，因此在后续的子请求中将被排除出去。

注：处于contextPrefix 或alreadySearched 中的名(称)应是主可辨别名，且不应包含可替代可辨别名。

10.5 操作进展

OperationProgress 的值描述了执行一个操作的进展状态，该操作中有多个 DSA 参与。

```
OperationProgress ::= SET {
        nameResolutionPhase [0] ENUMERATED {
            notStarted (1),
            proceeding (2),
            completed (3) },
        nextRDNToBeResolved [1] INTEGER OPTIONAL }
```

不同组件的含义定义如下：

a) 组件nameResolutionPhase 指示了在处理某个操作的targetObject 名(称)时，已经到达了哪个阶段。当它指示名(称)解析尚未开始时(值为notStarted)，则表示迄今为止尚未接触到这样的 DSA，即在命名上下文中包含该名(称)的初始 RDN 的 DSA。如果名(称)解析是正在进行中(值为proceeding)，则表示名(称)的初始部分已经被可辨别，尽管尚未到达拥有目标客体的 DSA。组件nextRDNToBeResolved 指示已经识别了名(称)的多少部分(见 10.5 的 b))。如果名(称)解析已经完成(值为completed)，则表示已经到达了拥有目标客体的 DSA，且适当的操作正在执行中。

b) 组件nextRDNToBeResolved 向 DSA 指示了在targetObject 名(称)中的哪个 RDN 是下一个将被解析的客体。它的格式是一个整数，取值范围从 1 到名(称)中 RDN 的个数。该组件仅在组件nameResolutionPhase 取值为proceeding 时，才存在。

10.6 踪迹信息

踪迹信息TraceInformation 的值中携带了包含在某个操作执行过程中的 DSA 的记录。它可用于判断由于知识不一致或 DIT 中出现别名环回而可导致的环回的存在，或者避免环回的出现。

```
TraceInformation ::= SEQUENCE OF TraceItem
TraceItem ::= SET {
        dsa [0] Name,
        targetObject [1] Name OPTIONAL,
        operationProgress [2] OperationProgress }
```

每个将操作散播到其他 DSA 的 DSA 将在TraceItem 序列的尾部增加一个新的项。每个这样的

TraceItem 包含：

a) 增加该项的 DSA 的名(称)。

b) targetObject 的名(称)是增加该项的 DSA 在入请求中接收到的名(称)。如果被链接的请求来自一个 DUA(在这种情况下，它的隐含值为XOperation 中的object 或baseObject)，或者如果它的值与出请求中的ChainingArgument 内的targetObject 值相同(实际相同或隐含相同)，则该参数被忽略。

c) operationProgress 的值是增加该项的 DSA 在入请求中接收到的值。

dsa 应是主可辨别名，且不应包含可替代可辨别名。被处理的targetObject 中的每个 RDN 都应是一个主 RDN。

在 RDN 的AttributeTypeAndDistinguishedValue 内的组件valuesWithContext 中可以包含带有上下文的可替代可辨别名。

10.7 引用类型

引用类型ReferenceType 的值指示了在 GB/T 16264.2—2008 中定义的各种类型的引用之一。

```
ReferenceType ::= ENUMERATED {
        superior                (1),
        subordinate             (2),
        cross                   (3),
        nonSpecificSubordinate  (4),
        supplier                (5),
        master                  (6),
        immediateSuperior       (7),
        self                    (8),
        ditBridge               (9) }
```

10.8 访问点信息

有三种类型的访问点：

a) 一个AccessPoint 值标识了一个特定的点，在该点可以对目录，更明确来说是对一个 DSA 或一个 LDAP 服务器，进行访问。当标识的是一个 DSA 时，则访问点应具有相关 DSA 的名(称) Name，并且可有一个表示层地址PresentationAddress，用于该 DSA 的 OSI 或 IDM 通信，在这种情况下，不能出现labeledURI。当标识的是一个 LDAP 服务器时，则访问点应有一个LabeledURI，用于与该 LDAP 服务器的 LDAP 通信中。若LabeledURI 项存在，则Name 和PresentationAddress 都应被忽略，且SET OF protocolInformation 不应当出现。

```
AccessPoint ::= SET {
    ae-title [0] Name,
    address [1] PresentationAddress,
    protocolInformation [2] SET SIZE (1..MAX) OF ProtocolInformation OPTIONAL,
    labeledURI [6] LabeledURI OPTIONAL }
LabeledURI ::= DirectoryString{ub-labeledURI}
```

b) 一个MasterOrShadowAccessPoint 值标识了目录的一个访问点。访问点的category，取值或者为master，或者为shadow，依赖于它是指向一个命名上下文，还是指向一个公共可用的复制区。组件chainingRequired 指示对于此 DSA，链接是否是需要的，即不应当向该 DSA 返回一个转向推荐。

```
MasterOrShadowAccessPoint ::= SET {
    COMPONENTS OF AccessPoint,
```

category [3] ENUMERATED {
master (0),
shadow (1) } DEFAULT master,
chainingRequired [5] BOOLEAN DEFAULT FALSE }

c) 一个MasterAndShadowAccessPoints 值标识了目录的一系列访问点，即一系列相关的 DSA 和/或 LDAP 服务器。这些访问点共享特性，每个访问点都标识了一个 DSA 或 LDAP 服务器，这些 DSA 或 LDAP 服务器都拥有来自一个公共命名上下文的条目信息(或者如果值是属性nonSpecificKnowledge 的一个值时，则是来自一个由 DSA 主管的命名上下文的一个公共集合中)。MasterAndShadowAccessPoints 的一个值指示它所包含的每个AccessPoint 值的category。命名上下文的主 DSA 或 LDAP 服务器的访问点不必包含在此集合中。

注：实现者应当认识到对于一个 LDAP 服务器来说，这种情况是可能的，即使它被标识为shadow，它也可以根据接收到的 LDAP 更新操作来更新条目。

MasterAndShadowAccessPoints ::= SET SIZE (1..MAX) OF MasterOrShadowAccessPoint

一个AccessPointInformation 值标识了一个或多个目录的访问点。

AccessPointInformation ::= SET {
COMPONENTS OF MasterOrShadowAccessPoint,
additionalPoints [4] MasterAndShadowAccessPoints OPTIONAL }

若产生一个AccessPointInformation 值的 DSA 是按照第 1 版本实现的 DSA，则该集合中的可选组件不存在。若解释AccessPointInformation 值的 DSA 是按照第 1 版本实现的 DSA，则出现的任何MasterAndShadowAccessPoints 值都被忽略。

若 DSA 是按照第 2 版本和后续版本实现的，则为一个AccessPointInformation 值而产生的MasterOrShadowAccessPoint 值组件的种类可以是属主或影像，由产生该值的 DSA 的知识选择规程来决定。它可以被看作是一个建议的访问点，由产生该值的 DSA 向接收该值的 DSA 提供。可选的，还可以为一个AccessPointInformation 值产生一个MasterAndShadowAccessPoints 值。它由附加的信息组成，这些信息可以由接收 DSA 的知识选择规程来部署，用于决定一个替代的访问点。

10.9 DIT 桥接知识

一个ditBridgeKnowledge 值标识了一个特定的点，在该点可以对另一个 DIT，更明确来说是对一个 DSA 或一个 LDAP 服务器进行访问。ditBridgeKnowledge 指定了一个访问点accessPoint，在这点上可以访问到 DSA 或 LDAP 服务器。

DitBridgeKnowledge ::= SEQUENCE {
domainLocalID DirectoryString{ub-domainLocalID} OPTIONAL,
accessPoints MasterAndShadowAccessPoints }

domainLocalID 包含一个可读的描述符，标识了包含在引用中的 DIT。

10.10 排除

正如在 10.3 中定义的那样，ChainingArguments 中的exclusions 组件被用于限制某个搜索操作的范围，这是通过标识一系列的作为目标客体下级的条目以及它们的所有下级来实现的，所标识的这些条目都不能包含在该搜索操作的处理过程中。组件exclusion 被定义为 ASN.1 数据类型Exclusions 的一个值。

Exclusions ::= SET SIZE (1..MAX) OF RDNSequence

Exclusions 集中的每个RDNSequence 值都应当标识一个作为目标客体下级的命名上下文的上下文前缀。如果一个 DSA 接收到一个search 请求，且携带的某个RDNSequence 值不符合本限制，则该 DSA 可忽略此值。RDNSequence 是相对于目标客体的，因此不是上下文前缀的可辨别名。

Exclusions 应是主可辨别名。可替代可辨别名和上下文信息也可以包含在内。

除了可以作为某个用户请求中一部分，Exclusions 还能够被 DSA 使用，使得在影像信息存在的情况下，从搜索子请求中返回的复制信息最少。

图 5 举例说明了Exclusions 的一种用法示例。在这个例子中，一个 DSA 拥有两个复制区，其中一个在另一个的下面。一个起始于上下文前缀 X，另一个起始于上下文前缀 C。一个处于 Y 的条目拷贝拥有三个下级引用，分别指向命名上下文 A、B 和 C。

作为一个示例，如果该 DSA 执行了一个子树搜索，该搜索起始于命名上下文 X 内的一个基本客体，则 DSA 能够从复制区 X 和 C 中提供信息。命名上下文 A 和 B 中的信息应通过下级引用才能够被提供。在执行请求分解时，可用于partialResults 或链接中的连续引用，将指定 Y 为目标客体，而 C 为一个Exclusions 集内的单独元素。

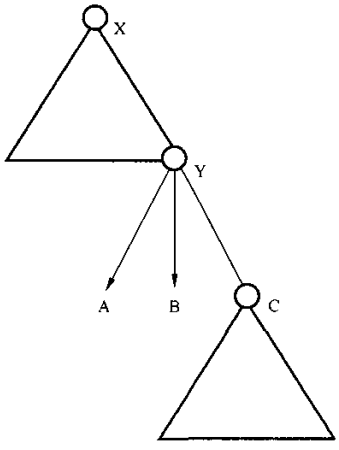


图 5　排除

10.11　连续引用

一个连续引用ContinuationReference 描述了某个操作的全部或部分是如何在一个不同的 DSA、LDAP 服务器或它们的某种组合中继续执行的。典型地，当包含在内的 DSA 不能够或不愿意自己去传播请求时，连续引用是作为一个转向推荐来返回的。

```
ContinuationReference ::= SET {
    targetObject [0] Name,
    aliasedRDNs [1] INTEGER OPTIONAL, ——仅出现在第 1 版本系统中
    operationProgress [2] OperationProgress,
    rdnsResolved [3] INTEGER OPTIONAL,
    referenceType [4] ReferenceType,
    accessPoints [5] SET OF AccessPointInformation,
    entryOnly [6] BOOLEAN DEFAULT FALSE,
    exclusions [7] Exclusions OPTIONAL,
    returnToDUA [8] BOOLEAN DEFAULT FALSE,
    nameResolveOnMaster [9] BOOLEAN DEFAULT FALSE }
```

不同组件的含义定义如下：

a) 组件targetObject 指示了在继续执行操作中所要使用的客体名。它可以不同于入请求中接收到的targetObject 中的名(称)，例如在一个别名被解除引用，或者搜索中的基本客体已经被定位的情况下。在targetObject 中的 RDN 应是主 RDN(对于已经处理过的 RDN)。带上下文的

可替代可辨别值也可以包含在内。

b) 组件aliasedRDNs 指示了在目标客体名中有多少个 RDN(如果有的话)已经被别名解除引用处理过了。当且仅当别名已经被解除引用时该变元存在。

注:提供本组件是为了与目录第 1 版本的实现相兼容。根据后续的目录规范版本实现的 DUA(和 DSA)应当总是忽略后续请求的CommonArguments 中的此参数。在这种情况下,如果别名解除引用到其他别名时,目录不会发出差错信号。

c) 组件operationProgress 指示了已经完成的名(称)解析的数量,该参数将控制指定 DSA 的后续操作的执行,假设接收ContinuationReference 的 DSA 或 DUA 希望在其后继续执行。

d) 组件rdnsResolved 的值(如果名(称)中的部分 RDN 不能进行完全的名(称)解析,但是被假设通过一个交叉引用而纠正了,在这种情况下,才需要出现该组件)指示了实际上有多少个 RDN 仅使用内部引用就已经被解析过了。

e) 组件referenceType 指示了什么类型的知识被用于产生此连续引用。

f) 组件accessPoints 指示了为了获得此连续引用而需要接触的访问点。仅在包含非特定下级引用时,才能够出现多个AccessPointInformation 项。

g) 当初始操作是一个搜索操作,且subset 变元被设置为oneLevel,并且遇到一个别名条目是 baseObject 的直接下级时,在这种情况下,组件entryOnly 被设置为TRUE。针对targetObject 名(称)成功地执行了名(称)解析的 DSA,应对此唯一的命名条目执行客体赋值。

h) 组件exclusions 标识了一系列下级命名上下文,这些下级命名上下文不能由接收方 DSA 部署。

i) 元素returnToDUA 是可选提供的,当创建此连续引用的 DSA 希望指示出它不愿意通过一个中介 DSA 来返回信息(如出于安全原因),并且希望指示出信息可以通过初始 DUA 或 LDAP 客户机和 DSA 之间的一个 DAP 或 LDAP 操作而直接可用时,则提供此组件。若returnToDUA 被设置为 TRUE,则referenceType 可以被设置为self。

j) 元素nameResolveOnMaster 是可选提供的,当创建此连续引用的 DSA 遇到 NSSR 时,则提供此组件。如果设置为TRUE,则它指示后续的名(称)解析,即与从nextRDNToBeResolved 开始的剩余 RDN 进行匹配时,不能使用条目拷贝信息,其中包括多主实现中的可写拷贝;对每个剩余 RDN 的后续解析应在该 RDN 所标识的条目的主 DSA 内进行(见 20.1)。

11 绑定和解绑定

DSABind 和DSAUnbind 分别被 DSA 用在访问另一个 DSA 的时间段的起始点和结束点。一个 DSP 联系的绑定和解绑定本身不能够导致在联系过程中所请求的任何分布式分页结果的丢失。

11.1 DSA 绑定

一个DSABind 操作被用于开始两个提供目录服务的 DSA 之间的合作阶段。

DSABind ::= BIND

ARGUMENT DirectoryBindArgument

RESULT DirectoryBindResult

BIND-ERROR DirectoryBindError

DSABind 中的组件与目录绑定DirectoryBind(见 GB/T 16264.3—2008)中的对等部分是相同的,除了下列所述的不同:

——DirectoryBindArgument 中的Credentials 允许标识始发 DSA 的AE-Title 的信息能够被发送到响应 DSA。AE-Title 的格式应为一个目录可辨别名。

——DirectoryBindResult 中的Credentials 允许标识响应 DSA 的AE-Title 的信息能够被发送到始发 DSA。AE-Title 的格式应为一个可辨别名。

——DSA 的名(称)或AE-Title 可以使用可替代可辨别名,并且可以包含上下文信息。

注 1:因为名(称)可被用于简单证书或强证书,因此可以使用可替代可辨别名,如果它们存在的话。然而,如果不使用主可辨别名,则基于名(称)的鉴别和访问控制可以不能按照期望的那样工作。随后对已鉴别的 BIND 操作的成功处理,无论在 BIND 操作中使用什么样的名(称),从此以后被绑定的实体相互之间应知道它们的主可辨别名,以便简化在 BIND 生效过程中访问控制的操作。

注 2:鉴别所要求的证书可以被安全交换服务元素(见 GB/T 16264.5—2008)所携带,在这种情况下,它们不出现在绑定变元或结果中。

11.2 DSA 解绑定

两个提供目录服务的 DSA 之间合作阶段结束时的解绑定,如果用于 OSI 环境,则在 GB/T 16264.5—2008 的 7.6.4 和 7.6.5 规定,如果用于 TCP/IP 环境,则在 GB/T 16264.5—2008 的 9.3.2 规定。

12 链接操作

对于每个用于访问目录抽象服务的操作,在合作的 DSA 之间都有一个一一对应的对等操作。所选择的操作名(称)要体现出这种对应性,因此在合作的 DSA 之间使用的操作名(称)前都增加一个前缀“链接”。

链接操作的变元、结果和差错都是从目录抽象服务的相应操作的变元、结果和差错中系统构建而成的(见 12.1 的描述),只有一个例外。这个例外便是ChainedAbandon 操作,它在语法上是等同于它相对应的目录抽象服务的(见 12.2 的描述)。

12.1 链接操作

如果一个 DSA 接收到来自一个 DUA 或 LDAP 客户机的操作,则它可以选择构建该操作的一个链接形式来将此操作传播到另一个 DSA。而接收到一个操作的链接形式的 DSA,也可以选择将此操作链接到另外一个 DSA。

调用一个操作的链接形式的 DSA 可以对操作的参数进行签名、加密,或签名并加密;而执行操作的 DSA,如果被这样请求的话,也可以对操作响应者所返回的结果或差错进行签名、加密或签名并加密。从某个 LDAP 客户机处接收到一个操作的 DSA,或者从另一个 DSA 处接收到一个 LDAP 操作的 DSA,可以选择将原始的 LDAP 客户机提供的操作传播到一个 LDAP 服务器。

一个操作的链接形式的规范是使用参数化类型chained { }。

```
chained { OPERATION : operation } OPERATION ::= {
    ARGUMENT OPTIONALLY-PROTECTED {
        SET {
            chainedArgument ChainingArguments,
            argument [0] operation. &ArgumentType } }
    RESULT OPTIONALLY-PROTECTED {
        SET {
            chainedResult ChainingResults,
            result [0] operation. &ResultType } }
    ERRORS { operation. &Errors EXCEPT referral | dsaReferral }
    CODE operation. &operationCode }
```

注 1:可以被用于chained { }中的实际参数的目录抽象服务的操作包括abandoned 差错。在一个链接操作的一系列可能差错中,此差错的出现体现了 12.2 讨论的可能性,即当一个链接的联系失效时,可以为chainedModify 操作产生一个chainedAbandon。

注 2:在附录 A 中对 DSA 抽象服务的明确规范中,应用了此参数化类型来构造所有的抽象服务的链接操作。派生出的操作的变元具有下列组件:

a) chainedArgument:这是ChainingArguments 的一个值,包含了除始发 DUA 或 LDAP 客户机

提供的变元之外的那些信息，为了让执行的 DSA 或 LDAP 服务器能够完成操作，需要这些信息。该信息类型在 10.3 定义。

b) argument：这是operation. &Argument 的一个值，包含了 DUA 提供的原始变元，如同在 GB/T 16264.3—2008 的相应章条规定的那样，或者包含了 LDAP 客户机提供的原始变元，如同在 RFC 2251 的相应章条规定的那样。

注 3：如果认为适当的话，它还可以封装除了源自 DAP 或 LDAP 的 PDU 类型之外的其他 PDU 类型。完成上述工作机制的规范尚待研究。

如果请求成功，则派生操作的结果具有如下组件：

a) chainedResult：这是ChainingResults 的一个值，包含了除将要提供给始发 DUA 的信息之外的那些信息，在某个链接中，之前的 DSA 可能需要这些信息。该信息类型在 10.4 中定义。

b) result：这是operation. &Result 的值，由该操作的执行者将要返回的结果组成，并且将要被放在返回给始发 DUA 的结果中而被传递回去。该信息在 GB/T 16264.3—2008 的适当章条中规定。

如果请求失败，除了dsaReferral 要替代referral 被返回之外，还将返回operation. &Errors 差错集中的一个差错。可被上报的差错集在 GB/T 16264.3—2008 中为相应操作描述。差错dsaReferral 在 13.2 中描述。

12.2 链接放弃操作

chainedAbandon 操作由一个 DSA 使用，向另一个 DSA 指示它不再对之前调用的某个分布式操作继续被执行感兴趣。这可以出于多种原因，其中示例如下：

——让 DSA 初始发起链接的操作本身已经被放弃了，或者由于联系的中断而被隐含中止了；

——DSA 已经通过另外的方法获得了必要的信息，例如已经从一个包含在并行多链接中的某个更快的响应 DSA 处获得了信息。

一个 DSA 永远都不能被迫发起一个chainedAbandon，或者实际上确实放弃了一个操作，如果被要求如此的话。

如果chainedAbandon 真正成功停止了某个操作的执行，则将有一个结果被返回，且相应的操作将返回一个abandoned 差错。如果chainedAbandon 没有成功停止某个操作，则它本身将返回一个abandonFailed 差错。

12.3 链接操作和协议版本

要求协议版本要高于第 1 版的操作（例如带某些变元的modifyEntry 操作），或者当与某个版本高于第 1 版的协议使用时，返回不同结果的操作（例如带一个签名变元的modifyEntry 操作），应仅被链接到与传送请求的联系具有相同版本或更高版本的联系上。

13 链接差错

13.1 引言

在大多数情况下，在目录抽象服务中返回的相同差错都能够在 DSA 抽象服务中返回。例外情况是 dsaReferral 差错被返回（见 13.2），而替代了Referral，同时下列服务问题具有相同的抽象语法但具有不同的语义：

a) invalidReference：返回此差错的 DSA 在呼叫方 DSA 的知识中检测到一个差错，呼叫方 DSA 的知识在链接变元referenceType 中指定。

b) loopDetected：返回此差错的 DSA 在目录的知识信息中检测到一个环回。

可以出现的差错的优先级同目录抽象服务中的优先级相同，在 GB/T 16264.3—2008 中指定。

如果在一个链接操作中出现了一个差错，则响应方 DSA 可对返回的差错进行签名、加密或签名并加密。

13.2 DSA 转向推荐

dsaReferral 差错由 DSA 产生，无论何种原因，当 DSA 不愿意将该操作链接到一个或多个其他 DSA 而继续执行该操作时，产生此差错。它可以返回一个转向推荐的环境在 8.3 描述。

```
dsaReferral ERROR ::= {
        PARAMETER OPTIONALLY-PROTECTED {
            SET {
                reference [0] ContinuationReference,
                contextPrefix [1] DistinguishedName OPTIONAL,
                COMPONENTS OF CommonResults } }
        CODE id-errcode-dsaReferral }
```

不同参数的含义如下所述：

a) ContinuationReference 包含调用者在将适当的请求传播到另一个 DSA 或一个 LDAP 服务器时所需的信息。该信息的类型在 10.11 规定。

b) 如果该操作的ChainingArguments 的returnCrossRefs 组件取值为TRUE，且该转向推荐是基于一个下级引用或交叉引用，则可以可选地包含参数contextPrefix。任意 DSA 的管理机构将决定哪种知识引用(如果有的话)能够以此种方式返回(例如，其他方式可以对该 DSA 是保密的)。

contextPrefix 或一个继续转向推荐应是主可辨别名。带上下文的可替代可辨别名可以包含在任意 RDN 的AttributeTypeAndDistinguishedValue 的valuesWithContext 组件中。

所提供的信息能够可选地通过使用CommonResults 中的notification 组件进行限定。

第五篇：分布式规程

14 概述

14.1 范围和限制

本章规定了 DSA 所执行的目录分布式操作的规程。每个 DSA 都独立地执行下面所描述的规程；所有 DSA 的集体行为将构成目录提供给用户的完整的服务集合。

14.2 一致性

本章对 DSA 规程的描述是基于 GB/T 16264.2—2008 中的第 8 章和第 9 章描述的模型，以及本目录规范的第 7 章和第 8 章描述的模型。流程图及其相应的文字描述是一种方式，将一个给定的向 DSA 输入的外部(DAP、LDAP 和/或 DSP)输入集合映射为该 DSA 所产生的一个或多个外部输出(即一个结果、差错、转向推荐或链接请求等)，依赖于该 DSA 所拥有的特定的 DSA 信息树。

目录很可能分布在各种不同的 DSA 之间，这些 DSA 可根据目录规范的不同版本实现，或者仅支持 LDAP 的实现等。发起请求的 DUA 或 LDAP 客户机不会关注满足 DUA 或 LDAP 客户机请求的一个或多个 DSA 是按照哪个版本实现的。因此，要在这样一个异构环境下执行操作，DSA 应根据 GB/T 16264.5—2008 的第 12 章中定义的扩展规则来实现。

注 1：仅支持 LDAP 而实现的 DSA，可根据，也可不根据该扩展规则来实现。

DSA 的实现应在功能上与这里所描述的这些规程所规定的外部行为是等价的。某个特定的 DSA 在实现时，如何根据给定的输入以及拥有的 DSA 信息树而导出正确的输出，所使用的算法是没有被标准化的。

注 2：伴随规程的流程图是辅助性的，用来帮助对规程的理解。但不能认为它们是对文字描述的一种精确替代。

对于某个特定的规程，如果在文字描述和流程图之间有分歧，则认为文字描述的优先级较高。

14.2.1 包含第 1 版本 DSA 的交互

如果修改操作是在跨 DSA 的边界进行评估的(即带有TargetSystem 的addEntry 操作，移除或重命名某个上下文前缀等操作)，则本目录规范仅规定了两个第 2 版本或后续版本的 DSA 应当具有何种行

为。两个第1版本的DSA之间的交互,以及第1版本的DSA和第2版本或后续版本的DSA之间的交互,不在本系列目录规范的定义范围之内。如果混合版本的DSA之间具有一个分等级的操作绑定,则彼此版本的知识可允许向用户给出一个一致性方面的差错。

14.3 概念模型

目录分布式操作的复杂性引发了这样的一个需求,即同时使用叙述性描述技术和图形描述技术来进行概念建模。然而,无论是叙述性方法还是图形图表方法都不能被认为是对分布式目录操作的一种正式描述。

14.4 DSA的单独操作和合作操作

该模型从两个不同的视角对DSA操作进行观察,这两个视角结合起来共同提供了一个完整的目录操作视图。

a) 以DSA为中心的视角:在这种视角下,支持目录的一系列规程都从一个单独DSA的视点进行描述。这就使得为每个规程都提供一个明确的规范成为可能,并且能够完整地解决它们之间的相互关系和全面的控制结构。第16章到第22章便从一个以DSA为中心的视角描述了DSA规程。

b) 以操作为中心的视角:以DSA为中心的视点提供了完整的细节,但对理解单个操作的结构变得困难,这些操作可被多个DSA所执行。因此在后面的第15章采用了一个主要以操作为中心的视点来介绍应用到每个操作的执行阶段。

为了支持目录的分布式操作,每个DSA都应执行两类动作,一类是实现每个操作的自身目的所需的动作,另一类是为了将这些实现在多个DSA间分布所需的附加动作。第15章研究了这两类动作之间的差异。第16章到第22章对这两类动作都给出了详细规定。

14.5 DSA之间的合作商定

所有的DSA如果根据它们所拥有的命名上下文而处于一种下级/上级关系中,则这样的DSA之间具有分等级操作绑定和/或非特定的分等级操作绑定,依赖于这些DSA所拥有的知识引用的类型。

两个DSA之间的分等级操作绑定和非特定分等级操作绑定可通过使用第24章和第25章中定义的规程来进行管理,或者通过其他的途径(如电话)来进行管理。

一个DSA,如果它拥有的条目处于其上级DSA的管理区内,则该DSA应管理子模式,应遵循控制搜索规则(如果有的话),且应按照管理机构的要求对条目的访问进行控制。管理区内条目的调整可按照GB/T 16264.2—2008中定义的那样执行,也可通过本地机制来执行。

15 分布式目录行为

15.1 操作的合作实施

每个DSA都配备有相应的规程来完整地实施所有的目录操作。如果一个DSA包含完整的DIB,则在这种情况下,实际上所有的操作都完全是在此DSA内部执行的。如果DIB是分布在多个DSA内的,则在这种情况下,一个典型操作的实现是分段的,在每个潜在的合作DSA内将仅执行该操作的一部分。

在分布式环境中,典型的DSA将每个操作都看作是一个临时事件:由一个DUA、一个LDAP客户机或某些其他DSA所调用的操作;DSA对相应客体进行处理,然后将其前向到另一个DSA以便作后续处理。

一个替代视点认为整个处理是在其实施过程中通过多个合作的DSA由一个操作所完成的。这种视角揭示了可以应用到所有操作的公共处理阶段。

15.2 操作处理的阶段

每个目录操作都可被认为由三个不同的阶段构成:

a) *名(称)解析(Name Resolution)*阶段,在此阶段,某个特定操作将要执行的条目所在的客体名被用来定位拥有此条目的DSA;

b) *赋值(Evaluation)阶段*,在此阶段,某个特定的目录请求所指定的操作(如一个阅读操作)被真正地执行;

c) *结果合并阶段(Results Merging phase)*,在此阶段,某个指定操作的结果被返回到发出请求的DUA或LDAP客户机。如果选择了交互的链接模式,则结果合并阶段可包含多个DSA,其中每个DSA都在之前的一个或两个阶段中将原始请求或子请求(如15.3.1中的定义——请求分解)链接到另一个DSA。

在操作为阅读、比较、列表、搜索、修改条目、修改DN和移除条目的情况下,名(称)解析针对的是在操作参数中所提供的客体名。在操作为增加条目的情况下,名(称)解析的目标条目是操作参数所提供的条目的直接上级。

条目——这可以很容易地通过将操作参数所提供的名(称)中的最后一个RDN去掉而得到(这是通过18.3.1中FindDSE过程中的本地参数m而实现的)。

针对某个特定条目的操作可被初始发送到目录内的任意DSA。该DSA,使用它的知识,还可联合其他的DSA一起,通过上述三个阶段来处理此操作。

15.2.1 名(称)解析阶段

名(称)解析是一个将某个声称名(称)中的每个RDN与DIT的弧(或顶点)进行顺序匹配的过程,逻辑上起始于DIT的根,并在DIT内一直向下进行。然而,由于DIT是分布在任意的多个DSA内的,因此每个DSA可仅能够执行名(称)解析过程中的一小部分。一个给定的DSA通过遍历它的本地DSA信息树来执行名(称)解析过程中属于它的那部分。该过程在第18章中描述,并且提供了相应的图(见图9到图12)。基于它的本地DSA信息树,以及包含在内的知识信息,一个DSA能够推断出名(称)解析是否可以被一个或多个其他DSA继续进行,或者名(称)是否是差错的。

如果服务控制选项manageDSAIT被设置,则名(称)解析阶段将被限制为仅在一个DSA信息树内工作。

15.2.2 赋值阶段

当名(称)解析阶段完成后,所要求的实际操作(如阅读或搜索操作)就开始执行了。

包含一个单独条目查询的操作——阅读和比较——可能完全是在此条目所在的DSA内执行的。

包含多个条目查询的操作——列表和搜索——需要定位目标客体的下级,而这些下级可处于同一个DSA,也可处于不同的DSA。如果它们不是全部都处于同一个DSA内,则操作需要被指引到其他的DSA来完成赋值过程,这些DSA是通过(适当的)下级引用、非特定下级引用、提供者引用、或主引用来指定的。

如果服务控制选项manageDSAIT被设置,则赋值阶段将被限制为仅在一个DSA信息树内工作。类似的,如果赋值过程起始于一个服务特定的管理区,则赋值将被限制在该管理区内。

15.2.3 结果合并阶段

一旦赋值阶段的某些结果可用,则进入到结果合并阶段。

如果操作仅受一个单独条目的影响,则在这种情况下,操作结果可以被简单地返回给请求DUA或LDAP客户机。如果操作受到多个DSA内的多个条目的影响,则在这种情况下,结果可以被组合。如果对结果执行了保护,则结果不能被组合。返回给DUA或LDAP客户机的结果应是没有执行合并的结果。

在结果合并后,允许返回给请求者的响应包括:

a) 操作的完整结果;

b) 不完整的结果,由于DIT的某些部分仍是不可查询的(仅应用于列表和搜索)。这样的一个部分结果中可以包含为那些不能查询的DIT部分的连续引用;

c) 一个差错(转向推荐是一种特殊的情况);以及

d) 如果请求者是一个DSA,则可以返回一个ChainingResults。

15.3 管理分布式操作

在DSA可被要求执行的每个操作的变元中包含的信息,指示了当操作在跨越目录的多个DSA时,

该操作的执行进展情况。这使得对于每个 DSA 来说可以执行处理所要求的适当方面，并且在将操作向外前向到其他 DSA 前，记录此方面工作的完成。

在 DSA 中还包括附加的规程以便在物理上分布操作，同时支持由于它们的分布而引发的其他需求。

15.3.1 请求分解

请求分解是在 DSA 与其他一个或多个其他 DSA 和 LDAP 服务器通信之前，由 DSA 内部执行的一个过程。一个请求被分解为多个子请求，因此每个子请求完成原始任务的一部分。例如在搜索操作中，在基本客体被发现后可以使用请求分解。在分解后，每个子请求可被单链接或多链接到其他 DSA 和/或 LDAP 服务器上，以便继续执行任务。

如果操作是由一个 DUA 发起的，则一个链接请求（见 12.1）或子请求的argument 应是未经修改的操作变元，而如果操作是由一个 LDAP 客户机发起的，则该argument 应是未经修改的 LDAP 消息。一个接收到链接请求的 DSA 在做请求分解时不应当修改argument。

注：下面的各条规定了argument 中的各个组件的需求。但不应当被解释为那些没有被显式提及的组件是可以修改的。

15.3.2 DSA 作为请求响应者

接收请求的 DSA 可以使用operationProgress 参数来检查该请求的进展情况。这可以判断操作是仍然处于名（称）解析阶段还是已经到了赋值阶段，以及 DSA 应准备满足操作的哪一部分。如果 DSA 不能够完全满足请求，则它应或者将请求（通过单链接或多链接）传递到其他的能够帮助实现请求的一个或多个 DSA 和/或 LDAP 服务器，或者返回另一个 DSA 或 LDAP 服务器的转向推荐，或者终止该请求并返回一个差错。

15.3.3 操作的完成

每个发起操作的 DSA，或者将操作传播到一个或多个其他 DSA 和/或 LDAP 服务器的 DSA，都应保持该操作的存在踪迹，直到每个其他的 DSA 和/或 LDAP 服务器都已经返回了一个结果或差错，或者操作的最大时间限制已经超时。此需求可应用到所有的操作、传播模式和处理阶段。它确保了传播到目录的分布式操作可以按顺序结束。

15.4 环回的处理

DIT 可处于某种状态而引起环回。例如，在名（称）解析过程中，当对一个或多个别名进行的解除引用，使得解析又回到 DIT 的相同分支上时，则出现环回。另一个引起环回的潜在原因是由于差错地配置了知识引用。

在一个特定的目录操作的上下文内，如果在任何时间，操作返回到之前的某个状态，则出现环回，这里，状态通过下面的组件来定义：

——当前处理操作的 DSA 的名（称）；

——包含在操作变元中的targetObject 的名（称）；

——包含在操作变元中的operationProgress，如 10.5 的定义。

这并不意味着一个操作不能被某个特定的 DSA 多次处理。然而，它意味着 DSA 不能多次处理处于同一状态的同一个操作。

可以使用 10.6 定义的traceInformation 变元来控制环回，该变元记录了某个特定操作已经经历过的状态序列。定义了两个策略来判断是否出现了环回，或者将要出现环回。这两个策略是环回检测和环回避免，它们分别在 15.4.1 和 15.4.2 中描述。

环回检测是必选的，而环回避免是可选的。

15.4.1 环回检测

一旦接收到一个目录操作，DSA 最初应对操作进行有效性判断以确保该操作是可被继续执行的。有效性判断的一个重要任务是检查环回，通过判断操作的当前状态是否已经存在于该操作的traceInformation 变元中所记录的之前状态序列中。检查环回的这一步骤便是环回检测。

15.4.2 环回避免

环回避免要求某个 DSA，作为链接规程的一部分在将某个请求前向到另一个 DSA 之前，就要判断由此而产生的操作状态(这是一个traceItem，接收的 DSA 在接收到它之后会将其增加到traceInformation 中)是否已经存在于该初始入操作的traceInformation 变元中记录的之前状态序列中。

若接收到转向推荐或对转向推荐进行操作，在这种情况下，就不能仅仅通过检查traceInformation 来完成环回避免和环回检测。在这种情况下，每次 DSA 对转向推荐进行操作时，它都应存储由此而产生的操作状态(即traceItem，接收的 DSA 在接收到该请求之后会将其增加到traceInformation 中)，并且将入请求记录也随之存储。在对某个转向推荐进行操作或返回某个转向推荐之前，DSA 都需要检查此列表，以便检查当准备为入操作服务时，相同的请求在之前并未发送过。

15.5 分布式操作的其他考虑

15.5.1 服务控制

在分布式环境中，为了操作能够按照所请求的方式执行，某些服务控制需要特殊的考虑。

a) chainingProhibited：一个 DSA 在决定某个操作的传播方式时，需要考虑此服务控制。如果它被设置，则 DSA 将总是使用转向推荐方式。然而，如果它未被设置，则 DSA 能够依赖于它自身的能力，选择是使用链接方式，还是使用转向推荐方式。

b) timeLimit：一个 DSA 需要考虑此服务控制以确保该 DSA 中没有超出时间限制。被某个 DUA 请求执行操作的 DSA，最初应注意 DUA 所表达的timeLimit，该时间限制以秒计，作为完成该操作的可用的持续时间。如果要求链接，则timeLimit 包含在将要传递到下一个 DSA 的链接变元中。在这种情况下，同样的限制值将用于每个链接请求，且在此时间(UTC)之前，操作应完成以满足最初规定的限制。在接收到指定了timeLimit 的ChainingArguments 时，接收的 DSA 应遵守此限制。

c) sizeLimit：一个 DSA 需要考虑此服务控制以确保结果列表没有超出指定的尺寸限制。该限制包含在初始请求的公共变元中，在请求被链接时将被无改变地传递。如果要求进行请求分解，则相同的值包含在将要传递到下一个 DSA 的变元中，对每个子请求都应用完全的限制。当结果返回时，请求者 DSA 分析多个结果，并将此限制应用于全体结果以便确保仅有请求的数量被返回。如果超出了该限制，将在答复中指出。

d) priority：在所有的传播方式下，每个 DSA 都有责任确保操作的处理过程是有序的，以便支持此服务控制，如果此服务控制存在的话。

e) localScope：操作被限制在一个本地定义的范围内，且每个 DSA 都不能将请求传播到该范围之外。

f) scopeOfReferral：如果 DSA 针对某个列表或搜索操作返回了一个转向推荐或部分结果，则所包含的连续引用应处于被请求的范围之内。

所有其他的服务控制都需要被遵守，但是它们的使用在分布式环境中不需要任何特别的考虑。

15.5.2 扩展

如果一个 DSA 在处理过程的名(称)解析阶段遇到一个扩展操作，且判断操作应被链接到一个或多个 DSA，则它应在链接操作中无变化地包含任何出现的扩展。

注：一个管理机构，如果不希望传播一个扩展时，它可判断返回一个问题为unwillingToPerform 的serviceError 是合适的。

如果一个 DSA 在处理过程的赋值阶段遇到一个它不支持的扩展，则有两种可能性出现。如果扩展不是关键的，则 DSA 应忽略此扩展。如果扩展是关键的，则 DSA 应返回一个问题为unavailableCriticalExtension的serviceError。一个关键扩展，如果扩展到多客体操作，可能会导致出现多样的结果和服务差错。合并这些结果和差错的 DSA 应抛弃这些服务差错，并且使用PartialOutcomeQualifier 的组件unavailableCriticalExtension，正如在 GB/T 16264.3—2008 中描述的那样。

15.5.3 别名解除引用

别名解除引用是创建一个新的目标客体名(称)的过程，它使用别名条目的AliasedEntryName 的属

性值来替换原始目标客体名的别名条目的可辨别名部分。在操作中的object 名(称)不受别名解除引用的影响。

15.5.4 解析上下文变体的名(称)

在名(称)解析阶段,当 RDN 被处理时,通过确保 RDN 中的每个AttributeTypeAndDistinguishedValue 都使用该属性的主可辨别值作为其 value,则可以创建一个新的目标客体名。通过这种方法,目标客体名被处理为一个主可辨别名。这样做是为了提供一致的名(称)处理,尤其是在名(称)解析中可以包含第 3 版本之前的 DSA 时。在操作中的object 名(称)不受该替代的影响。

15.5.5 分页结果

当 DUA 在搜索或列表操作的请求(见 GB/T 16264.3—2008 的 7.9)中包含PagedResultsRequest 时,分页可以是由直接与 DUA 绑定 DSA 来执行的,这种 DSA 又被称为绑定 DSA;或者可以是由拥有搜索或列表操作请求中的baseObject/object 条目的 DSA 来执行的(可以在一个或多个别名解除引用之后),这样的 DSA 又被称为初始执行者。如果分页是由绑定 DSA 执行的,该 DSA 也可以是初始执行者,则这样的分页被称为绑定 DSA 的分页结果。如果分页是由初始执行者执行的,且初始执行者不同于绑定 DSA,则这样的分页被称为 DSP 分页结果。

一个支持 DSP 分页结果的 DSA 应:

——支持绑定 DSA 的分页结果;

——作为一个绑定 DSA 时,支持 DSP 分页结果;

——作为一个初始执行者时,支持 DSP 分页结果;且

——支持PartialOutcomeQualifier 的子组件entryCount。

如果一个绑定 DSA 接收到一个包含PagedResultsRequest 的搜索或列表请求,且绑定 DSA 不是该请求的初始执行者时,则绑定 DSA 可选择在ChainingArguments 中包含参数dspPaging。初始执行者可选择完成 DSP 分页结果。这种情况会通过在PartialOutcomeQualifier 中包含一个queryReference 参数而向绑定 DSA 发出信号。返回给 DUA 的queryReference 将被用于检索下一页。

如果初始执行者不支持 DSP 分页结果,或者选择不去执行 DSP 分页结果,则绑定 DSA 可执行正常的绑定 DSA 分页。

一个作为执行者但不是初始执行者的 DSA,应忽略chainingArguments 中可能的dspPaging 组件,并且它应遵守服务控制sizeLimit,如果该服务控制存在的话。

