

**Zeitschrift:** Technische Mitteilungen / Schweizerische Post-, Telefon- und Telegrafienbetriebe = Bulletin technique / Entreprise des postes, téléphones et télégraphes suisses = Bollettino tecnico / Azienda delle poste, dei telefoni e dei telegrafi svizzeri

**Herausgeber:** Schweizerische Post-, Telefon- und Telegrafienbetriebe

**Band:** 67 (1989)

**Heft:** 5

**Artikel:** Das Getränk im OSI-Weinglas

**Autor:** Pitteloud, Joseph

**DOI:** <https://doi.org/10.5169/seals-874934>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

**Download PDF:** 30.04.2026

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

# Das Getränk im OSI-Weinglas

Joseph PITTELOUD, Berne

## Präsentations- und Anwendungsschichten sowie Zielsetzungen

Zusammenfassung. Der Autor gibt einen Überblick über den Ende 1988 erreichten Normierungszustand der Präsentations- und Anwendungsschichten in der OSI-Architektur und versucht die Zielsetzungen und die Anwendbarkeit der Normen kurz zu erläutern. Einige Entscheidungselemente sind dargestellt.

### Le breuvage du verre à vin OSI – Couches de présentation et d'application et enjeux

Résumé. L'auteur aborde de manière synthétique l'état, à fin 1988, de la normalisation des couches de présentation et d'application dans l'architecture OSI et tente d'esquisser les enjeux d'OSI et l'applicabilité des normes. Quelques éléments de décision sont proposés.

### Gli strati OSI di «presentazione e applicazione» – obiettivi

Riassunto. L'autore presenta una sintesi dello stato di normazione – alla fine del 1988 – degli strati di presentazione e applicazione nell'architettura OSI e illustra brevemente gli obiettivi e l'applicabilità delle norme. Fornisce inoltre alcuni elementi decisionali.

## 1 Allgemeines

Ein vorausgehender Artikel (Techn. Mitt. PTT Nr. 2/1989) [1] hat versucht, das synthetische Werk der OSI-Architektur (Open System Interconnection) durchzukämmen. Dort wurden die Transport- und Kommunikationssteuerschichten erläutert.

Im vorliegenden Heft beleuchtet der Autor die beiden obersten Schichten und den Einsatz von OSI. Die Lektüre des ersten Artikels wird empfohlen, da auf dort verwendete Vergleiche hingewiesen wird.

## 2 Die Präsentationsebene

### 21 Die Syntaxen

#### 211 Der Begriff der Syntax

Die Syntax definiert die binäre Darstellung der auszutauschenden Datenelemente. Sie unterstützt die Semantik, die die konzeptuelle Bedeutung definiert, die für eine bestimmte Anwendung der verschiedenen Daten spezifisch ist. Man unterscheidet vier Syntaxen (Fig. 1 und 2):

- Eine *abstrakte Syntax*, die die Anwendung unabhängig von der Programmiersprache definiert. Es ist eine Art Konzeptsprache, eine Metasprache auf der *Bedeutungsebene* des Sprachlichen. Sie erlaubt den Sinn der Informationen unabhängig vom Träger zu beschreiben. Dieser Sinn ist die Semantik, die von den zwei Offizieren geteilt und verstanden wird.
- Die *konkrete Syntax des offenen Systems A* ist jene der lokal verwendeten Programmiersprache (z. B. Pascal oder die französische Sprache des Offiziers).
- Die *konkrete Syntax des offenen Systems B* ist in den meisten Fällen eine andere als im System A (also z. B. die UNIX-Sprache C oder eben die deutsche Sprache des anderen Offiziers).
- Eine *konkrete Transfersyntax*, die die gemeinsamen Regeln des Austausches zwischen den Systemen definiert, d. h. Regeln für die Codierung und Decodierung, die jedes offene System beherrschen muss, um lokal seine eigene Syntax verwenden zu können, damit es in der Lage ist, für eine gegebene Anwendung mit der gegenüberliegenden Seite zu kommunizieren.

Die Transfersyntax definiert den Inhalt des «User-data»-Oktetts der Kommunikationssteuerung, welche den Inhalt der Anwendungsschicht transportiert (in der Analogie die englische Sprache).

Im sprachlichen Bereich befinden sich die drei Konkretsyntaxen auf der Ebene der *Bedeutung*, das heisst der materiellen Darstellung eines Zeichens, als Folge von Lauten oder Buchstaben, die den Sinn ergeben.

Gegenwärtig sind eine einzige Abstraktsyntax (X.208) und nur eine Transfersyntax (X.209) definiert. Man darf hoffen, dass die Normierer es damit bewenden lassen, sonst sind die Aussichten für ein syntaktisches Englisch-Esperanto zwischen offenen Systemen gering.

#### 212 Die Abstraktsyntax (CCITT X.208)

Die Abstraktsyntax, genannt ASN.1 (Abstract Syntax Notation One) ist eine formale Schreibweise, verwendet zur Beschreibung von Anwendungen unabhängig von der binären, konkreten Darstellung der «types», d. h. der zu übertragenden Datenelemente.

