

# Netzwerkmanagement

→ Grundlagen: Abstract Syntax Notation One

**Prof. Dr. Norbert Pohlmann**

Fachbereich Informatik

Verteilte Systeme und Informationssicherheit



Fachhochschule  
Gelsenkirchen

# Inhalt

---

- **Einführung**
- **Verwendung von ASN.1 beim Netzmanagement**
- **Module**
- **Typen und Werte**
- **Zusammenfassung**

# Inhalt

---

- **Einführung**

- Verwendung von ASN.1 beim Netzmanagement

- Module

- Typen und Werte

- Zusammenfassung

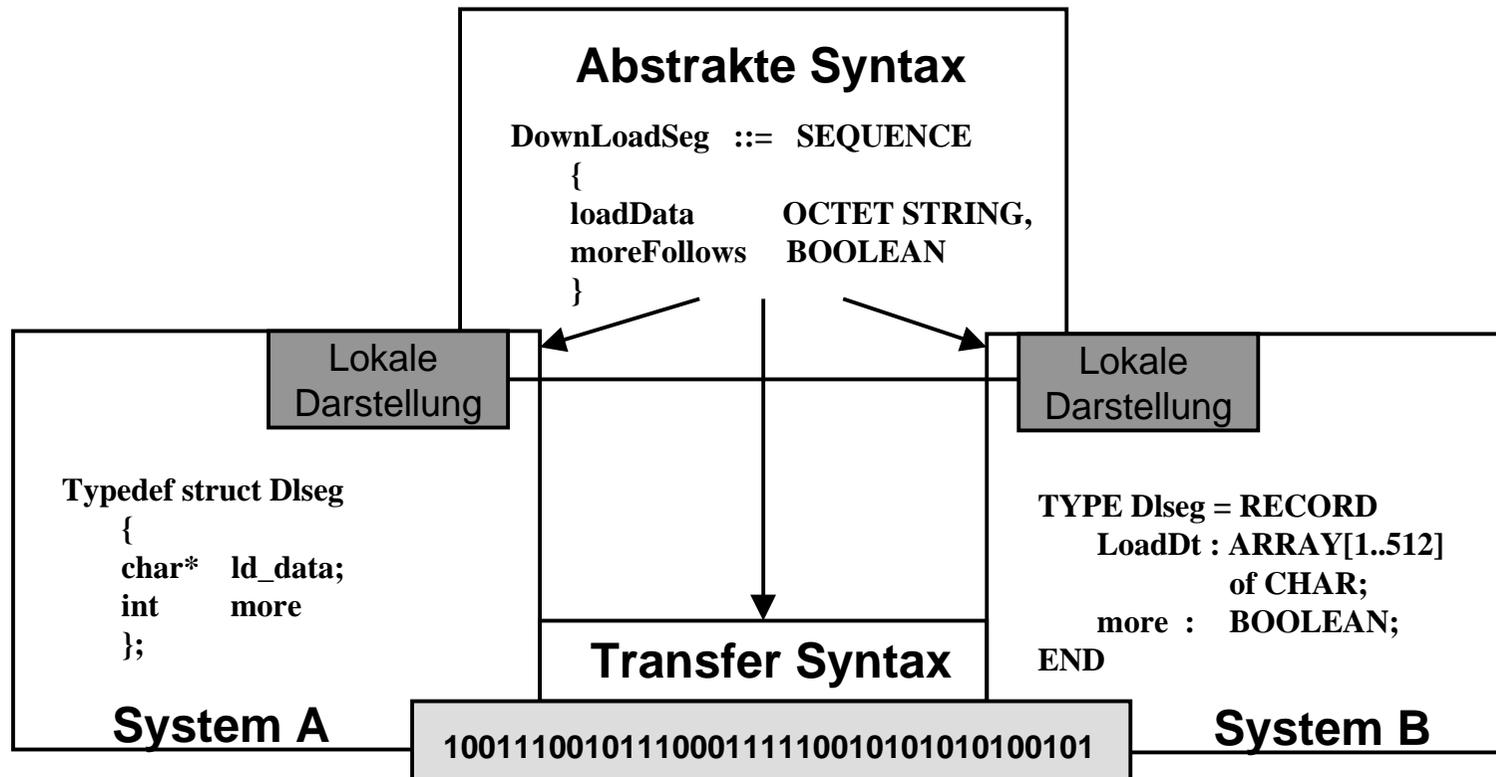
# Einführung

## → Übersicht

- Die Darstellungsebene (Schicht 6) des OSI-Referenzmodells hat die Aufgabe, die Beschreibung und Kodierung von Informationen in einer gemeinsamen Sprache des offenen Systems zu unterstützen.
- Während die einzelnen Anwendungsinstanzen lokal über unterschiedliche und systemabhängige Datenrepräsentationen (z.B. ASCII oder EBCDIC) verfügen können, muss beim Datentransfer jeder Kommunikationspartner seine lokalen Darstellungen der Daten (**lokale Syntax**) in eine von beiden Seiten als gemeinsam akzeptierte Darstellungsform umwandeln.
- Diese kodierungsunabhängige Beschreibung der Protokolldateneinheiten (PDU) für eine Anwendung wird als **abstrakte Syntax**, die Sprache zur Spezifikation dieser abstrakten Datenstrukturen und Werte als **abstrakte Syntaxnotation** bezeichnet.
- Für die Kommunikation in offenen Systemen wurde eine derartige abstrakte Syntaxnotation mit der Bezeichnung **Abstract Syntax Notation One (ASN.1)** von der ISO als Norm ISO 8824 festgelegt.

# Einführung

## → Funktionalitäten der Darstellungsschicht



# Einführung

## → Transfersyntax

---

- Die Tatsache, dass die Instanzen der verteilten Anwendungen von der Syntax der auszutauschenden Daten ein gemeinsames Verständnis haben, reicht zur Kommunikation noch nicht aus.
- Erforderlich ist darüberhinaus die Übermittlung der Daten zwischen den Instanzen der Anwendungen.
- Nötig sind also weitere Definitionen, die beschreiben, wie jeder Wert einer abstrakt beschriebenen Datenstruktur bitweise kodiert wird (**Transfersyntax**).
- Zur Ergänzung der Sprache ASN.1 für die Definition einer abstrakten Syntax wurden im ISO-Standard ISO 8825 Kodierungsregeln (**Basic Encoding Rules, BER**) definiert, durch die die Transfersyntax festgelegt ist.

# Inhalt

---

- Einführung
- **Verwendung von ASN.1 beim Netzmanagement**
- Module
- Typen und Werte
- Zusammenfassung