15.6 分布式操作的鉴别

目录的用户,以及提供目录服务的管理机构可根据它们自己的决定,要求目录操作被鉴别。对于任意一个特定的目录操作,鉴别过程的特性依赖于当时生效的安全策略。

有两个鉴别规程的集合可用,两者联合起来共同满足一定范围内的鉴别需求。一个规程集是由绑定提供的:这些规程简化了两个目录应用实体之间为了建立一个联系而所需的鉴别。绑定规程包含了一定范围的鉴别交互,从身份的简单交换到强鉴别。

除了绑定所提供的建立联系时对等实体之间的鉴别外,在目录内还定义了另外的规程使得独立的操作也可被鉴别。定义了两个不同的目录鉴别规程集。一个是发起者鉴别服务,由一个 DSA 对原始服务请求的发起者进行鉴别。第二个是结果鉴别服务,由发起者对返回的任意结果进行鉴别。

对于发起者鉴别,定义了两个规程,一个是基于身份的简单交换,被称为基于身份的鉴别;另一个是基于数字签名技术,被称为基于签名的鉴别。前一个规程从本质上说是最初级的,因为身份交换是基于可辨别名的交换,而可辨别名是明文传送的。

对于结果鉴别,定义了一个单独的结果鉴别规程,是基于数字签名技术的;由于一般从本质上来说结果核对是复杂的,因此没有定义简单的,基于身份的规程。

差错响应的鉴别可被这些规程所支持。

下面所描述的服务被认为是对绑定服务所提供服务的增强;假设绑定规程在对目录操作进行鉴别之前就已经被成功实现了。

DSA 所实现的提供发起者鉴别和结果鉴别的规程在第 22 章规定。

16 操作调度程序

操作调度程序是 DSA 内主要的控制规程。它引导每个操作都经过请求处理的三个阶段。因此，操作调度程序使用一系列的规程来完全地处理请求，如图 6 所示。

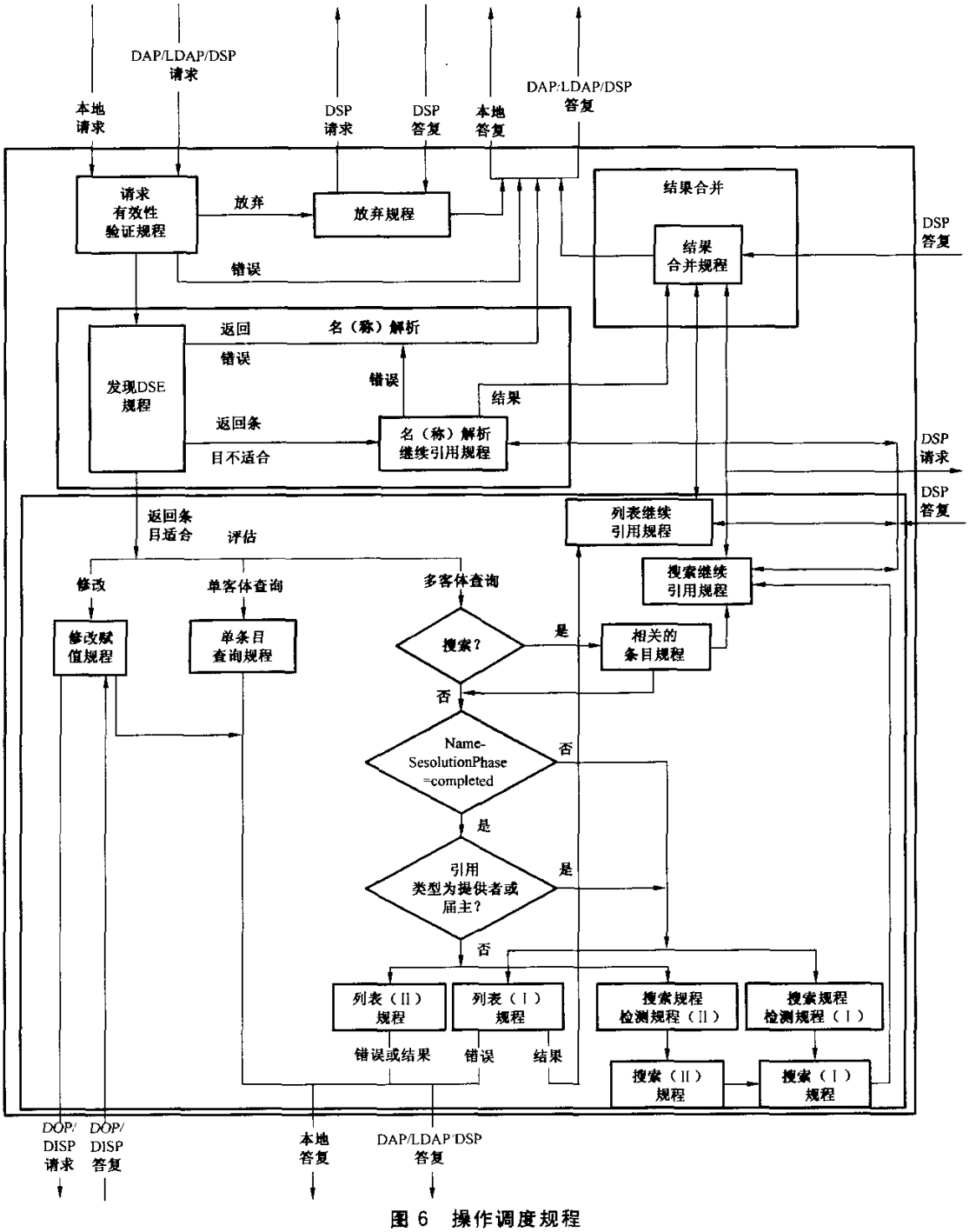


图 6 操作调度规程

16.1 通用概念

16.1.1 规程

操作调度程序所部署的每一个规程都包含一个概念性的接口定义，这是根据其参数来定义的，即变元、结果和差错等；同时还包含一个关于规程步骤本身的描述。这些规程的行为通过流程图和文字来描述。在一个流程图内所使用的符号具有如下语义(见图 7)。

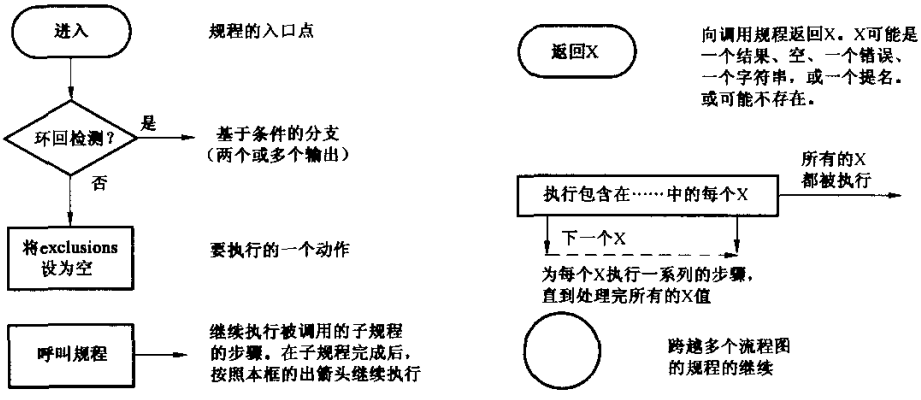


图 7 流程图中使用的符号

16.1.2 公共数据结构的使用

所有的规程都使用一些数据结构，这些数据结构在操作调用程序内，在某个操作的处理过程中是可用的。这些数据结构用于协调操作调用程序内的数据流。大多数这些数据结构都直接与操作的变元和为操作而产生的结果相关联。变元和结果组件的表示都是在相关的 ASN.1 定义中使用它们的名(称)来完成的(例如链接变元的operationProgress 组件)。如果这些结构中的任意一个是复合结构，则该结构的一个组件可被表示为compound.component (例如operationProgress.nameResolutionPhase)。

下面数据结构在操作调用程序中定义：

——NRcontinuationList：所创建的连续引用的列表，在名(称)解析连续引用规程中使用。

——SRcontinuationList：所创建的连续引用的列表，在列表或搜索连续引用规程中使用。

——admPoints：名(称)解析过程中收集到的类型为管理点的 DSE 的引用列表。

——referralRequests：由于运行转向推荐而被链接的请求或子请求的列表。每个这样的请求/子请求都以TraceItem 的形式进行概括。该列表被 15.4.2 中定义的环回避免规程使用。

——emptyHierarchySelect：一个布尔型的变量，其值可在等级选择规程中被设置。在一个搜索操作过程中，当第一次进入等级选择规程时，假设该变量被重置。

——streamedResultsOK：一个布尔型的变量，其值在名(称)解析规程中被设置，用来指示该操作可接收流结果。该变量的缺省值为 false。

另外，一个规程还可使用本地定义的一系列变量。

16.1.3 差错

在处理的每个阶段，执行任何子规程时，都可检测到差错。在本子规程中标识的差错一般会作为一个相应的协议差错返回给请求者。在这种情况下，操作调度程序将被立即终止。如果在接收到多个差错的情况下，本地规程可选择返回其中的一个。

可选的，一个规程也可选择在操作处理的某个点来处理差错(例如，如果一个带有问题为busy 的serviceError 被返回给一个链接搜索子请求)。在这种情况下，规程将继续它的运行，而不会有差错返回给请求者。

可选的，DSA 可基于所选择的DirQOP 和所请求的差错保护，对某个分布式操作中返回的差错进行签名、加密，或签名并加密。

16.1.4 异步事件

在操作调度程序内对某个操作请求的处理过程中，可发生多个异步事件。下面的段落规定了如何去处理时间限制或尺寸限制或管理限制的越限、联系的丢失，以及对一个正在进行的操作发起放弃请求等事件。所有其他异步事件的处理，如本地策略的决定等，不在本目录规范的定义范围之内。

16.1.4.1 时间限制

在CommonArguments 中规定的timeLimit 可以在操作过程中的任意时间点到期。在这种情况下，一般会有一个问题为timeLimitExceeded 的serviceError 返回给发出请求的 DUA、LDAP 客户机或DSA，且操作调度程序将被终止。可选的，一个规程可以选择某种不同的方式来处理本事件(如在一个搜索请求的处理过程中)。

如果一个 DSA 接收到从另一个 DSA 发来的请求，且超越了时间限制，则它应发送一个问题为timeLimitExceeded 的serviceError，而不再对请求进行任何后续处理。

如果一个 DSA 正在处理一个(子)请求，且当 timeLimit 到期时，尚无结果可用，则它应向请求者返回一个问题为timeLimitExceeded 的serviceError。

如果一个 DSA 正在处理一个子请求，且当timeLimit 到期时，有结果可用，则它应向请求者返回一个结果，并具有如下内容：

a) 一直到timeLimit 到期前，所有收集到的结果；

b) 结果参数partialOutcomeQualifier 中的limitProblem 组件应被设置为timeLimitExceeded；

c) 结果参数partialOutcomeQualifier 中的unexplored 组件应为每个 DSA 集包含一个连续引用值，这些 DSA 是子请求发向的目的地，但是它们的结果没有被包含在返回给请求者的结果中，该组件中还应当包括本 DSA 不愿意发送子请求的那些 DSA 的连续引用值。

16.1.4.2 联系丢失

如果与请求者之间的联系丢失，则所有可以返回的结果都会丢失。DSA 可选地为每个正在执行的查询(子)请求发送一个chainedAbandon 请求，除非与正在讨论的 DSA 的联系也丢失了。这些chained-Abandon 请求的所有响应，以及正在执行的(子)请求的所有响应都应被丢弃。在 DSP 分页结果的情况下，绑定 DSA 应当取消当前的分页结果，这是通过选择PagedResultsRequest 中的abandonQuery 选项而产生一个新的分页结果请求来实现的。

如果与当前链接的子请求的其中一个的联系丢失，但与请求者的联系尚未丢失，则仅对于查询操作而言，DSA 可以可选地对指向其他能够处理链接请求的 DSA 的任何可替代引用进行尝试(例如，当与主 DSA 的联系丢失后，可以尝试指向某个影像 DSA 的引用)。如果这样也不成功，则 DSA 应按照如下所述：

a) 如果operationProgress. nameResolution 的值被设置为notStarted 或proceeding，则或者向请求者返回一个问题为unavailable 的serviceError，或者返回一个转向推荐差错，其连续引用中包含能够继续处理操作的 DSA 集。如果在名(称)解析过程中使用了非特定下级引用，且不是所有的当前联系都丢失，则可选地可以尝试进行名(称)解析，但不包括联系丢失的 DSA。如果这样也失败，则返回一个问题为unavailable 的serviceError，或者返回一个包含 NSSRs 完整集合的转向推荐差错。

   如果使用本地知识的 DSA 已知需要链接到丢失了联系的 DSA 上，这可体现在适当的Master-OrShadowAccessPoint 值中，则它应选择发送一个问题为unavailable 的serviceError，且数据结构CommonResults 的notification 组件中应包含：

   ——一个dSAProblem 通知属性，且值为id-pr-targetDsaUnavailable；以及

   ——一个distinguishedName 属性，且值为 DSA 的可辨别名。

b) 如果operationProgress. nameResolution 的值被设置为completed，且该请求是一个单客体操作，则向请求者返回一个问题为unavailable 的serviceError。

c) 如果operationProgress. nameResolution 的值被设置为completed，且该请求是一个多条目查询操作，则 DSA 应在操作结果的partialOutcomeQualifier. unexplored 中增加一个连续引用，并携带AccessPointInformation 信息，用以标识可以继续处理该操作的 DSA 集，其中包括联系已经丢失了的那些 DSA。

16.1.4.3 放弃操作

在某个操作的处理过程中，可接收到对该操作的一个放弃请求。在这种情况下，在处理放弃请求的过程中，针对此要被放弃的操作，Abandon 规程被调用。

16.1.4.4 管理限制

可由本地 DSA 管理机构或者 DSA 实现本身施加某些限制，例如在处理一个请求时所花费的时间量，或者返回数据的最大尺寸等。如果这些限制中的任意一个被越限，则 DSA 应或者返回一个问题为administrativeLimitExceeded 的serviceError，或者返回一个部分结果(从已经收集的结果集中得到)，且limitProblem 被设置为administrativeLimitExceeded。

在通知属性dSAProblem 中应返回的附加信息包括如下：

a) 如果限制是由管理机构施加的，则通知属性dSAProblem 应取值为id-pr-administratorImposedLimit；

注：这并不意味着要求某种实现对实施管理限制的管理机构具有客户化的能力。

b) 如果限制是由某个实现约束引起的，且问题被认为是一个永久性的问题，则通知属性dSAProblem 应取值为id-pr-permanentRestriction；

c) 如果限制是由某个实现约束引起的，且问题被认为是一个临时性的问题，如临时拥塞，则通知属性dSAProblem 应取值为id-pr-temporaryRestriction。

16.1.4.5 尺寸限制

在CommonArguments 中规定的尺寸限制可在列表或搜索操作处理过程中的任意时间点被越限。在这种情况下，应向请求者返回一个部分结果(从已经收集的结果集中得到)，且limitProblem 被设置为sizeLimitExceeded。另外，可使用unexplored 组件来返回未访问到的 DSA 的连续引用。

如果这是一个搜索操作，且设置了entryCount 搜索控制选项，则 DSA 应作一个最佳估计，即在考虑访问控制但不是分等级选择时，如果没有尺寸限制，可以有多少个条目被潜在地返回，然后如果没有尚未访问到的 DSA，则使用bestEstimate 选择，并将此数值在PartialOutcomeQualifier 的entryCount 组件中返回，否则它应使用lowEstimate 选择。然后操作调度程序被终止。

16.2 操作调度程序的规程

操作调度程序为处理每个接收到的请求(通过 DAP、LDAP 或 DSP)而执行的规程由下面的步骤来定义。由于有别名解除引用，该规程也可以调用自己(一个本地请求)，在这种情况下，将返回一个本地答复(而不是一个 DAP、LDAP 或 DSP 答复)。

a) 对操作变元的多个方面进行有效性验证(请求有效性验证规程)。如果在有效性验证过程中遇到一个差错，则通过本地方式或通过 DAP/LDAP/DSP 返回此差错。

b) 如果接收到的操作是一个放弃操作，则调用放弃规程，并随后返回一个答复。

c) 通过执行发现 DSE 规程来对目标客体的名(称)进行解析(该规程中包括目标发现和目标未发现子规程)。如果被请求的条目被发现且适用(根据服务控制、链接变元和本地策略决策等的设置)，则继续第 f)步的赋值阶段。如果在名(称)解析过程中遇到一个差错，则被返回。如果条目被发现但不适用，继续执行步骤 d)。

d) 调用名(称)解析连续引用规程来处理NRcontinuationList 中存储的连续引用列表。为了处理这些连续引用，可向其他 DSA 发起链接请求(如果服务控制和本地策略决策允许的话)。
在出现差错的情况下，该差错或者通过本地方式，或者通过 DAP/LDAP/DSP 被直接返回。如果链接请求产生了一个结果，则继续执行步骤 e)。

e) 调用结果合并规程来将本地结果与接收到的链接结果合并起来。如果链接结果中包含内嵌的连续引用，则它们可以首先被解析，若服务控制和本地策略允许或要求的话。
这样可导致发起附加的链接请求(其链接结果也可以包含内嵌的连续引用)。
合并后的结果被返回给调用者，同时对请求的处理停止。
如果对结果执行了保护，则不能对结果执行合并。

f) 如果操作是一个修改操作，继续执行步骤 g)。
如果操作是一个单独条目查询操作，则继续执行步骤 h)。
如果操作是一个多条目查询操作，则继续执行步骤 i)。

g) 在执行一个修改规程时，作为操作执行的结果，操作绑定可需要被建立、修改或终止，或者影像可以需要被更新等。这些工作与初始操作的执行是同步进行的还是异步进行的，依赖于不同的修改操作(以及本地策略)。一个本地或 DAP/LDAP/DSP 结果或差错被返回给调用者。

h) 单独条目查询操作的结果，作为一个本地或 DAP/LDAP/DSP 结果被直接返回给调用者。

i) 如果操作是一个多条目查询操作，则检查操作的nameResolutionPhase。如果它的值不是completed，则分别调用列表(Ⅰ)规程或搜索(Ⅰ)规程，否则分别调用列表(Ⅱ)或搜索(Ⅱ)规程。

j) 调用列表(Ⅱ)规程的输出(结果或差错)以及调用列表(Ⅰ)规程的输出(在这种情况下，该结果为一个差错)能够直接返回给调用者(作为一个本地结果或 DAP/LDAP/DSP 结果)。
如果被调用的规程是列表(Ⅰ)规程，则结果中可以包含连续引用，该连续引用应被解除引用(依赖于服务控制和本地策略)。这可导致链接列表操作被发送到不同的 DSA。为了合并结果，继续执行第 e)步骤，并呼叫结果合并规程。

k) 如果操作是一个搜索操作，则任何连续引用都被搜索连续引用规程所解析(如果被要求或被允许的话)。这可导致链接搜索请求被发送到不同的 DSA。结果合并规程(见第 e)步骤)被调用来合并搜索结果，同时可以对包含的连续引用进行解除引用，如果有的话。

16.3 规程概述

本条对操作调度程序所部署的规程的基本功能给出了概述，这些规程在第 17 章到第 22 章中定义。

16.3.1 请求有效性验证规程

该规程在第 17 章描述，用于在执行本地名(称)解析之前执行环回检测、限制检测和安全检测等。如果请求是来自 DUA 或 LDAP 客户机，且 DAP 或 LDAP 客户机没有提供ChainingArgument 中的某些参数的值时，该规程还为这些参数提供缺省的设置。另外，该规程挑选任意的abandon 请求，并且将其通知给操作调度程序。

16.3.2 放弃规程

该规程在 20.5 描述，试图发现那些被放弃的操作，并且终止这些操作。如果有任意一个正在进行的子请求，则可以在发出请求后发送链接放弃操作。该规程可以向调用者返回一个空结果，或者返回一个差错指示(例如，问题为tooLate 的abandonError)。

16.3.3 发现 DSE 规程

该规程在 18.2 和 18.3 描述，对目标客体的名(称)组件与本地拥有的 DSE 进行匹配，以便解析目标客体名。如果遇到一个别名 DSE，则该别名被解除引用(如果允许的话)，且该规程被重新执行来解析新的名(称)。

如果目标客体没有被发现，则该规程继续执行目标未发现子规程。如果目标客体被发现，则该规程继续执行目标发现子规程。

注：目标未发现和目标发现是发现 DSE 规程的继续。

该规程可引起各种差错，在这种情况下，相关的协议差错将返回给请求者，且操作调度程序被终止。

16.3.3.1 目标未发现子规程

该规程在18.3.2中定义，对已定位的中介DSE执行赋值，并且基于在发现DSE规程中检测出的知识引用集，在NRcontinuationList中创建一个连续引用集。然后，该引用集在名(称)解析连续引用规程中被进行后续处理。

该规程可引起各种差错，在这种情况下，相关的差错将返回给请求者，且操作调度程序被终止。

16.3.3.2 目标发现子规程

该规程在18.3.3中定义，检查已经发现的DSE是否适合于所请求的操作，即在它是影像信息的情况下。在多客体操作的情况下(如子树搜索)，这可包括对目标客体下的整个影像信息子树的适宜性进行检查。

如果被定位的条目是适合的，则将调用适当的操作赋值规程。否则，一个指向信息的提供者(或属主)的ContinuationReference将在NRcontinuationList中创建，且名(称)解析连续引用规程被调用。

16.3.4 单条目查询规程

该规程在19.2描述，被调用来实际地执行那些仅影响一个单独条目的操作，如阅读和比较操作。在操作完成后，该规程产生的一个答复(结果或差错)将被返回给发起请求的DSA/DUA/LDAP客户机。

16.3.5 修改规程

这些规程在19.1描绘，被调用来处理修改操作，即增加条目、移除条目，修改条目和修改DN等。这是通过执行为每个这样的操作定义的特定子规程来完成的。在这些子规程过程中(或结束后)，可向其他DSA发起DOP和DISP请求。在成功完成后，一个结果(由子规程创建)被返回给发起请求的DSA/DUA/LDAP客户机。

16.3.6 多条目查询规程

这些规程在19.3描述，被调用来处理影响多个条目的操作，这些条目可以位于，也可不位于同一个DSA内。这是通过执行为了完成请求分解，而为每个搜索和列表操作定义的特定子规程来完成的。这些规程创建了操作赋值的一个本地结果，并且可选地，在SRcontinuationList中创建了一系列连续引用。如果在本规程结束时，SRcontinuationList为空，则被创建的结果将直接返回给请求的DSA/DUA/LDAP客户机。如果这是一个搜索操作，且结果为空并且变量emptyHierarchySelect被设置，则在PartialOutcomeQualifier的notification组件中返回：

——一个searchServiceProblem通知属性，且取值为id-pr-emptyHierarchySelection。

如果SRcontinuationList不为空，则根据操作类型，通过调用列表或搜索连续引用规程来处理这些连续引用。

16.3.7 名(称)解析连续引用规程

该规程在20.4.1中描述，对在名(称)解析阶段创建的NRcontinuationList中的连续引用进行处理。这些连续引用或者被用来发起链接子请求，或者在一个转向推荐差错中被返回。在链接的情况下，从链接请求返回的结果或差错被返回，以便由结果合并规程作进一步的处理。

16.3.8 列表和搜索连续引用规程

这些规程在20.4.2和20.4.3中描述，对在多条目查询规程中创建的SRcontinuationList中的连续引用进行处理，或者通过发起链接子请求来完成对它们的解析，或者是在partialOutcomeQualifier.unexplored内创建相应连续引用。当接收到所有正在进行的子请求的结果或差错时，它们将被返回，以便由结果合并规程作进一步的处理。

16.3.9 结果合并规程

该规程在第21章描述，或者检查从链接请求返回的结果，或者将本地操作结果与从链接子请

求中接收到的结果进行组合。如果一个子请求返回了一个差错，则该规程将决定此差错将被如何处理。

如果在结果中还留有连续引用，则它们将(如果本地策略允许这样，且服务控制要求这样)相应地被名(称)解析连续引用、列表连续引用或搜索连续引用等规程来解除引用。如果没有被签名的话，复制的信息将从结果中移除。

合并后的结果(包括所有的合并结果和未解析的连续引用)被返回给发起请求的 DUA/LDAP 客户机/DSA。如果对结果执行了保护，则不应对结果执行合并。

17 请求有效性验证规程

17.1 引言

请求有效性验证规程是操作调度程序对于来自 DUA、LDAP 客户机和 DSA 的输入的入口点，为这些输入进行名(称)解析处理作准备。该规程的功能包括检测放弃操作；执行安全检查；调整从 DUA 或 LDAP 客户机接收到的输入，使得它们可以与从 DSA 接收到的输入一样以相同的方法进行处理；检查请求中变元的有效语法和语义；执行环回检测；以及执行其他各种检查等。请求有效性验证的流程如图 8 所示。

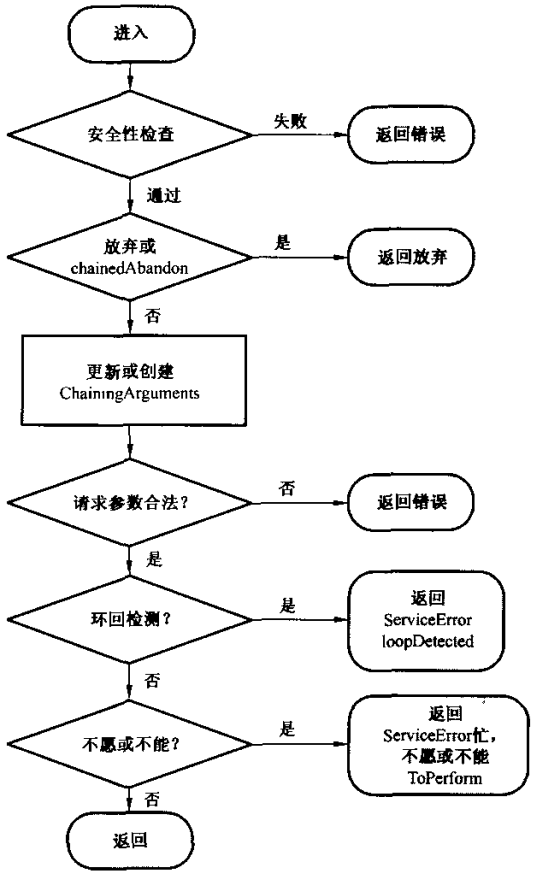


图 8 请求有效性验证规程

17.2 规程参数

17.2.1 变元

如果从 DSA 处接收到请求，以及请求的发起者所给出的变元等，请求有效性验证的输入变元包括 ChainingArguments（除了在chainedAbandon 操作的情况下）。

17.2.2 结果

请求有效性验证的输出结果包括五种可能性。

a) 如果安全检查失败，则向请求者返回一个差错。

b) 如果输入是一个abandon 或chainedAbandon 操作，则输出为该操作的变元。

c) 如果请求的变元非有效，则向请求者返回一个差错。依赖于本地策略，DSA 可以选择或者返回一个serviceError，或者返回一个securityError。

d) 如果检测到一个环回，则向请求者返回带有问题loopDetected 的serviceError。

e) 基于资源问题或策略方面的考虑，如果 DSA 不能或不愿意执行操作，则向请求者返回一个 serviceError（问题为busy，unavailable，或unwillingToPerform）。如果相关，则可以返回带有问题dataSourceUnavailable 的serviceError。

f) 在所有其他情况下，如果输入是从 DUA 或 LDAP 客户机接收到的，则有效性验证后的输入通过加入ChainingArguments 而被传送；如果输入是从 DSA 接收到的，则有效性验证后的输入通过更新ChainingArguments. traceInformation 而被传送，这些有效性验证后的输入是本规程的输出，并且随后将作为名(称)解析规程的输入。

17.3 规程定义

17.3.2 中描述的安全检查被执行。这可导致返回一个差错，并且终止该操作调度程序。

如果输入是一个abandon 或chainedAbandon 操作，则后续仅执行 17.3.1 中定义的步骤，否则 17.3.3～17.3.5 中定义的步骤都被执行。17.3.5 描述了环回检测规程，该规程可导致返回一个差错，并且终止该操作调度程序。

其次，17.3.6 中描述的检查被执行。它们可导致返回一个差错，并且终止该操作调度程序。

如果 17.3.2～17.3.6 中描述的检查没有导致操作调度程序的终止，则 17.3.7 中定义的步骤被执行，且其结果被传送到名(称)解析规程，本规程终止。

17.3.1 放弃处理

一个abandon 或chainedAbandon 的变元被传递到放弃规程(见 20.5)，来处理放弃请求。

17.3.2 安全检测

如果操作的变元被签名、加密或签名并加密，则签名可以被检测。如果签名非有效，或者加密失败，或者应存在的时候没有存在，则会向请求者返回一个差错。可选地，一个 DSA 还可执行任何其他本地定义的动作。

17.3.3 输入准备

17.3.3.1 DUA 或 LDAP 客户机请求

如果操作是从一个 DUA 或 LDAP 客户机处接收到的，则被创建ChainingArguments 值如下：

a) ChainingArguments. originator 按照 10.3 的描述被设置。

b) ChainingArguments. operationProgress 被设置为 CommonArguments. operationProgress 的值。

c) ChainingArguments. traceInformation 被设置为一个序列，该序列包含一个单独的TraceItem 值。该值的构造如下所述。TraceItem. dsa 被设置为执行请求有效性验证的 DSA 的名(称)。TraceItem. targetObject 应被忽略。TraceItem. operationProgress 被设置为输入值。

d) 如果操作的服务控制规定了一个时间限制(操作完成的可用时间段,以秒计数),则ChainingArguments. timeLimit 被设置为该(UTC)时间,到该时间为止操作应被完成以满足用户指定的时间限制。

e) ChainingArguments. AuthenticationLevel 和ChainingArguments. UniqueIdentifier 根据本地安全策略被设置。

f) ChainingArguments. nameResolveOnMaster 拷贝 CommonArguments. nameResolveOnMaster。

g) ChainingArguments. exclusions, ChainingArguments. entryOnly 和ChainingArguments. referenceType 拷贝自CommonArguments. exclusions, CommonArguments. entryOnly 和CommonArguments. referenceType,如果它们都存在的话,否则它们被忽略。

h) 如果manageDSAIT 选项在ServiceControls 中被设置,则:

——operationProgress 的组件nameResolutionPhase 应被设置为completed;

——operationProgress 的组件nextRDNToBeResolved 应被忽略;

——referenceType 应取值为self;

——entryOnly 应取值为FALSE;

——nameResolveOnMaster 应取值为FALSE;且

——ServiceControls 中的chainingProhibited 选项应被设置;

——ChainingArguments 中的剩余可选元素被忽略,如果指定则假设为缺省值。

i) 如果在ServiceControls 中没有设置manageDSAIT 选项,则ChainingArguments 中的剩余可选元素被忽略,如果指定则假设为缺省值。

j) ChainingArguments. SecurityParameters. ProtectionRequest 被用来指示准备应用于结果的保护级别(签名、加密、或签名并加密)。

17.3.3.2 LDAP 请求

如果操作是从一个 LDAP 客户机处接收到的,则创建的ChainingArguments 值如 17.3.3.1 中所描述的,但有一个例外,即ChainingArguments. operationProgress 应被设置为nameResolutionPhase notStarted,且 ChainingArguments. exclusions, ChainingArguments. entryOnly 和 ChainingArguments. referenceType 的值应被忽略。

17.3.3.3 DSA 请求

如果操作是从一个 DSA 处接收到的,则ChainingArguments. traceInformation 的值被更新,这是通过在TraceItem 序列的结尾处附加一个值实现的。该值的构成如下所述:

a) TraceItem. dsa 被设置为执行请求有效性验证的 DSA 的名(称);

b) TraceItem. targetObject 被设置为ChainingArguments. targetObject 的值,除非请求变元的 object(或搜索操作中的baseObject)与ChainingArguments. targetObject 相同,在这种情况下,TraceItem. targetObject 应被忽略;

c) TraceItem. operationProgress 被设置为ChainingArguments. operationProgress 的值。

如果操作是从一个 DSA 处接收到的,且如果ChainingArguments. streamedResults 包含一个大于或等于 1 的值,则当且仅当 DSA 理解流结果,并且愿意为此操作接收流结果时,才为ChainingArguments. streamedResults 的值加 1。

17.3.4 有效性声明

操作变元的语法和语义的有效性应被检查,这是根据在定义每个操作的章条中所包含的规则来实施的(例如,应检查nextRDNToBeResolved 不能提供一个超出了targetObject 中的 RDN 数量的数目)。

如果请求被检测出包含了非法变元，则操作被终止，并向用户返回一个差错，依赖于所检测出的非法类型。

17.3.5 环回检测

如果ChainingArguments. traceInformation 中的任意两个TraceItem 值(如 17.3.3 中准备的那样)是相同的，则操作的处理将返回到一个之前的状态，即检测出一个环回。在这种情况下，应向请求者返回一个serviceError (问题为loopDetected)，且操作调度程序终止。

17.3.6 不能或不愿执行

请求有效性验证可评估可用的资源，并决定操作不能被执行。它还可基于策略的考虑，决定操作不应当被执行。在这些情况下，可向请求者返回一个serviceError (问题为busy，unavailable，或unwilling-ToPerform)，且操作调度程序终止。

如果一个 DSA 通过本地方式能够判断出问题是与本地 DIB 资源的不可用相关的，则它应发送一个问题为unavailable 的serviceError，且数据类型CommonResults 的notification 组件中应包含：

——dSAProblem 通知属性，且取值为id-pr-dataSourceUnavailable；以及

——distinguishedName 属性，取值为 DSA 的可辨别名。

17.3.7 输出处理

在请求有效性验证的最后一个阶段，如果输入是从 DUA 或 LDAP 客户机接收到的，则有效性验证后的输入将通过加入ChainingArguments 中而被传送；如果输入是从 DSA 接收到的，则有效性验证后的输入将通过更新ChainingArguments. traceInformation 而被传送，这些有效性验证后的输入被返回，并且随后将作为名(称)解析规程的输入。

18 名(称)解析规程

18.1 引言

本章描述了名(称)解析规程，以及它的变元、结果和可能的差错条件。如图 6 中所示(操作调度程序)，名(称)解析规程包含两个规程：

——发现 DSE 规程；

——名(称)解析连续引用规程。

发现 DSE 规程在三个流程图中描述，分别为发现 DSE，目标发现和目标未发现。发现 DSE 规程将目标条目名与本地存储的 DSE 进行匹配，一个组件接着一个组件地进行。如果目标条目在本地发现，则发现 DSE 继续执行目标发现子规程，然后调用检查适宜性规程来检测所发现的 DSE 是否适合赋值。如果目标条目没有在本地发现，则发现 DSE 规程继续执行目标未发现子规程，准备要加入到NRcontinuationList 中的连续引用，然后由名(称)解析连续引用规程来调度。

注 1：当判断一个匹配时，名(称)解析应对多个由上下文所区分的可辨别值执行名(称)匹配，如在 GB/T 16264.2—2008 的 9.4 中描述的那样。

注 2：如果一个第 3 版本之前的上级 DSA 拥有一个下级引用，其指向的条目由一个后续版本的 DSA 所拥有，且该条目的 RDN 中包含有上下文，则名(称)解析可能会失败。当一个可替代名(称)被用作声称的名(称)，而影像条目由一个第 1 版本或第 2 版本的 DSA 所拥有时，则针对此条目的影像拷贝的名(称)解析会失败。

18.2 发现 DSE 规程参数

18.2.1 变元

本规程使用如下变元：

a) ChainingArguments. traceInformation；

b) ChainingArguments. aliasDereferenced；

c) ChainingArguments. aliasedRDNs;

d) ChainingArguments. excludeShadows;

e) ChainingArguments. nameResolveOnMaster;

f) ChainingArguments. operationProgress (nameResolutionPhase, nextRDNToBeResolved);

g) ChainingArguments. referenceType;

h) ChainingArguments. targetObject;

i) ChainingArguments. relatedEntry;

j) ChainingArguments. streamedResults;

k) 操作类型;

l) 操作变元。

注:如果没有实际值存在时,将使用缺省的或隐含的值,如 10.3 中的规定。

18.2.2 结果

有两种从Find DSE 成功的结果(分别由条目适合或条目不适合来指示):

第一种成功的结果是在NRcontinuationList 中返回连续引用(从目标未发现子规程中返回),然后该连续引用被传递到名(称)解析连续引用规程,继续执行名(称)解析阶段。

第二种成功的结果是返回一个指向 DSE 的引用(从目标发现子规程中返回),该引用被继续传递到某个赋值规程。

18.2.3 差错

下列差错可以被返回:

a) serviceError: unableToProceed, invalidReference, unavailableCriticalExtension, requested-ServiceNotAvailable;

b) nameError: noSuchObject, aliasDereferencingProblem, contextProblem。

18.2.4 全局变量

本规程使用下列全局变量:

——NRcontinuationList 列表存储了在名(称)解析连续引用规程中继续进行名(称)解析所需要的连续引用;

——StreamedResultsOK 存储了一个决定,即该 DSA 是否可以在此操作的响应中链接流结果。

18.2.5 本地和共享变量

该规程使用下列本地变量:

a) I 索引,用于标识正在被操作的目标名(称)的组件。

b) m 目标客体名的长度,在名(称)解析中使用。对于那些名(称)要解析到其父条目的操作,即增加条目操作,m 被设置为(目标客体中 RDN 的个数)−1。对于所有的其他操作,m 被设置为目标客体中的 RDN 的个数。

c) lastEntryFound 索引,表示此DSE(lastEntryFound)是最后一个匹配的类型为entry的 DSE。

d) lastCP 索引,表示此DSE(lastCP)是所遇到的最后一个被影像的上下文前缀。

e) candidateRefs 连续引用的集合。

共享变量admPoints (在操作调度程序中定义)也被使用。为了简便,目标客体名中的组件 i 表示为 N(i)。

18.3 规程

注：在流程图中有一些仅与特定操作相关的文字描述。这没有在流程图中显示，但是在相应的文字中进行了描述。

18.3.1 Find DSE 规程

见图 9。

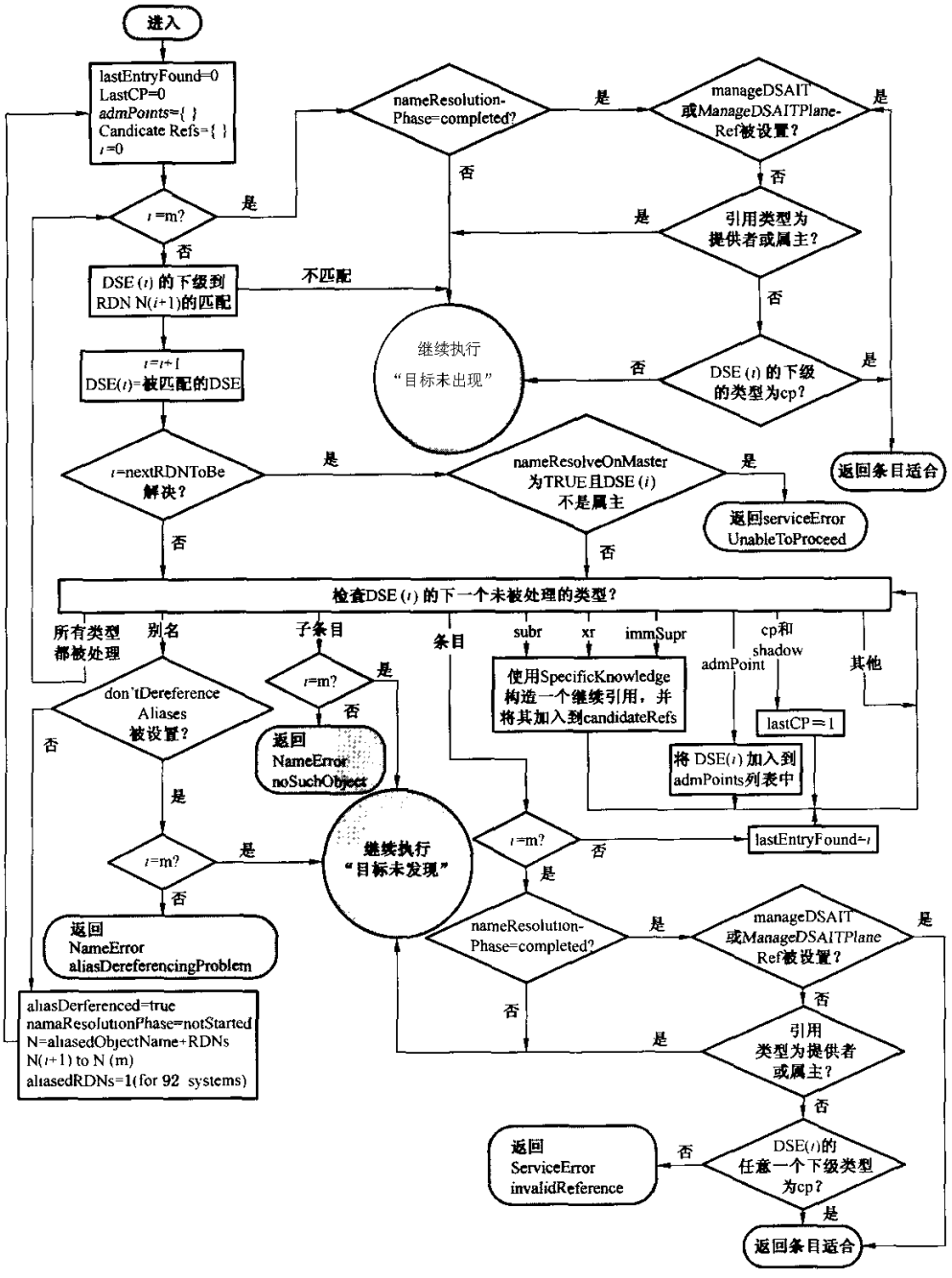


图 9 发现 DSE 规程

目标客体名的判断如下所述：

a) 如果在ChainingArguments 中存在targetObject，则使用该组件的值。

b) 如果在ChainingArguments 中存在relatedEntry，而不是targetObject，则使用relatedEntry 所标识的JoinArgument 中的baseObject 组件。

