Chapter 3

Basic Foundations:

Standards, Models, and Language

Introduction

- Standards
  - Standards organizations
  - Protocol standards of transport layers
  - Protocol standards of management (application) layer
- Management Models
- Language

Table 3.1 Network Management Standards

Standard	Salient Points
OSI / CMIP	■ International standard (ISO / OSI) ■ Management of data communications network - LAN and WAN ■ Deals with all 7 layers ■ Most complete ■ Object oriented ■ Well structured and layered ■ Consumes large resource in implementation
SNMP / Internet	■ Industry standard (IETF) ■ Originally intended for management of Internet components, currently adopted for WAN and telecommunication systems ■ Easy to implement ■ Most widely implemented
TMN	■ International standard (ITU-T) ■ Management of telecommunications network ■ Based on OSI network management framework ■ Addresses both network and administrative aspects of management
IEEE	■ IEEE standards adopted internationally ■ Addresses LAN and MAN management ■ Adopts OSI standards significantly ■ Deals with first two layers of OSI RM
Web-based Management	■ Web-Based Enterprise Management (WBEM) ■ Java Management Application Program Interface (JMAPI)

OSI Architecture and Model

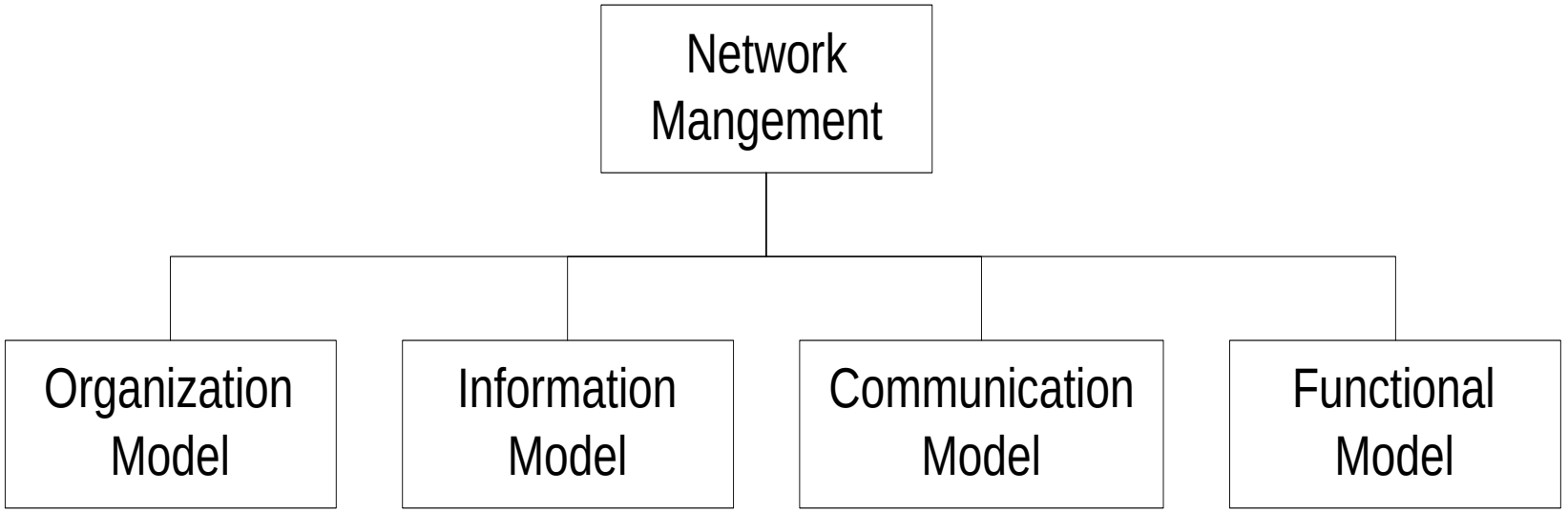


Figure 3.1 OSl Network Management Model

SNMP Architecture and Model

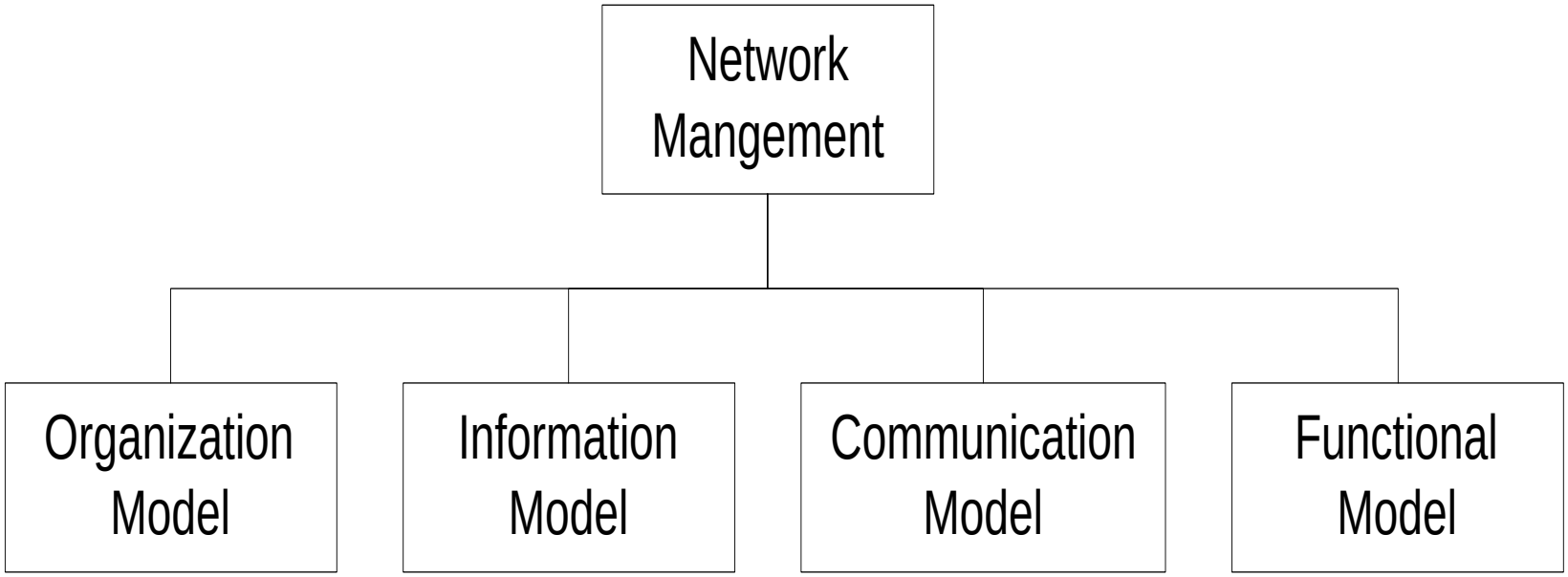


Figure 3.1 OSI Network Management Model

TMN Architecture

- Addresses management of telecommunication networks
- Based on OSI model
- Superstructure on OSI network
- Addresses network, service, and business management

Organizational Model

- Manager
  - Sends requests to agents
  - Monitors alarms
  - Houses applications
  - Provides user interface

- Agent
  - Gathers information from objects
  - Configures parameters of objects
  - Responds to managers' requests
  - Generates alarms and sends them to mangers

- Managed object
  - Network element that is managed
  - Houses management agent
  - All objects are not managed / manageable