# Verwendung von ASN.1

## → Netzwerkmanagement

---

- Für das Netzwerkmanagement mit SNMP wird ASN.1 für die folgenden zwei Aspekte eingesetzt:
  - Zum einen werden sämtliche Managementinformationen (Managed Objects) für das Netzmanagement in ASN.1 beschrieben.
  - Zum anderen dient ASN.1 als Beschreibungssprache für das eigentliche SNMP-Protokoll.
- ASN.1 ist eine formale Sprache, sie wird also mit Hilfe einer Grammatik beschrieben.
- Damit ist es möglich, ASN.1 Compiler zur Verfügung zu stellen, um die Syntax zu überprüfen oder auch den Code zu generieren.
- Das Internet Standard Network Management Framework benutzt nur einen Teil der Möglichkeiten von ASN.1.

# Inhalt

---

- Einführung
- Verwendung von ASN.1 beim Netzmanagement
- **Module**
  - Typen und Werte
  - Zusammenfassung

# ASN.1

## → Module (1/4)

- Eine Sammlung von ASN.1-Beschreibungen, die zu einem gemeinsamen Thema gehören, wird als Modul bezeichnet.
- Die oberste Syntax für ein Modul ist einfach:

```
<<module>> DEFINITIONS ::=
    BEGIN
        <<linkage>>
        <<declarations>>
    END
```
- Der Ausdruck <<module>> bezeichnet das Modul sowohl informell als auch möglicherweise zuverlässig.
- Der Modulname muss zuverlässig (eindeutig) sein, falls das Modul im <<linkage>> andere Module importieren soll.

# ASN.1

## → Module (2/4)

---

- Als Beispiel für das Importieren von Definitionen aus anderen ASN.1 Modulen, sei hier das OBJECT-TYPE Makro zur Definition von Managementobjekten erwähnt.
- Dieses Makro muss von allen MIB's, die Managementinformationen beschreiben, folgendermaßen importiert werden:  
    IMPORT OBJECT-TYPE  
        FROM RFC-1212;
- RFC-1212 ist dabei der Name des Moduls, in dem das OBJECT-TYPE Makro definiert wurde.
- Der Ausdruck <<declarations>> enthält die tatsächliche ASN.1 Definition.

# ASN.1

## → Module (3/4)

---

- Drei Arten von Objekten sind mit Hilfe von ASN.1 definiert:
- **types:** definiert neue Datenstrukturen und beginnt grundsätzlich mit einem Großbuchstaben z.B. DisplayString
- **values:** sind Instanzen (Variablen) eines bestimmten Typs und beginnen grundsätzlich mit einem Kleinbuchstaben z.B. internet
- **macros:** werden benutzt, um die Grammatik der ASN.1 Sprache zu erweitern und werden komplett in Großbuchstaben geschrieben z.B: OBJECT-TYPE
- Die Schlüsselwörter der ASN.1 Sprache erscheinen grundsätzlich alle in Großbuchstaben.