注 1：这仅与某个受保护的搜索请求相关。

c) 如果在ChainingArguments 中既没有存在relatedEntry，也没有存在targetObject，则使用操作变元中的base(baseObject)组件。

本规程试图在本地解析目标客体名。

a) 初始化本地变量lastEntryFound 和lastCP 为 0；admPoints 和candidateRefs 为空集，且初始化 i 为 0。

b) 比较 i 和 m。如果它们不相等，则继续执行步骤 5)。

c) 如果它们相等，则检查nameResolutionPhase 是否为completed。如果不是completed，则继续执行目标未发现子规程。

如果nameResolutionPhase 的值为completed，且manageDSAIT 关键扩展被设置，则返回条目适合。

d) 如果nameResolutionPhase 的值为completed，则检查 DSE(i)的任意一个直接下级是否是一个上下文前缀(类型为 cp)。

——如果一个(或多个)直接下级 DSE 的类型为 cp，则返回条目适合。

注 2：这种情况针对列表(Ⅱ)和搜索(Ⅱ)子请求。

——如果没有直接下级 DSE 的类型为 cp，则继续执行目标未发现子规程。

e) 尝试在目标客体名的第(i+1)个组件与最后匹配的 DSE 的某个下级名(称)之间找到一个匹配。在 i=0 的情况下，尝试与作为根 DSE 直接下级的某个 DSE 进行匹配。如果没有发现匹配，则继续执行目标未发现子规程。如果发现了一个单独的匹配，则 i 增加 1，并且在已发现的 DSE 的向量中增加该匹配的 DSE，作为第 i 个元素。

注 3：名(称)匹配包括对已知的，由上下文所区分的多个可辨别名的处理，如在 GB/T 16264.2—2008 的 9.4 中的描述。

如果发现了多个匹配，则返回一个问题为contextProblem 的nameError。

注 4：例如，可以是这样一种情况，当一个声称的名(称)中的AttributeTypeAndDistinguishedValue 包含了由上下文所区分的多个可辨别属性值，且这些不同的值与不同目标名(称)中的值相匹配。

f) 如果 i 等于nextRDNToBeResolved，则检查下面的两个条件是否都满足：

——ChainingArgument. nameResolveOnMaster 取值为TRUE；

——DSE(i)不是一个主条目。

如果这两个条件都满足，则返回一个问题为unableToProceed 的serviceError。

注 5：这指示了使用nameResolveOnMaster 可以避免对同一目标客体出现多条路径。

g) 检查 DSE(i)中的所有 DSE 类型比特。对于每一个类型比特，需要某些潜在的处理。为每种发现的类型要执行的动作如下所述：

——如果cp 和shadow 比特都被设置，则记住lastCP 中的索引 i。

——如果admPoint 比特被设置，则检查administrativeRole 操作属性。如果这是一个自治管理区的开始，则清空admPoints 列表。如果这是一个或多个特定管理区的开始，则检查

admPoints 列表，并删除任何已存在的但不再相关的点(即它们的作用已经被新的管理点所替代)。在列表中存储 DSE (i)。

——如果subr，xr，immSupr，或ditBridge 比特中的其中一个被设置，则产生一个连续引用，方法是使用specificKnowledge 属性，其中operationProgress. nameResolutionPhase 被设置为proceeding，nextRDNToBeResolved 被设置为 i，targetObject 是由使用主 RDN(可替代可辨别值可以也包含在 RDN 中)的已解析组件与剩余的尚未解析的组件级联起来构成的，且accessPoints 和referenceType 都被适当地设置。在candidateRefs 的连续引用列表中增加该连续引用。

——如果entry 比特被设置，则检查 i 是否等于 m(因此目标客体名正在被完整匹配)。如果 i 不等于 m，则通过将lastEntryFound 设置为 i 而记住此发现的条目，并且继续处理 DSE (i)的类型比特。如果 i 与 m 相等，则继续执行步骤 8)。

——如果subentry 比特被设置，则检查 i 是否等于 m(因此目标客体名正在被完整匹配)。如果它们相等，则继续执行目标发现规程；如果它们不相等，则返回一个问题为*noSuchObject* 的nameError。

——如果alias 比特被设置，则检查dontDereferenceAliases 是否被设置。

——如果dontDereferenceAliases 未被设置，则别名可被解除引用。因此，设置chainingArguments. aliasDereferenced 为TRUE，nameResolutionPhase 为notStarted，且目标客体名为别名条目所提供的aliasedEntryName 与前一个目标客体名中的剩余未匹配组件级联起来组成(即，与前一个目标客体名中的第 (i+1)个到第 m 个组件级联起来)。第 2 版本和后续版本的 DSA 不设置aliasedRDNs (反之，第 1 版本的 DSA 将aliasedRDNs 设置为 aliasedEntryName 中的 RDN 的个数)。再次开始名(称)解析，继续执行步骤 a)。

如果dontDereferenceAliases 被设置，则别名不能被解除引用。通过比较 i 和 m 是否相等，来检查目标客体名是否已经被完整地处理。如果它们相等(因此名(称)完全匹配)，则继续执行目标发现子规程。如果它们不相等(因此名(称)不是完全匹配)，则返回一个问题为aliasDereferencingProblem 的nameError。

——对于其他所有可能的 DSE 类型，不需要任何其他动作。在内部标注已经处理过的 DSE 类型，并且继续处理 DSE(i)中尚未处理的 DSE 类型比特。

——如果 DSE(i)的所有类型比特都被处理过，则继续步骤 b)。

h) 检查nameResolutionPhase 是否为completed。如果不是，则继续执行目标发现子规程。

i) 如果nameResolutionPhase 已经为completed，且manageDSAIT 关键扩展被设置，则返回条目适合。

j) 否则，检查作为 DSE(i)直接下级的任何 DSE 是否是一个上下文前缀(即类型为 cp)。如果有一个或多个，则返回条目适合。如果没有直接下级条目的类型为上下文前缀，则返回一个问题为invalidReference 的serviceError。

注 6：这种情况是针对列表(Ⅱ)和搜索(Ⅱ)子请求。

18.3.2 目标未发现子规程

见图 10。

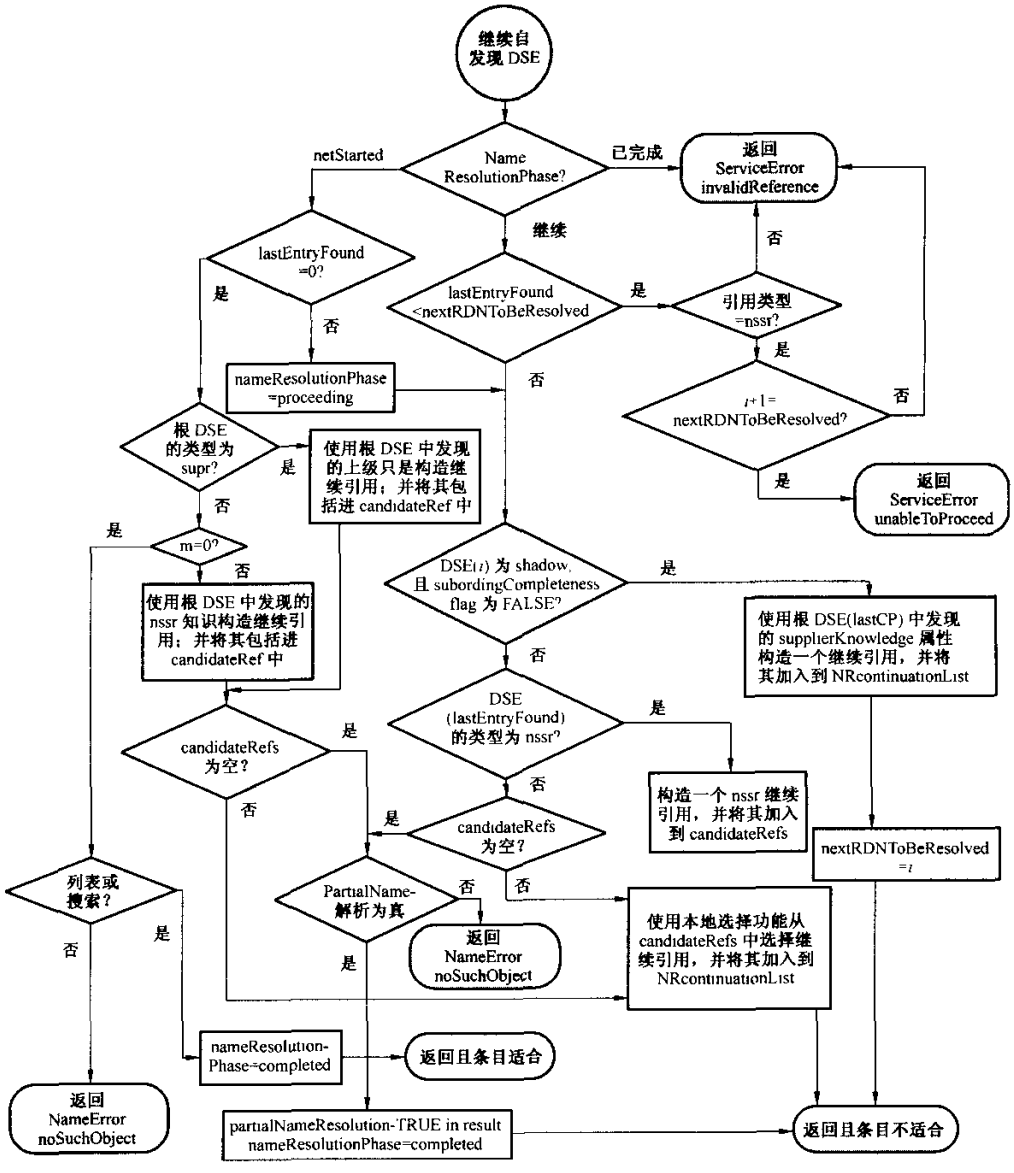


图 10　目标未发现子规程

当目标客体名没有在本地 DSA 中发现时，调用本子规程。本子规程将决定用来继续进行名(称)解析的知识引用的最佳类型，除非检测到一个差错，在这种情况下，将返回差错。

a) 当继续自发现 DSE 规程时，要区分名(称)解析阶段的三个可能的阶段：

——如果nameResolutionPhase 为notStarted，则继续执行步骤 b)。

——如果nameResolutionPhase 为proceeding，则继续执行步骤 h)。

——如果nameResolutionPhase 为completed，则继续执行步骤 l)。

b) 如果发现了一个条目(lastEntryFound 不等于 0)，则设置nameResolutionPhase 为 proceeding，并且继续执行步骤 i)。

c) 如果没有发现条目(lastEntryFound=0)，则检查该 DSA 是否为第一级的 DSA。

如果它是第一级的 DSA，则根 DSE 不包含一个上级引用，因此其类型不是 supr。在这种情况

下，继续执行步骤 d)。

如果该 DSA 不是第一级的 DSA，则根 DSE 包含一个上级引用，因此其类型为 supr。在这种情况下，使用在根 DSE 中找到的上级知识产生一个连续引用，设置：

——targetObject 为目标客体名，该名(称)是使用主 RDN(可替代辨别值也可包含在 RDN 中)的已解析组件与剩余的尚未解析的组件级联起来构成的；

——operationProgress. nameResolutionPhase 为notStarted；

——referenceType 为superior；以及

——适当的accessPoints。

在candidateRefs 的连续引用列表中增加该连续引用。继续执行步骤 f)。

d) 检查该操作是否被直接引导到根条目(m=0?)。如果是，则继续执行步骤 e)。如果不是，则使用根 DSE 中找到的任意 NSSR 知识产生一个连续引用，设置：

——targetObject 为目标客体名，该名(称)是使用主 RDN(可替代辨别值也可包含在 RDN 中)的已解析组件与剩余的尚未解析的组件级联起来构成的；

——operationProgress. nameResolutionPhase 为proceeding；

——operationProgress. nextRDNToBeResolved 为 1；

——referenceType 为nonSpecificSubordinate；以及

——适当的accessPoints。

在candidateRefs 的连续引用列表中增加该连续引用。继续步骤 f)。

e) 在第一级的 DSA 中，仅有列表或搜索操作可以将根条目作为基本客体来执行。因此，如果操作不是一个列表或搜索操作，则返回一个问题为noSuchObject 的nameError。如果是一个列表或搜索操作，则设置nameResolutionPhase 为completed，并返回，且为条目适合。

f) 检查在 candidateRefs 中是否有任意一个连续引用。如果 candidateRefs 为空且 partial NameResolution 为 FALSE，则返回一个问题为 noSuchObject 的 nameError。如果 candidate Refs 为空且 partialNameResolution 为 TRUE，则在结果中，设置 partialName 为 TRUE，nameResolutionPhase 为completed，并返回，且为条目适合。否则继续执行步骤 g)。

g) 使用一个本地选择功能从candidateRefs 的连续引用列表中选择一个连续引用，并将其增加到 NRcontinuationList 的连续引用列表中，并返回，且为条目不适合。

h) 如果 DSA 不能继续执行名(称)解析(在这种情况下，lastEntryFound 小于nextRDNToBeResolved)，则继续执行步骤 k)。否则继续下一步骤。

i) 如果 DSE(*i*)是一个影像 DSE，且具有不完整的下级知识(subordinateCompletenessFlag 取值为 FALSE)，则根据从DSE (lastCP)中找到的supplierKnowledge 属性产生一个连续引用。设置：

——targetObject 为目标客体名，该名(称)是使用主 RDN(可替代可辨别值也可包含在 RDN 中)的已解析组件与剩余的尚未解析的组件级联起来构成的；

——operationProgress. nameResolutionPhase 为proceeding；

——operationProgress. nextRDNToBeResolved 为lastEntryFound；

——referenceType 为supplier；以及

——适当的accessPoints。

在NRcontinuationList 的连续引用列表中增加该连续引用，并返回，且为条目不适合。

j) 如果最后发现的条目包含一个 NSSR(即DSE (lastEntryFound)的类型为nssr)，则根据从 DSE(lastEntryFound)中发现的 NSSR 知识产生一个连续引用。设置：

——targetObject 为目标客体名，该名(称)是使用主 RDN(可替代可辨别值也可包含在 RDN 中)的已解析组件与剩余的尚未解析的组件级联起来构成的；

——operationProgress. nameResolutionPhase 为proceeding；

——operationProgress. nextRDNToBeResolved 为lastEntryFound+1；

——referenceType 为nonSpecificSubordinate；以及

——适当的accessPoints。

在candidateRefs 的连续引用列表中增加该连续引用。继续步骤 h)。

如果 DSE(lastEntryFound)不是类型nssr，则继续步骤 g)。

k) 如果chainingArguments. referenceType 是类型nssr，则继续步骤 m)，否则继续步骤 l)。

l) 返回一个问题为invalidReference 的serviceError。

m) 如果 $i+1$ 等于nextRDNToBeResolved，则根据一个 NSSR，请求被引导到一个不能继续名(称)解析的 DSA；在这种情况下，返回一个问题为unableToProceed 的serviceError；否则继续执行步骤 l)。

18.3.3 目标发现子规程

当目标客体名与本地的某个条目 DSE 相匹配，则进入本子规程。本子规程检查所发现的条目是否适合在本地对请求进行处理(如图 11 所示)：

a) 调用检查适宜性规程。

b) 如果条目是适合的(即条目适合)，则执行下述动作：

——设置*nameResolutionPhase* 为*completed*；

——比较ChainingArguments. streamedResults 的值(如果存在)与ChainingArguments. traceInformation 中的元素个数；如果相等，则将StreamedResultsOK 设置为真；并且

——返回条目适合。

c) 如果条目是不适合的(即条目不适合)，则根据从DSE (lastCP)中发现的supplierKnowledge 属性产生一个连续引用。设置：

——targetObject 为目标客体名，该名(称)是使用主 RDN(可替代可辨别值也可包含在 RDN 中)的已解析组件与剩余的尚未解析的组件级联起来构成的；

——*operationProgress. nameResolutionPhase* 为*proceeding*；

——operationProgress. nextRDNToBeResolved 为 m；

——referenceType 为supplier；以及

——适当的accessPoints。

在NRcontinuationList 的连续引用列表中增加该连续引用。返回条目不适合。

注：如果设置了服务控制选项localScope，然而，基于本地策略，DSA 能够决定将此条目认为是适合的，并按照第 b)步骤来继续执行。

d) 如果不支持关键扩展(不支持的关键扩展)，则返回一个问题为unavailableCriticalExtension 的serviceError。

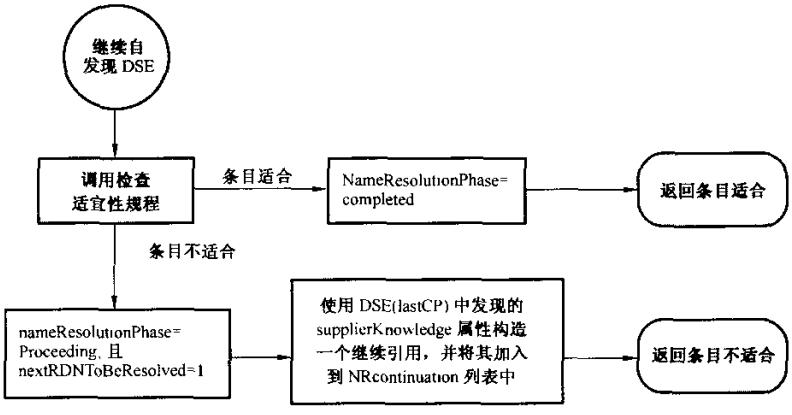


图 11 目标发现子规程

18.3.4 检查适宜性规程

调用该规程是用来决定一个发现的 DSE 是否适合执行所请求的操作(见图 12)。它考虑Chaining-Arguments,ServiceControls,用户所提供的变元,操作类型,以及 DSE 的特性(影像,下级知识,存在的属性等)。

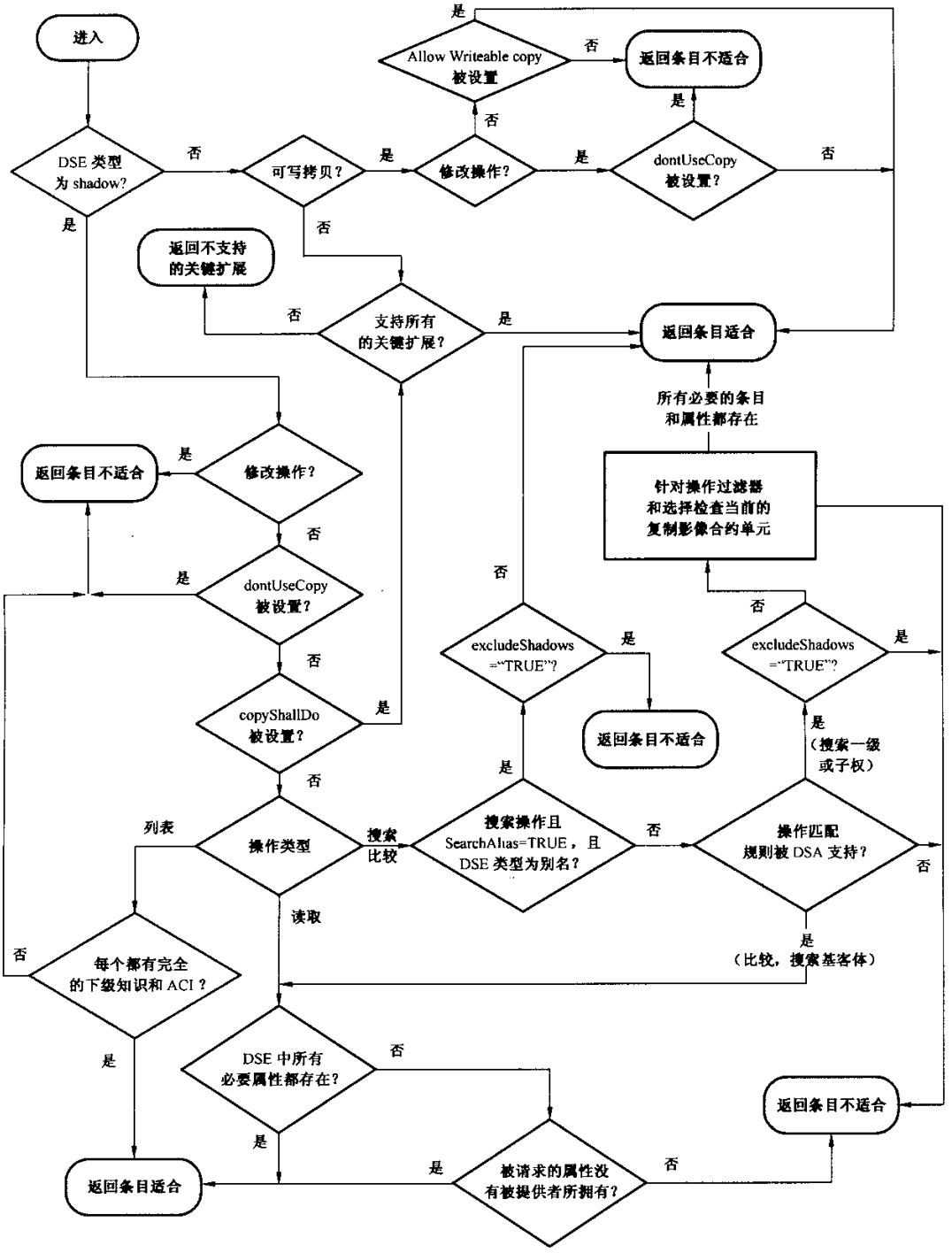


图 12 检查适应性规程

18.3.4.1 规程参数

本规程的输入变元为：

——一个指向某 DSE 的引用；

——操作类型，为了此操作类型，DSE 的适宜性将被检查；

——ChainingArguments；以及

——操作变元。

输出可以是条目适合，条目不适合，或者不支持的关键扩展。

a) 如果 DSE 不是 shadow 类型，也不是writeableCopy 类型，则检查是否所有的criticalExtensions 都被支持。如果是，则返回条目适合，否则返回不支持的关键扩展。

b) DSE 的类型为shadow。如果下面的任意一条为真，则返回条目不适合：

——被请求的操作类型为一个修改操作。

——服务控制dontUseCopy 被设置。

否则，继续下一步骤。

c) 如果 DSE 的类型为writeableCopy，且下面的任意一条为真，则返回条目不适合：

——被请求的操作类型为一个修改操作且服务控制dontUseCopy 被设置。

——被请求的操作类型为一个查询操作且服务控制allowWriteableCopy 没有被设置。

否则，返回条目适合。

d) 如果服务控制copyShallDo 被设置，则检查是否所有的criticalExtensions 都被支持。如果是，则返回条目适合，否则返回不支持的关键扩展。

e) 如果服务控制copyShallDo 没有被设置，则检查是否所有的criticalExtensions 都被支持。如果是，则转到步骤 e)，否则返回条目不适合。

f) 区分几类操作类型：

如果为列表操作，则继续执行步骤 f)。

如果为阅读操作，则继续执行步骤 g)。

如果为搜索或比较操作，则继续执行步骤 h)。

g) 如果条目具有完整的下级知识，则列表操作可以被执行。在这种情况下，返回条目适合，否则返回条目不适合。

h) 如果所有被请求的属性都出现在 DSE 中，则返回条目适合。如果某些属性缺失，则通过本地方式判断影像拷贝是否保存了属主所拥有的所有属性(例如，通过参考影像商定可以得知)。如果是，则条目是适合的(返回条目适合)。否则，提供者可以拥有影像中不存在的那些被请求的属性；在这种情况下，请求应被链接(返回条目不适合)。

i) 如果操作为search，且searchAliases 被设置为TRUE，同时 DSE 的类型为alias，则如果chaining Arguments. excludeShadows 为FALSE，则返回条目适合，如果为TRUE 则返回条目不适合。

j) 如果 DSA 支持所要求的用于比较或搜索的匹配规则，且操作为compare 或search 操作，其中 subset 为baseObject，则继续执行步骤 g)。如果 DSA 支持匹配规则，且操作为search，其中 subset 为oneLevel 或subtree，则继续执行步骤 j)。否则返回条目不适合。

k) 如果chainingArguments. excludeShadows 为TRUE，则返回条目不适合。否则，根据操作过滤器和选择来检查影像信息规范的本地理解。如果所有的必要条目和属性都存在，则返回条目适合。如果有任意条目或属性缺失，则返回条目不适合。

19 操作赋值(evaluation)

本章定义了如果(在名(称)解析阶段)已经在本地找到了某个操作的目标条目，则一个 DSA 应遵循的规程。

根据操作的类型，下面的规程之一将被调用：

——对于一个addEntry，chainedAddEntry，removeEntry，chainedRemoveEntry，modifyEntry，chainedModifyEntry，modifyDN或chainedModifyDN等操作，应遵循19.1中定义的规程。

——对于一个read，chainedRead，compare或chainedCompare操作，应遵循19.2中定义的规程。

——对于一个search，chainedSearch，list或chainedList操作，应遵循19.3中定义的规程。

19.1 修改规程

根据修改操作的类型，应遵循19.1.1到19.1.4中定义的相应规程。

19.1.1 增加条目操作

a) DSA应检查发起者是否有足够的访问权限，如同GB/T 16264.3—2008的11.1.5定义的那样。如果没有，则返回一个适当的差错。

b) DSA应确保要被加入的条目的名(称)所对应的条目尚不存在。否则，应返回一个问题为entryAlreadyExists的updateError。如果上级DSE具有附加的类型nssr，则DSA应遵循19.1.5(修改操作和NSSR)中定义的规程，以便确保新条目的名(称)是无二义性的。如果要加入的条目的名(称)中，最后一个RDN的某些属性包含了由上下文所区分的多个可辨别值，则DSA应确保在可能构建的可替代RDN中，没有一个会产生和已经存在的条目名相同的名(称)(不考虑上下文)。

c) 如果targetSystem存在，且AccessPoint不是当前DSA的访问点，则转到步骤d)。如果targetSystem不存在，或存在但AccessPoint是当前DSA的访问点，则转到步骤e)。

d) 如果该条目是一个子条目，则DSA应返回一个问题为affectsMultipleDSAs的updateError。如果该条目不是一个子条目，则DSA有一个本地选择，即它是否愿意与指定的DSA之间建立一个HOB。

如果不愿意，则DSA应返回一个问题为unwillingToPerform的serviceError，否则DSA应与指定的下级DSA之间建立一个分等级的操作绑定(HOB)。如果支持DOP，则应遵循24.3.1.1中定义的规程。否则，将使用本地方式建立该HOB。如果下级DSA不愿意建立此操作绑定，则addEntry操作将返回一个问题为unwillingToPerform的serviceError。如果HOB被成功建立，则继续执行步骤g)。

注1：规程中的本步骤不能应用于在一个下级DSA中创建自治管理区。

e) DSA应确保新的条目符合子模式，或者新的子条目或其他类型的DSE符合系统模式(例如一个子条目的直接上级DSE的类型为admPoint)。如果不符合，则DSA应返回一个适当的updateError或attributeError，否则它应加入此新的DSE。如果是条目，则继续执行步骤g)。如果是子条目，则继续执行步骤f)。否则，应当执行其他DSE类型的适当的知识管理规程。见第六篇。

f) DSA应在一个适当的时间点，将一个修改操作绑定前向到所有相关的下级DSA，本DSA与这些下级DSA之间具有分等级操作绑定或非特定的分等级操作绑定。相关的绑定是与作为上级DSE的下级的命名上下文相关的绑定。上下文前缀符合自治管理点的命名上下文是不相关的。如果支持DOP，则应遵循24.3.2.1和25.3.2中规定的规程。如果不支持DOP，则应使用本地方式来修改RHOB。

注2：一个适当的时间由DSA管理者来指定，其范围可以包括从操作结果刚刚返回(甚至在返回之前)一直到某个周期性策略(例如在某个指定的小时)。时间可根据修改的原因而变化，例如对ACI的更新会立即生效，而对模式的修改可周期性进行。

g) 如果增加的条目或子条目是在一个或多个影像商定的UnitOfReplication内，则影像消费者应被更新，方法是使用ISO/IEC 9594-9:2005中规定的目录信息影像服务规程。

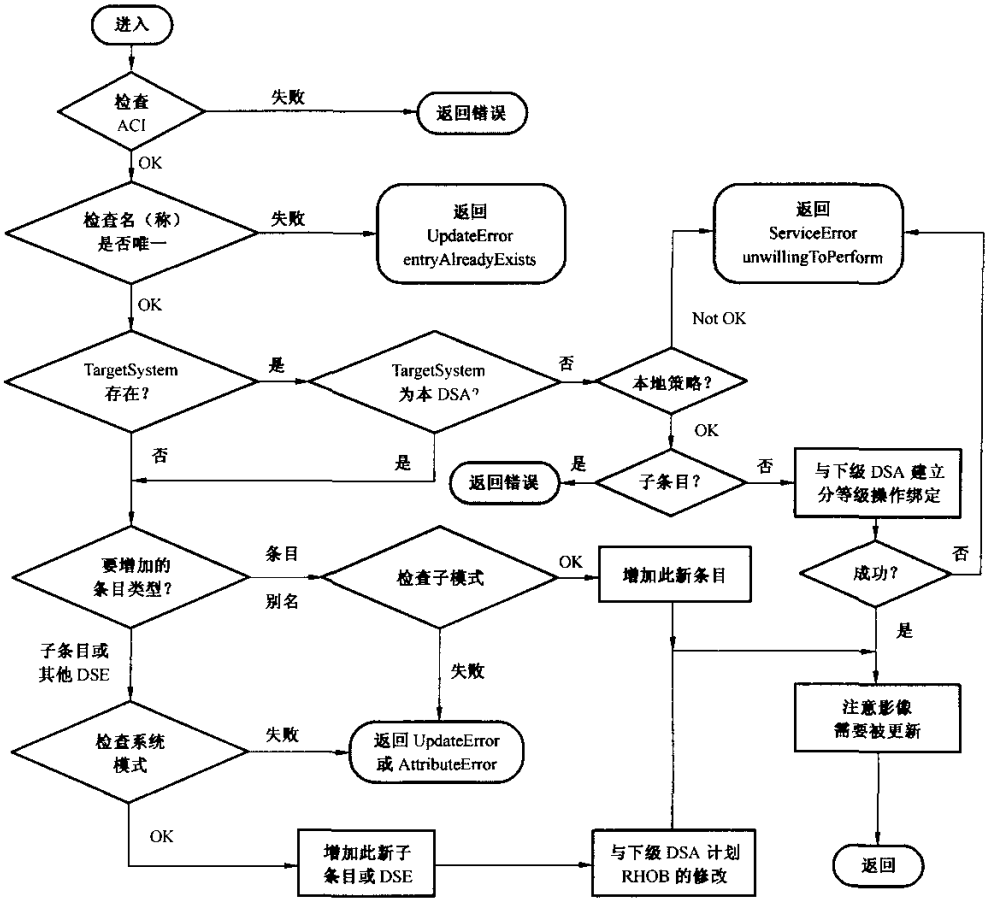


图 13 增加条目规程

19.1.2 移除条目操作

a) DSA 应检查发起者是否有足够的访问权限，如同 GB/T 16264.3—2008 的 11.2.5 定义的那样。如果没有，则返回一个适当的差错。

b) DSA 应确保要移除的条目是一个叶条目。否则，DSA 应返回一个问题为notAllowedOnNonLeaf 的updateError。

c) 要移除的条目的 DSE 类型被检查。如果是subentry，则继续执行步骤 e)。如果是 cp，则继续执行步骤 f)。如果是entry 或alias，则继续执行步骤 d)。否则，应当执行其他 DSE 类型的适当的知识管理规程。见第六篇。

d) 移除条目或别名条目，并继续执行步骤 g)。

e) 移除子条目。在一个适当的时间点，修改所有相关的下级 DSA 的操作绑定，本 DSA 与这些下级 DSA 之间具有分等级操作绑定或非特定的分等级操作绑定。相关的绑定是与作为上级 DSE 的下级的命名上下文相关的那些绑定。

上下文前缀符合自治管理点的命名上下文是不相关的。如果支持 DOP，则应遵循 24.3.2.1 和 25.3.2 中规定的规程。否则应使用本地方式。继续执行步骤 g)。

f) 移除命名上下文。如果该 DSA 有一个此命名上下文的分等级操作绑定，则它应终止与其直接上级 DSA 之间的分等级操作绑定。如果该 DSA 有一个此命名上下文的非特定分等级操作绑

定，且这是非特定分等级操作绑定的最后一个命名上下文，则它应终止与其直接上级 DSA 之间的非特定分等级操作绑定。如果支持 DOP，则应遵循 24.3.3.2 和 25.3.3.2 中规定的规程。否则，应使用本地方式来终止 RHOB。

g) 如果已移除的命名上下文、条目、别名条目或子条目等是在一个或多个影像商定的 UnitOfReplication内，则影像消费者应被更新，方法是使用 ITU-T X.525 建议书 | ISO/IEC 9594-9：2005 中规定的目录信息影像服务规程。

如果直接上级 DSA 内(其 RHOB 被终止)已移除的下级引用或非特定下级引用，是在一个或多个影像商定的UnitOfReplication 内，则影像消费者应被更新，方法是使用 ITU-T X.525 建议书 | ISO/IEC 9594-9:2005 中规定的目录信息影像服务规程。

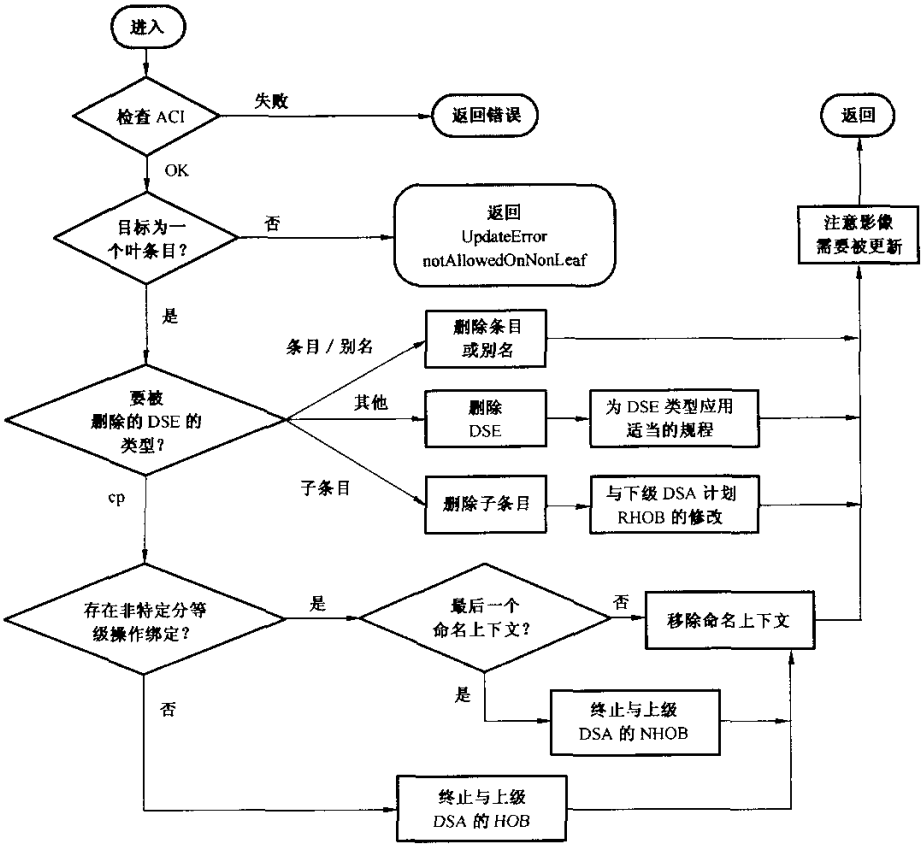


图 14 移除条目规程

19.1.3 修改条目操作

a) DSA 应检查发起者是否有足够的访问权限，如同 GB/T 16264.3—2008 的 11.3.5 定义的那样。如果没有，则返回一个适当的差错。

b) 对条目或别名的修改应符合子模式。对 DSE 的其他类型的修改，包括子条目，应符合系统模式。否则，DSA 应返回一个适当的updateError 或attributeError。在执行了修改之后，如果目标 DSE 的类型为subentry，则继续执行步骤 c)；如果目标 DSE 的类型为entry 或alias，则继续执行步骤 d)；

否则，应当执行其他 DSE 类型的适当的知识管理规程。见第六篇。

c) DSA 应在一个适当的时间点，修改与所有相关的下级 DSA 的操作绑定，本 DSA 与这些下级 DSA 之间具有分等级操作绑定或非特定的分等级操作绑定。相关的绑定是与作为管理点的下级的那些命名上下文相关的绑定，被修改的子条目位于此管理点之下。上下文前缀符合自治管理点的命名上下文是不相关的。如果支持 DOP，则应遵循 24.3.2.1 和 25.3.2 中规定的规程。否则，将使用本地方式。

d) 如果被修改的条目、别名条目或子条目等是在一个或多个影像商定的UnitOfReplication 内，则影像消费者应被更新，方法是使用 ISO/IEC 9594-9:2005 中规定的目录信息影像服务规程。

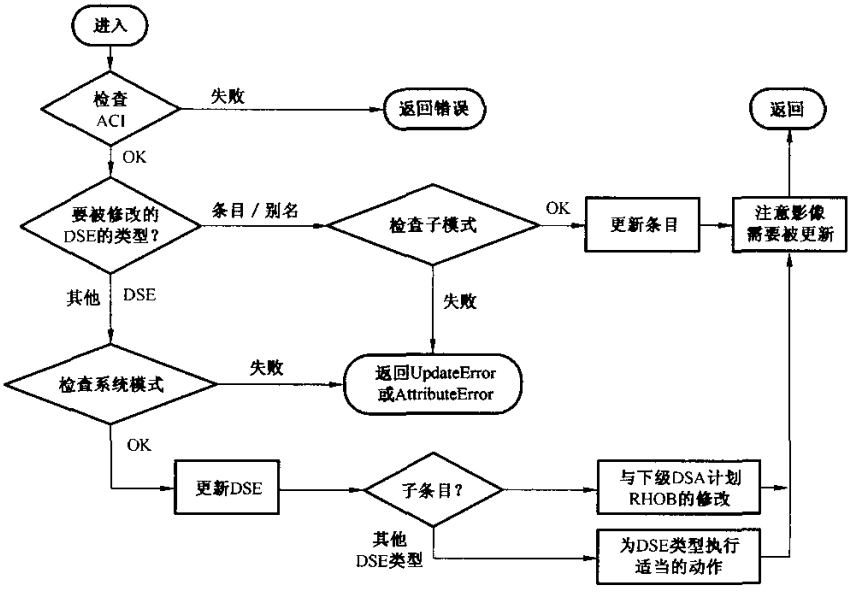


图 15 修改条目规程

19.1.4 修改 DN 操作

a) DSA 应检查发起者是否有足够的访问权限，如同 GB/T 16264.3—2008 的 11.4.5 定义的那样。如果没有，则返回一个适当的差错。

b) 如果操作是移动一个条目，或者是既移动条目又修改其相对可辨别名，则转到步骤 c)。如果操作仅仅是修改一个条目的相对可辨别名，则转到步骤 d)。

c) 操作应按照 GB/T 16264.3—2008 的 11.4.1 中的定义来执行。如果旧的上级、新的上级、本条目或其任意下级中的一个不在本 DSA 内，或者如果新的上级有 NSSR，则该操作应被拒绝，且返回问题为affectsMultipleDSAs 的updateError。DSA 应确保不存在具有此新名(称)的其他条目。否则，它应返回一个问题为entryAlreadyExists 的updateError。DSA 应确保条目的新名(称)符合子模式。否则，它应返回一个适当的attributeError 或updateError。如果没有发生上述这些问题，则移动该条目(如果需要，则修改 RDN)，并且转到步骤 i)。

d) 下面的文字应用于修改条目的相对可辨别名，该条目可以是，也可以不是一个叶条目；而且可以在一个或多个 DSA 中具有或不具有一个或多个下级。对要重命名的条目的 DSE 类型进行检查。如果为subentry，则继续执行步骤 g)。如果为 cp，则继续执行步骤 f)。如果为entry 或 alias，则继续执行步骤 e)。

e) DSA 应确保具有此新名(称)的其他条目尚不存在。否则，应返回一个问题为entryAlreadyExists 的updateError。如果被重命名的条目的上级 DSE 具有附加的类型nssr，则 DSA 应遵循