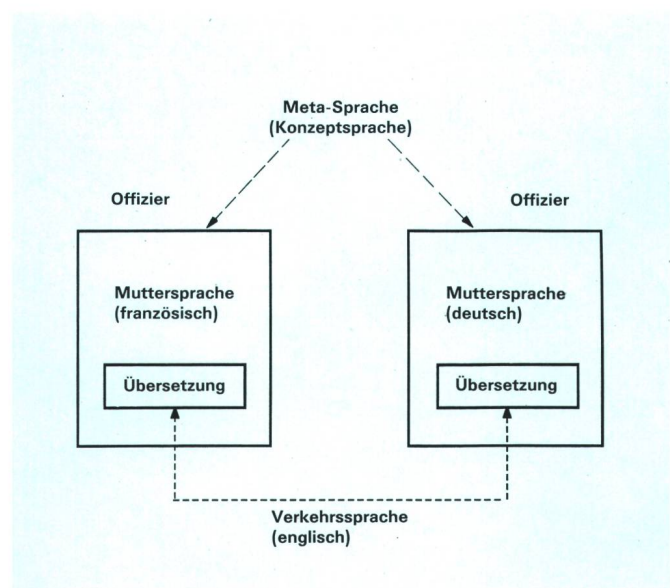


Fig. 1 Sprachen

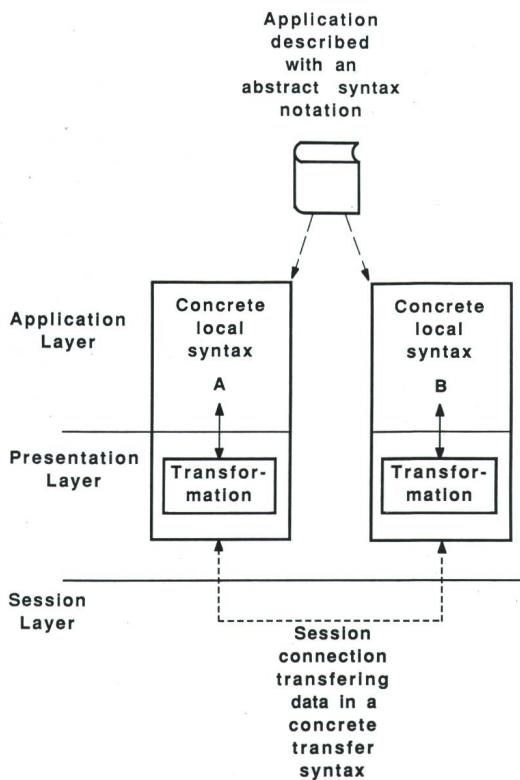


Fig. 2  
Syntaxen

Diese Schreibweise braucht die gleichen Definitionsprinzipien wie eine Programmiersprache. Jeder Datentyp ist identifiziert durch einen zugeordneten Klassennamen (tag class) und seine Nummer innerhalb der Klasse (tag number). Den Typen können auch Werte zugeordnet werden. Man unterscheidet vier Klassen von «tags»:

- *Universelle Klasse:* Sie ist bei allen Anwendungen einsetzbar, zum Beispiel:
  - einfache Typen wie BOOLEAN, INTEGER, BIT-STRING, OCTET STRING usw.
  - Typen zur Definition von Zeichenketten, die durch normierte Charaktersätze spezifiziert sind wie IA5STRING, TELETEXSTRING, VIDEOTEXSTRING usw.
  - Typen, die aus einfachen Typen zusammengesetzt sind wie eine geordnete Folge (SET), eine Auswahl (CHOICE) unter Typen.
- *Für eine ganze Anwendung gültige Klasse:* Diese Typen sind anwendungsspezifisch. Jede Anwendung kann sich so eigene «tags» festlegen, die einen für die zwei offenen Systeme ganz präzisen Sinn haben: So kann z. B. der «tag» [APPLICATION 0] bei einer bestimmten Bankanwendung eine Kontonummer bedeuten, während er im Mitteilungsdienst eine Adresse des Empfängers ist.
- *Umgebungsabhängige Klasse (context specific):* Sie erlaubt in einer konkreten Anwendungsumgebung, einen Typ vom anderen zu unterscheiden: Wenn im Mitteilungsdienst der Meldungskopf übertragen wird,

kann man mit diesem Typ von «tag» die verschiedenen Elemente des Kopfes in ihrer Reihenfolge unterscheiden (wie z. B. die Meldungsreferenz, auf die geantwortet wird, der Titel der Meldung usw.).

- *Private Klasse:* Diese Typen sind für betriebsinterne Anwendungen vorgesehen; sie sind weder von ISO noch vom CCITT normiert.

Der rekursive Charakter der Schreibweise ermöglicht es, komplexe Baumstrukturen für den Datenaustausch zwischen den Anwendungen zu definieren. Typischerweise findet man in diesen abstrakten Beschreibungsstrukturen des Datenaustausches die Typen der für eine ganze Anwendung gültigen Klasse im Stamm und in den Hauptästen (Fig. 3) (A[0], A[1] und A[2]). Die Verzweigungen vom Stamm über die Hauptäste bis zu den Zweigen werden durch universelle Typen gebildet (SET, SEQUENCE, CHOICE). Die kontextspezifischen Typen dienen z. B. in einem SET zur Identifizierung der verschiedenen Äste (C[0], C[1]). Die semantischen Inhalte dagegen werden von den Blättern getragen, welche durch einfache Typen definiert sind (NUMERICSTRING, IA5STRING usw.).

### 213 Die Transfersyntax (CCITT X.209)

Heute ist nur eine konkrete Transfersyntax für die Abstraktsyntax ASN.1 definiert. Jeder Datenwert wird in der in Figur 4 gezeigten Form codiert und übermittelt:

- 1 oder mehrere Identifikations-Oktetts
- 1 oder mehrere Oktetts zur Längenangabe
- 1 oder mehrere Oktetts als Inhalt.

Die Identifikation definiert die Klasse des Datentyps (tag class von ASN.1) und in dieser Klasse die Nummer des

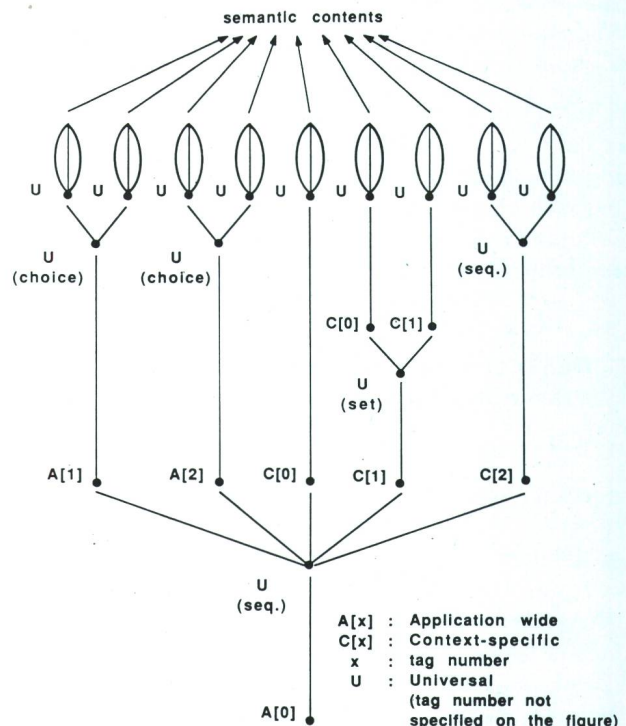


Fig. 3  
Durch die Abstraktsyntax dargestellte semantische Baumstruktur

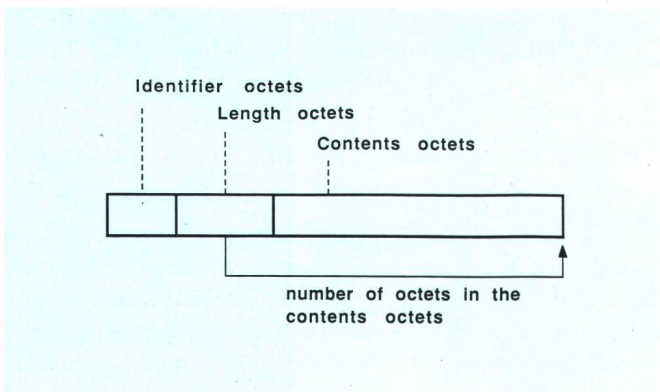


Fig. 4  
Codierstruktur der Transfersyntax

entsprechenden Typs. Wenn in einer Bankanwendung [APPLICATION 0] die Kontonummer ist, kann [APPLICATION 1] der Name des Kontoinhabers sein. Im Falle eines einzigen Identifikations-Oktetts bestimmen die Bits acht und sieben die Klasse (UNIVERSAL, APPLICATION, CONTEXT-SPECIFIC, PRIVATE), die Bits fünf bis eins die Nummer innerhalb der Klasse. Das Bit sechs bestimmt, ob die Codierung als «primitive encoding» oder «constructed encoding» ausgelegt ist. Im ersten Fall hat der Wert des Inhalts für die Anwendung eine Bedeutung (Blatt im Baum). Im zweiten Fall ist der Inhalt eine rekursive Zusammenstellung, die man in seine Elemente zerlegen muss, um am Ende der Kette «primitive» Typen zu finden (Stamm und Äste des Baumes, Fig. 3 und 5).