# ASN.1

## → Kommentare (4/4)

---

- Kommentare werden in ASN.1 mit zwei aufeinanderfolgenden Bindestrichen eingeleitet „--“.
- Alle nachfolgenden Anweisungen bis zum Zeilenende werden damit als Kommentar betrachtet.

# Inhalt

---

- Einführung
- Verwendung von ASN.1 beim Netzmanagement
- Module
- **Typen und Werte**
- Zusammenfassung

# ASN.1

## → Typen und Werte

- Ein neuer ASN.1-Typ wird mit einer einfachen Syntax definiert:

NameOfType ::= TYPE

- Ähnlich wird auch der Wert (oder besser eine Instanz eines Datentyps) definiert:

nameOfValue NameOfType ::= VALUE

- Das heißt, zuerst wird die Variable mit einem Namen versehen (nameOfValue), dann wird ihr Typ festgelegt (NameOfType) und schließlich ein Wert zugewiesen.
- Eine Wertdefinition kann z.B. zur Spezifikation bestimmter, fest vorgeschriebener Werte (Konstanten) benutzt werden:

internet            OBJECT-IDENTIFIER ::= { 1 3 6 1 }

mgmt                OBJECT-IDENTIFIER ::= { internet 2 }

mib-2                OBJECT-IDENTIFIER ::= { mgmt 1 }

# ASN.1: Typen und Werte

## → Einfache Typen (1/2)

---

- Die einfachen Datentypen, die im Management-Rahmen benutzt werden, sind:
- **INTEGER:**
  - Ein Datentyp, der eine natürliche Zahl als Wert hat.
  - Da ASN.1 das Konzept eines Objekts beschreibt, ist keine Bitgenauigkeit definiert.
  - Oftmals ist es praktisch, symbolische Namen für Werte vergeben zu können, z.B.:

```
Status ::= INTEGER { up(1), down(2), testing(3) }  
myStatus Status ::= up -- entspricht myStatus ::= 1
```
  - Als Kovention im Management-Rahmen gilt, falls es symbolische Werte für einen ganzzahligen Datentyp gibt, dass nur diese Werte für Instanzen diese Datentyps gültig sind.

# ASN.1: Typen und Werte

## → Einfache Typen (2/2)

---

### ■ **OCTET STRING:**

- Ein Datentyp, der kein oder mehrere Oktette als Wert hat.
- Jedes Byte in einem Oktett-String kann einen Wert zwischen 0 und 255 annehmen.

### ■ **OBJECT-IDENTIFIER:**

- Ein Datentyp, der die Objekte in der MIB eindeutig identifiziert.
- Ein OBJECT-IDENTIFIER ist eine Folge von nicht negativen ganzzahligen Werten, die einen Baum (Management Information Tree) durchlaufen.

### ■ **NULL:**

- Ein Datentyp, der als Platzhalter fungiert.
- Obwohl der Management-Rahmen die Existenz dieses Datentyps zugelassen hat, wird er momentan nicht benutzt.

# ASN.1: Typen und Werte

## → Zusammengesetzte Typen

---

- Die zusammengesetzten ASN.1-Typen, die im Management-Rahmen benutzt werden sind:
- **SEQUENCE:**
  - Ein Datentyp, der eine geordnete Liste aus anderen ASN.1-Datentypen enthält.
  - Dies entspricht in etwa einer Struktur in der Programmiersprache C.
- **SEQUENCE OF:**
  - Ein Datentyp, der eine geordnete Liste eines ASN.1-Typs enthält.
  - Dies entspricht einem dynamischen ARRAY in den meisten P-Sprachen.
  - Die Anzahl der Elemente ist dabei normalerweise erst bekannt, wenn das ARRAY erzeugt wird.
- **CHOICE:**
  - Der Datentyp CHOICE enthält eine Auswahl von mehreren ASN.1 Typen und entspricht dem union Datentyp in der Programmiersprache C.

# ASN.1: Typen und Werte

## → Angehängte Datentypen (1/3)

---

- ASN.1 bietet zusätzlich noch die Möglichkeit der Definition neuer Typen durch das „Anhängen“ (tagging) an einen vorher definierten Typ.
- Der neuen und alten Typen sind durch unterschiedliche Tags zu unterscheiden, beziehen sich aber auf denselben Datentyp.
- Tags werden zur Beschreibung von ASN.1 Datentypen verwendet.
- Ein solcher ASN.1 Tag setzt sich aus einer TAG-Klasse und einer Tag-Nummer zusammen.
- Bei der Kodierung das Abstrakten ASN.1 Syntax in die entsprechende Transfersyntax (BER) wird anstelle des Datentyps die Tag-Klasse und die Tag-Nummer dieses Datentyps übertragen.
- Sowohl dem Sender als auch dem Empfänger muss die Bedeutung von Tag-Klasse und -Nummer bekannt sein.