19.1.5(修改操作和 NSSR)中定义的规程,以便确保条目的新名(称)是无二义性的。如果新名(称)中某个 RDN 的某些属性包含了由上下文所区分的多个可辨别值,则 DSA 应确保在可能构建的 RDN 中,没有一个会产生和已经存在的条目名相同的名(称)(不考虑上下文)。DSA 应确保条目的新名(称)符合子模式。否则,它应返回一个适当的attributeError 或updateError。重新命名条目或别名条目。如果条目是一个非叶条目,且有下级在其他的 DSA 中,则继续执行步骤 h),否则继续执行步骤 i)。

f) DSA 应确保命名上下文的新名(称)符合子模式;否则,它应返回一个适当的attributeError 或updateError。

如果 DSA 与上级 DSA 之间有一个 HOB,则下级 DSA 应在响应修改 DN 操作之前就尝试修改 HOB。上级 DSA 应在接受修改之前,确保没有具有新名(称)的其他条目存在。如果支持 DOP,则应遵循 24.3.2.2 中规定的规程。如果不支持 DOP,则 HOB 如何被修改,且如何检查新名(称)的唯一性等属于本地事务。如果 HOB 被成功修改,且命名上下文具有下级命名上下文在其他 DSA 中,则转到步骤 h);否则转到步骤 i)。如果 HOB 不能被修改,则返回一个问题为affectsMultipleDSAs 的updateError。

如果 DSA 与其上级 DSA 之间有一个关于该命名上下文的 NHOB,则如何检测出复制的条目不在本目录规范的定义范围之内。重命名此条目。如果命名上下文具有下级命名上下文在其他 DSA 中,则转到步骤 h);否则转到步骤 i)。

g) DSA 应确保子条目的新名(称)符合系统模式。否则,它应返回一个适当的attributeError 或updateError。DSA 应确保没有具有新名(称)的其他子条目已经存在。否则,它应返回一个问题为entryAlreadyExists 的updateError。

h) DSA 应在一个适当的时间点,修改与所有相关的下级 DSA 的操作绑定,本 DSA 与这些下级 DSA 之间具有分等级操作绑定或非特定的分等级操作绑定。相关的绑定指的是与作为重命名条目的下级的所有命名上下文相关的那些绑定,或者是与作为管理点的下级的命名上下文相关的那些绑定,此管理点的子条目被重命名。上下文前缀符合自治管理点的命名上下文是不相关的。如果支持 DOP,则应遵循 24.3.2.1 和 25.3.2 中规定的规程。否则,将使用本地方式来更新 RHOB。

i) 如果被重命名的命名上下文、条目或其任意下级、别名条目或子条目等是在 DSA 所拥有的一个或多个影像商定的*UnitOfReplication* 内,则影像消费者应被更新,方法是使用 ITU-T X.525 建议书 | ISO/IEC 9594-9:2005 中规定的目录信息影像服务规程。

如果条目、别名条目或子条目等是在 DSA 所拥有的一个或多个影像商定的UnitOfReplication 内,但重命名的条目、别名条目或子条目等的上级不在此UnitOfReplication 内,则影像消费者应被更新,方法是使用 ITU-T X.525 建议书 | ISO/IEC 9594-9:2005 中规定的目录影像服务规程;在这种情况下,被影像的条目及其所有下级都应被移除。

如果条目、别名条目或子条目等不在 DSA 所拥有的一个或多个影像商定的UnitOfReplication 内,但重命名的条目、别名条目或子条目等目前在此UnitOfReplication 内,则影像消费者应被更新,方法是使用 ITU-T X.525 建议书 | ISO/IEC 9594-9:2005 中规定的目录影像服务规程;在这种情况下,被影像的条目及其所有下级都应被影像。

如果处于直接上级 DSA 内的被重命名的下级引用(其 HOB 在上述的第 f)步被修改)是在一个或多个它的影像商定的UnitOfReplication 内,则影像消费者应被更新,方法是使用 ISO/IEC 9594-9:2005 中规定的目录信息影像服务规程。

如果与某个下级 DSA 绑定的一个 RHOB 的组件(在上述的第 h)步被修改)是在下级 DSA 所拥有的一个或多个影像商定的UnitOfReplication 内,则影像消费者应被更新,方法是使用 ISO/IEC 9594-9:2005 中规定的目录信息影像服务规程。

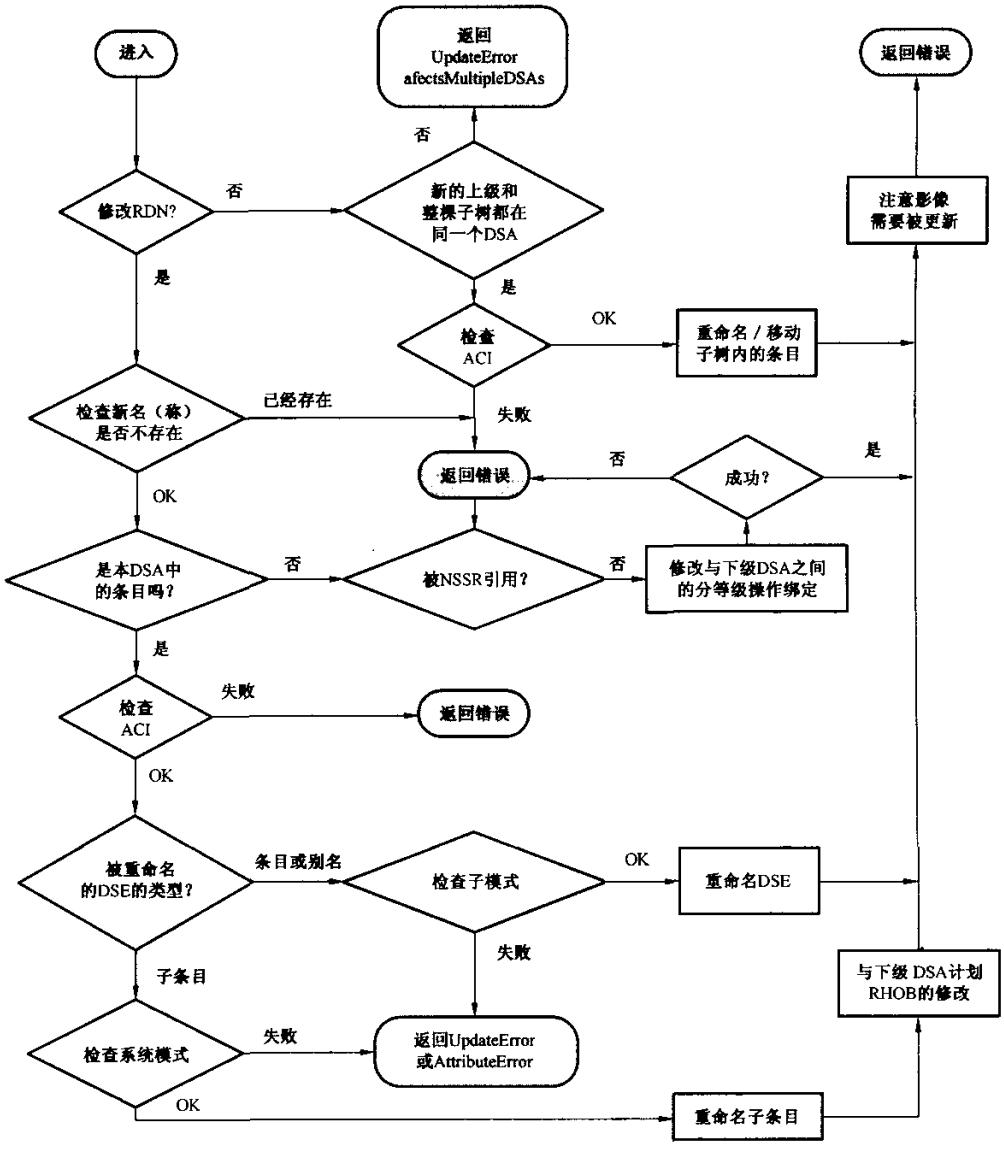


图 16 修改 DN 规程

19.1.5 修改操作和非特定下级引用

如果一个 DSA 具有 NSSR 且不知道某个条目的所有下级的完整名(称)集合，对于该 DSA，或者：

a) 有一个addEntry 操作已经被指向到该 DSA；或者

b) 有一个modifyDN 操作已经被指向到该 DSA。

则该 DSA 可以在执行操作前执行下列的规程集。

a) 如果chainingProhibited 服务控制选项在addEntry 或modifyDN 操作中被设置，则返回一个问题为affectsMultipleDSAs 的updateError。

b) 如果该 DSA 不愿意或不能够多链接出请求，则分别返回一个问题为unwillingToPerform 或 unavailable 的serviceError。

c) DSA 应将一个chainedReadEntry 操作多链接到 NSSR 的accessPointInformation 集合中的每个主 DSA 上(由于影像所引起的临时不一致性,DSA 应仅使用来自每个MasterAndShadowAccessPoints 的主 DSA)。ReadArgument 的参数应按照如下所述来设置:

object 或者设置为要增加的条目名(在addEntry 的情况下),或者设置为某个存在条目的声称名(在 modifyDN 的情况下)。

Selection 客体类属性。

CommonArguments 的参数应按照如下所述来设置:

——设置dontDereferenceAliases 服务控制选项;

——设置OperationProgress. nameResolutionPhase 为completed。

ChainingArguments 的参数应按照如下所述来设置:

——设置originator 为发起者的名(称);

——targetObject 被忽略;

——设置 OperationProgress. nameResolutionPhase 为 proceeding,且设置 nextRDNToBeResolved 为(客体名中的 RDN 数量)−1;

——设置traceInformation 为一个空序列;

——设置referenceType 为nonSpecificSubordinate;

——timeLimit,根据入请求进行适当地设置。

其他参数,例如SecurityParameters,可以被适当地设置,例如由本地策略来设置。

d) DSA 等待响应的完整集合。如果有任意一个响应是一个ReadResult,则应返回一个差错,如下面的 f)中所述。

e) 如果所有的响应是问题为unableToProceed 的serviceError,则操作赋值可继续进行。

f) 如果返回了一个ReadResult,则应为原始操作返回一个问题为entryAlreadyExists 的updateError;

g) 如果有任意一个其他的差错返回给readEntry 请求,则应返回一个问题为unwillingToPerform 的serviceError。

接收到chainedRead 请求的 DSA 应根据条目的存在与否,以及它的访问控制策略来给出一个响应。

19.2 单条目查询规程

阅读、ChainedRead、比较和ChainedCompare 等操作被划分为单条目查询规程组。这些规程仅包含下列三个步骤:

a) 检查访问控制,如同 GB/T 16264.3—2008 第 9 章中的描述。如果该操作不允许,则返回相应的安全差错。

b) 在发现的 DSE 上执行操作,如同 GB/T 16264.3—2008 第 9 章中的描述。

c) 准备答复,然后返回。

19.3 多条目查询规程

根据查询操作的类型(列表或搜索),应遵循 19.3.1 和 19.3.2 中定义的相应规程。

19.3.1 列表规程

本条规定了特定于*list* 和*ChainedList* 操作的评估规程。

当列表请求的operationProgress. nameResolutionPhase 组件被设置为notStarted 或proceeding,且当 DSA 在执行完名(称)解析后,发现它拥有基本客体时,应遵循列表(Ⅰ)规程。当列表请求的nameResolutionPhase 组件被设置为completed 时,应遵循列表(Ⅱ)规程。

19.3.1.1 **规程参数**

19.3.1.1.1 **变元**

本规程使用的变元为：

——ListArgument；

——目标 DSE e；

——chainingArgument 的operationProgress。

19.3.1.1.2 **结果**

如果本规程被成功执行，则它返回：

——一系列 e 的下级，在listInfo. subordinates 中；

——limitProblem，在partialOutcomeQualifier 中指示；

——一系列的连续引用，在SRcontinuationList 中。

19.3.1.2 **规程定义**

19.3.1.2.1 **列表(Ⅰ)规程**

列表(Ⅰ)规程包含下列步骤，如图 17 所示：

a) 如果服务控制subentry 被设置，则转到步骤 e)；否则转到步骤 b)。

b) 如果 DSE e 的类型为nssr，则向SRcontinuationList 中增加一个连续引用，其组件被设置如下：

——targetObject 被设置为 DSE e 的主可辨别名(可替代可辨别值也可包含在该 RDN 中)；

——aliasedRDNs 缺失；

——nameResolutionPhase 的 operationProgress 被设置为 completed，且 nextRDNtoBeResolved 缺失；

——rdnsResolved 缺失；

——referenceType 被设置为nonSpecificSubordinate；

——accessPoints 被设置为一系列accessPointInformation，每个值都源自 DSE e 的nonSpecificKnowledge 属性的一个值。

c) 对于作为 DSE e 直接下级的每个 DSE e'，执行下列步骤：

1) 检查 e'中的 ACI 是否可用。如果 ACI 不允许列出 e'的 RDN，则跳过该 DSE。如果 ACI 不可用(例如在下级引用和粘接的情况下)，则由本地策略来决定是否继续。

2) 检查 e'的所有 DSE 类型。

i) 如果 e'的类型为subr，则有两种情况。在第一种情况下，下级条目的 ACI 和客体类是本地可用的，在这种情况下，基于本地策略和 ACI 的准许，将 e'的 RDN 加入到listInfo. subordinates 中，其中aliasEntry 被设置为TRUE，如果 e'的类型为sa，则fromEntry 被设置为FALSE。另一种情况是条目的 ACI 在 e'中不可用，在这种情况下，向 SRcontinuationList 中增加一个连续引用，其组件被设置如下：

——targetObject 被设置为 DSE e 的主可辨别名(可替代可辨别名也可包含在 RDN 中)；

——aliasedRDNs 缺失；

——nameResolutionPhase 的operationProgress 被设置为completed，且 nextRDNtoBeResolved 缺失；

——rdnsResolved 缺失；

——referenceType 被设置为subordinate；

——accessPoints 被设置为包含在 DSE e'的specificKnowledge 属性中的值。

ii) 如果 DSE e'的类型为entry 或glue，则将 e'的 RDN 加入到listInfo. subordinates 中，其中 aliasEntry 被设置为FALSE，同时根据 e'是否是一个拷贝，而设置fromEntry 的值。

注：在 e'的类型是glue 的情况下，它应拥有一个或多个下级，这意味着它不能是主 DSA 中的一个别名。另外，任何与列表操作相关的 ACI 都存储在此 DSE 中，通过影像协议来提供。

iii) 如果 DSE e'的类型为alias，则将 e'的 RDN 加入到listInfo. subordinates 中，其中aliasEntry 被设置为TRUE，同时根据 e'是否是一个拷贝，而设置fromEntry 的值。

3) 检查时间、尺寸或管理限制是否被超越。如果被超越，则在partialOutcomeQualifier 中设置相应的limitProblem 并返回。

4) 继续执行步骤 c)中的 1)，直到所有的下级 DSE 都被处理过。

d) 如果所有的下级 DSE 都被处理，则返回到操作调度程序。

e) 对于作为 DSE e 直接下级的每个子条目 e'，执行下列步骤：

1) 检查 e'中的 ACI。如果 ACI 不允许列出 e'的 RDN，则跳过此 DSE。否则，将 e'的 RDN 加入到listInfo. subordinates 中，其中aliasEntry 被设置为FALSE，同时根据 e'是否是一个拷贝，而设置fromEntry 的值。

2) 检查时间、尺寸或管理限制是否被超越。如果被超越，则在partialOutcomeQualifier 中设置相应的limitProblem 并返回。

f) 返回到操作调度程序。

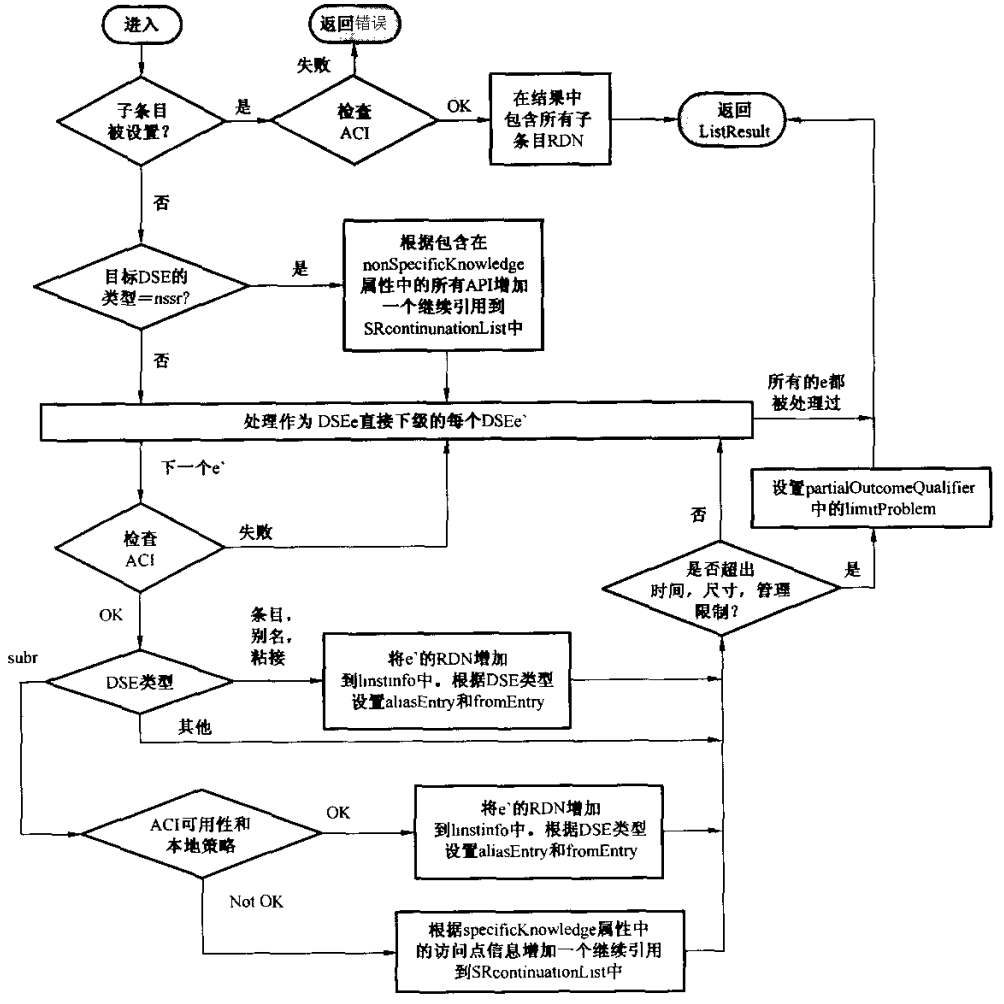


图 17 列表(Ⅰ)规程

19.3.1.2.2 列表(Ⅱ)规程

列表(Ⅱ)规程包含下列步骤,如图 18 所示:

a) 对于作为 DSE e 直接下级的每个 DSE e',执行步骤 a)中的 1)到 4):

1) 如果 e'不是一个条目或别名,则继续下一个直接下级。

2) 检查 e'中的 ACI。如果 ACI 不允许该操作,则继续 e 的下一个直接下级。

3) 将 DSE e'的 RDN 加入到listInfo. subordinates 中,其中根据 e'是否是一个别名来设置listInfo. subordinates 的aliasEntry 组件的值,同时根据 e'是否是一个拷贝,而设置fromEntry 组件的值。如果excludeShadows 为 TRUE,则忽略那些类型为 shadow 或 writableCopy 的 DSE。

4) 检查时间、尺寸或管理限制是否被超越。如果被超越,则在partialOutcomeQualifier 中设置相应的limitProblem 并返回。

5) 继续执行步骤 a)中的 1)直到所有的下级 DSE 都被处理过。

b) 如果所有的下级 DSE 都已经被处理,则检查该子请求是否来自 DAP 或 DSP。如果该子请求是通过 DAP 提交的,且ListResult 为空,则向操作调度程序返回一个问题为invalidReference 的serviceError。否则,返回ListResult。

注:当用户不能访问到上级条目时,invalidReference 被用作一个安全预警。如果上级的条目 ACI 可用(通过 RHOB 来提供),则如果允许的话,可返回一个空结果。

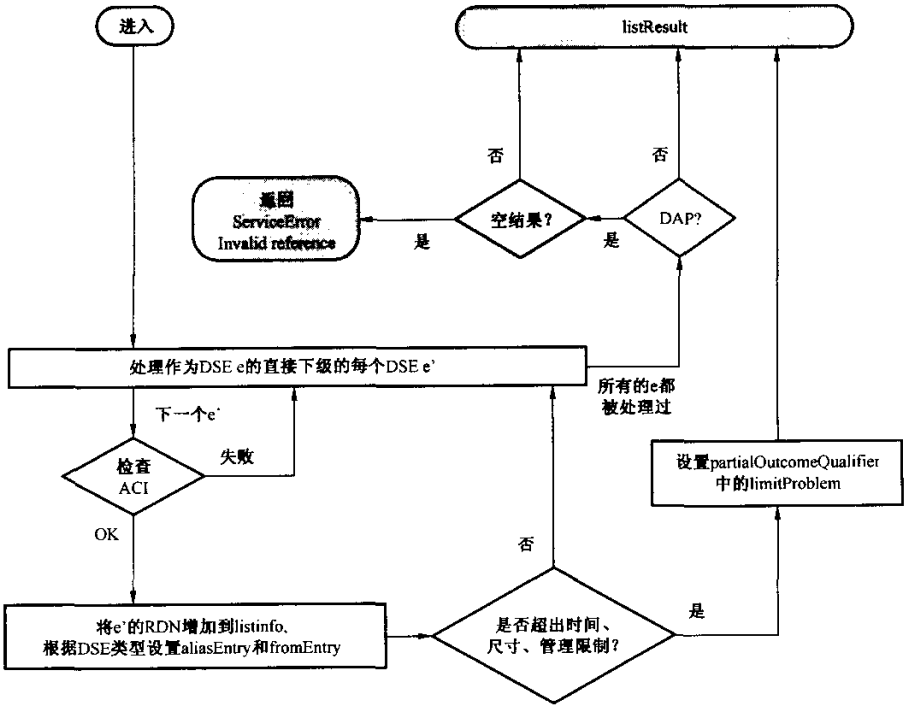


图 18 列表(Ⅱ)规程

19.3.2 搜索规程

本条规定了特定于Search 和Chained Search 操作的赋值规程。

当搜索请求的operationProgress. nameResolutionPhase 组件被设置为notStarted 或proceeding，且当 DSA 在执行完名(称)解析后，发现它拥有一个目标客体时，应遵循搜索规则检查(Ⅰ)规程。如果该规程返回了一个差错，则返回此差错。否则，应遵循Search (Ⅰ)规程。

当搜索请求的nameResolutionPhase 组件被设置为completed 时，应遵循搜索规则检查(Ⅱ)规程。如果该规程返回了一个差错，则返回此差错。否则，应遵循Search (Ⅱ) 规程。

注：若nameResolutionPhase 为completed，则期望目标客体是一个上下文前缀的直接上级。

19.3.2.1 **规程参数**

19.3.2.1.1 **变元**

本规程所使用的变元包括：

——SearchArgument；

——目标 DSE e；

——ChainingArguments 的operationProgress；

——ChainingArguments 的exclusions (要从搜索中排除出去的 RDN 列表)；

——ChainingArguments 的traceInformation；

——ChainingArguments 的searchRuleId；

——ChainingArguments 的chainedRelaxation；以及

——ChainingArguments 的relatedEntry。

19.3.2.1.2 **结果**

如果本规程被成功执行，它将返回：

——一系列匹配的条目，在searchResult. entryInformation 中；

——alreadySearched，在ChainingResults 中；

——根据条件的一个数目，在partialOutcomeQualifier. entryCount 中；以及

——一系列的连续引用，在SRcontinuationList 中。

19.3.2.2 **规程定义**

19.3.2.2.1 **相关的条目变元规程**

只有当搜索请求有一个joinArguments 组件，且ChainingArguments (如果有的话)没有relatedEntry 组件时，本规程才相关。

a) 如果搜索请求被保护，则为joinArguments 组件中的每个元素产生一个 DSP 请求，每个请求都包含了初始的 DAP 请求或 LDAP 消息。ChainingArguments 应如下所述：

——如果入请求有一个ChainingArguments，且带了originator 组件，则该组件的值将被拷贝到所产生的请求的originator 组件中；否则该组件的使用由本地安全策略所决定；

注：接收端 DSA 可能不能够使用本组件中给定的名(称)，如果它是来自一个不同的 DIT 的话。

——组件operationProgress 应被忽略或者被设置为缺省值；

——组件 traceInformation、aliasDereferenced、aliasedRDNs、returnCrossRefs、entryOnly、exclusions、nameResolutionOnMaster、searchRuleId、chainedRelaxation 等应被忽略；且

——组件relatedEntry 被设置为一个值，该值相对应于应用到 DSA 的JoinArgument 的相对位置，而该 DSA 是请求要前向的 DSA；当第一个JoinArgument 被给定值为 0 时，下一个值则为 1，依此类推。

b) 如果入请求没有被保护，则为joinArguments 组件中的每个元素产生一个 DSP 请求，此时 SearchArgument 应按照如下所述产生：

——组件baseObject 的值应拷贝自相应的JoinArgument 中的joinBaseObject 组件；

——组件subset 应拷贝自相应的JoinArgument 中的joinSubset 组件；

——组件filter 应拷贝自相应的JoinArgument 中的filter 组件；且

——剩余的组件应与原始请求中相应组件的值相同，除了joinArguments 和joinType 组件应被忽略。

ChainingArguments 应与上面被保护的请求的要求相同，除了relatedEntry 组件应被忽略以外。

c) 为每个要本地继续执行的请求调用操作调度程序。

d) 如果操作调度程序返回一个referral 差错，或忙，或不可用差错等，则在SearchResult 的partialOutcomeQualifier 中增加(或产生并增加)一个连续引用，并返回。

e) 如果操作调度程序返回其他差错，则丢弃并返回。

f) 如果操作调度程序返回一个SearchResult，则：

1) 如果结果被签名、加密或签名并加密，则将该结果增加到SearchResult 的uncorrelatedSearchInfo 中。

2) 如果结果没有被签名、加密或签名并加密，则根据 GB/T 16264.3—2008 中的规定执行合并过程。

19.3.2.2.2 搜索规则检查规程(Ⅰ)

当且仅当 DSA 支持服务特定的管理区时，本规程才相关。

如果searchRuleId 组件出现在ChainingArguments 中，则该操作是之前的赋值阶段中的一个别名解除引用规程的结果。然后，如果目标 DSE 处于一个具有不同dmdId 的服务特定管理区内，或者如果目标 DSE 处于某个服务特定管理区之外，则返回一个 unwillingToPerform 服务差错。否则基于searchRuleId 中的信息选择适当的搜索规则，并返回。

注 1：服务管理已经被定义为一个关键扩展。当一个不支持服务管理的 DSA 接收到一个链接搜索请求，且带有一个searchRuleId 组件时，它将返回一个问题为unavailableCriticalExtension 的serviceError。

如果 searchRuleId 组件不存在，且目标 DSE 处于某个服务特定管理区之外；或者如果它处于这样的一个管理区内但没有子条目与此管理区相关联，则返回。

如果目标 DSE 处于某个服务特定管理区内，且traceInformation 显示该操作已经在之前的赋值阶段出现过，则返回一个unwillingToPerform 服务差错。

注 2：这是这样一种情况，即一个搜索操作已经在某个服务特定管理区外启动了它的初始赋值，而目前准备扩展到一个不同的服务特定管理区内。

否则，应遵循下述规程：

a) 查找定位所有的与目标 DSE 相关联的搜索规则，即在服务子条目中，其子树规范内具有目标DSE 的所有搜索规则(例如通过使用searchRulesSubentry 操作属性)。这些搜索规则在后续被称为候选搜索规则。如果没有这样的搜索规则，则产生一个问题为requestedServiceNotAvailable 的服务差错，并在CommonResults 的notification 组件中，包括一个取值为id-pr-unidentifiedOperation 的searchServiceProblem 属性，并返回。

b) 如果在搜索请求中包含了serviceType 和/或userClass 服务控制，则从候选搜索规则中除去所有的与这些服务控制不符合的搜索规则。如果这样使得剩余的列表为空，则产生一个问题为requestedServiceNotAvailable 的服务差错；并在CommonResults 的notification 组件中，包括下列详述的信息，并返回：

——一个取值为id-pr-unidentifiedOperation 的searchServiceProblem 属性；

——如果在搜索请求中包括了serviceType 服务控制，则返回一个取值为该服务控制的ser-

viceType 属性。

c) 将候选搜索规则划分为四个列表(某些列表可以是空的):

——一个GoodPermittedSR 列表,包含了所有这样的候选搜索规则,即请求者具有对这些搜索规则的调用许可,且根据 GB/T 16264.3—2008 中第 13 章规定的搜索有效性验证规程,搜索请求符合这些搜索规则;

注 3:如果此列表非空,则没有理由去创建其他列表。

——一个MatchProblemSR 列表,包含了所有这样的候选搜索规则,即请求者具有对这些搜索规则的调用许可,且除了在一个或多个请求属性表中的matchingUse 外,搜索请求符合这些搜索规则;

——一个BadPermittedSR 列表,包含了所有这样的候选搜索规则,即请求者具有对这些搜索规则的调用许可,但搜索请求不符合这些搜索规则;

——一个DeniedSR 列表,包含了所有这样的候选搜索规则,即请求者不具有对这些搜索规则的调用许可。

d) 如果GoodPermittedSR 列表中包含了一个或多个空搜索规则,则使用本地算法选择这些空搜索规则中的一个作为控制搜索规则,然后返回。

e) 如果GoodPermittedSR 列表非空,则除了那些具有最高userClass 指示的搜索规则外,丢弃所有其他的搜索规则。

f) 使用本地算法,在GoodPermittedSR 列表中,选择剩余的搜索规则中的一个作为控制搜索规则,然后返回。

注 4:如果在上述列表中有多个搜索规则可供选择,则在实现时应当将事件记入日志,以便为管理之用,因为搜索规则的定义可需要改写。

g) 如果MatchProblemSR 列表非空,则根据与上述 e)和 f)中规定的算法类似的一种算法选择其中一个搜索规则;产生一个服务差错以及如 GB/T 16264.3—2008 的 13.4 详述的相关信息,并且返回。

h) 如果DeniedSR 列表为空,继续执行步骤 j);否则,将那些搜索请求不符合的所有搜索规则从列表中丢弃,并且丢弃所有的空搜索规则。如果此时列表为空,则继续执行步骤 j);否则产生一个问题为requestedServiceNotAvailable 的服务控制;同时在CommonResults 的notification 组件中包含下面详述的子组件,并且返回:

——一个取值为id-pr-unavailableOperation 的searchServiceProblem 属性;

——如果在DeniedSR 列表中的所有剩余搜索规则的serviceType 组件都具有相同的值,则包括一个取值为该值的serviceType 属性。

i) 如果BadPermittedSR 列表为空,则产生一个问题为requestedServiceNotAvailable 的服务差错;同时在CommonResults 的notification 组件中包含下面详述的子组件,并且返回:

——一个取值为id-pr-unidentifiedOperation 的searchServiceProblem 属性。

j) 对于 GB/T 16264.3—2008 的 13.1 定义的规程中的每个按顺序标号的项,根据BadPermittedSR 中剩余的搜索规则检查搜索请求,并且对于每个项:

——如果搜索操作符合该项中的某些搜索规则,但不是所有的搜索规则,则丢弃那些不符合的搜索规则;

——如果BadPermittedSR 中目前仅拥有一个搜索规则,则执行 GB/T 16264.3—2008 的第 13 章规定的规程,然后返回;

——否则,检查下一项。

k) 根据到此为止的规程，如果BadPermittedSR 中目前仅含有搜索操作不符合的那些搜索规则，则产生一个问题为requestedServiceNotAvailable 的服务差错；同时在CommonResults 的notification 组件中包含下面详述的子组件，并且返回：

——一个取值为id-pr-unidentifiedOperation 的searchServiceProblem 属性；

——如果BadPermittedSR 中的所有搜索规则都规定了同一个服务类型，则包含一个取值为该服务类型的serviceType 属性。

l) 对于 GB/T 16264.3—2008 的 13.2 定义的规程中的每个按顺序标号的项，根据BadPermittedSR 中剩余的搜索规则检查搜索请求，并且对于每个项：

——如果搜索操作符合该项中的某些搜索规则，但不是所有的搜索规则，则丢弃那些不符合的搜索规则；

——如果BadPermittedSR 中目前仅拥有一个搜索规则，则执行 GB/T 16264.3—2008 的第 13 章规定的规程，然后返回；

——否则，检查下一项。

m) 对于 GB/T 16264.3—2008 的 13.3 定义的规程中的每个按顺序标号的项，根据BadPermittedSR 中剩余的搜索规则检查搜索请求，并且对于每个项：

——如果搜索操作符合该项中的某些搜索规则，但不是所有的搜索规则，则丢弃那些不符合的搜索规则；

——如果BadPermittedSR 中目前仅拥有一个搜索规则，则执行 GB/T 16264.3—2008 的第 13 章规定的规程，然后返回；

——否则，检查下一项。

n) 产生一个问题为requestedServiceNotAvailable 的服务差错；同时在CommonResults 的notification 组件中包含下面详述的子组件，并且返回：

——一个取值为id-pr-unidentifiedOperation 的searchServiceProblem 属性；

——如果BadPermittedSR 列表中的所有搜索规则都指定了同一个服务类型，则包含一个取值为该服务类型的serviceType 属性。

19.3.2.2.3 搜索规则检查规程(Ⅱ)

当且仅当 DSA 支持服务特定的管理区时，本规程才相关。

如果searchRuleId 不存在，且目标 DSE 的所有直接下级条目(上下文前缀)都是服务特定管理点，则返回一个问题为unwillingToPerform 的serviceError。然而，如果某些下级条目不是服务特定管理点，则为搜索赋值选择相应的命名上下文并且返回。

如果searchRuleId 存在，目标 DSE 的每个下级条目都被检查以验证它是否与目标 DSE 处于同一个服务特定管理区内。如果不是，则相应的命名上下文从搜索中排除。如果还有搜索操作可以继续进行的剩余命名上下文(包含正在执行的 DSA 中的那些命名上下文)，在这些上下文中，则选择在searchRuleId 中指定的搜索规则并且返回。如果没有搜索操作可以继续进行的剩余命名上下文，则产生一个问题为unwillingToPerform 的serviceError 并且返回。

注：如果知识信息在 DSA 与拥有上级命名上下文的 DSA 之间是一致的，则后一种情况不可以发生。

19.3.2.2.4 条目信息选择

对于已经匹配的条目，以及被选择作为分等级选择中的一部分的条目而言，所选择的属性信息为下列信息的交集：

a) 由searchArgument. selection 所规定的信息，可以被缺省的上下文规范修改，对于已经匹配的

条目，还包括由searchArgument. matchedValuesOnly 所规定的信息；

b) 由控制搜索规则(如果有的话)所决定的信息。

该条目信息被加入到searchResult. entryInformation 中的条目列表中。

仅将尺寸(类型和所有的值)不大于attributeSizeLimit 的属性加入。

19.3.2.2.5 **搜索(Ⅰ)规程**

这是一个递归规程，应用于一个起始于某个给定的目标条目 e 的搜索请求。它搜索目标条目 e，然后处理作为 e 的直接下级的那些 DSE。当整个子树都需要被搜索时，该规程被自己递归调用。该规程包括如下步骤，如图 19 所示：

a) 如果 DSE e 的类型为 cp(即一个作为上下文前缀的 DSE)，则检查exclusions 变元中是否有任意一个元素是 e 的 DN 的一个前缀。

1) 如果是，则返回。

2) 否则，调用检查适宜性规程。

i) 如果 e 不适合，则按照如下所述构建一个continuationReference，并且将其加入到SR ContinuationList 中：

——targetObject 被设置为 DSE e 的 DN；

——nameResolutionPhase 的 operationProgres 被设置为 proceeding，且 nextRDNto BeResolved 被设置为 e 中的 RDN 的数量；

——continuationReference 中的所有其他组件都不变。

然后返回。

注 1：当一个搜索子请求被链接到一个影像提供者时，这是唯一的情况。换句话说，这样一个链接子请求的目标客体总是一个上下文前缀。

ii) 否则，将 e 的可辨别名加入到ChainingResults 的alreadySearched 中。

注 2：alreadySearched 仅包含上下文前缀。

b) 如果 e 的类型为alias，且SearchArgument 中的searchAliases 为TRUE，则调用搜索别名规程，然后返回。

c) 如果subset 是oneLevel，则继续执行步骤 f)。

注 3：e 在这个点上，不可能是不完整的下级，因为针对上下文前缀检查适宜性已经保证了这种情况不会发生。

d) 如果subset 为baseObject，或者如果entryOnly 为TRUE，则继续本步骤；否则转到步骤 e)。

如果下列情况之一是正确的：

1) e 的类型为subentry，且服务控制subentry 被设置；或者

2) e 的类型不是subentry，且服务控制subentry 未被设置，则执行下列步骤：

i) 检查 ACI。如果本操作不被允许，则返回。

ii) 将SearchArgument. filter 中指定的过滤器变元应用到 DSE e。确保对过滤器中使用的所有属性的访问是被准予的，如同 GB/T 16264.2—2008 中的定义。如果过滤器匹配，且根据分等级选择，条目未被排除，则按照 19.3.2.2.3 中的规定增加属性信息。

iii) 如果在搜索请求中包含hierarchySelection 搜索控制(可以被某个搜索规则规范所修改)，同时本条目是一个拥有多个成员的分等级分组中的一部分，且不仅self 指示被设置，则调用分等级选择(Ⅰ)规程。

然后返回。

e) 如果subset 为subtree (且entryOnly 不为TRUE)，并且下列中的一个是正确的：

1) e 的类型为subentry，且服务控制subentry 被设置；或者

2) e 的类型不是subentry，且服务控制subentry 未被设置，则执行下列步骤：

i) 检查 ACI。如果本操作不被允许，则转到步骤 f)。

ii) 将SearchArgument. filter 中指定的过滤器变元应用到 DSE e。确保对过滤器中使用的所有属性的访问是被准予的，如 GB/T 16264. 2—2008 中的定义。如果过滤器匹配，且根据分等级选择，条目未被排除，则按照 19. 3. 2. 2. 3 中的规定增加属性信息。

iii) 如果在搜索请求中包含hierarchySelection 服务控制（可以被某个搜索规则规范所修改），同时本条目是一个拥有多个成员的分等级分组中的一部分，且不仅self 指示被设置，则调用分等级选择（Ⅰ）规程。

iv) 继续执行步骤 f)。

f) 如果 e 的类型为nssr，则向SRcontinuationList 中增加一个连续引用，其组件如下所述：

——targetObject 被设置为 DSE e 的主可辨别名（可替代可辨别值可包含在 RDN 中）；

——aliasedRDNs 缺失；

——nameResolutionPhase 的operationProgress 被设置为completed，且nextRDNtoBeResolved 缺失；

——rdnsResolved 缺失；

——referenceType 被设置为nssr；

——accessPoints 被设置为AccessPointInformation，该值根据nonSpecificKnowledge 属性中发现的值推导而来。

g) 对处于目标 DSE e 的直接下级的所有 DSE e'进行处理，直到所有的下级 DSE 都处理完成。如果 e 是处于某个服务特定管理区内，则仅对那些作为同一服务特定管理区内的直接下级 DSE 才进行处理。如果 e 处于服务特定管理区外，则不应当对那些作为某一个服务特定管理区内的直接下级 DSE 进行处理。

在这个环回过程中，如果在searchResult. entryInformation 中的已匹配条目的列表超出了尺寸限制，或者时间限制或管理限制被超越，则在partialOutcomeQualifier 中设置相应的limit Problem，并返回。

注 4：在每次searchResult 被更新时，都隐含应用了对尺寸限制的检查。

1) 如果 DSE e'的类型为subr，不是 cp，且不表示一个下级条目，该下级条目是一个服务特定的管理点，则向SRcontinuationList 中增加一个连续引用，其组件如下所述：

——targetObject 被设置为 DSE e 的主可辨别名（可替代可辨别值可包含在 RDN 中）；

——aliasedRDNs 缺失；

——nameResolutionPhase 的operationProgress 被设置为completed，且nextRDNtoBeResolved 缺失；

——rdnsResolved 缺失；

——referenceType 被设置为subr；

——accessPoints 被设置为包含在 DSE e'的specificKnowledge 属性中的访问点信息。

注 5：如果 e'的类型既是 cp 又是subr，则能够潜在地生成一个搜索子请求，或者根据下级引用，或者根据上级知识，但不会两个都是。本规程使用后一种（即根据 cp 中发现的提供者引用）。