Two-Tier Model

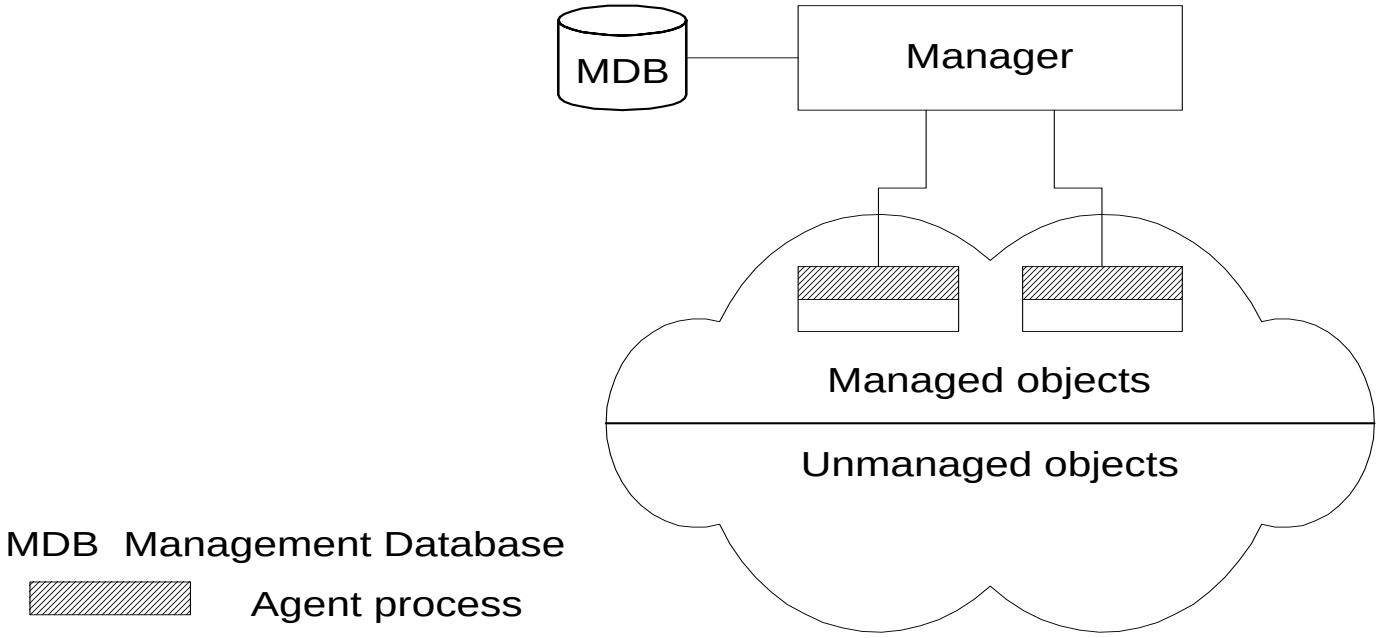


Figure 3.2 Two-Tier Network Mangement Organization Model

Three-Tier Model

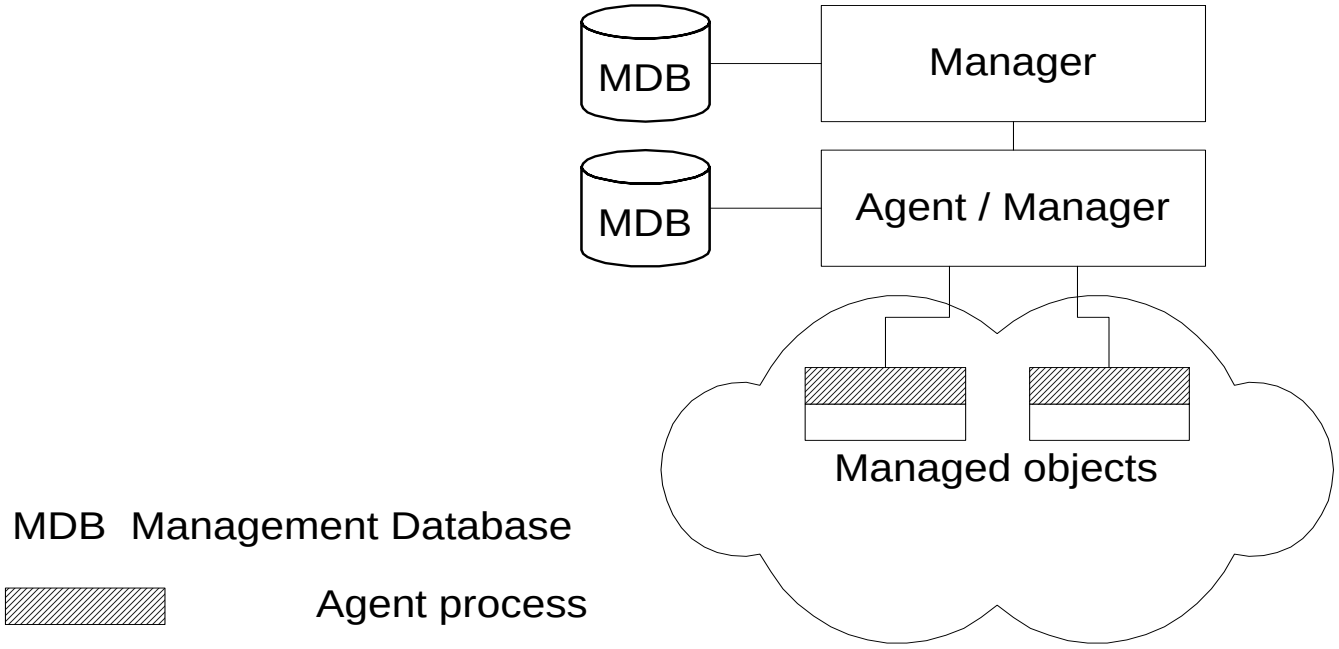


Figure 3.3 Three-Tier Network Mangement Organization Model

Manager of Managers

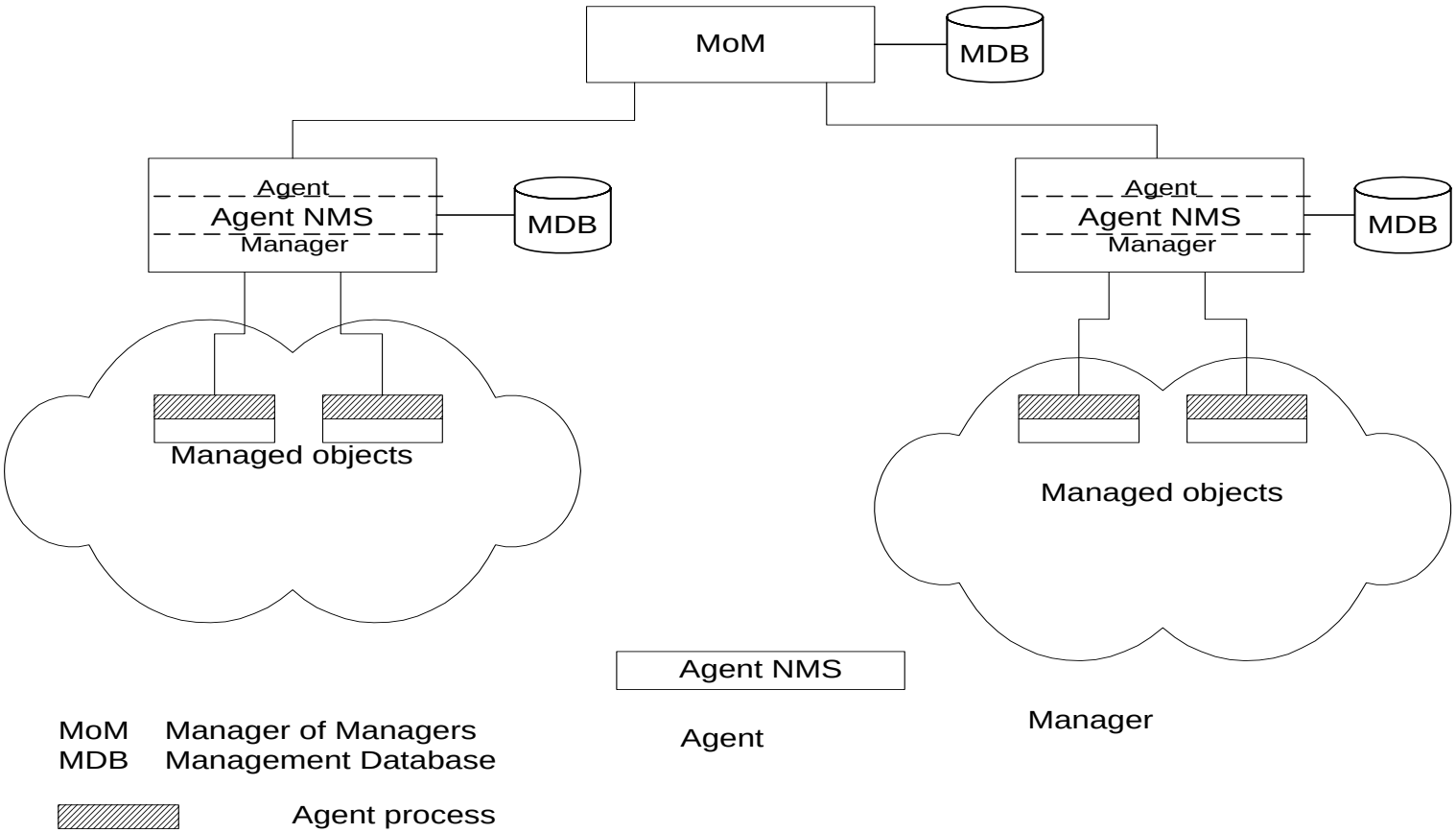


Figure 3.4 Network Mangement Organization Model with MoM

Peer NMSs

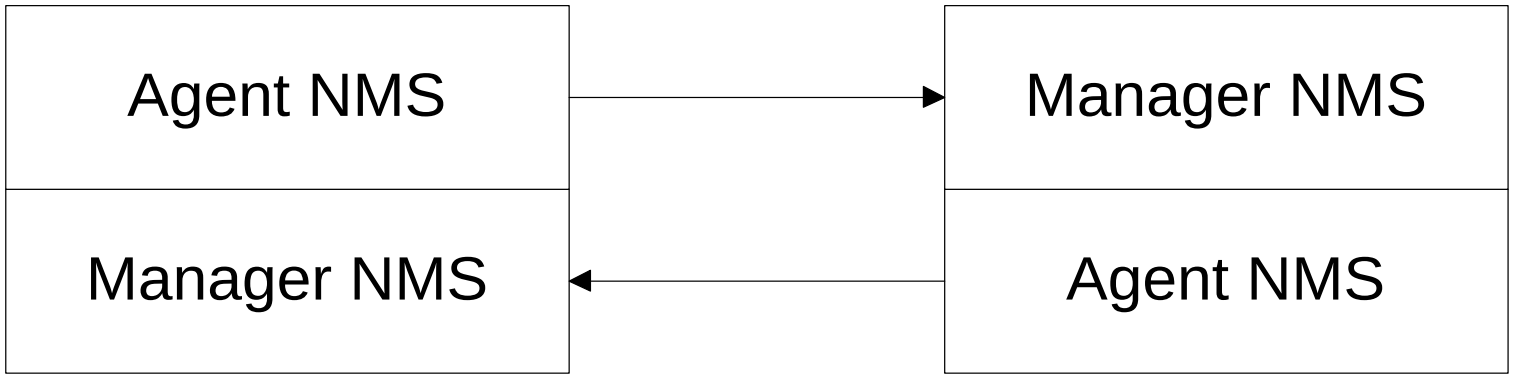


Figure 3.5 Dual Role of Management Process

Information Model: Analogy

- Figure in a book uniquely identified by
  - ISBN, Chapter, and Figure number in that hierarchical order
- ID: {ISBN, chapter, figure}
- The three elements above define the syntax
- Semantics is the meaning of the three entities according to Webster's dictionary
- The information comprises syntax and semantics about an object

Structure of Management Information (SMI)

- SMI defines for a managed object
  - Syntax
  - Semantics
  - plus additional information such as status

- Example

  sysDescr: { system 1 }
  Syntax: OCTET STRING
  Definition: "A textual description of the entity. "
  Access: read-only
  Status: mandatory

Management Information Base (MIB)

- Information base contains information about objects
- Organized by grouping of related objects
- Defines relationship between objects
- It is NOT a physical database. It is a *virtual* database that is compiled into management module

Information Base View: An Analogy

- Fulton County library system has many branches
- Each branch has a set of books
- The books in each branch is a different set
- The information base of the county has the view (catalog) of all books
- The information base of each branch has the catalog of books that belong to that branch. That is, each branch has its view (catalog) of the information base
- Let us apply this to MIB view

MIB View and Access of an Object

- A managed object has many attributes - its information base
- There are several operations that can be performed on the objects
- A user (manager) can view and perform only certain operations on the object by invoking the management agent
- The view of the object attributes that the agent perceives is the MIB view
- The operation that a user can perform is the MIB access