# ASN.1: Typen und Werte

## → Angehängte Datentypen (2/3)

- ASN.1 definiert die folgenden vier Tag-Klassen:
  - UNIVERSAL, PRIVATE
  - APPLICATION, CONTEX SPECIFIC
- Von diesen vier Tag-Klassen werden nur die UNIVERSAL und die APPLICATION Tag-Klassen für das Netzmanagement verwendet.
- **UNIVERSAL-TAGS** sind im ASN.1-Standard vordefinierte und weltweit eindeutige. Beispielsweise wird der Typ INTEGER mit [UNIVERSAL 2] beschrieben, wobei „2“ die Tag-Nummer darstellt.
- Ausschnitt der Universal Tags:

Tag-Nummer	ASN.1 Typ
1	BOOLEAN
2	INTEGER
3	BIT STRING
4	OCTET STRING
5	NULL
6	OBJECT IDENTIFIER

# ASN.1: Typen und Werte

## → Angehängte Datentypen (3/3)

- **APPLICATION-Tags** können innerhalb bestimmter Anwendungsklassen bzw. Standards zugewiesen werden und müssen innerhalb dieser Standards eindeutig sein.
- Für das Internet-Netzmanagement werden neue Datentypen (z.B. Counter oder IpAdress) unter Verwendung der Tag-Klasse APPLICATION definiert.  
NameOfType ::= [Tag-Class Tag-Nummer] IMPLICIT TYPE
- Damit wird ein neuer Datentyp NameOfType basierend auf dem bereits definierten TYPE erzeugt.
- Der neue Datentyp besitzt die Tag-Klasse Tag-Class und die Tag-Nummer Tag-Nummer und ist somit vom bisherigen TYPE unterscheidbar.
- Vom Konzept her führt das Anhängen von Datentypen zu einem „Verpacken“ des bestehenden Datentyps, des Tags und Inhalts in einen Datentyp.
- Das Schlüsselwort IMPLICIT ist optional und wird benutzt, falls der neue Datentyp entsp. den Regeln des alten Typs verpackt werden soll.

# ASN.1: Typen und Werte

## → Untertypen

- Zur Verfeinerung der Datentypen in ASN.1 können sogenannte Untertypen definiert werden.
- Von den vielen Möglichkeiten von Untertypen verwendet der Management-Rahmen nur die zwei folgenden:

```
IpAddress ::= - - in network byte order
            [ APPLICATION 0 ]
            IMPLICIT OCTET STRING ( SIZE (4))
```

- In diesem Beispiel wird ein neuer Datentyp IpAddress definiert, der eine Zeichenkette von Oktetten enthält.
- Über den SIZE Operator wird die Länge der Zeichenkette auf genau 4 Zeichen festgelegt.
- Counter ::=
 [ APPLICATION 1 ]
 IMPLICIT INTEGER (0 ... 4294967295)

Hier wird ein neuer ganzzahliger Datentyp definiert, dessen Instanzen Werte von 0 bis  $2^{32}$  annehmen kann.

# Inhalt

---

- Einführung
- Verwendung von ASN.1 beim Netzmanagement
- Module
- Typen und Werte
- **Zusammenfassung**

# Zusammenfassung

---

- Mit der Hilfe von **Abstract Syntax Notation One (ASN.1)** kann eine kodierungsunabhängige Beschreibung der Protokolldateneinheiten (PDU) erreicht werden.
  - **Abstrakte Syntax** beschreibt die Kodierungsunabhängigkeit.
  - **Syntaxnotation** ist die Sprache zur Spezifikation der abstrakten Datenstrukturen und Werte.
- Die **Basic Encoding Rules (BER)** definieren die Transfersyntax, d.h. wie jeder Wert einer in ASN.1 beschriebenen Datenstrukturen bitweise kodiert wird.
- Mit Hilfe einer Teilmenge von ASN.1 werden alle Managementinformationen (Managed Objects) für das Netzwerkmanagement beschrieben.

# Netzwerkmanagement

→ Grundlagen: Abstract Syntax Notation One

**Vielen Dank für Ihre Aufmerksamkeit**

**Fragen ?**

[norbert.pohlmann@informatik.fh-gelsenkirchen.de](mailto:norbert.pohlmann@informatik.fh-gelsenkirchen.de)