2) 对于所有的情况：

i) 如果subset 为oneLevel，则设置entryOnly 为TRUE。

ii) 为目标 DSE e'递归执行Search（Ⅰ）规程。

h) 如果所有的下级都已经被处理，则返回到操作调度程序以便执行进一步的处理。

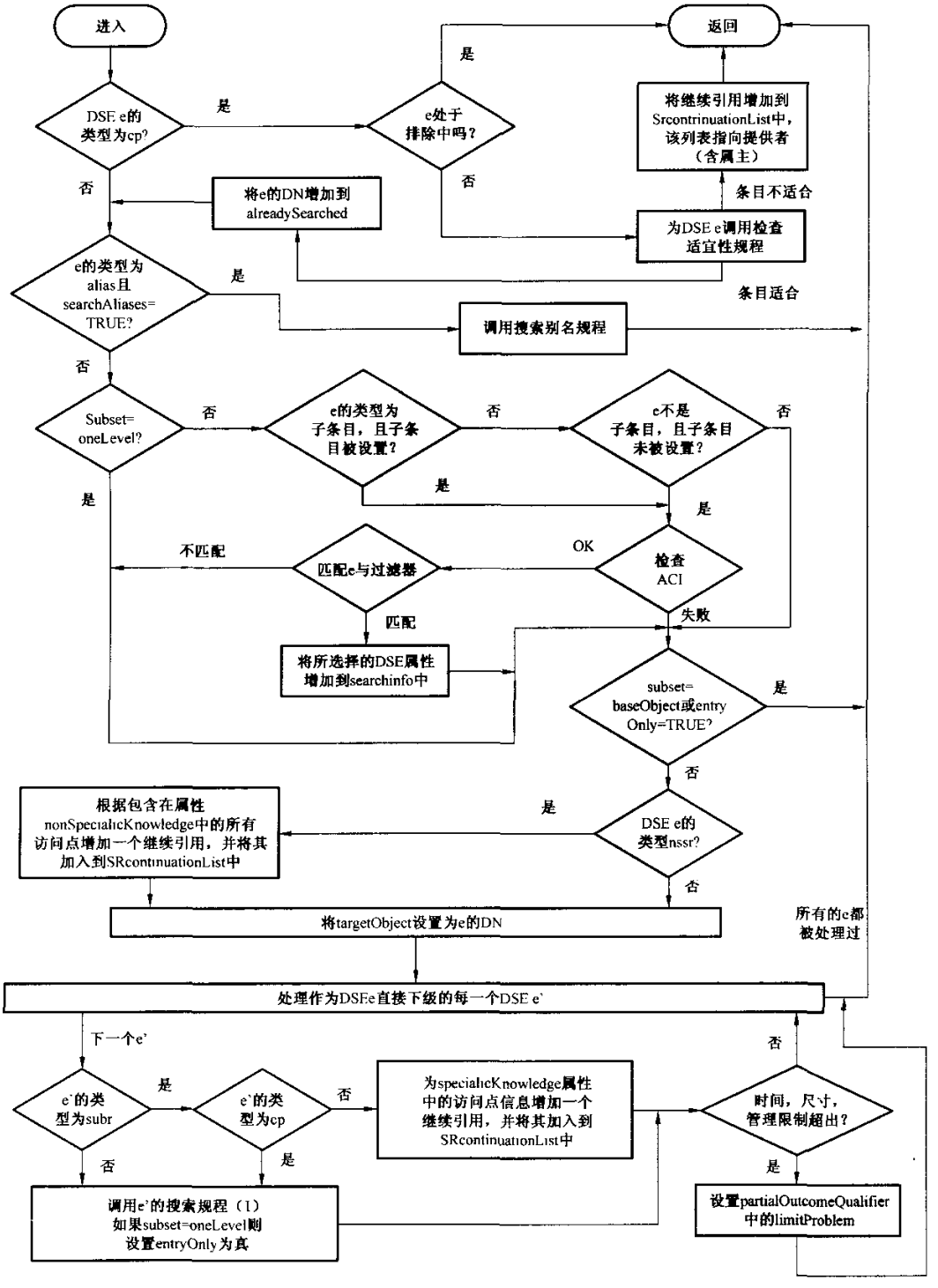

进入
返回
DSE e的类型为cp?
是
e处于排除中吗?
是
否
将继续引用增加到SrcontrinuationList中，该列表指向提供者（含属主）
否
将e的DN增加到alreadySearched
条目不适合
为DSE e调用检查适宜性规程
e的类型为alias且searchAliases=TRUE?
是
条目适合
调用搜索别名规程
否
Subset=oneLevel?
否
e的类型为子条目，且子条目被设置?
否
e不是子条目，且子条目未被设置?
否
是
是
是
不匹配
匹配e与过滤器
OK
检查ACI
匹配
失败
将所选择的DSE属性增加到searchinfo中
subset=baseObject或entryOnly=TRUE?
是
否
根据包含在属性nonSpecialicKnowledge中的所有访问点增加一个继续引用，并将其加入到SRcontinuationList中
是
DSE e的类型nssr?
否
将targetObject设置为e的DN
所有的e都被处理过
处理作为DSEe直接下级的每一个DSE e`
下一个e'
否
e`的类型为subr
是
e`的类型为cp
否
为specialicKnowledge属性中的访问点信息增加一个继续引用，并将其加入到SRcontinuationList中
时间，尺寸，管理限制超出?
否
是
是
调用e'的搜索规程（I）如果subset=oneLevel则设置entryOnly为真
设置partialOutcomeQualifier中的limitProblem

图 19　搜索（Ⅰ）规程

19.3.2.2.6 搜索(Ⅱ)规程

如果一个搜索请求是源自某个 DSA 处的一个请求分解,而该 DSA 是接收请求的 DSA,则当该请求被处理时,应用本规范。本规程处理目标 DSE e 之下的 DSE,并且为每个客体条目调用搜索(Ⅰ)规程:

a) 对处于目标 DSE e 的直接下级的所有 DSE e'进行处理,直到所有的下级 DSE 都处理完成。如果所有的下级都已经被处理,则返回到操作调度程序以便执行进一步的处理。

b) 如果 DSE 的类型不是 cp,则忽略该 DSE。返回到步骤 a)。

c) 调用检查适宜性。如果条目适合,则转到步骤 d);否则忽略此 DSE 并返回到步骤 a)。

d) 对 DSE e'执行搜索规程(Ⅰ),如 19.3.2.2 中的描述。如果 DSE 的类型为alias 且subset 参数的值被设置为oneLevel,则在调用搜索(Ⅰ)规程时,设置ChainingArguments. entryOnly 为 TRUE。返回到步骤 a)。

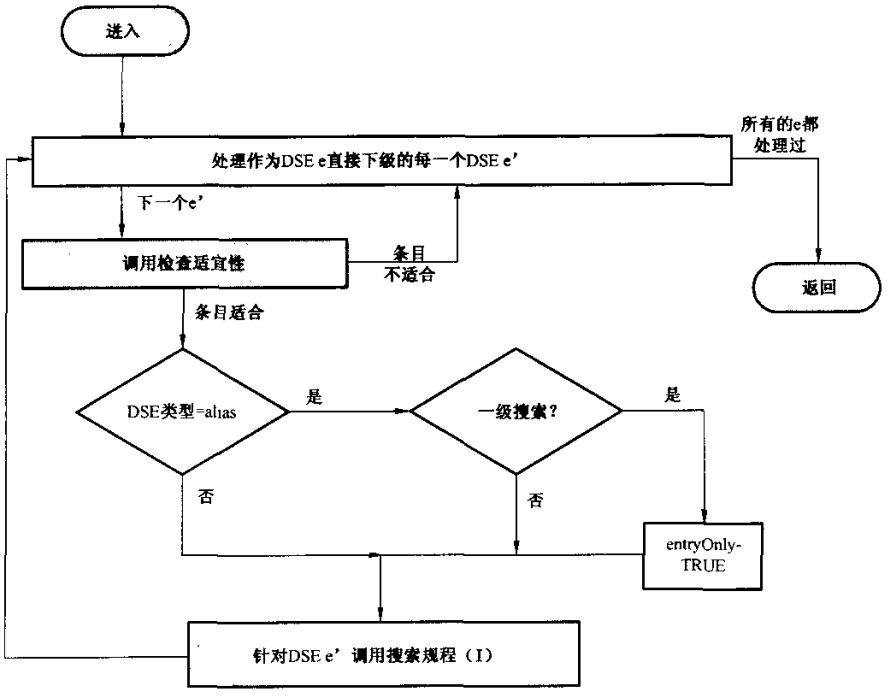


图 20 搜索(Ⅱ)规程

19.3.2.2.7 搜索别名规程

在一个搜索请求的处理过程中,当遇到一个类型为 alias 的 DSE 时,运行本规程(见图 21):

a) 如果subset 为baseObject 或oneLevel,则转到步骤 d)。

b) 如果 aliasedEntryName 是 targetObject 或 baseObject 的一个前缀,或者是 ChainingArguments. traceInformation 中的targetObject 的任意一个之前值,则该别名被排除出搜索操作,因为这会引起一个递归搜索导致复制的结果。

c) 如果targetObject 或baseObject 或ChainingArguments. traceInformation 中的targetObject 的任意一个之前值是aliasedEntryName 的一个前缀,则不需要对此别名作任何特殊的处理,因为无论如何被起别名的子树都将被搜索。

注:对于上述的两种情况,由于别名解除引用,baseObject 可以不是targetObject 的前缀。

d) 如果该搜索是在一个服务特定管理区内执行的,且如果服务特定管理点不是aliasedEn-

tryName 的一个前缀，则不需要对此别名作任何特殊的处理，因为被起别名的条目是在服务特定管理区之外。

e) 构建一个 DSP 请求，其中targetObject 被设置为aliasedEntryName。如果subset 为oneLevel，则设置entryOnly 为TRUE。然后为该请求调用操作调度程序，在本地继续执行。

f) 如果操作调度程序返回了一个referral 差错，或忙，或不可用差错，则在SearchResult 的partial OutcomeQualifier 中增加(或构建并增加)一个连续引用，然后返回。

g) 如果操作调度程序返回了其他差错，则丢弃并返回。

h) 如果操作调度程序返回了一个SearchResult，则：

1) 如果结果被签名、加密，或签名并加密，则将其加入到SearchResult 中的uncorrelated SearchInfo。

2) 如果结果没有被签名、加密，或没有签名并加密，则将其加入到SearchResult 中的search Info。

然后返回。

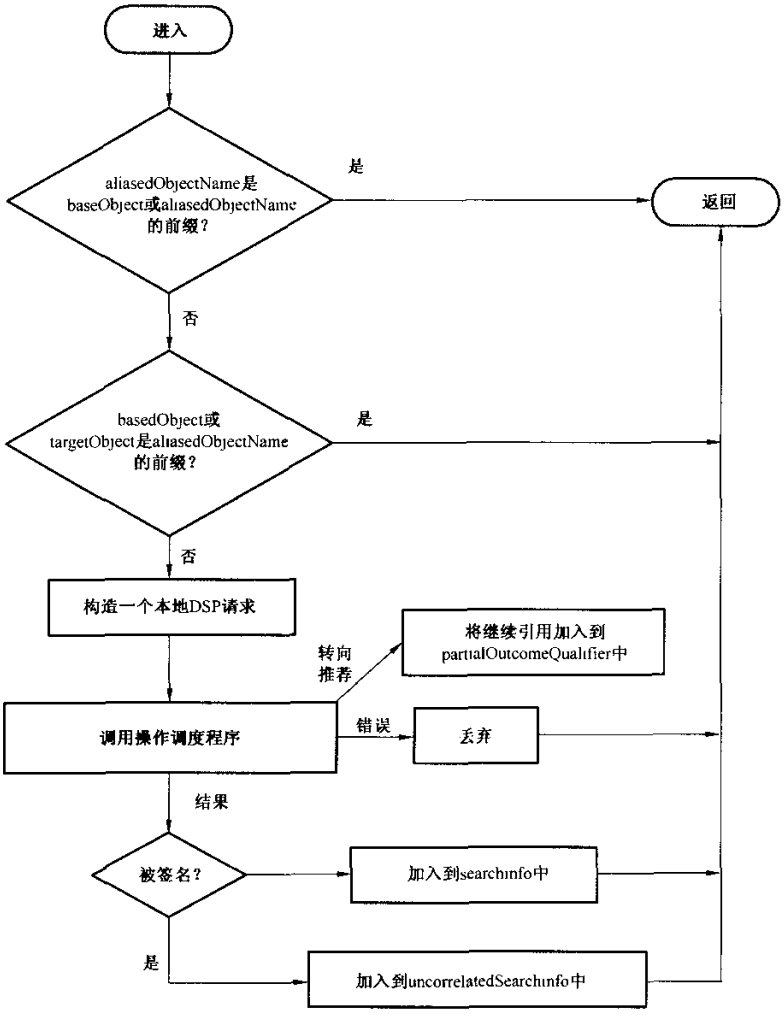


图 21 搜索别名规程

19.3.2.2.8 **分等级选择规程(Ⅰ)**

在一个规定了分等级选择的搜索请求的处理过程中,当遇到一个分等级分组中的某个成员时,运行本规程。

a) 如果 DSA 不支持的一个分等级选择存在,则返回如下信息:
——一个问题为requestedServiceNotAvailable 的serviceError;
——一个取值为id-pr-unavailableHierarchySelect 的通知属性searchServiceProblem;
——一个通知属性serviceType,其取值与搜索规则中的serviceType 组件的值相同;以及
——一个指示了非法选择的通知属性hierarchySelectList。

b) 否则,按照 19.3.2.2.4 中定义的那样,加入分等级选择所定义的所有条目。如果这样导致没有任何条目被加入,即分等级选择仅仅指定了一些不存在的条目,则设置全局变量emptyHierarchySelect。

20 连续引用规程

调用本章中的规程可以处理由其他规程所创建的连续引用列表(NRcontinuationList 或SRcontinuationList)。

连续引用规程由图 24、图 25 和图 26 中显示的步骤组成。第一阶段是从连续引用列表中标识那些具有公共目标客体组件的连续引用集。这些引用集合是根据与 DIT 中同一个条目相关的下级引用集或非特定下级引用集而创建的。在每个引用集中,可有的连续引用不止出现一次。这些集合应被扫描,并且发现的任何复制引用都应被丢弃。

这些集合(每个都拥有一个不同的 targetObject 组件)可独立地被 DSA 处理,或者是顺序处理,或者是并行处理,因为不会有从任意两个集合中返回同一个结果的风险存在。然而,在一个集合中对每个连续引用的处理,在一个连续引用中对每个AccessPointInformation 的处理,以及在一个AccessPointInformation 中对每个访问点的处理,都应被控制,否则可出现复制结果,如 20.1 中的描述。

在 APInfo 规程中采纳的规程是一个接一个地处理包含在一个单独AccessPointInformation 中的访问点集合。这些都指向同一个命名上下文(或其拷贝)(或者在 NSSR 的情况下,可指向一个 DSA 所拥有的一个命名上下文集)。如果第一个访问点产生了一个结果或一个硬差错,则后续的访问点不需要再被处理。然而,如果差错是一个软差错,即一个serviceError(问题可以为busy、unavailable、unwillingToPerform、invalidReference 或administrativeLimitExceeded),则作为一个本地选项,该 DSA 可从集合中选择另一个访问点来处理。

对一个连续引用集中的AccessPointInformation 值的处理应当以一种统一的方式进行,而不考虑连续引用源自何处(这是因为处于一个单独条目之下的两个类型为subr 的 DSE 将产生两个连续引用,其中每个都包含一个AccessPointInformation 值,然而对于一个类型为nssr 的 DSE,如果nssr 指向同样的这两个下级 DSE,假设它们由不同的 DSA 所拥有,则将产生一个连续引用,该引用包含了两个AccessPointInformation 值)。

accessPointInformation 值的处理可以是按顺序的,也可以是并行的,在 20.1 中描述。并行策略更易于产生复制结果。复制结果应总是被丢弃。

20.1 影像存在时的链接策略

影像存在时,当某个 DSA 应将一个请求多链接到多个 DSA 时,该 DSA 可在两个不同的策略之间进行选择。如果 DSA 应处理针对同一个targetObject 的多个连续引用时,总是会发生此选择。这可在两种情况下发生,一种是在名(称)解析过程中由于 NSSR 分解而引起的多链接情况(如图 22 所示),另一种是在一个多客体操作的赋值过程中由于请求分解而引起的情况(如图 23 所示)。

这些策略的目的是为了解决在请求的多链接中使用影像信息时所出现的复制结果和复制处理的问题(由于 NSSR 分解或者请求分解而引起)。例如,在图 22 中,由于在 DSE B 中所拥有的 NSSR,DSA 1 将一个请求多链接到 DSA 2 和 DSA 3。如果允许使用影像信息,则 DSA 2 和 DSA 3 可以将链接操作

应用到分别起始于 X 和 Y 的两棵子树上。

类似的，在图 23 中(由于请求分解的结果)DSA 1 将请求多链接到两个下级引用，这两个下级引用分别由 DSE X 和 DSE Y 所拥有。同样的，如果允许使用影像信息，则 DSA 2 和 DSA 3 可以将链接操作应用到分别起始于 X 和 Y 的两棵子树上。

为了处理这种复制问题，当多链接到多个 DSA 的请求具有同一个 targetObject 时，一个 DSA 可以选择下面的策略之一。

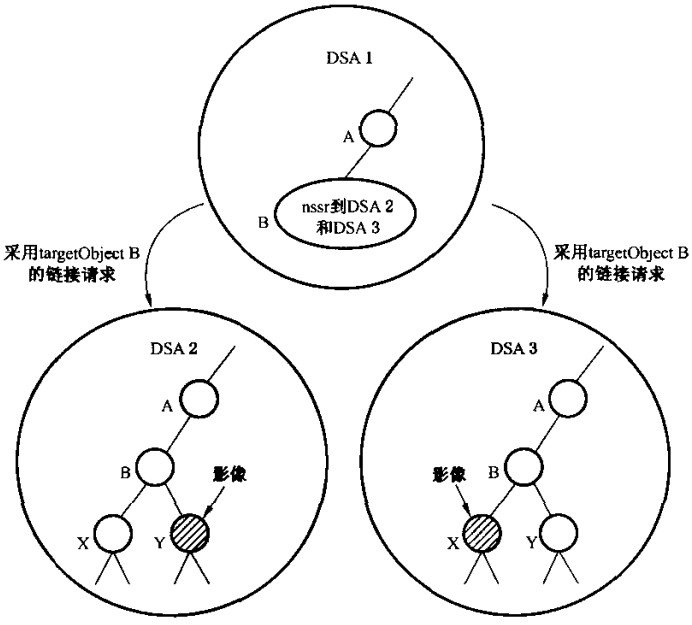


图 22 在名(称)解析阶段由于 NSSR 引起的多链接

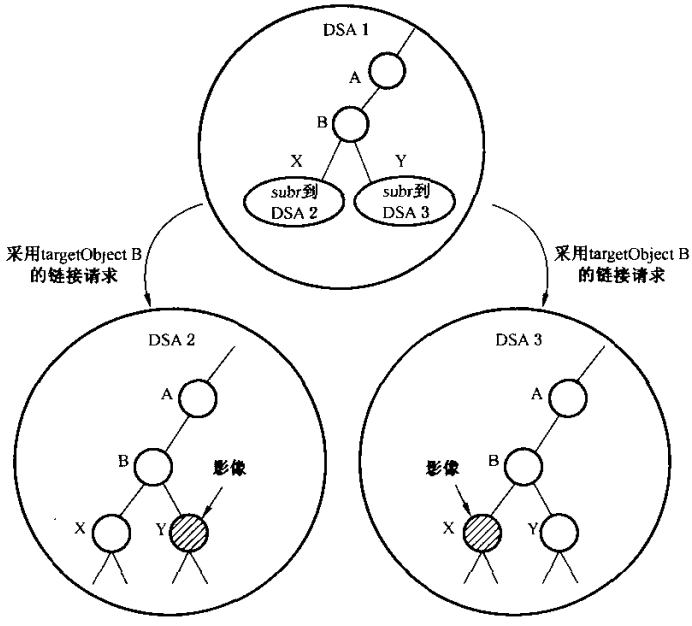


图 23 使用下级引用的多链接请求分解

20.1.1 仅使用属主策略

在某个搜索或列表赋值过程中，当执行一个由于 NSSR 分解或请求分解而引起的并行或顺序多链接时，DSA 可选择本策略来防止使用影像信息。为了使用本策略，在某个搜索或列表操作赋值过程中，ChainingArguments 的组件 excludeShadows 被设置为 TRUE。如果在名(称)解析过程中遇到了 NSSR，则 DSA 可设置 nameResolveOnMaster 为 TRUE 来确保仅遵循一个单独的路径。如果遇到了 NSSR，且操作是目录修改操作中的一种，则 nameResolveOnMaster 应被设置为 TRUE。在任何一种情况下，仅有拥有与操作相关的主条目的 DSA 才应当执行此操作。在并行和顺序多链接过程中都可以使用此"仅使用属主"策略。

注：设置 nameResolveOnMaster 为 TRUE 就排除了在名(称)解析过程中出现多路径的可能性，这是通过如下方式实现的：

a) 忽略影像条目和条目的可写拷贝；以及

b) 确保仅有一个 DSA 可以继续进行名(称)解析过程，否则在一种复杂 DIT 分布的情况下，可允许多条路径来继续处理。

这是通过仅允许一个 DSA 来继续进行名(称)解析而实现的，该 DSA 拥有与目标客体名中的第一个 nextRDNToBeResolved RDN 相应的主条目。其他任何 DSA 都不能够继续处理，即使它们可拥有与更多的目标客体名相匹配的主条目。

20.1.2 并行策略

使用本策略，一个 DSA 将所有的链接请求通过并行的多链接发出。在搜索或列表赋值阶段，以及 NSSR 的名(称)解析阶段，可使用本策略。这将会允许使用影像信息来处理链接请求，但是可引起对操作的复制处理和产生复制结果。如果一个 DSA 选择了本策略，则它应从返回的操作结果中将复制结果移除。

如果请求的是一个签名结果，则移除复制结果是不可能的，因此如果在搜索赋值阶段要求签名结果时，DSA 不能使用本策略，除非 excludeShadows 也被设置。

20.1.3 顺序策略

本策略通过使用顺序多链接来处理一个搜索分解或一个 NSSR 分解的链接请求(或链接子请求)，可以避免产生复制结果。每个链接请求都被一个接着一个地处理。

在 NSSR 分解的情况下，如果某个请求有一个结果或一个硬差错返回，则后续的请求不必再被链接。如果有一个软差错返回，依据本地策略，则可以有一个后续的请求被链接，或者将该软差错返回给请求者。

在搜索赋值的情况下，ChainingArguments 的 exclusions 组件被设置为已经被处理过的 RDN 集合。这是通过将 ChainingResults. alreadySearched 中的元素合并到下一个链接请求的 exclusions 变元中来实现的。这是唯一的一种策略可以在搜索赋值阶段完全地避免复制。

没有为列表评估定义顺序策略(尽管也可使用顺序多链接)，因为一个上级 DSA 没有办法将特定的下级从后续列表子请求的返回中排除出去(注意 excludeShadows 不排除特定的下级，但它是一种更粗糙的方式将所有的影像和可写拷贝都排除出去)。

20.2 向一个远端 DSA 发起链接子请求

在发起一个子请求之前，当 DSA 应建立一个与远端 DSA 之间的联系时，它应先运行一个 dSABind 操作。对联系的管理不在本系列目录规范的定义范围之内。当联系不能够被建立，或者 DSA 由于本地原因决定不建立联系时，与另一个 DSA 之间的联系被认为是不可用的。在这种情况下，dSABind 失败。何时停止对联系建立的尝试并宣称一个联系是不可用的，由本地决策来决定。

当一个 DSA 尝试向另一个 DSA 发起 dSABind，但接收到一个 directoryBindError 时，则子请求的

发起失败。

20.3 规程的参数

20.3.1 变元

这些规程使用如下变元：

——要处理的连续引用的列表，在NRcontinuationList（对于名（称）解析连续引用规程）和SRcontinuationList（分别对于列表连续引用和搜索连续引用规程）中；

——操作变元的CommonArguments；

——ChainingArguments。

20.3.2 结果

这些规程产生如下结果：

——如果选择了链接，则针对所发起的链接请求接收到的结果/差错列表；

——一个更新后的未处理的连续引用的列表，在continuationList 中。

20.3.3 差错

这些规程能够返回如下差错之一：

——若一个转向推荐已经被产生但它不在scopeOfReferral 内时，则产生一个问题为outOfScope 的serviceError；

——若一个非法的知识引用已经被检测到，则返回一个问题为ditError 的serviceError；

——若所有的来自 NSSR 分解的子请求都返回unableToProceed 时，则返回一个问题为noSuchObject 的nameError；

——某个链接子请求返回的任何其他差错；

——若链接未被选择且operationProgress. nameResolutionPhase 被设置为notStarted 或proceeding 时，则返回一个referral。

20.4 规程的定义

如果operationProgress. nameResolutionPhase 被设置为notStarted 或proceeding，则应遵循 20.4.1（名（称）解析连续引用规程）中的规程。多条目查询操作列表和搜索将分别调用 20.4.2 和 20.4.3 的规程。

20.4.1 名（称）解析连续引用规程

名（称）解析连续引用规程由图 24 中显示的步骤组成。本规程的基本原则是顺序地处理在名（称）解析阶段创建的连续引用集。对于NRcontinuationList 中包含的每个连续引用 C，按照一个选定的顺序，应运行下面的步骤，直到所有的连续引用都被处理过或者有一个差错或结果已经返回。如果所有的引用都已经被处理，则返回到操作调度程序继续执行，并且使用结果合并规程来处理所接收到的结果或转向推荐。

a) 检查chainingProhibited 是否被设置。如果已经被设置，则 DSA 不允许链接。根据本地策略，或者有一个问题为chainingRequired 的serviceError，或者一个转向推荐被返回到操作调度程序。

b) 如果chainingProhibited 没有被设置，则检查本地策略是否允许链接。如果链接不被允许，则返回一个转向推荐。如果本地策略允许链接，则继续下一步步骤。

c) 处理在NRcontinuationList 中发现的连续引用列表中的每一个连续引用。如果不再有未处理过的连续引用，则返回serviceError。

d) 处理NRcontinuationList 中的下一个连续引用 C。如果它是一个 NSSR，则继续执行步骤 e)。

如果它不是一个 NSSR，则调用APInfo 规程来处理它。区分对APInfo 规程的调用可返回的两种结果：

——如果 APInfo 规程返回了一个空结果，则继续执行步骤 c)，处理下一个连续引用。

——如果 APInfo 规程返回了一个差错、转向推荐或结果，则将其返回。

e) 在这种情况下，连续引用的类型为 NSSR，且 DSA 可以选择进行顺序的或并行的链接，依赖于策略的本地选择。如果 NSSR 将被顺序地处理，则继续执行步骤 f)。如果它将被并行地处理，则对于 NSSR 中的每个AccessPointInformation (API)，APInfo 规程都被调用，因此它们被并行地执行。等待所有的 API 都被处理，即等待所有对APInfo 规程的调用都返回。按照如下顺序检查所有的从APInfo 规程的调用中接收到的结果：

——如果所有的调用都返回一个问题为unableToProceed 的serviceError，且partialNameResolution 为FALSE，则返回nameError。

——如果所有的调用都返回一个问题为 unableToProceed 的 serviceError，且 partialNameResolution 为 TRUE，则在结果中设置 partialName 为 TRUE，nameResolutionPhase 为completed，且设置为条目适合(这将是为lastEntryFound 而设置)，然后转到适当的操作赋值。

——如果接收到一个或多个结果，则丢弃可能的复制并且返回结果。

——如果接收到一个不是serviceError 的差错(例如，一个nameError)，则返回该差错。

——否则根据本地选择，向操作调度程序返回一个转向推荐或serviceError。

f) 从 NSSR 的 API 集合中选择下一个未被处理的 API，然后继续执行步骤 g)。如果所有的 API 都已经被处理过了，则检查是否所有的对APInfo 规程的调用都返回一个问题为unableToProceed 的serviceError。

——如果是这样，且partialNameResolution 为FALSE，则不能发现条目且返回一个nameError。如果是这样，且partialnameResolution 为 TRUE，则在结果中设置partialName 为 TRUE，nameResolutionPhase 为completed，且设置为条目适合(这将是为lastEntryFound 而设置)，然后转到适当的操作赋值。如果不是这样，则根据本地选择，返回一个转向推荐或一个serviceError。

g) 调用APInfo 规程。区分对APInfo 规程的调用可返回的结果：

——如果接收到一个问题为unableToProceed 的serviceError，则尝试另一个访问点。继续执行步骤 f)。

——如果接收到一个问题为busy、unavailable、unwillingToPerform 或invalidReference 的serviceError，则被指示的差错可以是一个具有临时特性的差错，是否尝试将请求链接到另一个 DSA，是一个本地选择。如果选择要尝试另一个 DSA，则继续执行步骤 f)；否则根据本地选择，返回一个转向推荐或一个serviceError。

——如果接收到一个差错，但不是问题为busy、unavailable、unwillingToPerform、invalidReference 或unableToProceed 的serviceError，则该差错应被返回到操作调度程序。如果serviceError 为invalidReference，则在返回给请求者之前应被转换为ditError。

——如果接收到一个结果或一个转向推荐，则将其返回到操作调度程序。

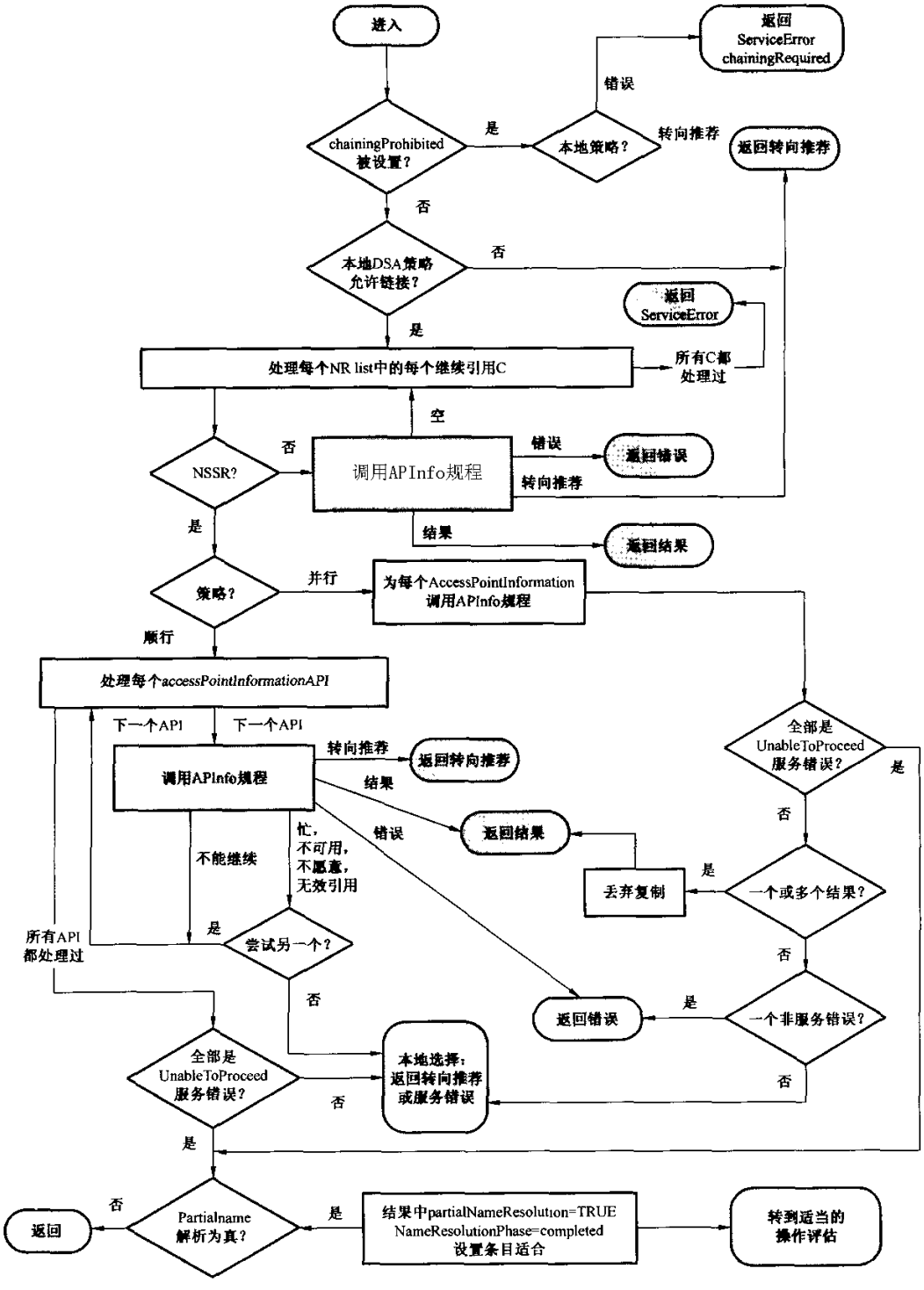


图 24 名(称)解析连续引用规程

20.4.2 列表连续引用规程

列表连续引用规程由图 25 中显示的步骤组成。当本地 DSA 不能够满足某个列表请求，且为了链接或转向推荐已经有一系列的连续引用被加入到SRcontinuationList 中时，本规程被调用。所有的这些连续引用（CR）都具有相同的targetObject。这些具有referenceType 为nssr 的 CR 具有一个或多个AccessPointInformation 值（API），而同时其他类型的 CR 仅具有一个 API。这些 API 中的每一个都被抽取出来，并考虑用作链接或转向推荐。

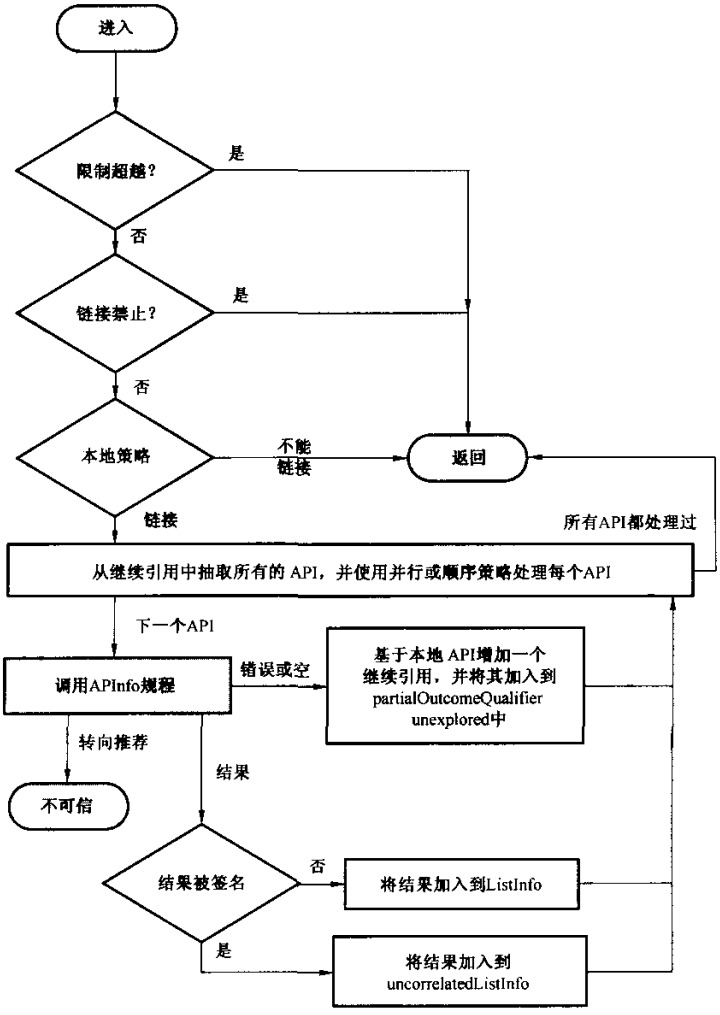


图 25 列表连续引用规程

下述步骤应被执行：

a) 迄今为止，如果有任意一个限制问题被超越，则返回到操作调度程序来继续进行结果合并规程。

b) 如果CommonArguments. serviceControls 中的chainingProhibited 标志被设置，或者由于 DSA 的本地操作策略，DSA 决定不做任何链接，则 DSA 应直接返回到操作调度程序来继续进行结果合并规程。

c) 根据SRcontinuationList 中的每个连续引用的accessPoints 组件，创建一个AccessPointInformation 值的集合。

使用并行或顺序策略来处理每个 API，如下所述：

1) 对集合中的下一个 API 调用 APInfo 规程。

2) 如果有一个结果返回，且没有被签名的话，则将其加入到listInfo 中，或者如果结果被签名的话，则将其加入到uncorrelatedListInfo 中。

3) 如果返回的是一个差错或空结果，则它意味着APInfo 已经尝试了 API 中的所有访问点而没有成功。则基于本地操作和安全策略，或者忽略并继续下一个 API，或者向partialOutcomeQualifier 中增加一个基于此 API 的连续引用。

注：从APInfo 中取回一个转向推荐是不可信任的。任何"转向推荐"应以partialOutcomeQualifier 中的unexplored 形式出现。

d) 当所有的 APIs 都被处理后，返回到操作调度程序。

20.4.3 搜索连续引用规程

搜索连续引用规程由图 26 中显示的步骤组成。当本地 DSA 不能够满足某个搜索请求，且为了链接或转向推荐已经有一系列的连续引用被加入到SRcontinuationList 中时，本规程被调用。本规程非常类似于列表连续引用规程。区别是在本规程中，SRcontinuationList 中的连续引用（CR）可以具有不同的targetObject 值。因此，这些连续引用被分类为不同的连续引用集合，每个集合拥有相同的targetObject。另外，在链接变元中的exclusions 的用法，以及链接结果中的alreadySearched 的用法都被定义，因为对于搜索来说这是一个重要的策略。exclusions 和alreadySearched 的用法被应用到对具有相同targetObject 的每个连续引用集合的处理中。

下述步骤应被执行：

a) 迄今为止，如果有任意一个限制问题被超越，则返回到操作调度程序来继续进行结果合并规程。

b) 如果CommonArguments. serviceControls 中的chainingProhibited 标志被设置，或者由于 DSA 的本地操作策略，DSA 决定不做任何链接时，则 DSA 应直接返回到操作调度程序来继续进行结果合并规程。

c) 将SRcontinuationList 中的连续引用分类为不同的集合，每个集合具有相同的targetObject。在这样的集合中不包括类型为ditBridge 的连续引用，但每个这种类型的连续引用组成了自己的一个集合。在每个集合中，移除任何复制内容。

注 1：如果有一个或多个targetObject 的值不是一个主 RDN，则这种分类可以是不正确的。分类应考虑可替代的可辨别 RDN，如果已知的话。

d) 对于连续引用的每个子集，根据子集中的每个连续引用的accessPoints 组件，创建一个Access PointInformation 值集，并且选择顺序或并行策略来做后续处理。如果选择的是并行策略，则跳过下列标示为仅应用于顺序策略的那些步骤。

1) 如果选择的是顺序策略，则为每个具有相同targetObject 的连续引用集合维护一个本地变量localExclusions。初始化时，localExclusions 被设置为入链接请求（如果存在的话）中的exclusions，且所有本地搜索过的子树直接置于targetObject 下。

2) 如果选择的是顺序策略，则比较targetObject 与localExclusions 中的所有元素，并且移除那些没有将targetObject 作为一个前缀的元素。这些是针对当前目标客体的相关排除。

3) 从当前目标客体集合中的所有连续引用中抽取出所有的 API。

4) 循环执行每个 API。对于每个 API：

i) 调用APInfo。

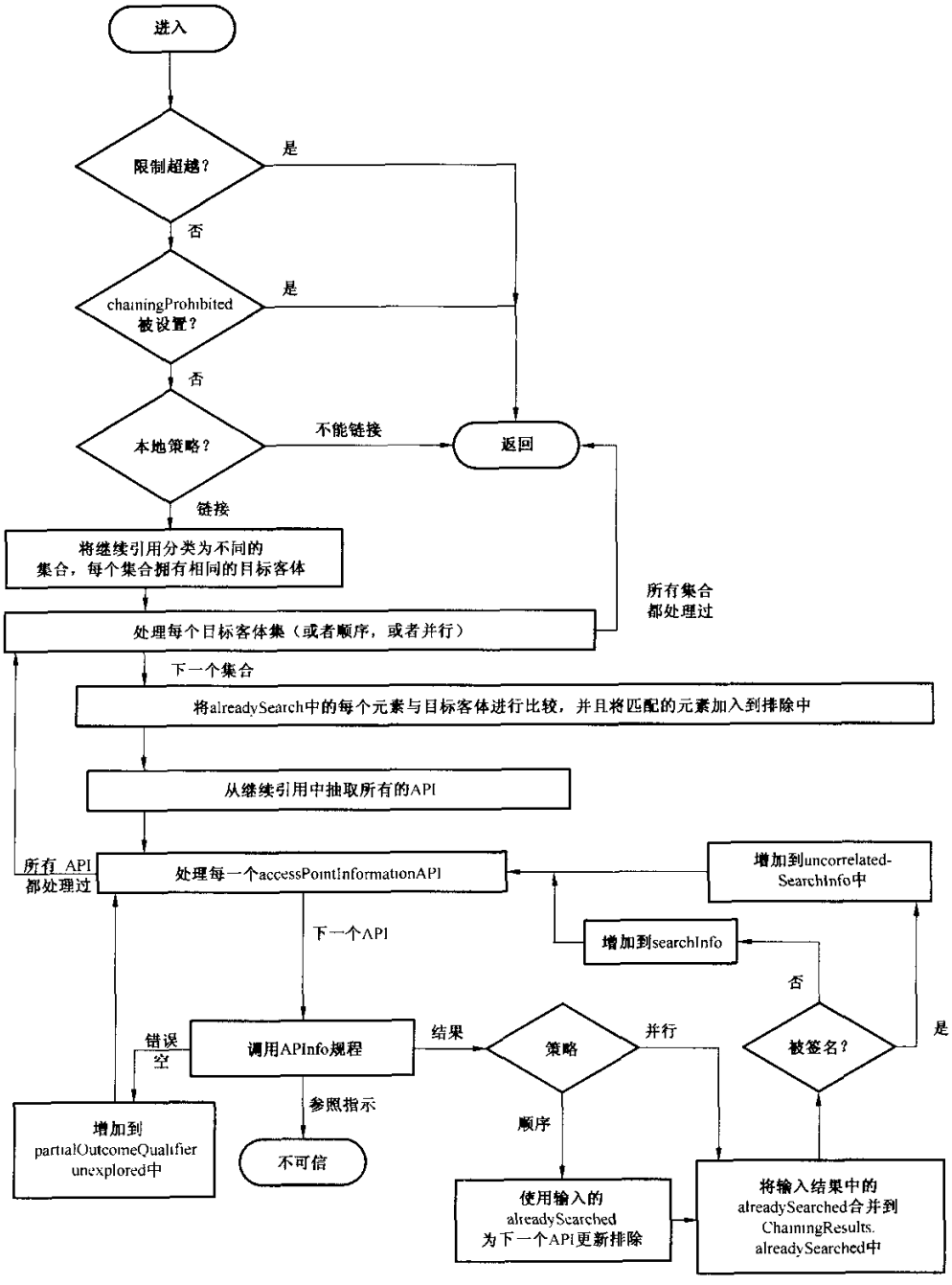


图 26　搜索连续引用规程

ii)　如果返回一个结果，则若结果未被签名，则将其增加到searchInfo 中，若结果被签名，则将其增加到uncorrelatedSearchInfo 中。如果使用了顺序策略，则使用入答复中的alreadySearched 更新localExclusions，并且将入答复中的alreadySearched 合并到该DSA 的ChainingResults. alreadySearched 中。然后继续下一个API。

iii) 如果返回一个差错或空结果，则它意味着APInfo已经尝试了API中的所有访问点而没有成功。则基于本地操作和安全策略，或者忽略并继续下一个API，或者向partialOutcomeQualifier中增加一个基于此API的连续引用。