Management Data Base / Information Base

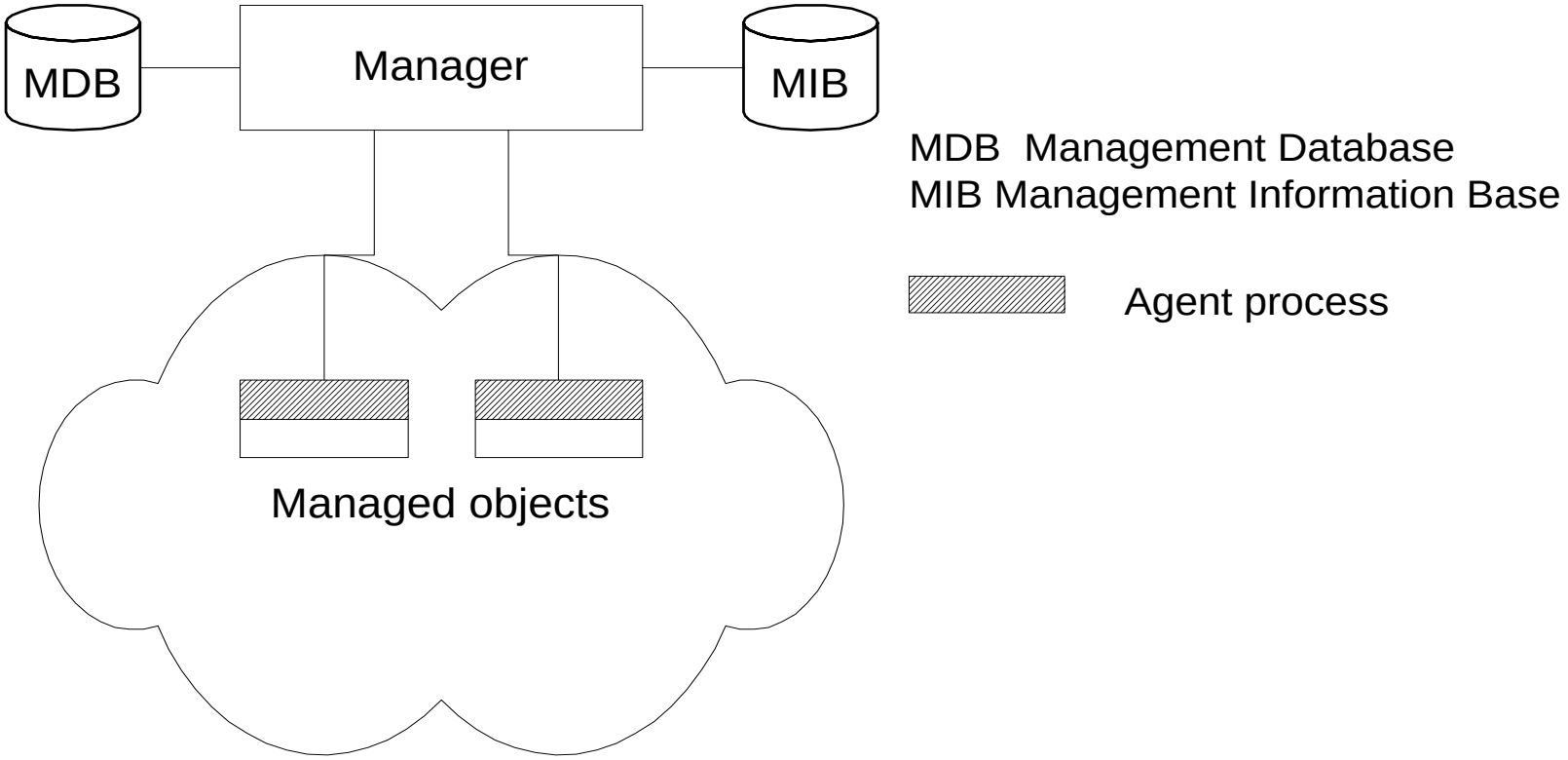


Figure 3.6 Network Configuration with Data and Information Base

Managed Object

- Managed objects can be
  - Network elements (hardware, system)
    - hubs, bridges, routers, transmission facilities
  - Software (non-physical)
    - programs, algorithms
  - Administrative information
    - contact person, name of group of objects (IP group)