Die Länge gibt die Anzahl Oktetts des Inhalts an. Sie kann auf drei Arten definiert werden:

- die kurze Form, die es erlaubt, Inhalte bis zu 127 Oktetts zu definieren
- die lange Form, die mehr als ein Oktett benötigt, um die Länge zu spezifizieren
- die unbestimmte Form, die es erlaubt, «Ende des Inhalts» durch ein Spezialfeld («tag» der universellen Klasse) zu definieren.

Der Inhalt enthält die zu übermittelnde Substanz. Wenn das Feld von der primitiven Form ist, kann die Anwendung den Inhalt semantisch interpretieren. Wenn nicht, muss das offene System den Inhalt auswerten, um die Semantik zu finden, das heißt die in den Blättern der Baumstruktur enthaltenen «primitiven» Werte (Fig. 3).

## 22 Der Dienst und das Protokoll der Präsentationsschicht (CCITT X.216, X.226)

### 221 Die Rolle der Präsentationsschicht

Entgegen ihrem Namen kümmert sich die Präsentationsschicht nicht um die Darstellung der Daten (z. B. auf einem Bildschirm), sondern um die gegenseitige Präsentation der Informationen zwischen beiden Anwendungen von zwei offenen Systemen. Damit zwei Anwendungsinstanzen (Offiziere) in der Lage sind, miteinander zu kommunizieren, müssen sie nicht nur die Anwendung definieren, die sie behandeln wollen (spezifische Rolle), sondern sie müssen auch über die entsprechenden Syntaxen für die Anwendungen verhandeln (Metasprache). Die Anwendungsschicht (höherer mentaler Prozess) in-

formiert die Präsentationsschicht (niedrigerer mentaler Prozess), wer später darüber verhandeln muss, welches für die vorhandenen Abstraktsyntaxen die gegenseitig akzeptablen Transfersyntaxen sind (z. B. die englische Sprache). Danach muss sie während der ganzen Verbindungsdauer die nötigen Codierungen und Decodierungen zwischen den ausgehandelten Transfersyntaxen und der lokalen Konkretsyntax des offenen Systems machen (in unserem Beispiel französische und deutsche Sprache, Fig. 1 und 2).

Die Verknüpfung zwischen einer von der Anwendung vorgegebenen Abstraktsyntax und einer durch die Präsentation verhandelten Transfersyntax heißt Präsentationskontext, oder «Presentation Context».

### 222 Die Mechanismen der Präsentation

Drei Mechanismen sind für die Definition des Präsentationskontextes vorgesehen (Fig. 6):

- Die vorausgegangene Genehmigung (prior agreement)

In diesem Fall sind die Werte des Präsentationskontextes den Anwendungsinstanzen der beiden offenen Systeme schon vor der Eröffnung der Präsentationsverbindung durch bilaterale, vorausgegangene Vereinbarungen bekannt.

- Der «Default Context»

Der Defaultkontext wird bei der Erstellung der Präsentationsverbindung definiert und während der ganzen Verbindungsdauer verwendet, ausser wenn der folgende Ablauf zum Einsatz kommt:

- Der «Defined Context Set», DCS

Im Falle einer Sammlung von im voraus definierten Kontexten (DCS) besitzt jedes System eine Liste von verhandelten Präsentationskontexten. Jeder so verhandelte Wert steht für unmittelbare Verwendung durch die Anwendung bereit (Englisch, Spanisch, oder Russisch aus unserem Beispiel). Die DCS-Liste wird am Anfang eines Präsentationskontextes verhandelt und kann im Laufe einer Verbindung verändert werden (Alter Context Service), das heißt, es ist möglich, der Liste Präsentationskontexte zuzufügen oder solche zu entfernen. Es wird also vorausgesetzt, dass die Offiziere eine beachtliche Sprachbegabung haben!

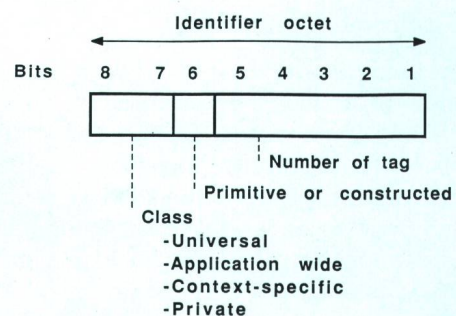


Fig. 5  
Identifikations-Oktett

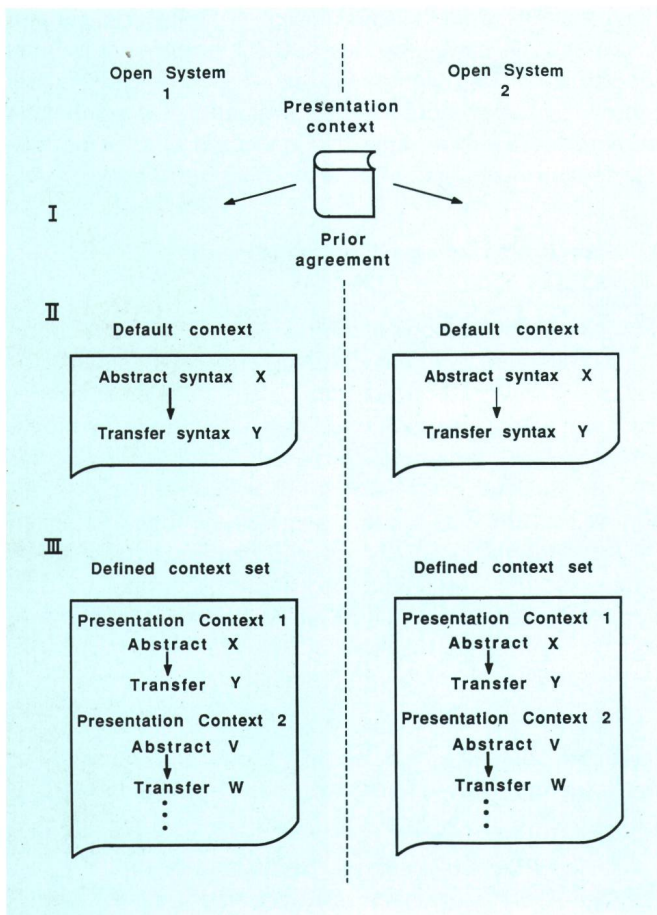


Fig. 6  
Präsentations-Mechanismen

## 223 Die Prinzipien des Protokolls

Der Defaultkontext und die anfängliche DCS-Liste werden durch den Austausch von «Presentation Protocol Data Units», PPDU, erstellt:

- Präsentationsverbindung, CP PPDU (Connect Presentation)
- Präsentationsverbindung angenommen, CPA PPDU (Connect Presentation Accept)
- Präsentationsverbindung zurückgewiesen, CPR PPDU (Connect Presentation Reject).