注2：从APInfo中取回一个转向推荐是不可信任的。任何"转向推荐"应以partialOutcomeQualifier中的unexplored的形式出现。

5) 当所有的APIs都被处理后，继续下一个具有相同targetObject的连续引用集合。

e) 当所有的连续引用都被处理后，返回到操作调度程序。

20.4.4 APInfo规程

本规程被调用来处理一个AccessPointInformation，该信息中包含了一个或多个访问点（见图27）。它们被一个接一个地处理直到返回一个结果或一个差错。如果差错是一个服务差错，因此尝试另一个访问点可以成功，因此只要本地操作策略允许，则可以一直尝试其他的访问点：

a) 执行环回检测。如果有一个环回被检测出，则返回一个问题为loopDetected的serviceError。否则继续执行步骤b)。

b) 根据访问点信息处理每个访问点。如果所有访问点都已经被处理，则返回一个空结果。如果还有任意一个访问点要处理，则继续执行步骤c)。

c) 检查本地策略是否允许链接到此访问点。这种检查应考虑到服务控制和链接变元的设置（例如chainingProhibited、preferChaining，该访问点是否处于localScope内以及excludeShadows等）。如果本地设置或相应的服务控制设置不允许使用这个特定的访问点，则忽略此访问点并继续执行步骤b)。如果能够使用此访问点，则继续执行步骤d)。

d) 如果本地策略选择了"仅使用属主"策略，则将链接变元excludeShadows设置为TRUE。

如果nameResolutionPhase不是completed，且策略是继续执行主条目的名（称）解析，则将nameResolveOnMaster设置为TRUE。

如果下列的任何一点为真，则链接变元nameResolveOnMaster应被设置为TRUE：

——在入链接变元中，nameResolutionPhase为proceeding，且nameResolveOnMaster为TRUE；或者

——操作是修改操作之一，要发出的链接请求的referenceType为NSSR，且使用了一个并行策略。

注：这种使用nameResolveOnMaster的方法是为了防止由于NSSR的存在，而多次应用修改操作。

e) 构建一个链接请求并尝试发起该请求：

1) 执行环回避免，方法是检查具有相同的targetObject和operationProgress的项是否出现在所接收到的ChainingArguments内的traceInformation中。如果接收到的请求（如步骤e)的第3)项所描述）将导致一个环回，则DSA或者向发起请求的DUA/LDAP客户机/DSA返回一个问题为loopDetected的serviceError，或者通过继续执行步骤2)来忽略此访问点，并尝试下一个访问点。

2) 如果将要被链接的请求或子请求是执行一个转向推荐的结果，则要求执行一个额外的环回避免检查。方法是检查具有相同的targetObject、operationProgress和目标DSA的项是否出现在referralRequests中。如果这样的话，则执行a)项中指定的动作。如果没有，则向referralRequests中增加一个新的TraceItem，并具有如下组件：

——targetObject和operationProgress被设置为与被链接的请求/子请求的值相同；

——dsa被设置为请求/子请求要被链接向的DSA的名（称）。

3) 在一个成功的绑定后，DSA应发起一个与被处理的操作具有相同操作类型的链接操作，并具有如下参数：

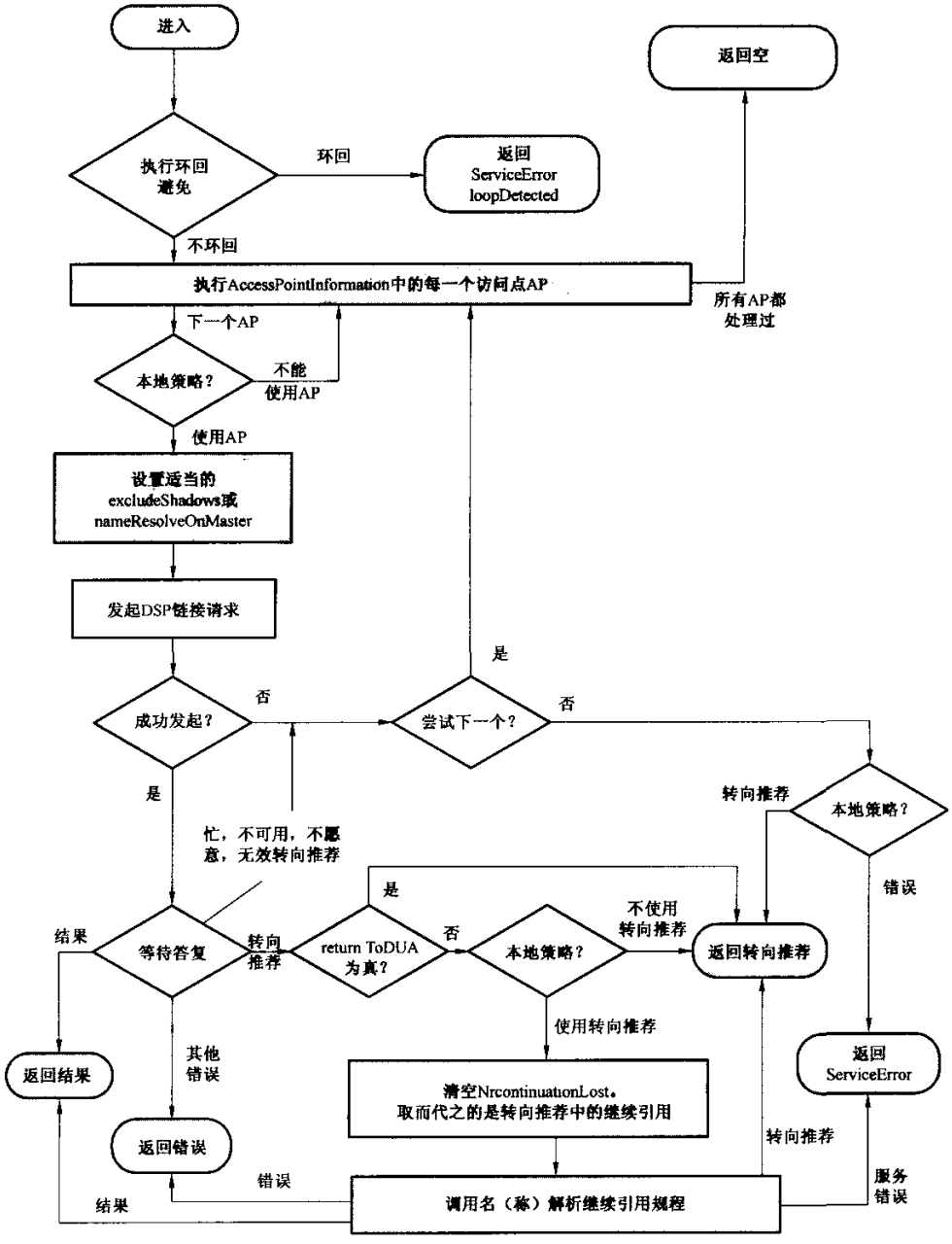


图 27　APInfo 规程

——链接操作中的操作变元被设置为与接收到的操作变元相同；

——ChainingArguments. originator 被设置为与接收到的相同；

——ChainingArguments. targetObject 被设置为连续引用的targetObject；

——ChainingArguments. operationProgress 被设置为连续引用的 operationProgress 的值；

——如果连续引用的类型不是ditBridge，则ChainingArguments. traceInformation 被设置为由请求有效性验证规程更新过的踪迹信息，否则该组件应缺失；

——ChainingArguments. aliasDereferenced 被设置为本地更新后的aliasDereferenced 的更新后的值；

——ChainingArguments. returnCrossRefs 的设置是一个本地选择；

——ChainingArguments. referenceType 被设置为连续引用的referenceType 的值；

——ChainingArguments. timeLimit 被设置为接收到的timeLimit 的值；

——如果被搜索连续引用规程所调用，则chainingArguments. exclusions 被设置为当前目标客体的相关排除，或者如果 APInfo 规程是由名(称)解析或列表连续引用规程所调用，则该chainingArguments. exclusions 缺失；

——SecurityParameters 被设置为接收到的SecurityParameters 的值。

f) 如果请求不能被成功发出，则继续执行步骤 g)。如果能够被成功发出，则继续执行步骤 h)。

g) 是否继续执行由本地选择。如果 DSA 选择继续，则此差错被忽略，下一个访问点将被尝试。继续执行步骤 b)。如果 DSA 决定不再尝试下一个访问点，则本地策略将选择是向规程的调用者返回一个相应的转向推荐还是一个serviceError。

h) 如果请求能够被成功发出，则 DSA 应等待答复，并对答复进行处理：

1) 如果接收到一个结果，则该结果被返回给规程的调用者。

2) 如果接收到一个问题为 busy、unavailable、unwillingToPerform 或 invalidReference 的 serviceError，则继续执行步骤 g)。

3) 如果接收到一个转向推荐，且returnToDUA 被设置为TRUE，则接收方 DSA 不应当对该转向推荐执行动作，而应将转向推荐返回给请求者。

4) 如果接收到一个转向推荐，且returnToDUA 被设置为FALSE，则应用同步骤 3)同样的本地策略进行考虑(考虑服务控制、链接变元、链接策略等)。如果决定不对转向推荐解除引用，则向调用者返回该转向推荐。如果决定对转向推荐解除引用，则清空NRcontinuationList，将接收到的转向推荐中的连续引用放置到NRcontinuationList 中，并且调用名(称)解析连续引用规程。这可产生一个结果、转向推荐、服务差错或其他差错。无论从名(称)解析连续引用规程的调用中接收到什么，都应被返回给调用者。

5) 如果出现任何其他差错，则应被返回给调用者。

20.5 放弃规程

如果接收到一个放弃请求，则本规程被调用。它由图 28 中显示的下列步骤组成。

a) 当接收到一个放弃请求，而该请求指向一个未知的操作，则应向请求者返回一个问题为noSuchOperation 的abandonFailed。

b) 如果要被放弃的请求已经被答复，且DSA 已经保留了需要知道的信息，则可向请求者返回一个问题为tooLate 的abandonError。

c) 如果要被放弃的请求是非有效的，即要求放弃的请求不是一个查询请求，则应向请求者返回一个问题为cannotAbandon 的abandonFailed。

d) 如果一个 DSA 在接收到一个有效的对原始请求的放弃请求时，还有链接请求(或子请求)正在进行中，而 DSA 决定尝试执行放弃，则它可向本操作的正在进行的请求(或子请求)的部分或全部发送放弃请求，并且等待放弃请求和正在进行的请求(或子请求)的答复。在本操作执行过程中的任何时间点，DSA 都可向请求者发送一个放弃结果和一个abandonFailed，然后当所发起的放弃请求和正在进行的请求(或子请求)的答复到达时，放弃它们。

如果 DSA 决定直到没有正在进行的请求(或子请求)时,才向请求者发送答复,则如果所有发起的abandon 请求都被答复为具有abandonedFailed 差错,且没有本地放弃操作被执行时,DSA 可以可选地向请求者发送一个abandonedFailed 差错。

如果向请求者返回了一个AbandonedFailed 差错,则原始请求应被处理,就好像从来没有接收过放弃操作一样。

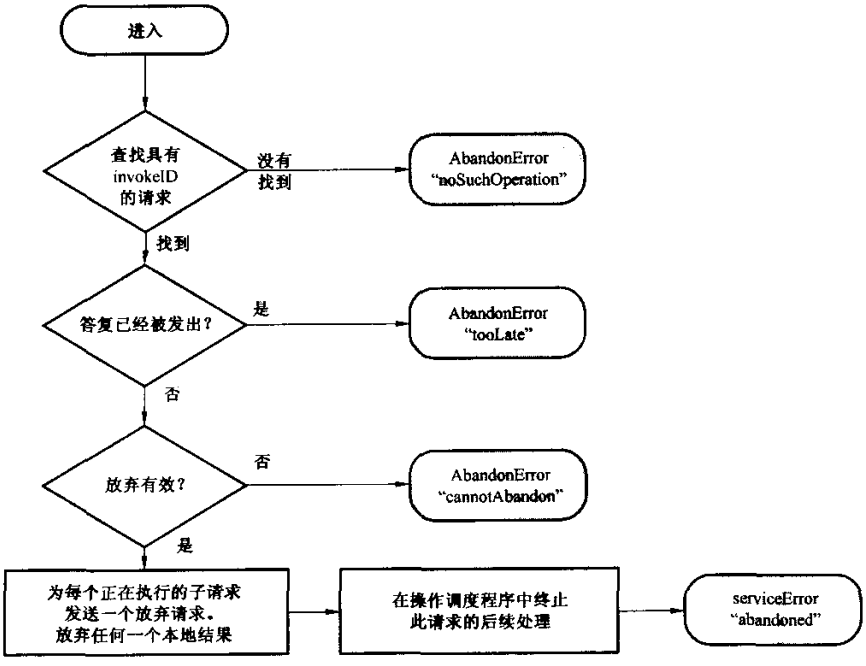


图 28 放弃规程

21 结果合并规程

在任意一个连续引用规程之后,图 29 中显示的结果合并规程被调用。如果结果没有被签名,且如果在partialOutcomeQualifier. unexplored 中有附加的连续引用时,本规程将移除复制。因此如果本地操作策略允许的话,相关的连续引用规程将被调用:

a) 如果操作是一个列表操作,则继续执行步骤 b);如果操作是一个搜索操作,则继续执行步骤 c);否则,返回作为结果合并规程的输入参数所提供的结果。

b) 操作是一个列表操作。移除所有的复制,且属主信息的优先级高于影像信息。

如果操作结果是本地产生的,且它包含连续引用,因此这些结果不能被用于链接而是要返回给用户。在这种情况下,继续执行步骤 f)。

如果接收到的操作结果是一个链接列表操作的结果,则该结果中可以包含连续引用。在这种情况下,检查preferChaining 服务控制是否被设置。如果被设置为TRUE,则 DSA 应使用此连续引用来进行链接。继续执行步骤 d)。

c) 操作是一个搜索操作。移除所有的复制,且属主信息的优先级高于影像信息。如果有一个限制问题,则返回此结果。否则继续执行步骤 d)。

d) 处理任意链接操作结果中partialOutcomeQualifier. unexplored 内的每一个连续引用。如果本地策略决定不使用它来进行链接,则忽略它并选择另一个连续引用。如果本地策略允许使用该连续引用来进行链接,则执行如下动作:

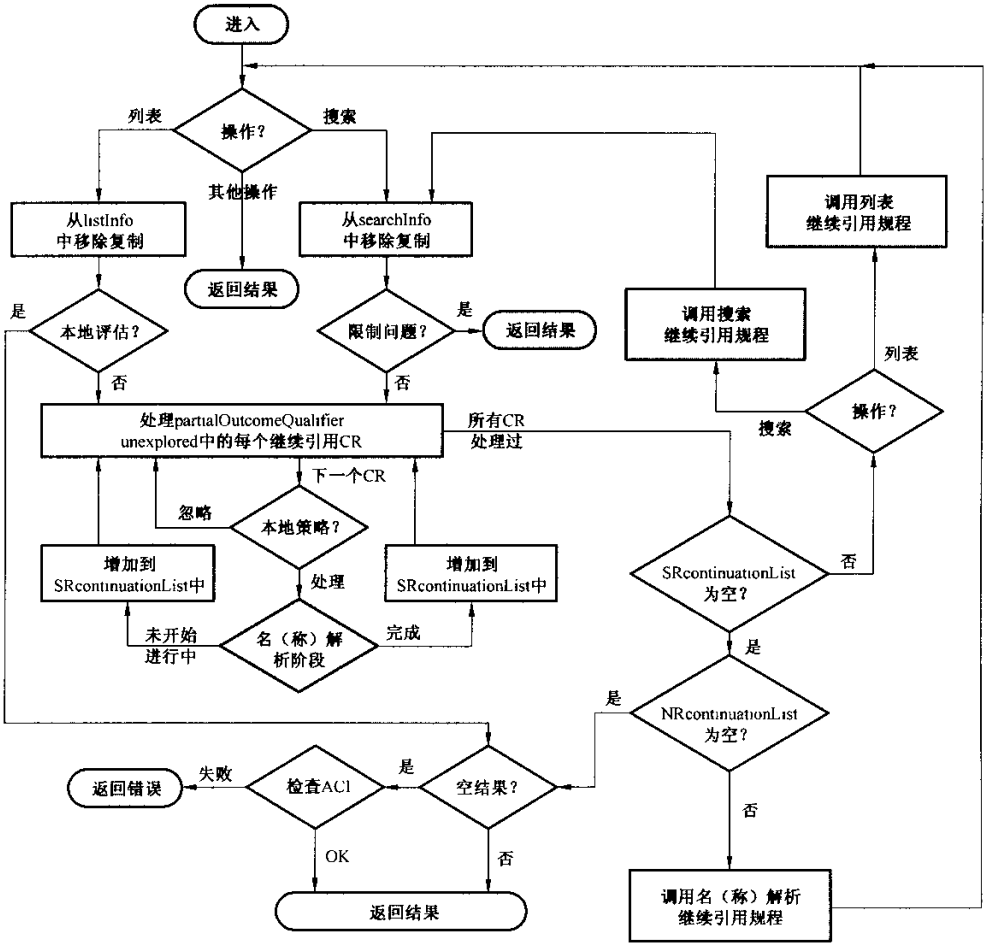


图 29　结果合并规程

检查连续引用中提供的nameResolutionPhase。如果它的值为notStarted 或proceeding，则将其加入到准备提供给名(称)解析继续规程的连续引用列表中(NRcontinuationList)。如果nameResolutionPhase 的值为completed，则将连续引用加入到提供给子请求继续规程的连续引用列表中(SRcontinuationList)。

继续执行，直到所有的连续引用都被处理过。

e) 如果在SRcontinuationList 中有要处理的连续引用，则检查操作类型。如果操作是一个列表操作，则调用列表连续引用规程，并继续执行步骤 b)。如果操作是一个搜索操作，则调用搜索连续引用规程，并继续执行步骤 c)。

如果SRcontinuationList 为空，则检查在NRcontinuationList 中是否有连续引用。如果有，则调用名(称)解析连续引用规程，并继续执行步骤 c)。

如果两个连续引用列表都为空，则继续执行步骤 f)。

f) 检查结果是否为空。如果非空，则返回该结果。如果为空，且访问控制和本地策略允许，则返回一个空结果，或者返回一个适当的差错。

当一个 DSA 从其他 DSA 处接收到搜索或列表结果，且这些结果中含有该 DSA 未知的参数时，则无关的结果应被返回。否则，如果搜索结果未被签名，或者如果 DSA 是一个允许移除签名的初始执行

者时(见 GB/T 16264.3—2008 的 7.9),则 DSA 应执行合并。

如果一个 DSA 从一个不能执行合并的 DSA 处接收到未被签名的、不相关的结果,且该 DSA 对这些不相关的结果的所有参数都拥有正确知识时,该 DSA 应执行合并。

如果一个 DSA 从另一个 DSA 处接收到未被签名的结果,同时它也可拥有一个本地结果,则该 DSA 将产生一个条目数量,该条目数量将放在 DSA 产生的PartialOutcomeQualifier 的entryCount 中被返回,这个条目数量将是所有接收到的entryCount 的值,本地结果,以及从没有返回entryCount 值的 DSA 处接收到的条目数量的总和,然后再将复制结果抵消后的值。如果该 DSA 是初始执行者,且分页结果被请求时,则它还应包括从其他 DSA 处接收到的签名结果中的条目数量。

如果分页结果被请求,且任何 DSA 都没有遇到限制问题,则 DSA 应为entryCount 参数执行exact 选择。应为每个返回的页给出同样的值。

如果一个或多个 DSA 遇到了某个限制问题,则:

——如果所有遇到限制问题的 DSA 都已经返回了一个entryCount,且具有exact 或bestEstimate 选择,则如果仅有一个 DSA 能够执行该选择,则它应执行bestEstimate 选择;否则它应执行 exact 选择;

——如果只有一个 DSA 遇到一个限制问题,且返回了一个选择为lowEstimate 的entryCount,或不返回一个entryCount,则它应执行lowEstimate 选择。

22 分布式鉴别规程

本章规定了支持目录分布式鉴别服务所必需的规程。这些服务以及相应的规程,被分类为:

——发起者鉴别,对它的支持或者采用一种未被保护(基于简单身份)的方式,或者采用一种安全(基于数字签名)的方式;以及

——结果鉴别,采用类似被保护的方式(也是基于数字签名)。

22.1 发起者鉴别

22.1.1 基于身份的鉴别

基于身份的鉴别服务使得 DSA 可以出于实现本地访问控制的目的,对信息的初始请求者进行鉴别。希望使用此服务的 DSA 应采用下列规程:

——对于一个要求对 DAP 或 LDAP 请求进行鉴别的 DSA,在一个 DUA 联系(DUA 到 DSA)或 LDAP 客户机联系(LDAP 客户机到 DSA)建立的时刻,该 DSA 通过绑定规程获得请求者的可辨别名。这些规程的成功结果在任何时候都不会损害后续使用此联系来处理操作时所要求的鉴别级别。

——与 DUA 或 LDAP 客户机存在联系的 DSA 应为链接到其他 DSA 的所有子请求,将请求者的可辨别名插入到ChainingArguments 的发起者字段中。

——一个 DSA,在接收到一个链接操作时,可以满足此操作,也可以不满足,依赖于访问权限的决定(一个本地定义的机制)。如果输出不满意,则可以返回一个问题为insufficientAccessRights 的securityError。

22.1.2 基于签名的发起者鉴别

基于签名的发起者鉴别服务使得 DSA 可以对一个特定的服务请求的发起者进行鉴别(以一种安全的方式)。DSA 在实现此服务时使用的规程在本条中描述。

基于签名的鉴别服务由一个 DUA 来调用,方法是使用一个可选的被保护请求中的PROTECTED 变元,其中DirQOP 取值为signed 或signedAndEncrypted。

一个 DSA,在从另一个 DSA 处接收到一个签名的请求后,应在处理此操作之前移除那个 DSA 的签名。假设任何签名有效性验证的结果证明是满足的,则 DSA 将继续执行此操作。如果在处理过程中,DSA 需要执行链接,则每个相关的链接操作的变元集应按照如下所述来构建:

——DSA 构造一个可以被签名的变元集：该变元集由输入的已签名变元集和一个修改后的ChainingArguments 共同组成。

如果 DSA 能够将信息发布给响应，则基于签名服务请求的发起者鉴别可以被用来决定对此信息的访问权限。

如果一个 DSA 接收到一个关于某信息的未被签名的服务请求，而根据发起者鉴别该信息只能被释放，则应返回一个问题为protectionRequired 的securityError。

22.2 结果鉴别

本服务的提供使得目录操作的请求者(DUA、LDAP 客户机或 DSA)能够验证(使用数字签名机制这样一个安全方式)结果的来源。可以要求使用结果鉴别服务，而不用考虑是否使用了发起者鉴别。

结果鉴别服务的发起是使用包含在目录操作变元集内的protectionRequest 组件的签名值来实现的：一个 DSA 如果接收到一个带有此选项的操作时，可以可选地对任何子请求结果进行签名。在保护请求中的签名选项，其作用是向 DSA 指示了请求者的优先级；实际上，DSA 可以对，也可以不对任意子请求结果进行签名。

在一个 DSA 执行链接的情况下，根据返回到请求者的结果形式，DSA 拥有一系列的选项，这些选项为：

a) 向请求者返回一个复合响应(签名的或未签名的)；

b) 向请求者返回两个或多个未被合并的部分响应的集合(签名的或未签名的)；在此集合中可以有零个或多个成员被签名，也可以有零个或一个未被签名。如果有一个未被签名的部分结果存在，则该成员实际上可以是一个或多个未被签名的部分响应的集合，这些部分响应可以是从其他 DSA 处接收到的，或者是由本 DSA 提供的，或者两种都有可能。

当一个 DSA 对相关条目进行合并时，执行合并的 DSA 可对结果进行签名。

第六篇：知识管理

23 知识管理概述

为了以一种一致性和性能可接受的程度来操作广泛分布的目录，需要有规程对每个 DSA 中拥有的知识进行创建、维护和扩展。下列机制被共同使用来管理 DSA 的知识。

a) 分等级操作绑定和非特定分等级操作绑定：这些规程和协议在第 24 章和第 25 章中定义。它们被用来维护下级引用、非特定下级引用、直接上级引用、以及命名上下文的上下文前缀信息等。这些操作绑定是在主 DSA 之间建立的，这些主 DSA 拥有彼此之间分等级相关的命名上下文，即具有直接下级与直接上级的关系。本规程的触发可以是修改条目 RDN、增加条目、或移除条目等操作的一个副作用，那些条目的直接上级不是由拥有该条目的同一个 DSA 所拥有的。

b) 影像操作绑定：这些规程和协议在 ISO/IEC 9594-9 中定义。它们被用来以两种方法创建和维护知识引用。第一种是作为建立(或终止)影像商定的一个副作用，访问点被加入到consumerKnowledge 中或可选的secondaryShadow 操作属性中，或从中移除。因此这些信息可以被上面讨论的规程和协议使用来更新上级主 DSA 中的下级引用，或者下级主 DSA 中的直接上级引用。第二种是 DISP 将主 DSA 中拥有的知识引用传播到影像消费者 DSA 处。

c) 交叉引用：交叉引用的分布是 DSP 的一个特性。它用于创建和维护交叉引用，在 23.2 给出了摘要。

注：初始化和维护上级引用以及myAccessPoint 的机制，不在本目录规范的定义范围之内。

23.1 知识引用的维护

本条描述了 DOP 是如何被用来维护表示知识的 DSA 操作属性的。关于知识属性和用于维护这些

知识属性的协议之间的关系，有一个简单示例在附录 E 中描述。

23.1.1 提供者和主 DSA 对消费者知识的维护

一个消费者引用，通过consumerKnowledge 属性的值来表示，由一个影像提供者 DSA 所拥有，并与一个命名上下文的上下文前缀相关联；一个提供者引用，通过 supplierKnowledge 属性的值来表示，由一个影像消费者 DSA 所拥有，并与一个命名上下文的上下文前缀相关联。这两个属性都由类型为 cp 的 DSE 所拥有。这些属性的每个值都是在影像操作绑定建立时被创建的，并且在影像操作绑定修改时更新。

一个提供者 DSA 可以获取信息来构建secondaryShadows 属性的值，这是在它与某个消费者的 ShadowingAgreementInfo 中的可选组件secondaryShadows 取值为TRUE 的情况下。在这种情况下，无论何时，当消费者 DSA 检测出拥有公共可用复制区拷贝的 DSA 集合（包括它的消费者，以及消费者的消费者等，可以携带任意深度的二次影像）被修改了（通过增加、修改或移除访问点），则它会通过一个 modifyOperationalBinding 操作来传达此新信息（一个SupplierAndConsumers 的集合），该操作在 ITU-T X.525 建议书 | ISO/IEC 9594-9:2005 中描述。

一个提供者 DSA 维护其自身的，与上下文前缀相关的secondaryShadows 属性，如下所述：

a) 通过modifyOperationalBinding 操作从一个消费者处接收到的SupplierAndConsumers 集合可以被用于创建或替代本属性的值。SupplierAndConsumers 中的提供者组件表示了一个消费者 DSA（或消费者的消费者等，依赖于二次影像的深度）的访问点；消费者组件表示消费者的消费者集合（或其消费者，依赖于二次影像的深度）。

b) 每个在modifyOperationalBinding 操作中提供其提供者的消费者，都包含一个 SupplierAndConsumers 集合，具有下列值：它的secondaryShadows 属性的值，以及一个新构造的值。这个新值是这样构造的：使用它自己的访问点myAccessPoint（作为提供者组件），以及包含在consumerKnowledge 属性中的表示拥有公共可用影像的消费者的访问点值（作为消费者组件）。

本规程的递归使用可以使一个命名上下文的主 DSA 能够知道它的所有的二次影像消费者 DSA，这些 DSA 拥有从该命名上下文派生的公共可用的复制区。于是，这些信息可用于对下级引用、非特定下级引用和直接上级引用等的维护。

23.1.2 主 DSA 中下级和直接上级知识的维护

一个下级引用通过specificKnowledge 属性的一个值来表示，该属性由拥有此下级引用的直接上级命名上下文的 DSA 中类型为subr 的 DSE 所拥有；一个直接上级引用通过specificKnowledge 属性的一个值来表示，该属性由拥有此上级引用的直接下级命名上下文的 DSA 中类型为immSupr 的 DSE 所拥有。这些属性的每个值都在相应的上级和下级主 DSA 中建立 HOB 时创建，并在 HOB 修改时更新。

一个下级主 DSA 通过它在 DOP 中传送到上级的SubordinateToSuperior 参数中的accessPoints 组件，向其上级主 DSA 提供信息来构造它的下级引用。包含在accessPoints 中的信息由下级 DSA 所拥有的属性值来决定，如下所述：

a) myAccessPoint 属性的值（由根 DSE 拥有）被用来构造accessPoints 中的元素，其中category 的取值为master。

b) consumerKnowledge 和secondaryShadows 的值（两个都由下级上下文前缀 DSE 拥有）被用于构造accessPoints 中的附加元素，其中category 的取值为shadow。

一个上级主 DSA 通过它在 DOP 中传送到下级的SuperiorToSubordinate 参数中的contextPrefixInfo 组件，向其下级主 DSA 提供信息来构造它的直接上级引用。该组件的值类型为SEQUENCE OF Vertex，包含了与 DIT 中的根到此下级上下文前缀之间的路径相对应的元素序列。对于这些元素中的其中一个与直接上级命名上下文的上下文前缀相应的元素，可选组件accessPoints 将存在。下级 DSA 将此信息作为类型为immSupr 的 DSE 中的一个specificKnowledge 属性，相应于contextPrefixInfo 中的这个元素。包含在上级 DSA 的accessPoints 中的信息由上级 DSA 所拥有的属性值来决定，如下所述：

a) myAccessPoint 属性的值(由根 DSE 拥有)被用来构造accessPoints 中的元素,其中category 的取值为 master。

b) consumerKnowledge 和secondaryShadows 的值(两个都由上级上下文前缀 DSE 拥有)被用于构造accessPoints 中的附加元素,其中category 的取值为shadow。

注:仅有那些与接收到公共可用复制区的消费者 DSA 相应的访问点才应当被上级 DSA 和下级 DSA 根据它们的consumerKnowledge 属性选中,并将其包含到accessPoints 中。构造secondaryShadows 的规程会确保这些访问点可以标识出拥有公共可用复制区的影像 DSA。

23.1.3 消费者 DSA 中下级和直接上级知识的维护

一个影像消费者 DSA 与其提供者之间有商定,可以接收到与某个复制单元相关的直接上级和下级知识,如果该商定有效,则也有商定由它的影像提供者 DSA 通过 DISP 来维护其直接上级和下级引用。

注:对于某些复制规范单元,对于消费者 DSA 来说,为了能够让它的提供者可以向其提供下级知识,可以有必要定义商定来接收extendedKnowledge。

23.2 请求交叉引用

为了改进目录系统的性能,可以使用普通的目录操作来扩展交叉引用的本地集合。如果一个 DSA 支持 DSP,它可请求另一个 DSA(该 DSA 也支持 DSP)返回那些包含了与一个普通目录操作的目标客体名相关的命名上下文的位置信息的知识引用。

如果 ChainingArguments 中的 returnCrossRefs 组件被设置为 TRUE,则 ChainingResults 中的 crossReferences 组件可存在,且包含了一个交叉引用项的集合。

如果一个 DSA 不能够将一个请求链接到下一个 DSA,则一个转向推荐会被返回到发起请求的 DSA。如果ChainingArguments 中的returnCrossRefs 组件为TRUE,则该转向推荐中另外还包含此转向推荐所指向的命名上下文的上下文前缀。如果该转向推荐是基于非特定下级引用,则contextPrefix 组件缺失。一个转向推荐所返回的交叉引用是基于产生该转向推荐的 DSA 所拥有的知识的。

在两种情况下(链接结果和转向推荐),某个管理机构,通过他的 DSA,可选择忽略那些返回交叉引用的请求。

23.3 知识的不一致性

目录应支持一致性检查机制来保证某种程度的知识一致性。

注:在某种环境中,一个知识引用是正确的(在下面描述的情况下不是非法的),但是对于某个 DSA 来说使用它是无效的,因为被引用的 DSA 的 DMD 根本不希望被引用它的 DSA 所接触(例如,一个 DSA 不知何故获得了一个指向被引用的 DSA 的交叉引用),或者不希望它以某种角色被接触(例如,作为某个命名上下文的主 DSA)。

23.3.1 知识不一致性的检测

不一致性的类型及其检测根据知识引用类型的不同而不同。

a) 交叉引用和下级引用——如果被引用的 DSA 没有拥有一个命名上下文或者一个复制区,该复制区所来自的命名上下文具有引用中所包含的上下文前缀时,这种类型的引用是非有效的。这种不一致性可以在名(称)解析过程中被检测出来,方法是通过检查ChainingArguments 中的operationProgress 和referenceType 组件。

b) 非特定下级引用——如果被引用的 DSA 没有拥有一个本地命名上下文,该命名上下文的上下文前缀是引用中所包含的上下文前缀减去最后一个 RDN 而形成的时,这种类型的引用是非有效的。一致性检查的应用如上所述。

c) 上级引用——一个非法的上级引用是不能够构成到根的一个引用路径的引用。上级引用的维护应通过外部方式来维护,且不在本目录规范的定位范围之内。

注:并不总是能够检测出一个非法的上级引用。

d) 直接上级引用——如果被引用的 DSA 没有拥有一个命名上下文或者一个复制区,该复制区所来自的命名上下文具有引用中所包含的上下文前缀时,这种类型的引用是非有效的。另外,仅当ChainingArguments 中的operationProgress 组件取值为notStarted 或proceeding 时,使用

这种类型的引用才是有效的。这种不一致性可以在名(称)解析过程中被检测出来,方法是通过检查ChainingArguments 中的operationProgress 和referenceType 组件。

e) 提供者引用——这种类型的引用,标识了某个复制区的提供者,并且可选地标识了该复制区所来自的命名上下文的属主,当被引用的 DSA 不是使用该引用的 DSA 的影像提供者时(当ChainingArguments 的referenceType 组件取值为supplier 时),或者当被引用的 DSA 不是该命名上下文的属主时(当referenceType 的取值为master 时),这种类型的引用是非法的。这种不一致性可以在操作处理的名(称)解析阶段和操作赋值阶段被检测出来,方法是通过检查ChainingArguments 中的referenceType 组件。

23.3.2 知识不一致性的上报

如果使用链接来执行一个目录请求,则所有的知识不一致性都将被拥有非法知识引用的 DSA 检测出来,方法是通过接收到一个问题为invalidReference 的serviceError。

如果一个 DSA 返回了一个基于某个非法知识引用的转向推荐,则请求者如果使用了该转向推荐的话,将被返回一个问题为invalidReference 的serviceError。差错条件是如何被传播到存储此非法引用的 DSA 的,不在本目录规范的定义范围之内。

23.3.3 不一致的知识引用的处理

当一个 DSA 在检测到一个非法引用后,它应尝试去重新建立知识的一致性。例如,这可以通过简单地删除一个非法交叉引用来实现,或者通过将其替换为一个使用returnCrossRefs 机制而获得的正确引用来实现。

实际上,DSA 如何处理非法引用的方法属于本地事务,不在本目录规范的定义范围之内。

23.4 知识引用和上下文

知识引用中的名(称)应是主可辨别名,对于任意 RDN 中使用的任意属性,也可以包含可替代的可辨别值以及在valuesWithContext 中拥有的上下文信息,如同 GB/T 16264.2—2008 的 9.3 中的描述。

依赖于一个知识引用是如何被获得的(尤其是对于一个第 3 版本之前的 DSA,其是否拥有该引用,或者是否是获取引用的链接中的一部分),可能一个知识引用中不会包含所有可能的可替代可辨别名。这可导致一个声称名不被该知识引用的所有者认为是同一个名(称),这在某种情况下,会导致需要在名(称)解析过程中执行额外的步骤,或者在某种情况下,会导致不一致的结果或名(称)解析失败。在主可辨别名已知时,通常使用主可辨别名,会优化目录处理名(称)中的上下文变元的能力。

24 分等级操作绑定

一个分等级操作绑定被用来表示两个 DSA 之间的关系,这两个 DSA 拥有两个命名上下文,其中一个是另一个的直接下级。在 HOB 的情况下,上级 DSA 拥有一个指向下级 DSA 所拥有的命名上下文的下级引用;下级 DSA 拥有一个指向上级 DSA 所拥有的命名上下文的直接上级引用。操作绑定确保了可以在两个 DSA 之间交换和维护适当的知识信息,因此,两个 DSA 都能够在名(称)解析和操作赋值的处理过程中,按照第 18 章和第 19 章的定义来运转。

24.1 操作绑定类型特性

24.1.1 对称与角色

分等级操作绑定类型是一种非对称的操作绑定类型。在这种类型的绑定中有两种角色,分别为:

a) 上级命名上下文的主 DSA 所承担的角色,称为上级 DSA(相关的抽象角色为"A");以及

b) 下级命名上下文的主 DSA 所承担的角色,称为下级 DSA(相关的抽象角色为"B")。

24.1.2 商定

在建立分等级操作绑定的过程中,所交换的商定信息是HierarchicalAgreement 的一个值。它包括新的上下文前缀的相对可辨别名(rdn 组件)以及该新的命名上下文的直接上级条目的可辨别名(immediateSuperior 组件)。此信息应被发起该 HOB 的 DSA 所提供。

```
HierarchicalAgreement ::= SEQUENCE {
    rdn                 [0]  RelativeDistinguishedName,
    immediateSuperior   [1]  DistinguishedName }
```

rdn 应是主 RDN，且immediateSuperior 应是一个主可辨别名。上下文信息以及所有的可替代可辨别值应被包含在任意 RDN 的AttributeTypeAndDistinguishedValue 的valuesWithContext 组件中，如同 GB/T 16264.2—2008 的 9.3 中的描述。

24.1.3 发起者

24.1.3.1 建立

一个分等级操作绑定的建立可以由任意一种角色来发起。上级 DSA 的发起可以是由于一个增加条目操作而引起的，其中在targetSystem 扩展中指定了下级 DSA，或者是由于管理干预而引起的。下级 DSA 的发起(将一个本地存在的条目或子树与全球 DIT 连接起来)是由于管理干预而引起的。

24.1.3.2 修改

一个分等级操作绑定的修改可以由任意一种角色来发起。由于上级上下文前缀信息的修改，使得上级 DSA 可以发起修改。这可以是任意修改操作的结果，或者是管理干预的结果。

如果下级命名上下文的上下文前缀条目的 RDN 被修改，则任意一个 DSA 都可以修改此商定。上级 DSA 发起此修改，是由于一个相对可辨别名被修改为高于此 DIT 而引起的，或者是由于管理干预而引起的。下级 DSA 发起此修改，是由于针对一个上下文前缀的ModifyDN 而引起的，或者是由于管理干预而引起的。

如果该命名上下文的访问点信息发生了变化，则任何一个 DSA 也都可以修改此 HOB。

24.1.3.3 终止

一个分等级操作绑定的终止可以由任意一种角色来发起。上级 DSA 的发起可以是由于管理干预而引起的。下级 DSA 的发起可以是由于一个移除条目操作而引起的，该操作移除了下级命名上下文的上下文前缀条目，或者是由于管理干预而引起的。

24.1.4 建立参数

一个 HOB 的两种角色，上级 DSA 和下级 DSA，它们的建立参数是不同的。上级 DSA 角色使用的建立参数是SuperiorToSubordinate 的一个值，下级 DSA 角色使用的建立参数是SubordinateToSuperior 的一个值。

24.1.4.1 上级 DSA 的建立参数

由上级 DSA 所发起的建立参数是SuperiorToSubordinate 的一个值，向下级 DSA 提供了关于新的命名上下文的上下文前缀的上级 DIT 顶点的相关信息(包含了直接上级引用)，可选的，还提供了下级上下文前缀条目的用户属性和操作属性，以及新上下文前缀的直接上级条目的用户属性和操作属性的拷贝。

```
SuperiorToSubordinate ::= SEQUENCE {
    contextPrefixInfo       [0]  DITcontext,
    entryInfo               [1]  SET SIZE (1..MAX) OF Attribute OPTIONAL,
    immediateSuperiorInfo   [2]  SET SIZE (1..MAX) OF Attribute OPTIONAL }
```