Management Information Tree

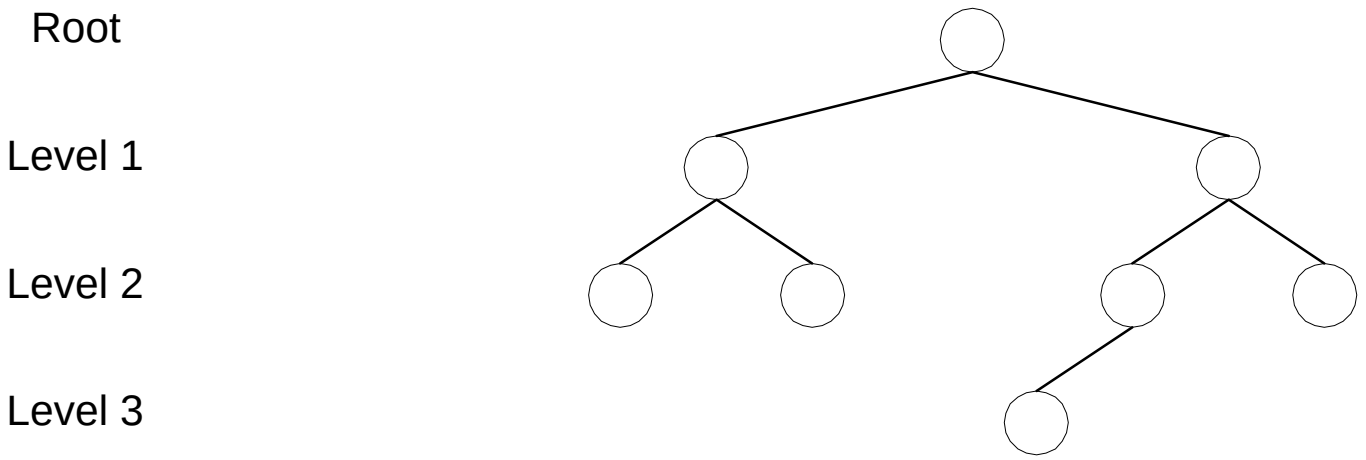


Figure 3.7 Generic Representation of Management Information Tree

OSI Management Information Tree

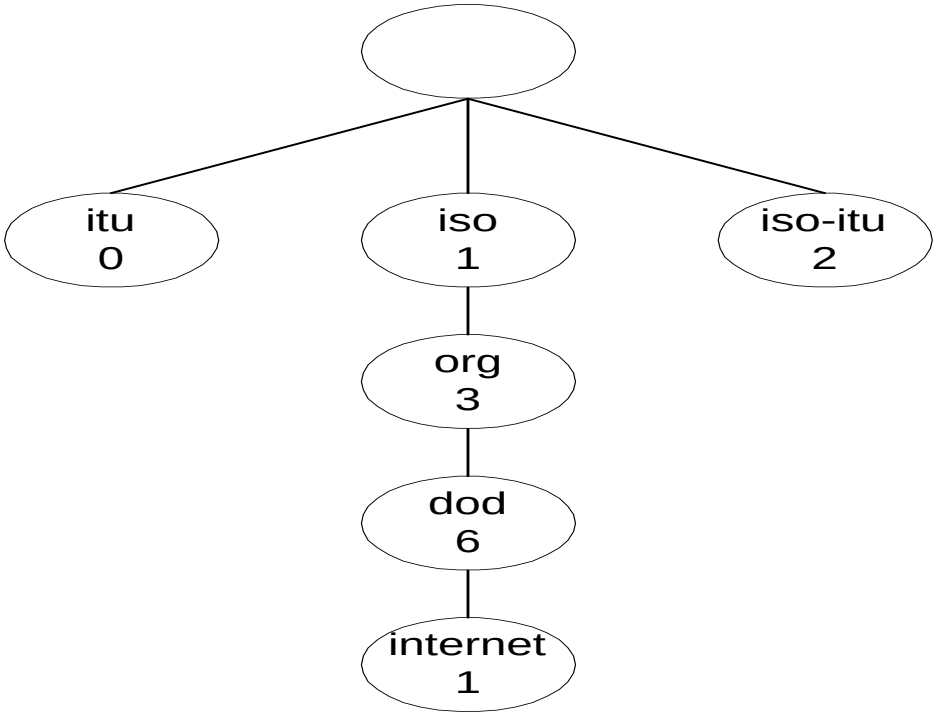


Figure 3.8 **OSI Management Information Tree**

Object Type and Instance

- Type
  - Name
  - Syntax
  - Definition
  - Status
  - Access
- Instance

Managed Object: Internet Perspective

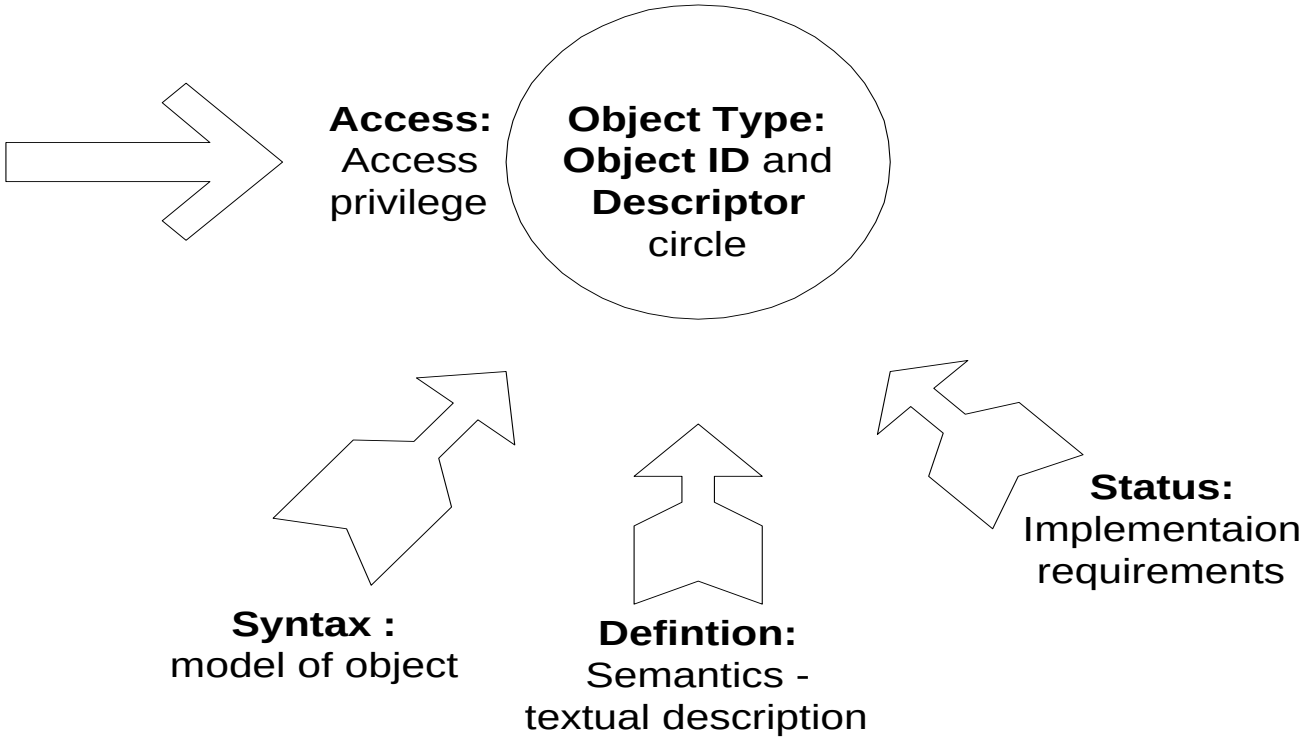


Figure 3.9(a) Internet Perspective

Managed Object: OSI Perspective

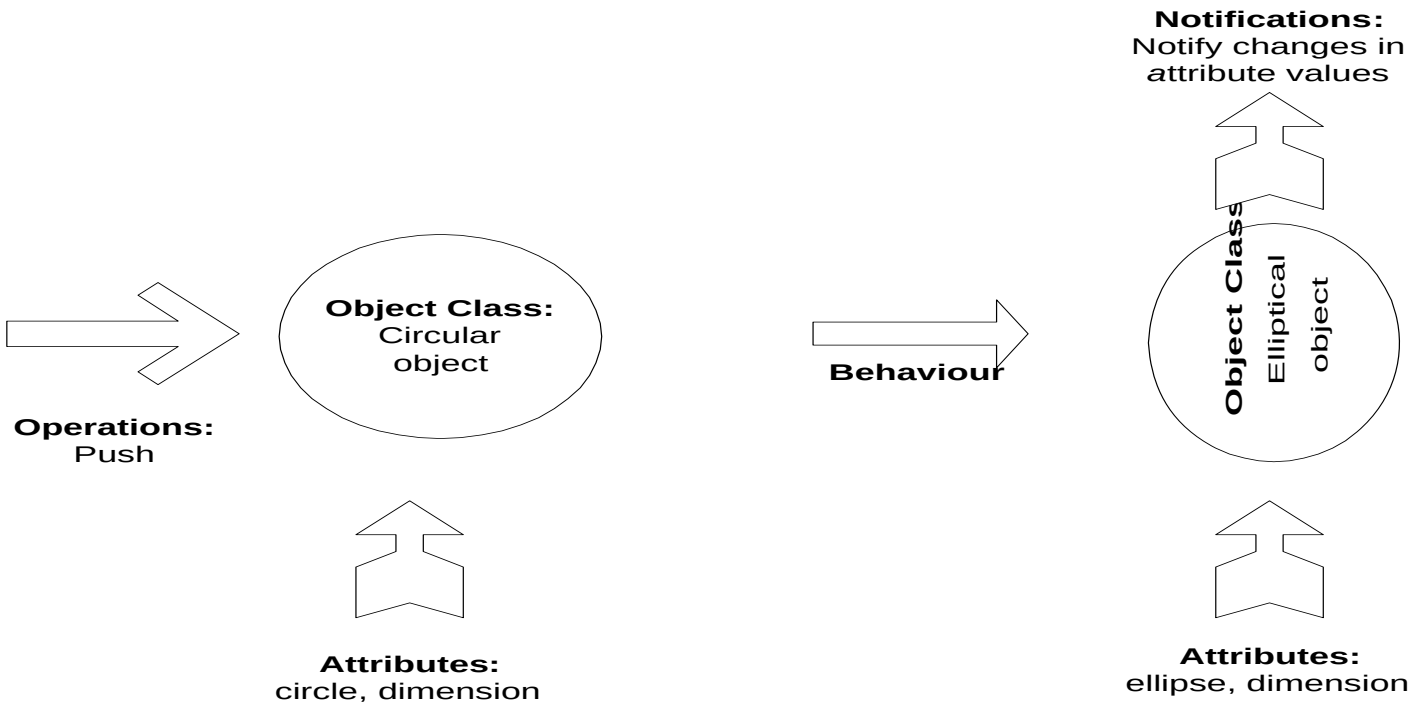


Figure 3.9(b) OSI Perspective

Packet Counter Example

Characteristics	Example
Object type	PktCounter
Syntax	Counter
Access	Read-only
Status	Mandatory
Description	Counts number of packets

Figure 3.10(a) Internet Perspective

Characteristics	Example
Object class	Packet Counter
Attributes	Single-valued
Operations	get, set
Behavior	Retrieves or resets values
Notifications	Generates notifications on new value

Figure 3.10 (b) OSI Perspective

Figure 3.10 Packet Counter As Example of Managed Object

Internet Vs OSI Managed Object

- Scalar object in Internet Vs Object-oriented approach in OSI
- OSI characteristics of operations, behaviour, and notification are part of communication model in Internet: get/set and response/alarm
- Internet syntax is absorbed as part of OSI attributes
- Internet access is part of OSI security model
- Internet status is part of OSI conformance application
- OSI permits creation and deletion of objects; Internet does not: Enhancement in SNMPv2