Änderungen der Kontextliste werden durch den Austausch folgender PPDU gemacht:

- Kontext geändert, AC PPDU (Alter Context)
- geänderter Kontext bestätigt, ACA PPDU (Alter Context Acknowledge).

Es gibt dabei keinen Verbindungsaufbau im eigentlichen Sinn: Die PPDU, CP, CPA, CPR sind im «User-Data»-Feld der Kommunikationssteuerungsschicht codiert (Fig. 10, P-connect, control information), dagegen werden die Protokolle AC und ACA als «Typed-Data»-Parameter der Kommunikationssteuerung transportiert. Auch wenn sie nicht zur Anwendung gehören, sind diese Parameter durch die ASN.1-Syntax (X.208) spezifiziert und können mit den Codierregeln von ASN.1 (X.209) codiert werden. Es ist zu hoffen, dass diese komplexen Mechanismen, die durch die Normengremien aufgestellt wurden, sich im Laufe der Zeit und mit der Erfahrung etwas festigen,

damit kein neuer Turm zu Babel mit Metasprachen und Transfersyntaxen gebaut wird.

## 3 Die Anwendungsebene

### 31 Die logische Struktur der Anwendung

In der OSI-Umgebung wird die Kommunikation zwischen zwei Anwendungsprozessen als eine Kommunikation zwischen Paaren von Anwendungsinstanzen (Application Entities, AE) dargestellt. Diese Instanzen (unsere Offiziere) sind getrennte logische Einheiten, denen normalerweise ein Softwareprozess entspricht, oder in der Protokollsprache: Eine Protokollmaschine (protocol machine). Sie stellen in unserer Analogie die mentalen Prozesse der Offiziere dar. Diese logischen Einheiten teilen sich auf in (Fig. 7 und 9):

- ein Benutzerelement (user-element), das den höchsten hierarchischen Teil des Anwendungsprozesses darstellt. Es macht von den logischen Elementen ASE (Application Service Elements) Gebrauch, um eine geforderte Anwendung zu verwirklichen. Es kennzeichnet in einer gewissen Weise die Rolle des Offiziers der betroffenen Anwendung (der Fahrzeugoffizier in unserer Analogie).
- eine Sammlung von ASE, die sich gegenseitig aufrufen können, um ihre Funktion zu erfüllen. Jedes dieser Elemente hat im Prinzip seine eigene Abstraktsyntax. Es ist, wie wenn jedes vorprogrammierte Verhalten des Offiziers durch eine eigene konzeptuelle Metasprache definiert wäre.

Eine besondere Einheit, genannt «single association controlling function», soll die einwandfreie Funktion unter all diesen Prozessen sichern. In unserer Analogie muss zwischen dem guten Funktionieren einer Einheit und dem IQ des Offiziers ein bestimmtes Verhältnis bestehen.

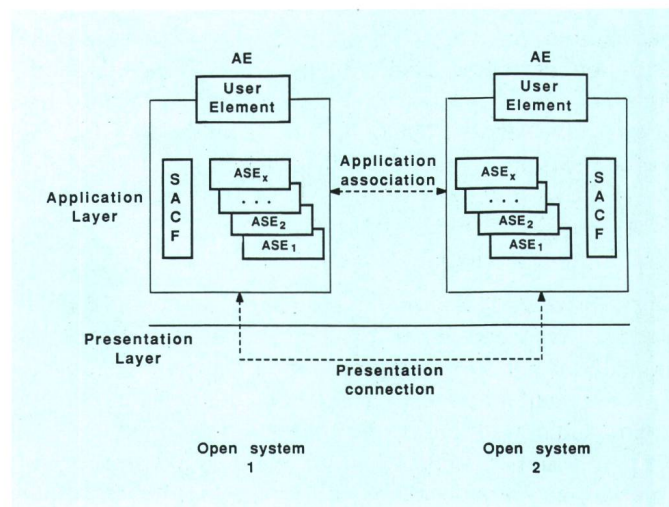


Fig. 7  
Struktur der Anwendungsebene  
AE Application Entity  
ASE Application Service Element  
SACF Single Association Controlling Function

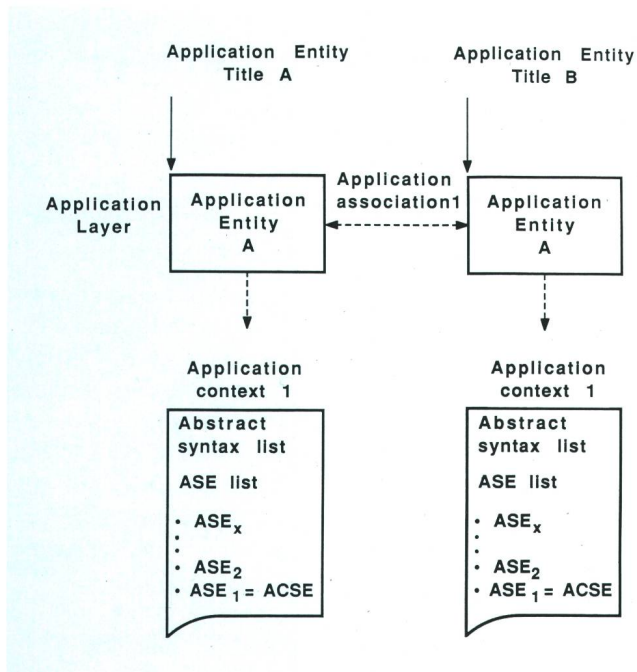


Fig. 8  
Anwendungselemente

Es scheint, dass die Softwarelösungen diese Konzepte nicht wie sie sind einsetzen können, und dass zwar gewisse Protokollabläufe konstruiert werden können, andere aber nur künstliche Konstruktionen ohne Sinn für den Anwender ergeben. Die Beschreibung dieser Anwendungsprozesse lässt noch zu wünschen übrig und wird sicherlich noch Änderungen erfahren.

Drei Konzepte sind wichtig für die Anwendungsschicht (Fig. 8):

- die Anwendungsverknüpfung (Application Association)
- der Anwendungskontext (Application Context)
- der Anwendungstitel (Application Title).