Vertex 或 SubentryInfo 中的rdn 应是主 RDN，且上下文信息和所有其他的可辨别值都应被包含在 RDN 的AttributeTypeAndDistinguishedValue 组件中，如同在 GB/T 16264.2—2008 的 9.3 中的描述。

24.1.4.1.1 上下文前缀信息

SuperiorToSubordinate 中的contextPrefixInfo 组件是类型为DITcontext 的一个值，这是一个Vertex 值的序列。

```
DITcontext ::= SEQUENCE OF Vertex
```

```
Vertex ::= SEQUENCE {
        rdn             [0]  RelativeDistinguishedName,
        admPointInfo    [1]   SET SIZE (1..MAX) OF Attribute OPTIONAL,
        subentries      [2]   SET SIZE (1..MAX) OF SubentryInfo OPTIONAL,
        accessPoints    [3]   MasterAndShadowAccessPoints OPTIONAL }
```

contextPrefixInfo 组件是 RDN 的序列，这些 RDN 构成了新的上下文前缀的直接上级的可辨别名，每个 RDN(由rdn 组件给定)可选地还伴随有附加的信息。

一个Vertex 中可选的admPointInfo 组件表示了该 DIT 顶点是一个管理点，并至少提供了它的administrativeRole 操作属性。

与一个管理点相关联的子条目信息由Vertex 中的subentries 组件提供，该组件是一个或多个SubentryInfo 值的集合。每个SubentryInfo 值由子条目的 RDN(rdn 组件)和子条目的属性(info 组件)组成。

```
SubentryInfo ::= SEQUENCE {
        rdn   [0]  RelativeDistinguishedName,
        info  [1]  SET OF Attribute }
```

一个 Vertex 中可选的accessPoints 组件表示了该顶点符合直接上级命名上下文的上下文前缀。上级使用此组件以便向下级提供其直接上级引用所需的信息。

注：accessPoints 中的主访问点与建立和修改操作绑定操作中传递到accessPoint 参数中的主访问点相同。

24.1.4.1.2 条目信息

SuperiorToSubordinate 中可选的entryInfo 组件是建立新的上下文前缀条目内容的一个属性集合。

24.1.4.1.3 直接上级条目信息

SuperiorToSubordinate 中可选的immediateSuperiorInfo 组件是新的上下文前缀的直接上级条目的属性集的一个拷贝，尤其是*objectClass* 和*entryACI*。

注：本组件可以被下级使用来优化一个列表请求的赋值，该列表请求针对某个基本客体产生了一个空的ListResult，而此基本客体是下级上下文前缀的直接上级(见 19.3.1.2.2 中 b)的注)。

24.1.4.2 下级 DSA 的建立参数

下级 DSA 发起的建立参数是SubordinateToSuperior 的一个值，向上级 DSA 提供了与下级命名上下文相关的信息。

```
SubordinateToSuperior ::= SEQUENCE {
        accessPoints    [0]  MasterAndShadowAccessPoints OPTIONAL,
        alias           [1]   BOOLEAN DEFAULT FALSE,
        entryInfo       [2]   SET SIZE (1..MAX) OF Attribute OPTIONAL,
        subentries      [3]   SET SIZE (1..MAX) OF SubentryInfo OPTIONAL }
```

SubordinateToSuperior 中的accessPoints 组件被下级使用，向上级提供其下级引用所需的信息。

注 1：accessPoints 中的主访问点与建立和修改操作绑定操作中传递到accessPoint 参数中的主访问点相同。

SubordinateToSuperior 中的alias 组件用于向上级表示该下级命名上下文包含一个单独的别名条目。

SubordinateToSuperior 中的entryInfo 组件包含了新的上下文前缀条目的属性集的一个拷贝，尤其是objectClass 和entryACI 属性，而且如果可应用的话，还包含administrativeRole 操作属性。

注 2：前两个属性可以被上级使用来优化一个列表请求或一个一级搜索请求的赋值，而这些请求的基本客体是下级上下文前缀的直接上级条目，同时，最后一个属性被用来避免对一个搜索操作进行不必要的处理，使其进入到某个服务特定的管理区或者从某个服务特定的管理区出来。

SubordinateToSuperior 中的subentries 组件被下级使用来向上级传递包含了预定 ACI 的子条目。

24.1.5 修改参数

对于 HOB 的修改，上级角色的修改参数SuperiorToSubordinateModification 是SuperiorToSubordinate，其中有一个限制为 entryInfo 组件可以不存在；下级角色的修改参数是SubordinateToSuperior。

```
SuperiorToSubordinateModification ::= SuperiorToSubordinate (
        WITH COMPONENTS { ..., entryInfo ABSENT})
```

这些参数与相应的建立参数是相同的(除了上面所标注的限制外)，并且被用于表示在 HOB 建立之后，建立参数中提供的信息所发生的变化。

如果 SuperiorToSubordinate（或者 SuperiorToSubordinateModification）、或者 SubordinateToSuperior 中的任何组件经历了一次变化(例如SuperiorToSubordinate 中的contextPrefixInfo 组件)，则修改参数中的相应组件(例如SuperiorToSubordinateModification 中的contextPrefixInfo 组件)应在修改操作绑定中被完整提供。

24.1.6 终止参数

当终止一个 HOB 时，两种角色都不提供终止参数。

24.1.7 类型标识

分等级操作绑定由客体标识符来标识，这些客体标识符是在 24.2 定义hierarchicalOperationalBindingOPERATIONAL-BINDING 信息客体时被分配的。

24.2 操作绑定信息客体类定义

本条使用 GB/T 16264.2—2008 中定义的OPERATIONAL-BINDING 信息客体类模板，定义了分等级操作绑定的类型。

```
hierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
    AGREEMENT       HierarchicalAgreement
    APPLICATION CONTEXTS {
                    {directorySystemAC} }
    ASYMMETRIC
        ROLE-A { --上级 DSA
            ESTABLISHMENT-INITIATOR      TRUE
            ESTABLISHMENT-PARAMETER      SuperiorToSubordinate
            MODIFICATION-INITIATOR       TRUE
            MODIFICATION-PARAMETER       SuperiorToSubordinateModification
            TERMINATION-INITIATOR        TRUE }
        ROLE-B {--下级 DSA
            ESTABLISHMENT-INITIATOR      TRUE
            ESTABLISHMENT-PARAMETER      SubordinateToSuperior
            MODIFICATION-INITIATOR       TRUE
            MODIFICATION-PARAMETER       SubordinateToSuperior
            TERMINATION-INITIATOR        TRUE }
    ID              id-op-binding-hierarchical }
```

24.3 分等级操作绑定管理的 DSA 规程

在下面的规程中，由一个 DSA 创建的一个新 DSE 或一个标记(即一个与信息的某些项相关联的状态指示)应被存储在一个静态存储器中。只有这样做，才能使得遵循下面规程的两个 DSA 在出现通信和终端系统失败时，有可能对 HOB 的参数维护一个一致的理解。

在下面所描述的establishment 和modification 两个规程中，承担了响应者角色的 DSA(即没有发起

建立或修改操作)可向承担了发起者角色的 DSA 提供信息(例如操作属性),而由于这样或那样的原因,这些信息是不可接受的。在这些情况下,发起者 DSA 可终止此操作绑定。

24.3.1 建立规程

24.3.1.1 上级 DSA 发起的建立

如果一个 DSA 与另一个在targetSystem 扩展中指定的不同 DSA 之间执行了一个增加条目操作,则该 DSA 应根据下列规程建立一个分等级的操作绑定。出于管理原因,如果一个 DSA 希望与一个下级 DSA 之间建立一个 HOB,且它支持 DOP HOB 协议,则应遵循下列规程:

a) 上级 DSA 创建一个类型为 subr 的新 DSE,其名(称)为新条目名,并且将这个新的 DSE 标记为"将被增加"。上级 DSA 产生一个唯一的 bindingID,并将其与新的 DSE 存储在一起。

b) 上级 DSA 应向下级 DSA 发送一个建立操作绑定操作,并包含下列参数:

1) bindingType 被设置为hierarchicalOperationalBindingID;

2) SuperiorToSubordinate 建立参数中,contextPrefixInfo 和entryInfo 组件存在;所有其他参数都是可选的;

3) HierarchicalAgreement 中,immediateSuperior 组件被设置为新条目的直接上级的可辨别名,且rdn 组件被设置为新条目的 RDN;以及

4) 适当的bindingID,myAccessPoint 和valid 参数。

c) 如果下级 DSA 接受此操作,则它将创建所需的类型相应为glue、subentry、admPoint、rhob 和 immSupr 的 DSE 来表示contextPrefixInfo;或者类型相应为cp 和entry 或alias 的 DSE 来表示新的上下文前缀客体或别名条目;以及类型相应为rhob 和entry 的 DSE 来表示immediateSuperiorInfo。它将bindingID 与新的上下文前缀条目的 DSE 存储在一起,并且向上级 DSA 返回一个SubordinateToSuperior 参数。

如果下级 DSA 拒绝此操作,则它会返回一个操作绑定差错,并且设置适当的问题值。

如果命名上下文已经存在,且已经存在的命名上下文与新的上下文的bindingID 值相同,即下级 DSA 已经创建了所请求的命名上下文,在这种情况下,下级 DSA 向上级 DSA 返回一个结果。如果这两个值不相同,则发送一个问题为invalidAgreement 的操作绑定差错;这就意味着上级 DSA 有一个永久的知识不一致,需要管理者的纠正。

d) 如果上级 DSA 接收到一个差错,则它移除类型为subr 的被标记 DSE,并且为增加条目操作返回一个差错。

如果上级 DSA 接收到一个结果,则它从表示subr 的 DSE 中移除标记,并且为增加条目操作返回一个结果。

如果有任何失败出现(如通信失败或终端系统失败),上级 DSA 都应从步骤 2)开始重复执行上述步骤,直到以该 DSA 为发起者的每个悬置的分等级操作绑定建立操作都接收到一个结果或差错为止。

如果建立是由于一个增加条目操作而引起的,且在建立完成之前由请求者中止了该操作(例如通过释放或中止应用联系),则上级 DSA 应忽略此事件,并且完成此次建立(可以成功,也可以不成功)。在这种情况下,增加条目操作的输出不会通知到用户。

注 1:对下级进行标注可以帮助恢复和进行同步控制。另一个用户不能增加一个已经被标注的条目,且在一次失败后,DSA 将为所有被标注的下级重复建立操作绑定。

注 2:通过上述规程,知识仅具有临时的不一致性。当下级引用被标注时,上级 DSA 如何处理那些对下级引用进行阅读的不相关操作属于本地事务。

24.3.1.2 下级 DSA 发起的建立

下级 DSA 可以发起建立一个分等级的操作绑定。这可以是由于某个管理者希望将该 DSA 内拥有的一棵条目子树与全球 DIT 中的某个点相关联起来而引起的。在这种情况下,下级 DSA 应根据下列

规程建立一个 HOB:

a) 下级 DSA 或者具有一个类型为cp 的 DSE 作为一个已经存在的命名上下文的一部分,或者创建一个新的这种 DSE。它将这个新的 DSE 标记为"将被增加",且产生一个唯一的bindingID,并将其与上下文前缀 DSE 存储在一起。

b) 下级 DSA 向上级 DSA 发起一个建立操作绑定操作,包含下列参数:

1) bindingType 被设置为hierarchicalOperationalBindingID;

2) 适当的*SubordinateToSuperior* 建立参数;

3) HierarchicalAgreement 中,immediateSuperior 组件被设置为新条目的直接上级的可辨别名,且rdn 组件被设置为新条目的 RDN;以及

4) 适当的bindingID,myAccessPoint 和valid 参数。

如果上级 DSA 拒绝此操作,则它返回一个操作绑定差错,并且设置适当的问题值。

c) 上级 DSA 检测出它是新的上下文前缀条目的直接上级的属主,或者返回一个问题为roleAssignment 的operationalBindingError。

d) 上级 DSA 检测被请求的新的上下文前缀的 RDN 是否尚未被使用。如果使用本地拥有的信息没有找到匹配的 RDN,但是直接上级 DSE 的类型为nssr,则遵循 19.1.5 中的规程。如果使用此规程没有发现匹配的 RDN,则上级 DSA 创建一个类型为 subr 的 DSE,并且将bindingID 与其存储在一起,然后返回一个结果。

如果发现一个下级引用与此 RDN 匹配,则比较两个bindingID 的值。如果它们相等,则返回一个结果。且由上级 DSA 所返回的*SuperiorToSubordinate* 参数中,不能够包含*entry* 组件。如果两个bindingID 值不相等,则发送一个问题为invalidAgreement 的operationalBindingError;这就意味着上级 DSA 具有一个永久的知识不一致,需要管理者的纠正。

如果通过部署一个 NSSR,发现一个匹配的 RDN,则发送一个问题为invalidAgreement 的operationalBindingError,这也意味着上级 DSA 具有一个永久的知识不一致,需要管理者的纠正。

e) 如果下级 DSA 接收到一个差错,它将移除新的上下文前缀 DSE 及其标记。如何决定该上下文前缀 DSE 所来源的条目信息的命运,属于本地事务。

如果下级 DSA 接收到一个结果,它将增加必要的相应类型为glue、subentry、admPoint、rhob和immSupr 的一个 DSE,来表示contextPrefixInfo;以及相应类型为 rhob 和 entry 的一个 DSE,来表示immediateSuperiorInfo。上下文前缀 DSE 的标记被移除。

如果有任何失败出现(如通信失败或终端系统失败),下级 DSA 都应从步骤 b)开始重复执行上述步骤,直到以该 DSA 为发起者的每个悬置的分等级操作绑定建立操作都接收到一个结果或差错为止。

24.3.2 修改规程

定义了下列规程来修改一个 HOB,该 HOB 是根据 24.3.1 详述的规程而建立的。

24.3.2.1 上级发起的修改规程

本规程的调用可以是由于修改操作引起的,如 19.1 中的描述,或者是由于管理干预而引起的(例如传递 HOB 的myAccessPoint、agreement 或valid 参数的变化)。另外,如果一个上级 DSA 检测出它提供给下级 DSA 的SuperiorToSubordinate 中的contextPrefixInfo 或immediateSuperiorInfo 组件发生了变化,则它应使用下列规程将新的信息传播给下级 DSA:

a) 将类型为subr 的 DSE 标记为"将被修改",并且如果这种修改是由于对下级上下文前缀条目的 RDN 进行修改而引起的,则一个新的 subr 类型为的 DSE 被增加,并被标记为"将被增加"。

b) 上级 DSA 根据已经存在的值产生一个新的bindingID 值,方法是增加其version 组件的值。使

用此新的bindingID，它向下级 DSA 发送一个修改操作绑定操作，且修改参数为SuperiorToSubordinateModification。

c) 下级 DSA 检查bindingID 中的identifier 组件。如果它与上级之间无此商定，或者如果version 组件小于 HOB 的版本，则它应返回一个问题为invalidAgreement 的operationalBindingError。

d) 下级 DSA 可接受对 HOB 的修改，即修改或重建表示上下文前缀信息的 DSE，更新它的bindingID 中的version 组件，并且返回一个结果。可选的，它也可返回一个差错并且终止此商定。

e) 如果上级 DSA 接收到一个结果，则修改完成。如果这种修改是由于对下级上下文前缀条目的 RDN 进行修改而引起的，则类型为subr，并被标注为“将被增加”的新的 DSE，其标注被移除，并且被标注为“将被修改”的老的 DSE 被移除。如果不是，则仅是“将被修改”的标注被简单移除。

如果上级 DSA 接收到一个差错，则修改失败。“将被修改”的标注被移除。如果这种修改是由于对下级上下文前缀条目的 RDN 进行修改而引起的，则类型为subr，并被标注为“将被增加”的新的 DSE 被移除。如果不是，则所采取的措施不在本目录规范的定义范围之内。

如果有任何失败出现(如通信失败或终端系统失败)，上级 DSA 都应从步骤 2)开始重复执行上述步骤，直到以该 DSA 为发起者的每个悬置的分等级操作绑定修改操作都接收到一个结果或差错为止。

如果修改是由于一个修改下级上下文前缀条目的 RDN 的ModifyDN 操作而引起的，且在修改完成之前由请求者中止了该操作(例如通过释放或中止应用联系)，则上级 DSA 应忽略此事件，并且完成此次修改(可以成功，也可以不成功)。在这种情况下，修改 DN 操作的输出不会被通知到用户。

24.3.2.2 下级发起的修改规程

本规程的调用可以是由于管理干预而引起的(例如传递 HOB 的myAccessPoint、agreement 或valid 参数的变化)。另外，如果一个下级 DSA 检测出它提供给上级 DSA 的SubordinateToSuperior 的值发生了变化，则它应使用下列规程将新的信息传播给上级 DSA：

a) 将类型为cp 的 DSE 标注为“将被修改”。

b) 下级 DSA 根据已经存在的值产生一个新的bindingID 值，方法是增加其version 组件的值。使用此新的bindingID，它向上级 DSA 发送一个修改操作绑定操作，且修改参数为Subordinate ToSuperior。

c) 上级 DSA 检查bindingID 中的identifier 组件。如果它与下级之间无此商定，或者如果version 组件小于 HOB 的版本，则它应返回一个问题为invalidAgreement 的operationalBindingError。

d) 上级 DSA 可接受对 HOB 的修改，即修改表示下级引用的 DSE，并返回一个结果。可选的，它也可返回一个差错并且终止此商定。

另外，如果这个将被重命名的 DSE(类型为subr)的上级 DSE 的类型为nssr，则 DSA 在响应 HOB 修改请求之前，应遵循 19.1.5 中定义的规程(修改操作和 NSSR)以确保条目的新名(称)是无二义性的。

e) 如果下级 DSA 接收到一个结果，则修改完成，并且移除了标记。如果它接收到一个差错，则所采取的措施不在本目录规范的定义范围之内。

如果有任何失败出现(如通信失败或终端系统失败)，下级 DSA 都应从步骤 b)开始重复执行上述步骤，直到以该 DSA 为发起者的每个悬置的分等级操作绑定修改操作都接收到一个结果或差错为止。

24.3.3 终止规程

定义了下列规程来终止一个 HOB,该 HOB 是根据 24.3.1 详述的规程而建立的。

24.3.3.1 上级 DSA 发起的终止

上级 DSA 发起的对分等级操作绑定的终止只能是由于管理干预而引起的。下列规程应被遵循:

a) 上级 DSA 将表示下级引用的 DSE 标注为"将被移除",因此在名(称)解析过程中,该下级引用不能再被使用。

b) 针对分等级操作绑定,上级 DSA 向下级 DSA 发出一个终止操作绑定的操作。bindingID 中的 version 组件被上级忽略。

c) 当下级 DSA 接收到该终止操作绑定,则它将移除关于此分等级操作绑定的所有信息,并且发送一个结果,除非bindingID 中的identifier 组件未知,在这种情况下,将返回一个问题为invalid ID 的operationalBindingError。如何决定与该下级命名上下文相关的条目信息的命运,属于本地事务。

d) 如果上级 DSA 接收到一个结果,或者一个问题为invalidID 的operationalBindingError,则它应移除标记为"将被移除"的 DSE,该 DSE 表示了与分等级操作绑定相关的下级引用,并且移除与操作绑定相关的所有信息。

如果有任何失败出现(如通信失败或终端系统失败),上级 DSA 都应从步骤 b)开始重复执行上述步骤,直到以该 DSA 为发起者的每个悬置的分等级操作绑定终止操作都接收到一个结果或差错为止。

24.3.3.2 下级 DSA 发起的终止

由下级 DSA 发起的终止规程可以是由于一个移除条目操作而引起的,该操作移除了下级命名上下文中的最后一个条目,即上下文前缀条目,或者是由于管理干预而引起的。应遵循下列规程:

a) 下级 DSA 将命名上下文的上下文前缀 DSE 标记为"将被移除"。

b) 下级 DSA 向上级 DSA 发起一个关于分等级操作绑定的终止操作绑定操作。bindingID 中的 version 组件被下级 DSA 忽略。

c) 当上级 DSA 接收到该终止操作绑定,则它将移除表示下级引用的 DSE,此下级引用是与分等级操作绑定相关联的,同时移除关于此操作绑定的所有信息,并且发送一个结果,除非binding ID 中的identifier 组件未知,在这种情况下,将返回一个问题为invalidID 的operationalBindingError。

d) 如果下级 DSA 接收到一个结果或者一个问题为invalidID 的operationalBindingError,则它应移除关于此操作绑定的所有信息。

注:命名上下文的条目信息的命运对下级 DSA 来说是一种本地事务。由于重新命名(即移动)一个命名上下文是不被修改 DN 操作所允许的,则一个管理者可终止 HOB,为命名上下文选择另一个上下文前缀,并且将其与 DIT 的另一部分重新连接起来(即建立一个新的 HOB)。

如果有任何失败出现(如通信失败或终端系统失败),下级 DSA 都应从步骤 2)开始重复执行上述步骤,直到以该 DSA 为发起者的每个悬置的分等级操作绑定终止操作都接收到一个结果或差错为止。

24.4 操作规程

在一个分等级操作绑定的合作状态中可以被执行的操作是在应用上下文 directorySystemAC 中定义的那些操作。

在一个分等级操作绑定中包含的 DSA 所应遵循的规程在第 16 章到第 22 章中定义。

24.5 应用上下文的用法

为了能够使用本目录标准中的协议和规程来建立、修改或终止一个分等级操作绑定,一个 DSA 应使用operationalBindingManagementAC 应用上下文。

25 非特定分等级操作绑定

一个非特定分等级操作绑定被用来表示两个 DSA 之间的关系，这两个 DSA 拥有两个命名上下文，其中一个是另一个的直接下级。在 NHOB 的情况下，上级 DSA 拥有一个指向下级 DSA 所拥有的命名上下文的非特定下级引用；下级 DSA 拥有一个指向上级 DSA 所拥有的命名上下文的直接上级引用。操作绑定确保了在两个 DSA 之间可以交换和维护适当的知识信息，因此，两个 DSA 都能够在名(称)解析和操作赋值的处理过程中，按照第 18 章和第 19 章的定义来运转。

25.1 操作绑定类型特性

25.1.1 对称和角色

分等级操作绑定类型是一种非对称的操作绑定类型。在这种类型的绑定中有两种角色，分别为：

a) 上级命名上下文的主 DSA 所承担的角色，称为上级 DSA(相关的抽象角色为“A”)；以及

b) 下级命名上下文的主 DSA 所承担的角色，称为下级 DSA(相关的抽象角色为“B”)。

25.1.2 商定

在建立非特定分等级操作绑定的过程中，所交换的商定信息是NonSpecificHierarchicalAgreement 的一个值。它仅包括新的命名上下文的直接上级条目的可辨别名(immediateSuperior 组件)。该信息应被发起该 NHOB 的 DSA 所提供。

```
NonSpecificHierarchicalAgreement ::= SEQUENCE {
        immediateSuperior    [1]    DistinguishedName }
```

注：下级 DSA 如何判断新的命名上下文的名(称)是无二义性的，不在本部分的定义范围之内。如果名(称)被有关命名机构正确分配，且没有其他的 DSA 拥有与此名(称)相同的一个主条目，则该名(称)是无二义性的。

25.1.3 发起者

25.1.3.1 建立

一个非特定分等级操作绑定的建立仅可以由下级 DSA 角色来发起。下级 DSA 的发起(将一个或多个本地存在的条目或子树与全球 DIT 连接起来)是由于管理干预而引起的。

25.1.3.2 修改

一个非特定分等级操作绑定的修改可以由任意一种角色来发起。由于上级上下文前缀信息的修改，使得上级 DSA 可发起修改。这可以是任何修改操作的结果，或者是管理干预的结果。

如果该命名上下文(或者在下级角色的情况下，它的直接下级命名上下文中的一个)的访问点信息发生了变化，则任何一个 DSA 也都可以修改此 NHOB。

25.1.3.3 终止

一个分等级操作绑定的终止可以由任意一种角色来发起。上级 DSA 的发起是由于管理干预而引起的。下级 DSA 的发起可以是由于一个移除条目操作而引起的，该操作移除了商定中 immediateSuperior 组件的直接下级所拥有的最后一个上下文前缀条目，或者是由于管理干预而引起的。

25.1.4 建立参数

上级 DSA 发起的建立参数是NHOBSuperiorToSubordinate 的一个值，它与相应的 HOB 建立参数相同，除了entryInfo 组件是缺失的。

```
NHOBSuperiorToSubordinate ::= SuperiorToSubordinate (
        WITH COMPONENTS { ..., entryInfo ABSENT})
```

下级 DSA 发起的建立参数是NHOBSubordinateToSuperior 的一个值，它与相应的 HOB 建立参数是相同的，除了alias 和entryInfo 组件是缺失的。

```
NHOBSubordinateToSuperior ::= SEQUENCE {
        accessPoints    [0]  MasterAndShadowAccessPoints OPTIONAL,
        subentries      [3]   SET SIZE (1..MAX) OF SubentryInfo OPTIONAL }
```

25.1.5 修改参数

这些参数与相应的建立参数是相同的，并且被用于表示在 NHOB 建立之后，建立参数中提供的信息所发生的变化。

如果NHOBSuperiorToSubordinate 或者NHOBSubordinateToSuperior 中的任何组件经历了一次变化(例如NHOBSuperiorToSubordinate 中的contextPrefixInfo 组件)，则修改参数中的相应组件(例如NHOBSuperiorToSubordinate 中的contextPrefixInfo 组件)应在修改操作绑定中被完整提供。

25.1.6 终止参数

当终止一个 NHOB 时，两种角色都不提供终止参数。

25.1.7 类型标识

非特定分等级操作绑定由客体标识符来标识，这些客体标识符是在 25.2 定义nonSpecificHierarchicalOperationalBinding OPERATIONAL-BINDING 信息客体时被分配的。

25.2 操作绑定信息客体类定义

本条使用 GB/T 16264.2—2008 中定义的OPERATIONAL-BINDING 信息客体类模板，定义了非特定分等级操作绑定的类型。

```
nonSpecificHierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
    AGREEMENT       NonSpecificHierarchicalAgreement
    APPLICATION CONTEXTS {
                { directorySystemAC } }
    ASYMMETRIC
        ROLE-A {        ——上级 DSA
            ESTABLISHMENT-PARAMETER     NHOBSuperiorToSubordinate
            MODIFICATION-INITIATOR      TRUE
            MODIFICATION-PARAMETER      NHOBSuperiorToSubordinate
            TERMINATION-INITIATOR       TRUE }
        ROLE-B {        ——下级 DSA
            ESTABLISHMENT-INITIATOR     TRUE
            ESTABLISHMENT-PARAMETER     NHOBSubordinateToSuperior
            MODIFICATION-INITIATOR      TRUE
            MODIFICATION-PARAMETER      NHOBSubordinateToSuperior
            TERMINATION-INITIATOR       TRUE }
    ID id-op-binding-non-specific-hierarchical }
```

25.3 非特定分等级操作绑定管理的 DSA 规程

在下面的规程中，如同 24.3 中描述的规程那样，由一个 DSA 创建的一个新 DSE 或一个标记应被存储在一个静态存储器中。

在下面所描述的建立和修改两个规程中，承担了响应者角色的 DSA(即没有发起建立或修改操作)可向承担了发起者角色的 DSA 提供信息(例如操作属性)，而由于这样或那样的原因，这些信息是不可接受的。在这些情况下，发起者 DSA 可终止此操作绑定。

25.3.1 建立规程

仅有下级 DSA 才可发起一个分等级操作绑定。这可以是由于一个管理者希望将 DSA 内拥有的一个或多个条目子树与全球 DIT 中的某个点相关联起来而引起的。在这种情况下，下级 DSA 应根据下列规程建立一个 NHOB：

a) 下级 DSA 或者具有一个类型为cp 的 DSE，该 DSE 是一个已经存在的命名上下文的一部分，或者创建一个新的这种 DSE。它将此 DSE 标记为“将被增加”，且产生一个唯一的bindingID，

并将其与上下文前缀 DSE 存储在一起。

b) 下级 DSA 向上级 DSA 发送一个建立操作绑定操作，并包含下列参数：

1) bindingType 被设置为nonSpecificHierarchicalOperationalBindingID；

2) 适当的NHOBSubordinateToSuperior 建立参数；

3) NonSpecificHierarchicalAgreement 中的immediateSuperior 组件被设置为新条目的直接上级的可辨别名；

4) 适当的bindingID、myAccessPoint 和valid 参数。

c) 上级 DSA 检测出它是新的上下文前缀条目的直接上级的属主，或者返回一个问题为roleAssignment 的operationalBindingError。

d) 上级 DSA 将 DSE 类型nssr（以及nonSpecificKnowledge 属性信息）加入到新条目的直接上级的 DSE 中，并将bindingID 与其一起存储起来，然后返回一个结果。

e) 如果下级 DSA 接收到一个差错，它将移除新的上下文前缀 DSE 及其标记。如何决定该上下文前缀 DSE 所来源的条目信息的命运，属于本地事务。

如果下级 DSA 接收到一个结果，它将增加必要的相应类型为glue、subentry、admPoint、rhob 和immSupr 的一个 DSE，来表示 contextPrefixInfo；以及相应类型为 rhob 和 entry 的一个 DSE，来表示immediateSuperiorInfo。上下文前缀 DSE 的标记被移除。

如果有任何失败出现（如通信失败或终端系统失败），下级 DSA 都应从步骤 2）开始重复执行上述步骤，直到以该 DSA 为发起者的每个悬置的分等级操作绑定建立操作都接收到一个结果或差错为止。

25.3.2 修改规程

如果上级 DSA 检测出在一个非特定分等级操作绑定内，它提供给下级 DSA 的NHOBSuperiorToSubordinate 信息发生了任何变化，则它应将发生变化的信息传播给下级 DSA。如果 NHOB 是使用 25.3.1 中的规程而建立的，则它的修改应根据 24.3.2.1 中为修改分等级操作绑定而定义的规程（其中，NHOBSuperiorToSubordinate 取代了SuperiorToSubordinateModification）。

类似的，如果下级 DSA 检测出它提供给上级 DSA 的NHOBSubordinateToSuperior 信息发生了任何变化，则它应将此变化传播给上级 DSA。如果 NHOB 是使用 25.3.1 中的规程而建立的，则它的修改应根据 24.3.2.2 中为修改分等级操作绑定而定义的规程（其中，NHOBSubordinateToSuperior 取代了SubordinateToSuperior）。

25.3.3 终止规程

定义了下列规程来终止一个 NHOB，该 NHOB 是根据 25.3.1 详述的规程而建立的。

25.3.3.1 上级 DSA 发起的终止

上级 DSA 发起的对分等级操作绑定的终止只能是由于管理干预而引起的。下列规程应被遵循：

a) 上级 DSA 将nonSpecificKnowledge 属性中对应于下级 DSA 的值标注为“将被移除”，该属性由直接上级条目的 DSE 所拥有。

b) 针对与下级 DSA 之间的 NHOB，上级 DSA 发送一个终止操作绑定操作。bindingID 中的version 组件被上级 DSA 忽略。

c) 当下级 DSA 接收到该终止操作绑定的操作时，它将移除关于此 NHOB 的所有信息，并且发送一个结果，除非bindingID 中的identifier 组件未知，在这种情况下，将返回一个问题为invalidID 的operationalBindingError。如何决定与该下级命名上下文相关的条目信息的命运，属于本地事务。

d) 如果上级 DSA 接收到一个结果，或者一个问题为invalidID 的operationalBindingError，则它应删除标记为“将被删除”的nonSpecificKnowledge 属性的值，该属性表示了与 NHOB 相关的访问点信息，并且删除与操作绑定相关的所有信息。如果这是nonSpecificKnowledge 属性的最

后一个值，则它从 DSE 中删除此nonSpecificKnowledge 属性和 DSE 类型nssr。如果有任何失败出现(如通信失败或终端系统失败)，上级 DSA 都应从步骤 2)开始重复执行上述步骤，直到以该 DSA 为发起者的每个悬置的 NHOB 终止操作都接收到一个结果或差错为止。

25.3.3.2 下级 DSA 发起的终止

由下级 DSA 发起的终止规程可以是由于一个移除条目操作而引起的，该操作移除了下级命名上下文中的最后一个条目，即由下级 DSA 所拥有的最后一个下级命名上下文的上下文前缀条目，或者是由于管理干预而引起的。

应遵循下列规程：

a) 下级 DSA 将命名上下文的上下文前缀 DSE 标注为“将被删除”。

b) 下级 DSA 向上级 DSA 发起一个关于分等级操作绑定的终止操作绑定操作。bindingID 中的 version 组件被下级 DSA 忽略。

c) 当上级 DSA 接收到该终止操作绑定的操作，它将移除表示与 NHOB 相关的访问点信息的 nonSpecificKnowledge 属性的值，同时移除关于此操作绑定的所有信息，从下级命名上下文的直接上级 DSE 中移除nonSpecificKnowledge 属性和 DSE 类型nssr（如果被移除的值是non-SpecificKnowledge 属性的最后一个值），并且发送一个结果，除非bindingID 中的identifier 组件未知，在这种情况下，将返回一个问题为invalidID 的operationalBindingError。

d) 如果下级 DSA 接收到一个结果或者一个问题为invalidID 的operationalBindingError，则它应移除关于此操作绑定的所有信息。如何决定与该下级命名上下文相关的条目信息的命运，属于本地事务。

如果有任何失败出现(如通信失败或终端系统失败)，下级 DSA 都应从步骤 2)开始重复执行上述步骤，直到以该 DSA 为发起者的每个悬置的终止 NHOB 操作都接收到一个结果或差错为止。

25.4 操作规程

在一个非特定分等级操作绑定的合作状态中可以被执行的操作是在应用上下文directorySystemAC 中定义的那些操作。

在一个非特定分等级操作绑定中包含的 DSA 所应遵循的规程在第 16 章到第 22 章中定义。

25.5 应用上下文的用法

为了能够使用本目录标准中的协议和规程来建立、修改或终止一个非特定分等级操作绑定，一个 DSA 应使用operationalBindingManagementAC 应用上下文。

附 录 A
（规范性附录）
分布式操作的 ASN.1 定义

本附录包含了本目录规范中的所有 ASN.1 类型和值定义，以 ASN.1 模块DistributedOperations的形式提供。

```
DistributedOperations {joint-iso-ITU-T ds(5) module(1) distributedOperations(3) 5}
DEFINITIONS ::=
BEGIN

--EXPORTS All --
```

——本模块中定义的所有类型与值都可被输出到本系列目录规范所包含的其他的 ASN.1 模块中，供其使用，

——也可以被其他应用所使用，这些应用将使用本模块中的定义来访问目录服务。

——其他应用可将本模块中的定义用于其自身目的，但是不会限制为了维护和改进目录服务而进行的扩展和修改。

```
--来自GB/T 16264.2—2008

    basicAccessControl, commonProtocolSpecification, directoryAbstractService,
    enhancedSecurity, informationFramework, selectedAttributeTypes,
    serviceAdministration, upperBounds
        FROM UsefulDefinitions {joint-iso-ITU-T ds(5) module(1) usefulDefinitions(0) 5}

    DistinguishedName, Name, RDNSequence
        FROM InformationFramework informationFramework

    MRMapping, SearchRuleId
        FROM ServiceAdministration serviceAdministration

    AuthenticationLevel
        FROM BasicAccessControl basicAccessControl

    OPTIONALLY-PROTECTED{ }
        FROM EnhancedSecurity enhancedSecurity

--来自GB/T 16264.3—2008

    abandon, addEntry, CommonResults, compare, directoryBind, list, modifyDN, modifyEn-
    try, read, referral, removeEntry, search, SecurityParameters
        FROM DirectoryAbstractService directoryAbstractService
```

——来自*GB/T 16264.5—2008*

```
        ERROR, id-errcode-dsaReferral, OPERATION
            FROM CommonProtocolSpecification commonProtocolSpecification
```

——来自*GB/T 16264.6—2008*

```
        DirectoryString{}, PresentationAddress, ProtocolInformation, UniqueIdentifier
            FROM SelectedAttributeTypes selectedAttributeTypes

        ub-domainLocalID, ub-labeledURI
            FROM UpperBounds upperBounds;

--派生链接操作所需的参数化类型--

        chained { OPERATION : operation }
        OPERATION ::= { ARGUMENT
        OPTIONALLY-PROTECTED {
            SET {
                chainedArgument ChainingArguments,
                argument        [0]
                    operation.&ArgumentType } } RESULT
                    OPTIONALLY-PROTECTED {
            SET {
                chainedResult   ChainingResults,
                result          [0]
             operation.&ResultType } } ERRORS
        { operation.&Errors EXCEPT referral | dsaReferral }
        CODE    operation.&operationCode }
--绑定和解绑定操作--
dSABind                 ::=directoryBind
--dSAUnbind             ::=directoryUnbind
--链接操作--
chainedRead             ::=chained { read }
chainedCompare          ::=chained { compare }
chainedAbandon          ::=abandon
chainedList             ::=chained { list }
chainedSearch           ::=chained { search }
chainedAddEntry         ::=chained { addEntry }
chainedRemoveEntry      ::=chained { removeEntry }
chainedModifyEntry      ::=chained { modifyEntry }
chainedModifyDN         ::=chained { modifyDN }
--差错和参数--
```

```
dsaReferral     ERROR ::= {
    PARAMETER
        OPTIONALLY-PROTECTED
        { SET {
            reference       [0]     ContinuationReference,
            contextPrefix   [1]     DistinguishedName OPTIONAL,
            COMPONENTS OF
    CommonResults } } CODE
                id-errcode-dsaReferral }
--公共变元和结果--
ChainingArguments   ::= SET {
        originator                  [0]     DistinguishedName OPTIONAL,
        targetObject                [1]     DistinguishedName OPTIONAL,
        operationProgress           [2]     OperationProgress
                                    DEFAULT { nameResolutionPhase notStarted },
        traceInformation            [3]     TraceInformation,
        aliasDereferenced           [4]     BOOLEAN DEFAULT FALSE,
        aliasedRDNs                 [5]     INTEGER OPTIONAL,
                        ——仅在第 1 版本系统中存在
        returnCrossRefs             [6]     BOOLEAN DEFAULT FALSE,
        referenceType               [7]     ReferenceType DEFAULTsuperior,
        info                        [8]     DomainInfo OPTIONAL,
        timeLimit                   [9]     Time OPTIONAL,
        securityParameters          [10]    SecurityParameters DEFAULT { },
        entryOnly                   [11]    BOOLEAN DEFAULT FALSE,
        uniqueIdentifier            [12]    UniqueIdentifier OPTIONAL,
        authenticationLevel         [13]    AuthenticationLevel OPTIONAL,
        exclusions                  [14]    Exclusions OPTIONAL,
        excludeShadows              [15]    BOOLEAN DEFAULT FALSE,
        nameResolveOnMaster         [16]    BOOLEAN DEFAULT FALSE,
        operationIdentifier         [17]    INTEGER OPTIONAL,
        searchRuleId                [18]    SearchRuleId OPTIONAL,
        chainedRelaxation           [19]    MRMapping     OPTIONAL,
        relatedEntry                [20]    INTEGER OPTIONAL,
        dspPaging                   [21]    BOOLEAN DEFAULT FALSE,
        nonDapPdu                   [22]    ENUMERATED { ldap (0) } OPTIONAL,
        streamedResults             [23]    INTEGER OPTIONAL,
        excludeWriteableCopies      [24]    BOOLEAN DEFAULT FALSE }
Time ::= CHOICE {
        utcTime
                UTCTime,
        generalizedTime
        GeneralizedTime }
```

```
DomainInfo    ::= ABSTRACT-SYNTAX.&Type

ChainingResults ::= SET {
        info                    [0]  DomainInfo OPTIONAL,
        crossReferences         [1]  SEQUENCE SIZE (1..MAX) OF CrossReference OPTIONAL,
        securityParameters      [2]  SecurityParameters DEFAULT { },
        alreadySearched         [3]  Exclusions OPTIONAL } CrossReference ::= SET {
CrossReference  ::= SET {
        contextPrefix           [0]  DistinguishedName,
        accessPoint             [1]  AccessPointInformation }
OperationProgress ::= SET {
        nameResolutionPhase     [0]  ENUMERATED {
          notStarted     (1),
          proceeding     (2),
          completed      (3) },
        nextRDNToBeResolved     [1]  INTEGER OPTIONAL }
TraceInformation ::= SEQUENCE OF TraceItem
TraceItem ::= SET {
        dsa                     [0]  Name,
        targetObject            [1]  Name OPTIONAL,
        operationProgress       [2]  OperationProgress }
ReferenceType ::= ENUMERATED {
        superior                     (1),
        subordinate                  (2),
        cross                        (3),
        nonSpecificSubordinate       (4),
        supplier                     (5),
        master                       (6),
        immediateSuperior            (7),
        self                         (8),
        ditBridge                    (9) }
AccessPoint ::= SET {
        ae-title                [0]  Name,
        address                 [1]  PresentationAddress,
        protocolInformation     [2]  SET SIZE (1..MAX) OF ProtocolInformation OPTIONAL
        labeledURI              [6]  LabeledURI OPTIONAL }
LabeledURI ::= DirectoryString{ub-labeledURI}
MasterOrShadowAccessPoint ::= SET {
        COMPONENTS OF           AccessPoint,
        category                [3]  ENUMERATED {
            master              (0),
            shadow              (1) } DEFAULT master,
```

```
        chainingRequired        [5]  BOOLEAN DEFAULT FALSE }
MasterAndShadowAccessPoints ::= SET SIZE (1..MAX) OF MasterOrShadowAccessPoint
AccessPointInformation ::= SET {
        COMPONENTS              MasterOrShadowAccessPoint,
        OF
        additionalPoints        [4]  MasterAndShadowAccessPoints OPTIONAL }
DitBridgeKnowledge ::= SEQUENCE {
        domainLocalID           DirectoryString{ub-domainLocalID}
        OPTIONAL, accessPoints      MasterAndShadowAccessPoints }
Exclusions ::= SET SIZE (1..MAX) OF RDNSequence
ContinuationReference ::= SET {
        targetObject            [0]  Name,
        aliasedRDNs             [1]  INTEGER OPTIONAL,——仅在第1版本系统中存在
        operationProgress       [2]  OperationProgress,
        rdnsResolved            [3]  INTEGER OPTIONAL,
        referenceType           [4]  ReferenceType,
        accessPoints            [5]  SET OF AccessPointInformation,
        entryOnly               [6]  BOOLEAN DEFAULT FALSE,
        exclusions              [7]  Exclusions OPTIONAL,
        returnToDUA             [8]  BOOLEAN DEFAULT FALSE,
        nameResolveOnMaster     [9]  BOOLEAN DEFAULT FALSE }

END -- DistributedOperations
```