Mgmt. Communication Model

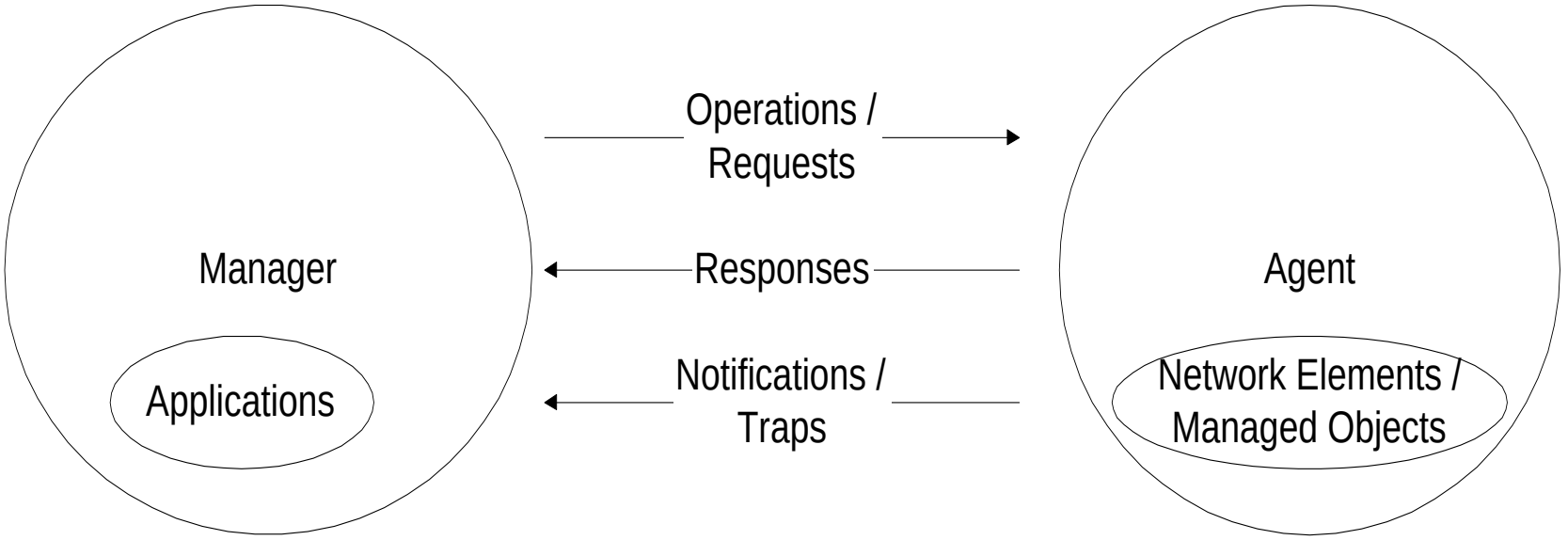


Figure 3.11 Management Message Communication Model

Transfer Protocols

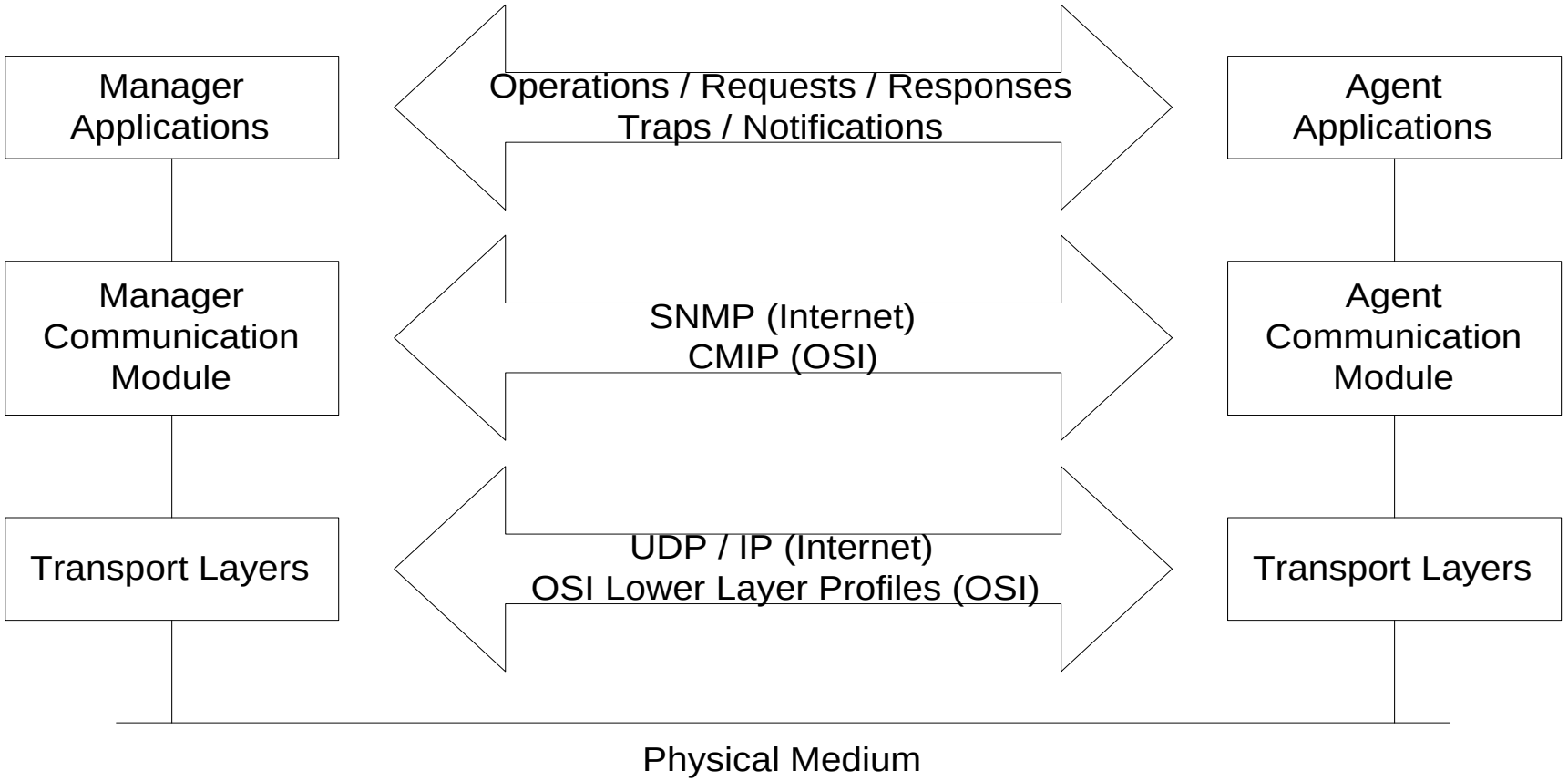


Figure 3.12 Management Communication Transfer Protocols

Abstract Syntax Notation One

- ASN.1 is more than a syntax; it's a language
- Addresses both syntax and semantics
- Two type of syntax
  - Abstract syntax: set of rules that specify data type and structure for information storage
  - Transfer syntax: set of rules for communicating information between systems
- Makes application layer protocols independent of lower layer protocols
- Can generate machine-readable code: Basic Encoding Rules (BER) is used in management modules