Die *Anwendungsassoziation* ist die Verknüpfung zweier AE, erstellt durch die Präsentationsverbindung. Während diese eine Pipeline für die Übertragung der syntaktischen Datenwerte bildet, spezifiziert die Anwendungsassoziation die vereinbarten Prozeduren zwischen AE und den verteilten Semantiken. Der darunterliegende mentale Prozess der Kommunikationssteuerung bestimmt die Art des Dialogs. Der nächsthöhere hat die Verkehrssprache ausgewählt (Präsentation). Die Anwendungsassoziation ist der darüberliegende mentale Prozess, welcher die Verbindung zwischen den zwei Rollen herstellt (Fahrzeugoffizier).

Der *Anwendungskontext* ist eine den Anwendungsinstanzen während einer ganzen Anwendungs-Assoziationsdauer zur Verfügung stehende Liste von «Application Service Elements». Diese Liste von logischen Protokollmaschinen definiert die Rolle einer Anwendungseinheit in einem offenen System (Liste der vorprogrammierten Verhaltensweisen, die nötig sind, um die Rolle des Fahrzeugoffiziers zu spielen).

Der *Anwendungstitel* wird durch die Informationselemente gebildet, die eine bestimmte Anwendungseinheit eines festgelegten offenen Systems irgendwo in der

Welt eindeutig identifizieren. Es ist z. B. der genaue Name und die Einteilung des deutschsprachigen Fahrzeugoffiziers im Kommandoposten Nr. 2. Diese Titel sollten einmal Elemente der Teilnehmerverzeichnisse (directories) werden, eingetragen durch eine namengebende Autorität, wie z. B. die Administrationen.

## 32 Der Assoziations-Kontrolldienst (CCITT X.217, X.227)

Der Assoziations-Kontrolldienst, genannt «Application Control Service Element» (ACSE), ist ein Basis-«Associations Service Element», das heißt, dass es systematisch von allen Spezifikationen der Anwendungskontexte verwendet werden sollte. Dieses einfache Werkzeug liefert die Basishilfsmittel für die Kontrolle einer Anwendungsassoziation zwischen zwei Anwendungsinstanzen (AE). Es definiert den Anfang und das Ende einer Assoziation für die durch den Anwendungskontext identifizierten ASE-Prozeduren. Es ist der «höhere» mentale Basisprozess des Offiziers, der über den Übermittlungssoldaten den Kontakt mit seinem Partner herstellt (Fig. 9).

Die Protokollmaschinen des ACSE tauschen hauptsächlich folgende «Application Protocol Data Units» (APDU) aus:

- AARQ APDU (A-Associate-Request, Fig. 10)
- AARE APDU (A-Associate-Response).

Für die Assoziationsfreigabe bezieht man sich auf folgende Protokolle:

- RLRQ APDU (A-Release-Request)
- RLRE APDU (A-Release-Response).

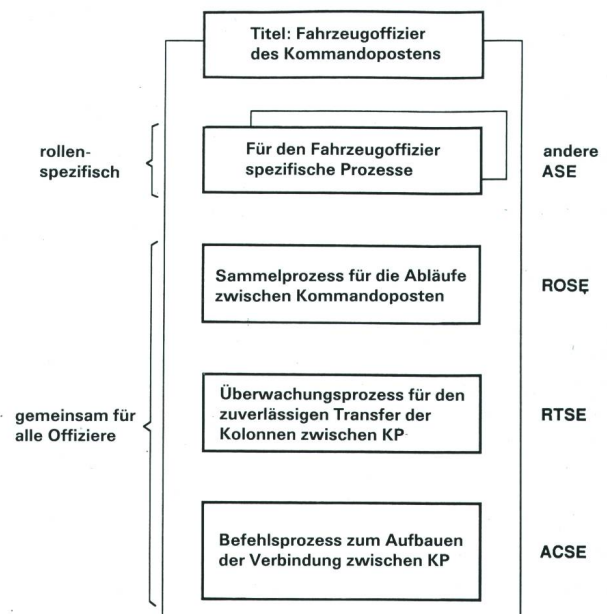


Fig. 9  
Einige mentale Prozesse des Offiziers  
ACSE Association Control Service Element  
RTSE Reliable Transfer Service Element  
ROSE Remote Operation Service Element  
ASE Application Service Element

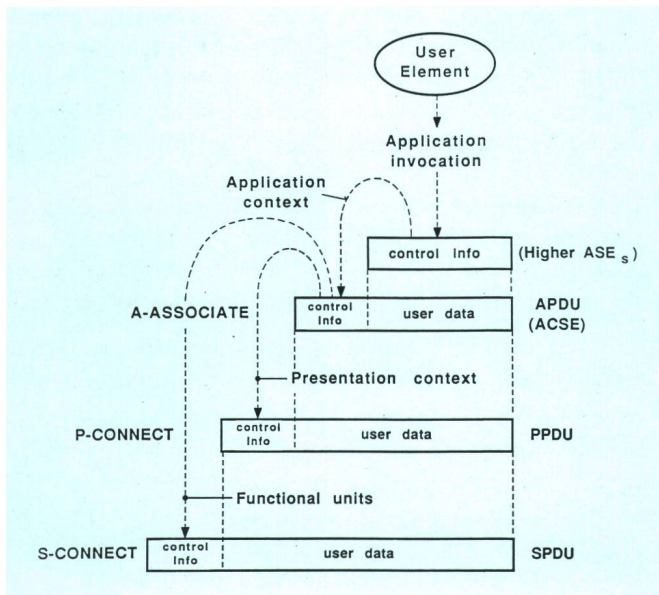


Fig. 10  
 Typischer Anwendungsassoziations-Aufbau  
 SPDU Session Protocol Data Unit  
 PPDU Presentation Protocol Data Unit  
 APDU Application Protocol Data Unit  
 ASE Application Service Element  
 ACSE Association Control Service Element (basic ASE)

Zur Erstellung der Assoziation enthält die AARQ APDU im wesentlichen (Fig. 10, A-Associate control information):

- den Namen des Anwendungskontextes, gültig für die Dauer der Assoziation (Fahrzeugoffizier)
- den Titel der rufenden Anwendung (Name des Offiziers von KP1)
- den Titel der gerufenen Anwendung (Name des Offiziers von KP2).

In der Antwort enthält die AARE APDU zusätzlich zum Anwendungskontext den Titel der effektiv antwortenden Anwendung (wenn nicht Offizier Meier, sondern Offizier Müller antwortet).

Der ACSE besitzt seine eigene Abstraktsyntax. Die APDU des Anwendungskontextes werden zurzeit durch die einzige definierte Abstraktsyntax (ASN.1) beschrieben. Die Codierregeln aus X.209 sind auf sie anwendbar.

Der Offizier könnte zuerst mit seinem Partner eine Kommunikationssteuerungs-Verbindung aufbauen, in der die Funktionseinheiten der Kommunikationssteuerung verhandeln würde, dann eine Präsentationsverbindung eröffnen, in der er die Verkehrssprache verhandeln würde (Präsentationskontext), dann endlich die Anwendungsassoziation mit dem mentalen Werkzeug ACSE erstellen, womit er den Anwendungskontext verhandeln würde.