附 录 B
（资料性附录）
分布式名（称）解析的示例

图 B.1 是一个关于如何使用分布式名（称）解析来处理不同的目录请求的示例。本例的基础是 GB/T 16264.2—2008 的附录 O（知识的建模）中所描述的一棵假设的 DIT 以及相应的 DSA 配置，为了便于阅读在这里复制如下。

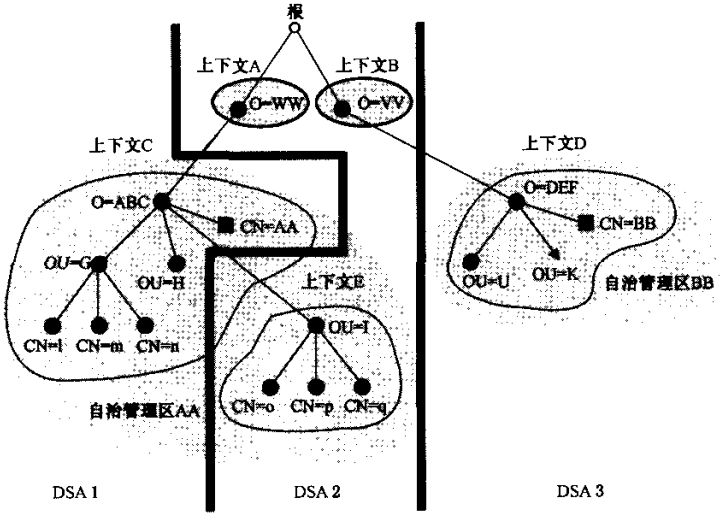


图 B.1 映射到三个 DSA 上的假设 DIT

假设一种链接的传播模式，发给 DSA 1 的下列请求应按如下处理：

a) 一个请求，其中可辨别名为{C = WW, O = ABC, OU = G , CN = l}

——名（称）解析将对目标名（称）中的每个 RDN 与 DSA 1 所拥有的 DSE 进行成功地匹配，直到目标 DSE 被定位。

b) 一个请求，其中可辨别名为{C = WW, O = JPR}

——DSA 1 中的名（称）解析规程将匹配到 DSE C = WW，但无法进行后续的匹配。在这点上，DSA 1 潜在地发现两个引用可以帮助它来继续进行：一个是在 DSE C=WW 中的immSupr 引用；另一个是在根 DSE 中的 supr 引用。在这个假设的示例中，两个都指向 DSA 2。因此，该请求被链接至 DSA 2。

——在 DSA 2 中，名（称）解析规程将匹配到 DSE C = WW，但无法进行后续的匹配。在这种情况下，由于 DSE C = WW 的类型为cp 和entry，且 DSA 2 是该条目的主 DSA，而在 C=WW 处没有nssr，因此 DSA 2 可以判断出在本目录中没有此名（称）。一个问题为 *noSuchObject* 的*nameError* 将被返回。

c) 一个请求，其中可辨别名为{C = VV, O = DEF, OU = K}

——DSA 1 中的名（称）解析规程不能匹配到任何一个 DSE。仅有的一个可用的引用是根 DSE 中的supr 引用，该引用指向 DSA 2。因此该请求被链接至 DSA 2。

——在 DSA 2 中，名（称）解析规程将匹配到 DSE C = VV，然后匹配到 DSE O = DEF，然后就无法进行后续的匹配了。由于 DSE O = DEF 被发现其类型为subr，这个特定的指向 DSA 3 的知识引用被使用，并将本请求链接至 DSA 3。

——在 DSA 3 中，名(称)解析规程将匹配到完整的目标客体名，并且发现这个被定位的 DSE 类型为alias。假设在这种情况下，别名将被解除引用，使用包含在已匹配 DSE 中的aliasedEntryName，可以构建一个新的名(称)。于是 DSA 3 将重新进入到名(称)解析规程中继续进行处理。

附 录 C
（资料性附录）
鉴别的分布式使用

C.1 摘要

安全模式在 GB/T 16264.2—2008 的第 17 章定义。下面是该模型的要点摘要：

a） 在 DSP 中支持强鉴别，方法是对请求、结果和差错等进行签名。

b） 在 DSP 中支持对请求、结果和差错等进行加密。

本附录描述了在分布式目录中如何实现这些要求。它使用了 ISO/IEC 9594-8:2005 中定义的术语和表示法。

C.2 分布式保护模型

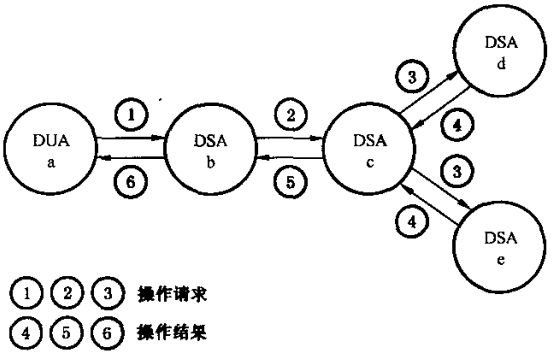


图 C.1 分布式保护

图 C.1 举例说明了在规定分布式保护过程中所使用的模型。该模型标识了在一般情况下列表操作或搜索操作的信息流序列。该操作被认为是由 DUA 'a'发起的，引用了一个位于执行操作的DSA 'c'内的目标客体，DSA 'b'、'c'、'd'和'e'也被包含在内。

最初，DUA 'a'与没有拥有目标客体的任何 DSA(DSA 'b')交互，但该 DSA 能够通过链接，将请求导航到拥有目标客体的一个 DSA(DSA 'c')。如果所有的 DSA 都可以以转向推荐方式运行，则该模型将被明显地简化，且每个 DSA/DSA 的交互，从保护来看，将等同于 DUA 'a'和 DSA 'b'之间的交互。

C.2.1 保护的质量

在应用联系的生命周期中使用的保护质量是在目录绑定操作过程中建立的。系统策略将对 DUA 和 DSA 应遵守的保护级别进行评估。DIRQOP 是一个信息客体类，可被用于规范与每个操作(请求、结果或差错)相关联的保护质量。DUA 在DirectoryBindArgument 中传送DIRQOP 信息客体类，而 DSA 在DirectoryBindResult 中接受此保护级别。保护质量可被用于提供下列类型的保护：签名的、加密的、或签名并加密的。

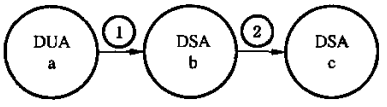
C.3 签名的链接操作

如果支持数字签名的链接操作，则由 DUA 来负责验证 DSA 所返回的列表或搜索结果中的数字签名。如果使用一个分布式环境来产生列表或搜索结果，则要求 DUA 有能力验证来自多个 DSA 的数字签名。将列表和搜索操作的结果相关联起来是 DUA 的责任。DSA 不必代表 DUA 来合并这些结果。

在某些情况下，DUA 可从多个不同的 DSA 中接收信息，而每个 DSA 支持不同级别的鉴别和数字签名。则由 DUA 来决定当数字签名非法时，是否使用所返回的信息。

C.3.1 链接的签名变元

如果 DUA 对一个 DAP 变元进行了签名，则该签名应在请求的生命周期内都被维护。当 DSA 执行访问控制验证时，该签名可被 DSA 验证和使用。如果 DSA 决定该请求需要链接到另一个 DSA 来处理时，它应当在必要的链接变元中，包含此 DUA 签名的请求。如果该 DSA 准备支持签名的 DSP 操作（DSA 到 DSA），则 DSA 的许可证将被用于对 DSP 的ChainingArguments 进行签名，且 DUA 的签名应当与原始的 DAP 请求一起被维护。

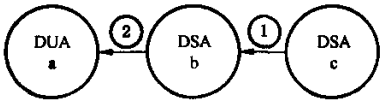


① DUA'a'用户对DAP请求签名

② DUA'b'对DSP链接变量签名
（由DUA'a'用户签名的DAP请求）

C.3.2 链接的签名结果

如果 DUA 用户希望从目录中接收到签名的结果，则SecurityParameters. ProtectionRequest 字段应当被设置为SIGNED。远端 DSA 应当有能力被配置为可以发送数字签名后的ChainingResults。可选地，远端 DSA 能够对 DAP 结果和 DSP 的ChainingResults 进行签名，以便支持端到端的签名。DSA 'b'将负责验证远端 DSA 的 DSP 签名，而 DUA 'a'将负责验证 DSA 的 DAP 结果签名。

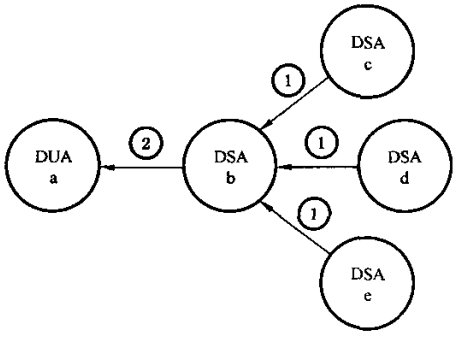


① DSA'c'对DSP链接结果和DAP结果签名

② DSA'b'返回由DSA'c'签名的DAP结果

C.3.3 合并签名的列表或搜索结果

如果使用一个分布式环境来产生列表或搜索结果，则要求 DUA 有能力验证来自多个 DSA 的数字签名。将列表和搜索操作的结果合并起来是 DUA 的责任。DSA 不必代表 DUA 来合并这些结果。在某些情况下，DUA 可从多个不同的 DSA 中接收信息，而每个 DSA 支持不同级别的鉴别和数字签名。这种时候由 DUA 来决定当数字签名非法时，是否使用所返回的信息。



① DSA'c','d','e'对DSP链接结果（由DSA'c','d','e'签名的DAP结果）签名

② DSA'b'返回由DSA'c','d'和'e'签名的部分DAP结果，DSA'b'不合并DAP结果

注：DSA 到 DSA 的 DSP 协议也能够被签名，被加密或被签名并加密。

C.3.4 多链接请求

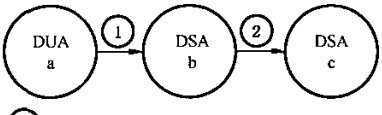
如果 DSA 判断 DAP 请求需要被链接到多个其他 DSA 时，它能够将请求进行多链接，或者以并行方式，或者以顺序方式。有两种分解模式被描述：非特定下级引用(NSSR)分解，或请求分解。在 NSSR 分解中，DSA 向其他指定的 DSA 发送相同的请求。在请求分解中，DSA 向其他的每个 DSA 发送一个部分的(可能是不同的)并发的请求。

C.4 加密的链接操作

如果支持加密，则在目录的每个组件之间需要提供等同的保护。要形成一个有关策略等同性的协定，需要使用映射，但此映射不在本目录规范的定义范围之内。

C.4.1 对请求的点到点(DUA→DSA 或 DSA→DSA)加密

如果一个 DUA 用户想对 DAP 请求进行加密，则加密仅能够基于点到点方式出现。DUA 将为 DSA 'b'加密该 DAP 请求；然而，DUA 用户不知道该请求最终是否会被链接到一个远端 DSA 来处理。DSA 'b'将解密该请求，并且尝试满足该请求。如果 DSA 'b'判断该请求应当被链接到另一个 DSA (DSA 'c')来处理，则 DSA 'b'为 DSA 'c'加密该链接操作。为 DSP 请求和响应(链接的操作变元和结果)选择点对点保护是由 DSA 'b'和 DSA 'c'之间在 DSP 绑定中建立的 dirqop 来指示的。

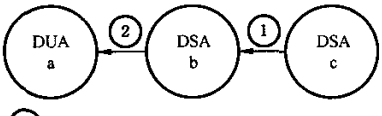


① DUA'a'用户为DSA'b'加密DAP请求

② DSA'b'对DSP链接操作变量加密

C.4.2 对结果的点到点(DUA←DSA 或 DSA←DSA)加密

如果 DUA 用户希望从目录中接收加密的结果或差错，则SecurityParameters. ProtectionRequest 字段应当被设置为ENCRYPTED，或者如果该字段不存在，则处于链接操作变元中的SecurityParameters. ProtectionRequest 字段将被设置为可以体现 DAP BindArgument 中的DIRQOP。远端 DSA (DSA 'c')应当有能力被配置为可以发送加密后的链接操作结果。在这个场景中，DSA 'c'系统判断出它能够满足该请求，于是它产生一个 DAP 结果和 DSP 链接操作结果。通过 DSA 'c'为 DSA 'b'加密 DSP 的链接操作结果，则能够实现点到点加密。DSA 'b'能够解密 DSP 链接操作结果，并且为 DUA 'a'用户加密 DAP 结果。这就提供了结果的点到点加密。DUA 'a'将负责对它的本地 DSA(DSA 'b')的 DAP 结果进行解密。



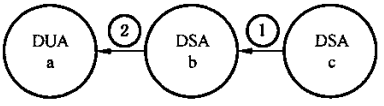
① DSA'c'对DSP链接操作结果加密

② DSA'b'为DUA'a'加密DAP结果

C.4.3 对 DAP 结果的端到端加密和对 DSP 链接结果的点到点加密

如果 DUA 'a'用户希望从目录中接收加密后的结果或差错，则SecurityParameters. ProtectionRequest 字段应当被设置为ENCRYPTED，或者如果该字段不存在，则处于链接操作变元中的SecurityParameters. ProtectionRequest 字段应当被设置为可以体现 DAP 绑定中的DIRQOP。远端 DSA 'c'应当有能力被配置为可以发送加密后的链接操作结果。在这个场景中，DSA 'c'系统判断出它能够满足该请求，于是它对 DAP 结果(为 DUA 用户)产生一个端到端加密，并对 DSP 链接操作结果产生一个点到点加密。端到端加密可被 DSA 'c'执行，因为它知道谁将是 DUA 'a'用户。通过 DSA 'c'为 DSA1 加密 DSP 链接操作结果，则能够实现对 DSP 链接操作结果的点到点加密。DSA 'b'能够解密 DSP，并且

将加密后的 DAP 结果传递到 DUA 'a'用户。DUA 'a'将负责解密它通过 DSA 'b'从 DSA 'c'接收到的 DAP 结果。

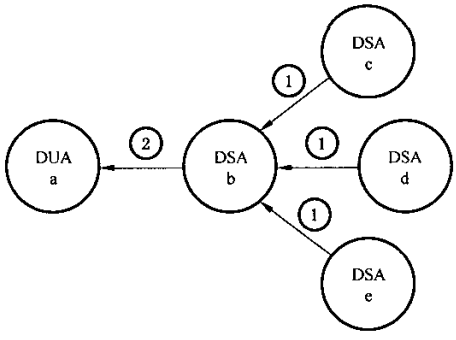


① DSA'c'为DSA'b'加密DSP链接操作结果；这包括来自DSA'c'的为DUA'a'加密的DAP结果

② DSA'b'返回DSA'c'为DUA'a'加密的DAP结果

C.4.4 合并列表/搜索结果(与 DSA 1 的重新加密合并)

如果 DUA 'a'用户希望从目录中接收加密的列表或搜索结果或差错，则SecurityParameters. ProtectionRequest 字段应当被设置为ENCRYPTED，或者如果该字段不存在，则处于链接操作变元中的SecurityParameters. ProtectionRequest 字段应当被设置为可以体现 DAP 绑定中的DIRQOP。本地 DSA(DSA 'b')可以选择将此列表/搜索请求多链接到多个其他 DSA 处(或者是并行的，或者是顺序的)。远端 DSA(DSA 'c'、'd'和'e')应当有能力被配置为可以发送加密后的链接列表/搜索结果。在本模型中，每个远端 DSA('c'、'd'和'e')都满足请求，并产生 DAP 结果和加密的 DSP 链接操作结果。由远端 DSA('c'、'd'和'e')产生的链接操作结果被传递到 DSA 'b'。DSA 'b'接收到每个链接操作结果，解密结果并将这些结果整理或合并为一个公共结果。DSA 'b'于是便加密此新的公共列表/搜索结果，并将其发送到 DUA 'a'用户。点到点加密的完成是通过远端 DSA 为 DSA 'b'加密 DSP 链接操作结果，然后 DSA 'b'为 DUA 'a'用户加密 DAP 结果而实现的。DUA 将负责解密一个合并后的 DAP 结果。



① DSA'c'、'd'、'e'加密DSP链接操作结果（包括DAP结果）

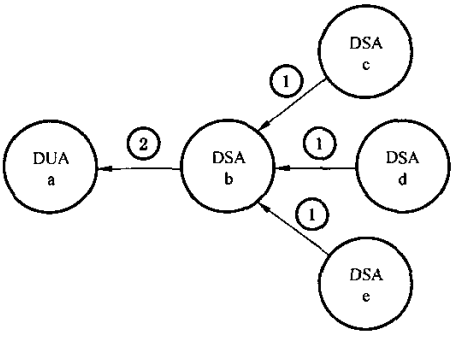
② DSA'b'解密来自DSA'c'、DSA'd'和DSA'e'的DSP链接操作结果，然后合并DAP结果，并为DUA'a'重新加密DAP结果

C.4.5 不允许合并列表/搜索结果

(提供端到端的 DAP 列表/搜索结果加密的 DSA 'b'不执行合并)

如果 DUA 用户希望从目录中接收加密的列表或搜索结果或差错，则SecurityParameters. ProtectionRequest 字段应当被设置为ENCRYPTED，或者如果该字段不存在，则处于链接操作变元中的SecurityParameters. ProtectionRequest 字段应当被设置为可以体现 DAP 绑定中的DIRQOP。本地 DSA 可以选择将此列表/搜索请求多链接到多个其他的 DSA 处(或者是并行的，或者是顺序的)。远端 DSA(DSA 'c'、'd'和'e')应当有能力被配置为可以发送加密后的链接列表/搜索结果。在本场景中，每个远端 DSA('c'、'd'和'e')都满足请求，并产生加密的 DAP 结果(为 DUA 'a'用户)和加密的 DSP 链接操作结果(为 DSA 'b')。由远端 DSA('c'、'd'和'e')产生的链接操作结果被传递到 DSA 'b'。DSA 'b'将列表/搜索结果(由'c'、'd'和'e'加密)不做任何改变地转送并发送到 DUA 'a'。端到端加

密的完成是通过远端 DSA 为 DUA 'a'用户加密 DAP 列表/搜索结果而实现的，而点到点加密的完成是通过远端 DSA 为 DSA 'b'加密 DSP 链接操作结果而实现的。DUA 'a'将负责解密每一个返回的 DAP 列表/搜索结果。

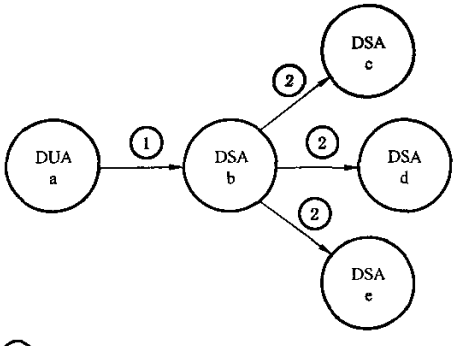


① DSA'c','d','e'为DSA'b'加密DSP链接操作结果：这包括那些已经为DUA'a'用户加密的内容

② DSA'b'解密来自DSA'c',DSA'd'和DSA'e'的DSP链接操作结果，然后不执行解密或合并便将DAP结果（该结果由'c','d'和'e'为DUA'a'加密）传递到DUA'a'

C.4.6 使用一个加密密钥(网钥)多链接一个 DAP 请求

如果 DUA 'a'用户希望从目录中接收加密的结果或差错，则SecurityParameters. ProtectionRequest 字段应当被设置为ENCRYPTED，或者如果该字段不存在，则链接操作变元中的SecurityParameters. ProtectionRequest 字段应当被设置为可以体现 DAP 绑定中的DIRQOP。本地 DSA 可以选择将此列表/搜索请求多链接到多个其他的 DSA 处(或者是并行的，或者是顺序的)。本地 DSA(DSA 'b')可以被配置为支持一个加密密钥或网钥。一个网钥是一个对称的加密密钥，由链接中的所有 DSA 共享。通过使用一个网钥，DSA 'b'仅需要对链接请求加密一次。每个远端 DSA 都知道此网钥，并能够使用此网钥来解密 DSP 链接操作变元。在本场景中，点到点加密的实现是通过 DUA 用户为 DSA 'b'加密 DAP 请求来实现的，而 DSA 'b'能够使用一个网钥来实现到远端 DSA 的点到点加密。



① DUA'a'为DSA'b'加密一个DAP变量

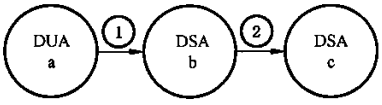
② DSA'b'解密此请求，并尝试满足此请求；如果DSA'b'不能满足请求，它使用一个“净密钥”来加密DSA链接操作请求（包括DAP请求）。链接请求被发送到DSA'c','d'和'e'

C.5 签名并加密的分布式操作

C.5.1 端到端签名与点到点加密

如果一个 DUA 'a'用户希望对 DAP 请求进行签名并加密，则签名可以被端到端提供，而加密仅能够基于点到点方式出现。DUA 'a'能够为 DSA 'b'签名并加密 DAP 请求；然而，DUA 'a'用户不知道

该请求最终是否会被链接到一个远端 DSA(DSA 'c')来处理。DSA 'b'将解密该请求并验证签名。然后它将尝试满足该请求。如果 DSA 'b'判断该请求应当被链接到另一个 DSA(DSA 'c')来处理,则 DSA 'b'为 DSA 'c'加密该 DSP ChainingArguments。原始签名的 DAP 请求将保持,并与加密的 DSP ChainingArguments 一起传递。

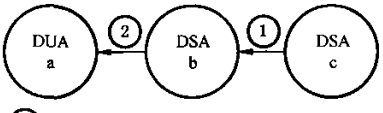


① DUA'a'用户为DSA'b'签名并加密DAP请求

② DSA'b'解密此DAP请求并验证签名;在尝试本地满足此请求后,DSA'b'判断该请求需要被链接到DSA'c'。DSA'b'发送原始签名的DAP请求(由DUA'a'用户签名),并为DSA'c'产生和加密DSP链接变量

C.5.2 对 DAP 结果的端到端签名并加密,对 DSP 的点到点签名并加密

如果 DUA 'a'用户希望从目录中接收签名并加密的结果,则SecurityParameters. ProtectionRequest 字段应当被设置为SIGNED-AND-ENCRYPTED,或者如果该字段不存在,则ChainingArguments 中的SecurityParameters. ProtectionRequest 字段应当被设置为可以体现 DAP 绑定中的DIRQOP。远端 DSA 应当有能力被配置为可以发送签名并加密后的链接操作。在本模型中,DSA 'c'系统能够满足请求,并且对 DAP 结果产生并执行一个端到端加密(为 DUA 'a'用户),同时对 DSP ChainingResults 产生并执行一个点到点加密。端到端的签名并加密能够由 DSA 'c'执行,因为它知道谁将是 DUA 'a'用户。对 DSP ChainingResults 进行点到点的签名并加密可以通过 DSA 'c'为 DSA 'b'签名并加密 DSP ChainingResults 来实现。DSA 'b'能够解密并验证 DSA 'c'对签名的DSPChainingResults 的签名,并且将签名并加密后的 DAP 结果传递到 DUA 'a'用户。DUA 'a'将负责对通过 DSA 'b'接收到的来自 DSA 'c'的 DAP 结果进行解密并对签名进行验证。

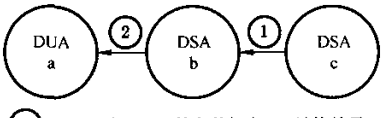


① DSA'c'为DSA'b'签名并加密DSP链接结果;这包括为DUA'a'用户签名并加密的DAP结果

② DSA'b'解密来自DSA'c'的DSP链接结果,并且为DUA'a'转发签名并加密后的DAP结果

C.5.3 对 DAP 的端到端签名,对 DSP 和 DAP 结果的点到点加密

如果 DUA 'a'用户希望从目录中接收签名并加密后的结果,则SecurityParameters. ProtectionRequest 字段应当被设置为SIGNED-AND-ENCRYPTED,或者如果该字段不存在,则ChainingArguments 中的SecurityParameters. ProtectionRequest 字段应当被设置为可以体现 DAP 绑定中的DIRQOP。远端 DSA(DSA 'c')应当有能力被配置为可以发送签名并加密后的链接操作结果。在此模型中,DSA 'c'系统能够满足请求,则它会产生一个签名后的 DAP 结果,并且为了 DSA 'b',对 DAP 结果和 DSP 的 ChainingResults 进行签名并加密。DSA 'b'能够解密并验证 DSA 'c'对 DSP ChainingResults 的签名,并且为 DUA 'a'用户重新加密已签名(由 DSA 'c'签名)后的 DAP 结果。DUA 'a'将负责对从 DSA 'b'接收到的 DAP 结果进行解密,并且对通过 DSA 'b'接收到的来自 DSA 'c'的 DAP 结果的签名进行验证。



① DSA'c'为DSA'b'签名并加密DSP链接结果;这包括DAP结果

② DSA'b'解密来自DSA'c'的DSP链接结果(以及在DSP链接结果中接收到的 DAP 结果),并且将签名后的 DAP 结果转发到DUA'a'

附 录 D
（规范性附录）
分等级和非特定分等级操作绑定类型的规范

本附录包含了本目录规范中引入的 ASN.1 信息客体类的定义，以 ASN.1 模块 HierarchicalOperationalBindings 的形式提供。

```
HierarchicalOperationalBindings
        {joint-iso-ITU-T ds(5) module(1)
hierarchicalOperationalBindings(20) 5} DEFINITIONS ::=
BEGIN

--EXPORTS All --
```

——本模块中定义的所有类型与值都可被输出到本系列目录规范所包含的其他的 ASN.1 模块中，供其使用。

——也可以被其他应用所使用，这些应用将使用本模块中的定义来访问目录服务。

——其他应用可以将本模块中的定义用于其自身目的，但是不会限制为了维护和改进目录服务而进行的扩展和修改。

```
IMPORTS
```

——来自*GB/T 16424.2*

```
        directoryOperationalBindingTypes, directoryOSIProtocols,
        distributedOperations, informationFramework,
        opBindingManagement
            FROM UsefulDefinitions {joint-iso-ITU-T ds(5) module(1) usefulDefinitions(0) 5}

        Attribute, DistinguishedName, RelativeDistinguishedName
            FROM InformationFramework informationFramework

        OPERATIONAL-BINDING
            FROM OperationalBindingManagement opBindingManagement
```

——来自*GB/T 16424.4*

```
        MasterAndShadowAccessPoints
            FROM DistributedOperations distributedOperations
```

——来自*GB/T 16424.5*

```
        directorySystemAC
```

```
        FROM DirectoryOSIProtocols directoryOSIProtocols

    id-op-binding-hierarchical, id-op-binding-non-specific-hierarchical
        FROM DirectoryOperationalBindingTypes directoryOperationalBindingTypes;

--类型--

HierarchicalAgreement ::= SEQUENCE {
    rdn                 [0]  RelativeDistinguishedName,
    immediateSuperior   [1]  DistinguishedName }
SuperiorToSubordinate ::= SEQUENCE {
    contextPrefixInfo   [0]  DITcontext,
    entryInfo           [1]  SET SIZE (1..MAX) OF Attribute OPTIONAL,
    immediateSuperiorInfo [2] SET SIZE (1..MAX) OF Attribute OPTIONAL }

DITcontext ::= SEQUENCE OF Vertex

Vertex ::= SEQUENCE {
    rdn                 [0]  RelativeDistinguishedName,
    admPointInfo        [1]  SET SIZE (1..MAX) OF Attribute OPTIONAL,
    subentries          [2]  SET SIZE (1..MAX) OF SubentryInfo OPTIONAL,
    accessPoints        [3]  MasterAndShadowAccessPoints OPTIONAL }
SubentryInfo ::= SEQUENCE {
    rdn                 [0]  RelativeDistinguishedName,
    info                [1]  SET OF Attribute }
SubordinateToSuperior ::= SEQUENCE {
    accessPoints        [0]  MasterAndShadowAccessPoints OPTIONAL,
    alias               [1]  BOOLEAN DEFAULT FALSE,
    entryInfo           [2]  SET SIZE (1..MAX) OF Attribute OPTIONAL,
    subentries          [3]  SET SIZE (1..MAX) OF SubentryInfo OPTIONAL }
SuperiorToSubordinateModification ::= SuperiorToSubordinate (
    WITH COMPONENTS { ..., entryInfo ABSENT})
NonSpecificHierarchicalAgreement ::= SEQUENCE {
    immediateSuperior   [1]  DistinguishedName }
NHOBSuperiorToSubordinate ::= SuperiorToSubordinate (
    WITH COMPONENTS { ..., entryInfo ABSENT})

NHOBSubordinateToSuperior ::= SEQUENCE {
    accessPoints        [0]  MasterAndShadowAccessPoints OPTIONAL,
    subentries          [3]  SET SIZE (1..MAX) OF SubentryInfo OPTIONAL }

--操作绑定信息客体--
```

```
hierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
        AGREEMENT       HierarchicalAgreement
        APPLICATION CONTEXTS {
                        {directorySystemAC} }
        ASYMMETRIC
           ROLE-A   {      ——上级DSA
              ESTABLISHMENT-INITIATOR          TRUE
              ESTABLISHMENT-PARAMETER      SuperiorToSubordinate
              MODIFICATION-INITIATOR           TRUE
              MODIFICATION-PARAMETER       SuperiorToSubordinateModification
              TERMINATION-INITIATOR          TRUE }
           ROLE-B {          ——下级DSA
              ESTABLISHMENT-INITIATOR          TRUE
              ESTABLISHMENT-PARAMETER   SubordinateToSuperior
              MODIFICATION-INITIATOR      TRUE
              MODIFICATION-PARAMETER          SubordinateToSuperior
              TERMINATION-INITIATOR      TRUE }
        ID                  id-op-binding-hierarchical }
nonSpecificHierarchicalOperationalBinding OPERATIONAL-BINDING ::= {
        AGREEMENT
                        NonSpecificHierarchicalAgreement
        APPLICATION CONTEXTS {
                        { directorySystemAC } }
        ASYMMETRIC
           ROLE-A   {      ——上级 DSA
              ESTABLISHMENT-PARAMETER NHOBSuperiorToSubordinate
              MODIFICATION-INITIATOR       TRUE
              MODIFICATION-PARAMETER    NHOBSuperiorToSubordinate
              TERMINATION-INITIATOR      TRUE }

           ROLE-B {       ——下级 DSA
              ESTABLISHMENT-INITIATOR          TRUE
              ESTABLISHMENT-PARAMETER   NHOBSubordinateToSuperior
              MODIFICATION-INITIATOR        TRUE
              MODIFICATION-PARAMETER          NHOBSubordinateToSuperior
              TERMINATION-INITIATOR      TRUE }
        ID              id-op-binding-non-specific-hierarchical }
END - HierarchicalOperationalBindings
```

附 录 E
（资料性附录）
知识维护示例

本附录以一个简单的示例举例说明了第 23 章定义的知识维护。在图 E.1 中，使用了下列符号来描述五个 DSA 的 DSA 信息树。

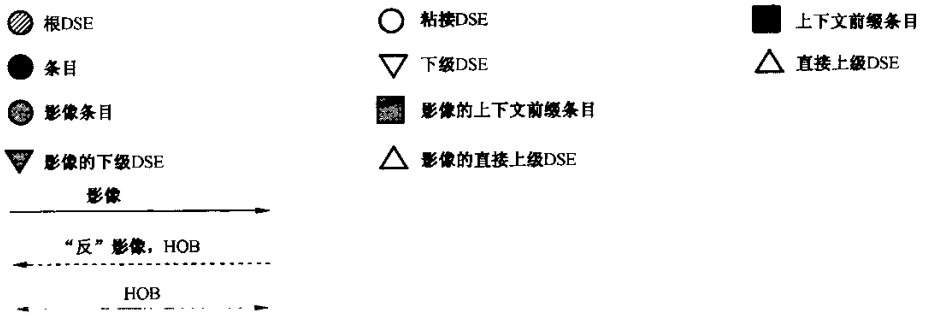


图 E.1 用于描述 DSA 信息树的符号

在图 E.2 中，DSA 1 是命名上下文{A}的属主，由两个条目{A}和{A，B}组成。DSA 1 拥有命名上下文{A，B，C}的一个下级引用，该引用通过一个与 DSA 3 之间的 HOB 来维护。DSA 1 是 DSA 2 的一个影像提供者，向其提供命名上下文{A}的用户信息的拷贝，以及指向命名上下文{A，B，C}的下级引用的拷贝，该引用标识了 DSA 3，DSA4 和 DSA5 的访问点，其中前者是下级命名上下文的属主。

DSA 3 是命名上下文{A，B，C}的属主。除了拥有该命名上下文的单独条目{A，B，C}外，DSA 3 还拥有命名上下文{A}的一个直接上级引用，该引用通过一个与 DSA 1 的 HOB 来维护。DSA 3 是 DSA 4 的一个影像提供者，向其提供命名上下文{A，B，C}的用户信息的拷贝，以及指向命名上下文{A}的直接上级引用的拷贝，该引用标识了 DSA 1 和 DSA 2 的访问点，其中前者是上级命名上下文的属主。DSA 4 是 DSA 5 的一个（二次）影像提供者，向其提供从 DSA 3 中接收到的信息的拷贝。

图 E.2 举例说明了用于表示和维护知识的 DSA 操作属性。

DSA 1 使用它的 myAccessPoint 属性的值（与其根 DSE 相关联）与它的 consumerKnowledge 属性的公共可用值（与上下文前缀{A}相关联）来构造 MasterAndShadowAccessPoints 类型的一个值，用于与 DSA 3 的 HOB 交互。DSA 3 随之使用它的 myAccessPoint 属性的值（与其根 DSE 相关联）与它的 consumerKnowledge 属性和 secondaryShadows 属性的公共可用值（两个值都与上下文前缀{A，B，C}相关联）来构造 MasterAndShadowAccessPoints 类型的一个值，用于与 DSA 1 的 HOB 交互。两个 DSA，共同使用 DOP，来维护DSA 1所拥有的一个下级引用，以及 DSA 3 所拥有的一个直接上级引用。DSA 1 的下级引用，由一个与处于{A，B，C}中的 DSE 相关联的 specificKnowledge 属性来表示，是基于它从 DSA 3 中接收到的 MasterAndShadowAccessPoints 值构造的；DSA 3 的直接上级引用，由一个与处于{A}中的 DSE 相关联的 specificKnowledge 属性来表示，类似的，是基于它从 DSA 1 中接收到的 MasterAndShadowAccessPoints 值构造的。

DSA 1 和 DSA 2 在影像操作绑定交互中使用它们的myAccessPoint 值来维护 DSA 1 中的consumerKnowledge 的值（该值标识了 DSA 2 的访问点）以及 DSA 2 中的supplierKnowledge 的值（该值标识了 DSA 1 的访问点），两个属性都与上下文前缀{A}相关联。两个 DSA，共同使用 DOP，维护 DSA 1 所拥有的消费者引用，以及 DSA 2 所拥有的提供者引用。

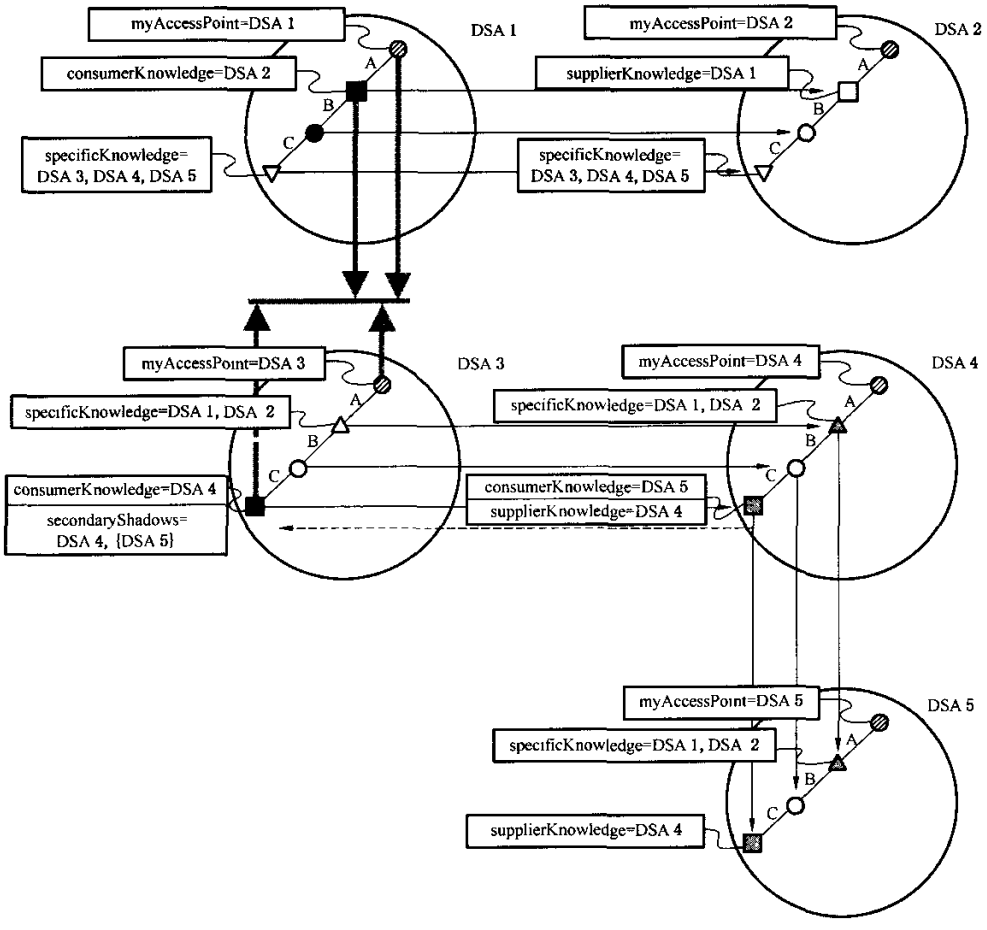


图 E.2　知识维护示例

DSA 2 在与 DSA 1 的 DISP 交互中，从 DSA 1 处接收到一个与上下文前缀{A，B，C}相关联的 specificKnowledge 属性的拷贝。这种交互用于维护 DSA 2 中的指向上下文前缀{A，B，C}的下级引用。

DSA 3 和 DSA 4(以及类似的 DSA 4 和 DSA 5)以一种同 DSA 1 和 DSA 2 之间的交互类似的方式，分别维护消费者引用和提供者引用。

DSA 4 在与 DSA 3 的 DISP 交互中，从 DSA 3 处接收到一个与上下文前缀{A4}相关联的specificKnowledge 属性的拷贝。这种交互用于维护 DSA 4 中的指向上下文前缀{A}的直接上级引用。

DSA 4 向 DSA 3 传送在它的myAccessPoint 和consumerKnowledge 属性中所发生的任何变化(以及secondaryShadows 属性，在本例中该属性值为空)，方法是使用 DOP 的修改操作绑定操作。DSA 4 向 DSA 3 提供了SupplierAndConsumers 的一个值，其中仅包含了consumerKnowledge 属性的一些值，那些值标识了具有公共可用影像的 DSA 的访问点；DSA 4 所提供的secondaryShadows 属性的值，如果有的话，将被设计为都是公共可用的(在本例中，DSA 5 被假设拥有处于{A，B，C}中的命名上下文的一个公共可用的拷贝)。DSA 3 使用该信息来维护它的secondaryShadows 属性中的与上下文前缀{A，B，C}相关联的一个值。正如上述描述的那样，这个属性被用于与 DSA 1 的 DOP 交互，来维护

DSA 1 中的指向上下文前缀{A，B，C}的下级引用。

DSA 5 使用与 DSA 4 的 DISP 交互来维护它的指向上下文前缀{A}的直接上级引用，采用一种同 DSA 3 和 DSA 4 之间的交互相类似的方式。

中 华 人 民 共 和 国
国 家 标 准
信息技术 开放系统互连 目录
第4部分：分布式操作规程
GB/T 16264.4—2008/ISO/IEC 9594-4:2005
*
中国标准出版社出版发行
北京复兴门外三里河北街16号
邮政编码：100045
网址 www.spc.net.cn
电话：68523946 68517548
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 7.5 字数 227 千字
2008年12月第一版 2008年12月第一次印刷
*
书号：155066·1-34754 定价 68.00 元

如有印装差错 由本社发行中心调换
版权专有 侵权必究
举报电话：(010)68533533