Backus-Nauer Form (BNF)

Definition:

<name> ::= <definition>

Rules:

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<number> ::= <number> | <digit> <number>

<op> ::= +|-|x|/

<SAE> ::= <number>|<SAE>|<SAE><op><SAE>

Example:

- 9 is *primitive* 9
- 19 is *construct* of 1 and 9
- 619 is *construct* of 6 and 19

Simple Arithmetic Expression

<SAE> ::= <number> | <SAE><op><number>

Example: 26 = 13 x 2

Constructs and primitives

Type and Value

- Assignments
  - <BooleanType> ::= BOOLEAN
  - <BooleanValue> ::= TRUE | FALSE

- ASN.1 module is a group of assignments

```
person-name     Person-Name::=
                {
                first         "John",
                middle    "I",
                last        "Smith"
                }
```

Data Type: Example 1

```
PersonnelRecord ::= SET
        {       Name,
         title      GraphicString,
         division   CHOICE
            marketing      [0]  SEQUENCE
              {Sector,
               Country},
        research       [1]  CHOICE
              {product-based   [0]  NULL,
               basic           [1]  NULL},
        production         [2]  SEQUENCE
              {Product-line,
               Country}                }
etc.
```

Figure 3.13 ASN.1 Data Type Definition Example 1

Data Type: Example 2

```
Trade-message ::= SEQUENCE
     {invoice-no          INTEGER
      name                GraphicString,
      details         SEQUENCE OF
                          SEQUENCE
                {part-no        INTEGER
                 quantity       INTEGER},
      charge            REAL,
      authenticator       Security-Type}

Security-Type ::= SET
     {     ...
           ...
           ...    }
```

Figure 3.14 ASN.1 Data Type Definition Example 2

ASN.1 Symbols

Symbol	Meaning
::=	Defined as
|	or, alternative, options of a list
-	Signed number
--	Following the symbol are comments
{}	Start and end of a list
[]	Start and end of a tag
()	Start and end of subtype
..	Range

Keyword Examples

- CHOICE
- SET
- SEQUENCE
- OF
- NULL

ASN.1 Data Type Conventions

Data Types	Convention	Example
Object name	Initial lowercase letter	sysDescr, etherStatsPkts
Application data type	Initial uppercase letter	Counter, IpAddress
Module	Initial uppercase letter	PersonnelRecord
Macro, MIB module	All uppercase letters	RMON-MIB
Keywords	All uppercase letters	INTEGER, BEGIN

Data Type: Structure & Tag

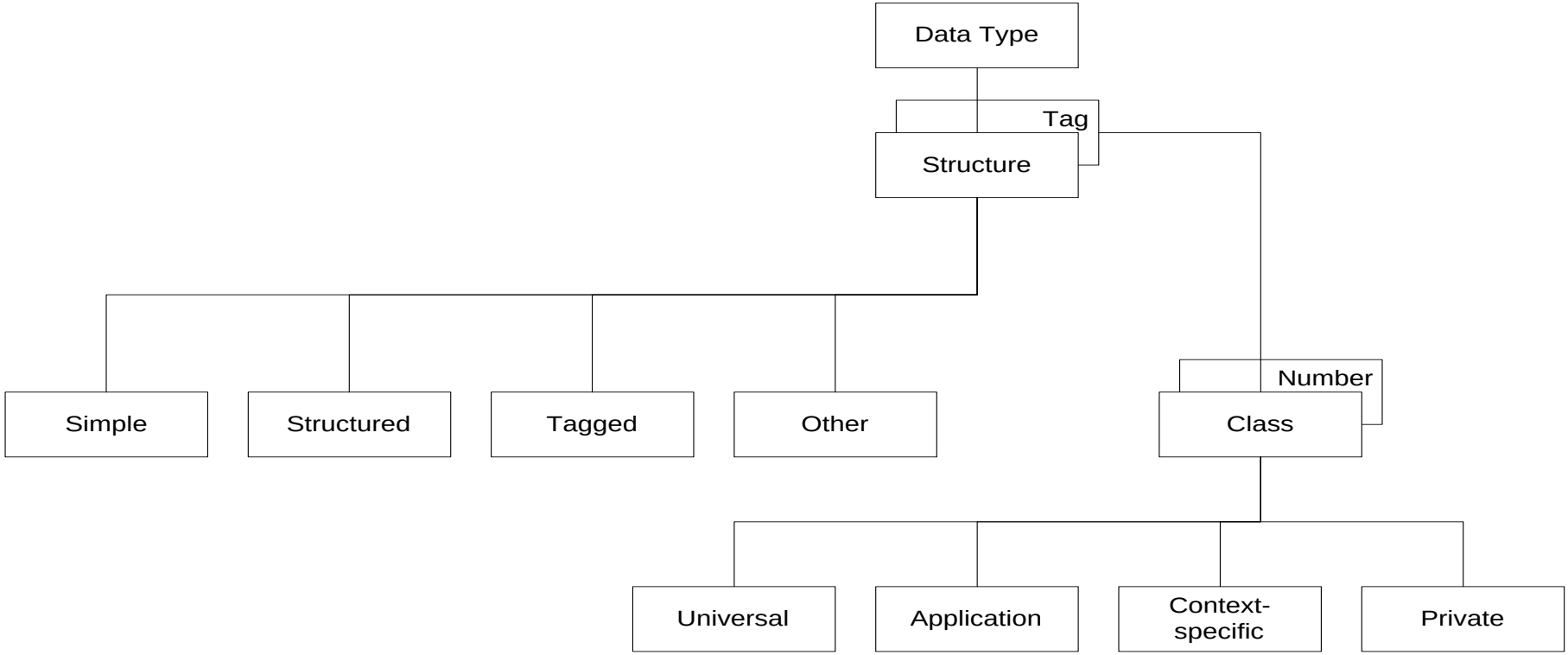


Figure 3.15 ASN.1 Data Type Structure and Tag

Structure

- Simple
  - PageNumber ::= INTEGER
  - ChapterNumber ::= INTEGER
- Structure / Construct
  - BookPageNumber ::=
    SEQUENCE
    {ChapterNumber, Separator, PageNumber
    Example: {1-1, 2-3, 3-39}
- Tagged
  - Derived from another type; given a new ID
  - In Fig. 3-14, INTEGER is either universal or application specific
- Other types:
  - CHOICE, ANY

- BookPages ::= SEQUENCE OF { BookPageNumber}

or

BookPages ::=
SEQUENCE OF
{
SEQUENCE
{ChapterNumber, Separator, PageNumber}
}

Tag

- Tag uniquely identifies a data type
- Comprises *class* and *tag number*
- Class:
    - Universal - always true
    - Application - only in the application used
    - Context-specific - specific context in application
    - Private - used extensively by commercial vendors

Enumerated Integer

```
RainbowColors ::= ENUMERATED
        {
                violet          (0)
                indigo          (1)
                blue            (2)
                green           (3)
                yellow          (4)
                orange          (5)
                red             (6)
        }
```

ASN.1 Module Example

```
IpNetMediaEntry ::=SEQUENCE{
        ipNetToMediaIfIndex             INTEGER
        ipNetToMediaPhysAddress         PhysAddress
        ipNetToMediaNetAddress          IpAddress
        ipNetToMediaType                INTEGER}
```

```
Name:                 John P Smith
Title:                Director
Employee Number       51
Date of Hire:         17 September 1971
Name of Spouse;       Mary T Smith
Number of Children    2
Child Information
        Name          Ralph T Smith
        Date of Birth 11 November 1957
Child Information
        Name          Susan B Jones
        Date of Birth 17 July 1959
```