Tatsächlich werden in den heutigen Anwendungen die drei oberen Schichten in einem mentalen Arbeitsgang vom Offizier *gleichzeitig* eröffnet (Fig. 10). Der Fahrzeugoffizier (user-element) entscheidet, eine Operation, genannt ROSE, zwischen KP1 und KP2 zu starten. Dieser höhere mentale Prozess übergibt das Argument der Operation ROSE dem unteren Prozess ACSE, indem er ihm mitteilt, zu welchem Zweck er die Anwendungsas-

soziation erstellen will (Anwendungskontext). Je nach Partner, den er im KP2 erreichen will, wird er die Verkehrssprache auswählen, die er zu gebrauchen gedenkt (Präsentationskontext). Dann, abhängig von der Rolle und der durchzuführenden Operation, wählt er die Dialogprozeduren, die er mit seinem Partner verwenden möchte (funktionelle Einheiten der Kommunikationssteuerung).

Erst wenn er in Gedanken Ordnung in alle diese Ideen gebracht hat, gibt der Offizier den Befehl zur Kommunikationssteuerungs-Verbindung (Abfahrbefehl für ersten Transport-Betriebspanzer); alle Einzelheiten der Kommunikationssteuerung, der Präsentation und Anwendung sind in dieser Protokolleinheit enthalten (Session Connect – Session Protocol Data Unit, SPDU, Fig. 10).

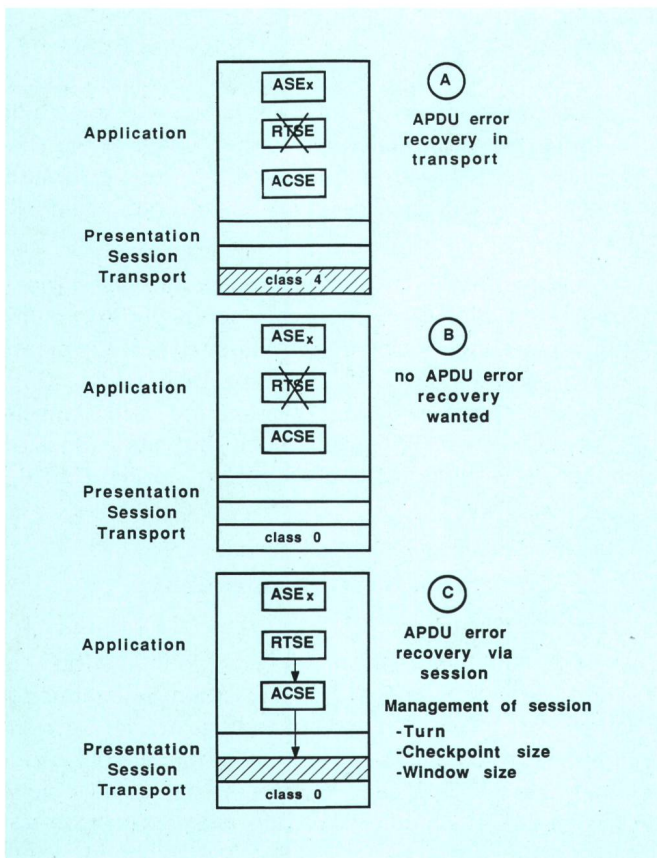
### 33 Der gesicherte Übertragungsdienst (CCITT X.218, X.228)

Der gesicherte Übertragungsdienst, genannt «Reliable Transfer Service Element» (RTSE), ist ein Baustein oder eher ein zusätzlicher mentaler Prozess in der Anwendungsschicht. Dieses Werkzeug bildet für sich ein «Application Service Element» (ASE) und hat seine eigene zugeordnete Abstraktsyntax. Der gesicherte Übertragungsdienst wird von der Anwendung nicht in jedem Fall benötigt:

Wenn die 7-Schichten-Säule einer OSI-Verbindung eine sichere Transportklasse verlangt (Klasse 4), die erlaubt, die Transport-«Protocol Data Units» zu nummerieren und eine «End-to-End»-Wiederholung auf der Transportebene zwischen zwei offenen Systemen zu veranlassen, ist der RTSE unnötig (dies ist der Fall bei den heutigen lokalen Netzen, Fig. 11a). Der Offizier muss im Falle von Übertragungsfehlern keine Wiederholungen machen, weil er dem Übermittlungssoldaten vertrauen kann.

Wenn andererseits eine Anwendung mit einer entsprechenden minimalen Transportschicht arbeitet (Klasse 0) akzeptiert sie, dass im Fehlerfall während der Übertragung von «Application Protocol Data Units» (APDU) von einem System zum andern keine Wiederholung möglich ist und dass die Übertragung vom Beginn der APDU wiederholt werden muss. Das Werkzeug RTSE ist also ebenfalls überflüssig (Fig. 11b). In unserer Analogie muss, sobald ein Panzer eine Panne hat, die ganze Panzerkolonne neu gesendet werden, sogar wenn der fehlende Panzer der letzte der Kolonne ist. Die Kommunikationssteuerung ist ebenfalls auf ein striktes Minimum reduziert (z. B. Kern/Kernel). Demzufolge werden die Mechanismen für Synchronisation und Wiederholung weggelassen, und man kann sich angesichts des mit administrativ-hierarchischer Trägheit zu bezahlenden Preises zurecht fragen, welches der Sinn der Dienste der Schichten 5 und 4 von OSI ist.

Dagegen hat RTSE seine volle Berechtigung bei der Übertragung von grossen Datenmengen (z. B. Kilo-Oktetts bis Mega-Oktetts je APDU). RTSE isoliert die anderen ASE der Anwendung von der Kommunikationssteuerung und bietet ihnen im Fehlerfall einen Wiederholungsmechanismus an, der die Menge zu wiederholender Information minimiert. Nach der transparenten Übertragung der APDU mittels des RTSE-Werkzeuges



**Fig. 11**  
**RTSE-Anwendung**  
 APDU Application Protocol Data Unit  
 RTSE Reliable Transfer Service Element

haben die anderen ASE die Garantie einer vollständigen und perfekten Übertragung jedes ihrer APDU zwischen den zwei offenen Systemen. Eine APDU von einigen Mega-Oktetts scheint riesig. Wenn man aber an das Übertragungsvolumen eines A4-Dokuments im Faksimile-Modus der Gruppe 4 denkt, versteht man, dass diese Datenmengen realistisch sind (Fig. 11c). Im letzteren Fall sichert sich der Offizier (Anwendung) nicht nur gegen die Fehler des Netzes, sondern auch gegen die seines Übermittlungssoldaten (Software der Schicht 4). Er hat die Garantie und die direkte Quittung von Offizier zu Offizier, von Gehirn zu Gehirn, ohne über seine Untergeordneten zu gehen. Auch bei Teletex, obschon dort der Übermittlungssoldat im Fehlerfall auf den Wiederholungsdienst der Paketebene im Netz (X.25) zurückgreifen kann, sendet der Offizier kleine Synchronisationspunkte nach jeder Seite des Dokuments und kontrolliert persönlich, dass das gesamte Dokument richtig empfangen wurde (grosser Synchronisationspunkt).

Der RTSE (oberer mentaler Prozess) benützt die Ressourcen der Kommunikationssteuerung (unterer mentaler Prozess) und kontrolliert den Dialog hauptsächlich durch die Definition folgender Funktionen:

- der zu verwendende Dialogmodus (monolog, two-way-alternate)
- die Kontrolle des «turn», um festzulegen, wer das Recht hat, die Anwendungsassoziation und die Kommunikationssteuerung zu starten und zu beenden

- die Grösse des «checkpoint size», das heisst die maximale Länge von Anwendungsdaten (in Einheiten von 1024 Oktetts), die versendet werden kann, bevor die Kommunikationssteuerung während des Austausches einen Synchronisationspunkt einfügt (minor synchronization point)
- das Fenster der Fehlerkontrolle (window size), das die Anzahl nicht quittierter kleiner Synchronisationspunkte definiert, die die Kommunikationssteuerung aussenden darf, bevor sie die Quittung erhalten hat.

Der RTSE benützt seinerseits den Anwendungskontroll-Dienst (ACSE) für die Erstellung und Freigabe der Anwendungsassoziation (Fig. 11c).

## 34 Der Fernbetriebsdienst ROSE (CCITT X.219, X.229)

### 341 Das Konzept der «Remote Operation»

Der Fernbetriebsdienst oder «Remote Operation Service Element» (ROSE) ist ein weiteres Werkzeug in der Werkzeugkiste der Anwendungsschicht (Gehirn des Offiziers) und von allen das ausgeklügeltste. Es dient hauptsächlich der Unterstützung von interaktiven Anwendungen in einer verteilten Umgebung. Eine Fernbetriebs-Elementaroperation wird von einer Anwendungsinstanz verlangt (Offizier). Sein Partner versucht sie auszuführen und informiert über das Ergebnis, wenn dies verlangt wird (Fig. 9).

Das ROSE-Werkzeug ist nicht nur eine Protokollmaschine wie RTSE und ACSE, sondern zusätzlich eine Notation zur Erzeugung von Operationen, die es der gewünschten Anwendung erlaubt, eigene ASE zu definieren (Fig. 12).

Die Stärke des ROSE-Werkzeuges liegt darin, dass es fähig ist, komplette Fernbetriebsoperationen unter Anwendung einer einzigen Makrooperation, definiert in Syntax ASN.1, festzulegen (wenn gewünscht mit Ergebnissen). Der Aufruf einer solchen Makrooperation verfolgt drei Ziele:

- den durch die Operation geleisteten Dienst zu beschreiben
- die notwendige APDU-Syntax abzuleiten, um die Operation zu verwirklichen
- ebenfalls die zwischen den APDU ausgetauschte Protokollsequenz abzuleiten.

Es handelt sich sicher um ein aufbauendes Werkzeug mit Zukunft. In unserer Analogie mit dem Offizier ist es nicht so ganz ein vorprogrammierter mentaler Prozess, sondern eher eine allgemeine *Wegleitung für alle «Operationen»*, die im militärischen Sinn zwei KP betreffen. Es ist der Leitfaden für alle KP-Offiziere, der auf alle Typen von Operationen anwendbar ist, sei es für die Verschiebung von Informationen, von Fahrzeugen, von Munition usw. Es genügt, anlässlich der konkreten Operation (gewünschte Anwendung) dieses generelle Werkzeug in Funktion der eigenen Bedürfnisse zu parametrisieren.

Ein Fernbetrieb kann in Funktion des erwarteten Ausführungsergebnisses analysiert werden (keine Rückmeldung, Rückmeldung nur im Fehlerfall, Rückmeldung im

Erfolgsfall oder Rückmeldung in jedem Fall). Er kann auch in Funktion des Betriebsmodus aufgeteilt werden:

- Synchron: Die Instanz, welche die Operation aufruft, erwartet vor dem nächsten Aufruf eine Antwort.
- Asynchron: Die aufrufende Instanz kann andere Operationen starten, auch bevor sie die Antworten der vorausgegangenen erhalten hat.

Der Fernbetrieb kann auch bezüglich Symmetrie betrachtet werden:

- Bei Symmetrie können die zwei ASE die gleiche Sammlung von Operationen aufrufen (Offizier der gleichen hierarchischen Ebene).
- Bei Asymmetrie spielt eine der ASE die Rolle des Lieferanten (supplier), die andere die Rolle des Konsumenten (consumer, Offiziere in unterschiedlichen hierarchischen Ebenen).

Man kann das Modell noch komplizierter machen, indem man gewisse Operationen unter sich verknüpft, die einen als erzeugende (parent-operation), die anderen als erzeugte (child-operations).

### 342 Die Notation «Remote Operation»

Die Notation «Remote Operation» basiert auf dem Konzept der in ASN.1 (X.208) definierten Makros. Sie stützt sich auf die Prinzipien der klassischen Programmierung, was erlauben sollte, automatisch Werkzeuge zu entwickeln die nach dieser Notation definierte Anwendungen in die Ausführungsumgebung dieser Anwendungen integrieren. Es sind vier Makros definiert:

- BIND. Dieses Makro erlaubt dem Anwendungs-«user-element», die Operation zu spezifizieren, um die Anwendungsassoziation zu erstellen. Durch diese Operation kann das «user-element» die Anwendungs-kontext-Parameter definieren, die dem untergeordneten

mentalenen Prozess zu übergeben sind. Das Argument der Operation BIND ist im Prinzip im Inhalt des «User-data»-Feldes des «A-Associate-Request» enthalten (Fig. 10).

- UNBIND. Dieses Makro erlaubt, die Assoziationsfreigabe-Operation zu spezifizieren, welche durch das «user-element» aufgerufen wird.
- OPERATION. Dieses Makro erlaubt, die durch das «user-element» aufgerufene ASE-Operation zu spezifizieren, sobald die Anwendungsassoziation erstellt ist.
- ERROR. Makro für die Ausnahmefälle.

Das Ergebnis einer Operation wird auf der gleichen Anwendungsassoziation wie der Operationsaufruf erzeugt. Es gibt im Prinzip keine Assoziationsauslösung, bevor alle aufgerufenen Operationen beendet sind.

### 343 Protokollelemente

Das Makro «Operation» erzeugt im Prinzip vier «Applications Protocol Data Units» (APDU):

- Remote Operation Invoke, ROIV APDU. Aufruf der Operation durch einen Offizier
- Remote Operation Response, RORS APDU. Übertragung der positiven Rückmeldung durch den gegenüberliegenden Offizier
- Remote Operation Error, ROER APDU. Übertragung einer Fehlermeldung im Falle eines negativen Resultats der Operation
- Remote Operation Reject, RORJ APDU. Rückweisung eines Antrags (invocation) oder einer Antwort (Resultat oder Fehler) von anderen Anwendungsinstanzen.