(a) Informal description of personnel record

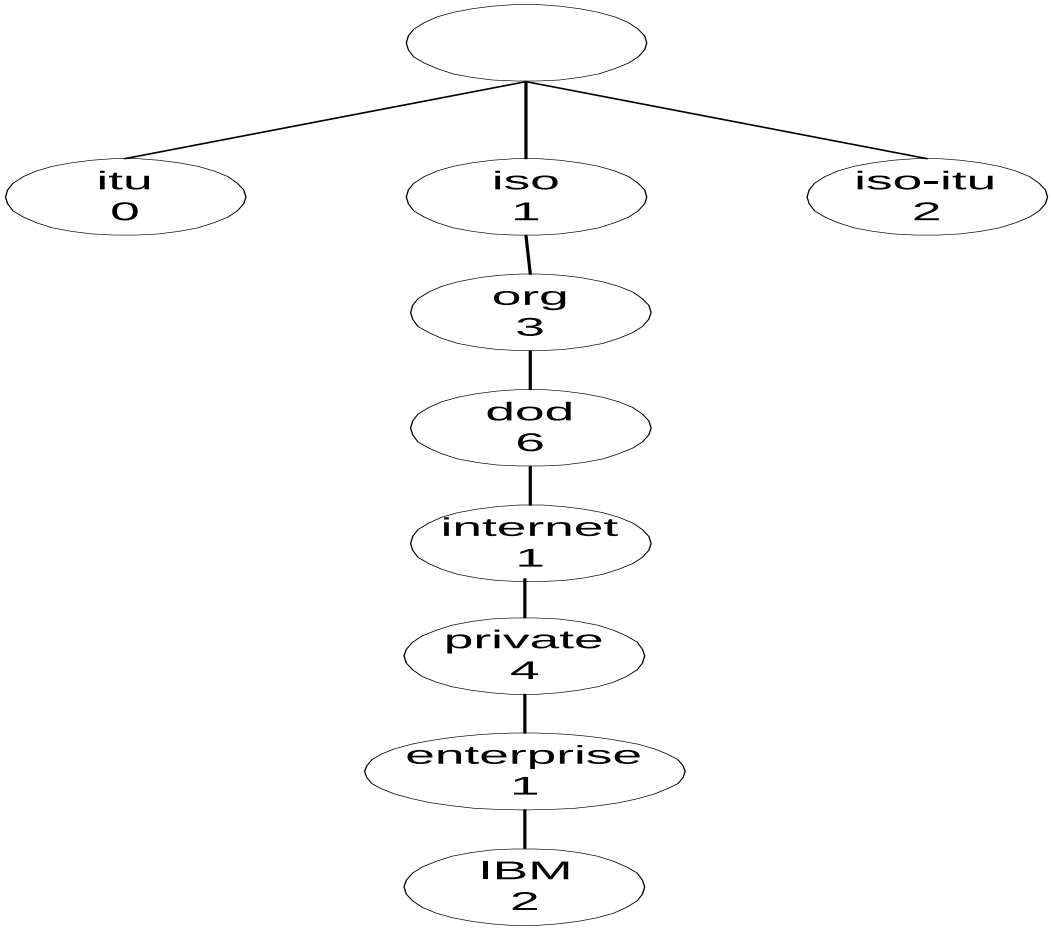
```
PersonnelRecord ::= [APPLICATION 0] IMPLICIT SET {
        Name,
        title [0] VisibleString,
        number EmployeeNumber,
        dateOfHire [1] Date,
        nameOfSpouse [2] Name,
        children [3] IMPLICIT SEQUENCE OF ChildInformation DEFAULT { } }
ChildInformation ::= SET {
        Name,
        dateOfBirth [0] Date }
Name ::= [APPLICATION 1] IMPLICIT SEQUENCE {
        givenName VisibleString,
        initial VisibleString,
        familyName VisibleString }

EmployeeNumber ::= [APPLICATION 2] IMPLICIT INTEGER

Date ::= [APPLICATION 3] IMPLICIT VisibleString -- YYYYMMDD
```

(b) ASN.1 description of the record structure

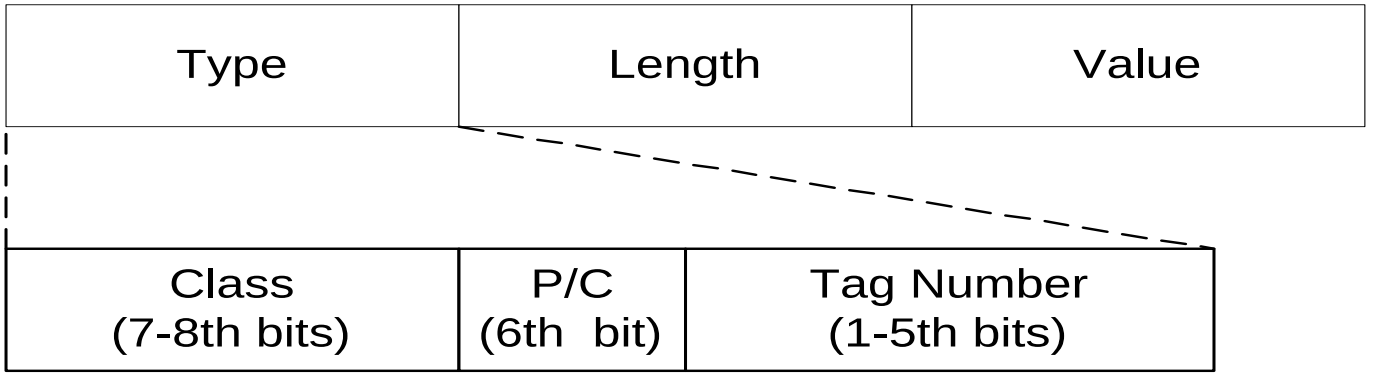
```
{                               {givenName "John", initial "T", familyName "Smith"},
        title                   "Director"
        number                  "51"
        dateOfHire              "19710917"
        nameOfSpouse            {givenName "Mary", initial "T", familyName "Smith"},
        children
        { {                     {givenName "Ralph", initial "T", familyName "Smith"},
           dateOfBirth                  "19571111"},
          {                     {givenName "Susan", initial "B", familyName "Jones"}
           dateOfBirth                  "19590717"}}}
```

(c) ASN.1 description of a record value

Object Name


itu
0
iso
1
iso-itu
2
org
3
dod
6
internet
1
private
4
enterprise
1
IBM
2

TLV Encoding



Class	8th bit	7th bit
Universal	0	0
Application	0	1
Context-specific	1	0
Private	1	1

Macro

```
<macroname> MACRO ::=
BEGIN
    TYPE NOTATION ::= <syntaxOfNewType>
    VALUE NOTATION ::= <syntaxOfNewValue>
    <auxiliaryAssignments>
END
```

Example:

```
CS8803 OBJECT-IDENTITY
    STATUS      current
    DESCRIPTION   "A graduate-level network
    management course offered every fall by
    College of Computing in Georgia Institute of
    Technology."
        ::= {csclasses 50}
```

Functional Model