Diese APDU werden als «user data» übertragen, entweder durch RTSE, wenn angewendet, oder direkt durch die Präsentationsinstanz.

Ein Anwendungsbeispiel dieser mentalenen Prozesse (ACSE, RTSE und ROSE) wird Gegenstand eines späteren Beitrags sein. So werden im Protokoll zwischen dem «user agent» und seinem «message store» ROSE drei höhere ASE angewendet:

- für die Verwaltung der Parameter unter sich
- für die Meldungsablage vom «user agent» durch den «message store»
- für das Auffinden abgelegter Meldungen im «message store».

## 4 Schlussfolgerungen

### 41 Synthese (CCITT X.220)

Nach dieser kurzen Betrachtung der Schichten 6 und 7 und ihrer Werkzeuge ist es sicher nötig, eine Synthese der oberen Schichten (OSI 4...7) aufzustellen. Die Norm CCITT X.220 macht einen Versuch. Es ist die Beschreibung der Protokollstapel (OSI Protocol Stacks) für die CCITT-Anwendungen. Sie zeigt, dass die Dienste nicht auf die totale Definition des OSI-Wolkenkratzers warten können, um in Betrieb zu gehen, und je mehr sich die generellen Werkzeuge der sieben Schichten konsolidieren, desto häufiger werden sie gebraucht, um endgül-

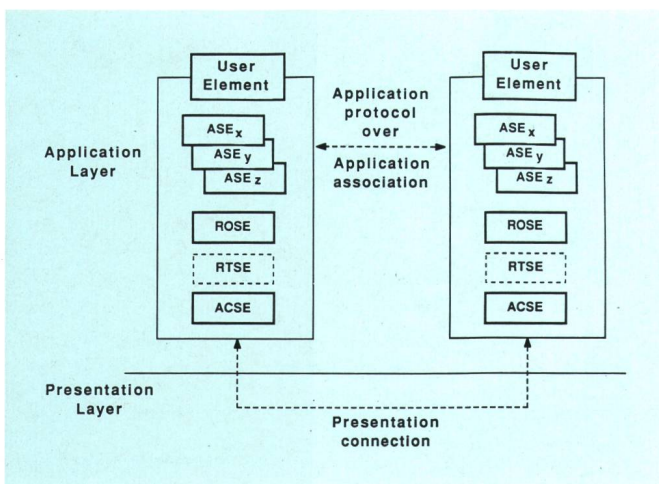


Fig. 12  
ROSE-Anwendung  
ASE<sub>x</sub>, ASE<sub>y</sub>, ASE<sub>z</sub> Werte der Elemente der Anwendungsdienste ASE<sub>s</sub>, die die RO-Schreibweise benutzen  
RO Remote Operation  
ROSE Remote Operation Service Element

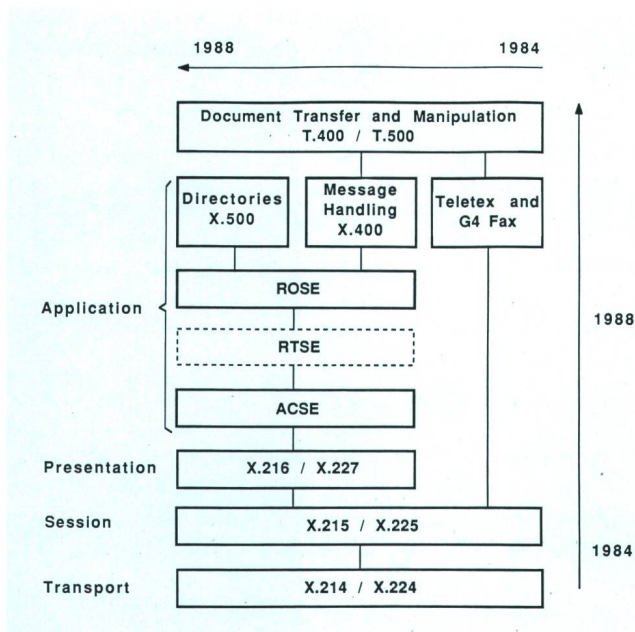


Fig. 13  
Entwicklung der Protokollsäule

tige, kommerzialisierte Dienste zu definieren (Fig. 13). Beispiele sind:

- der Teletex-Dienst, normiert 1980, für den es enormer Anstrengungen bedurfte, um die damals schlecht stabilisierte Transport- und Kommunikationssteuerschicht kompatibel mit jenen von 1984 und 1988 zu machen (CCITT T.62, T.70)
- der Mitteilungsdienst (message handling), normiert 1984, wo man mit Kunstgriffen die Rückwärtskompatibilität gewährleisten musste, um das Zusammenspiel der Versionen 1984 und 1988 zu sichern, weil 1984 die Präsentation und das ACSE-Konzept noch nicht definiert waren (CCITT X.400)
- der Teilnehmerverzeichnis-Dienst (directory), normiert 1988, wo alle Werkzeuge, die zurzeit in der Anwendungsschicht definiert sind, eingesetzt werden (CCITT X.500)
- Dokumentarchitektur (CCITT T.400/T.500) wendet ASN.1 an.

Die Entwicklung und die wachsende Komplexität des Prozesses sind als Abbild der Wandlungen unserer hochtechnologisierten Gesellschaft voraussehbar.

## 42 Kritische Bemerkungen

Die Konstruktion der OSI-Protokollsäule ist ein dynamischer Prozess, der sicher nicht beendet ist. Enorme Fortschritte wurden gemacht, und die Säule wird bereits in kommerziellen Diensten eingesetzt. Kritik hinterher ist immer einfacher als eine fehlerfreie Konstruktion; immerhin sollten einige Punkte noch bereinigt werden:

- der übertriebene Formalismus der Anwendung mit dem zu theoretischen Satz von mehreren Abstraktsyntaxen
- die Rolle, die den Schichten 5 und 6 in Zukunft tatsächlich gegeben werden soll, sobald sie mit echten Anwendungen konfrontiert werden. Wird sich in Zu-

kunft das Modell etwa in drei Makroschichten aufteilen?

- unterhalb des Transports
- der Transport (Übermittlungssoldat)
- die Anwendung (Offizier)
- die Art, mehr oder weniger voneinander abhängige mehrfache Prozesse zu beschreiben und zum Arbeiten zu bringen
- die Art, die Systeme zu öffnen, wenn mehr als zwei KP, das heisst zwei offene Systeme im Spiel sind.

## 43 Der Einsatz von OSI

Um die Analogie mit dem Weinglas wieder aufzunehmen: Dieses Glas wurde mit einem grossen Jahrgang gefüllt . . . , das Aroma entfaltet sich bereits, aber er hat noch nicht die richtige Temperatur. Die kostbare Flüssigkeit wird bald zum Degustieren bereit sein, das heisst um in unsere vertraute Umgebung gebracht zu werden. Sehen wir uns einmal an, wie man diesen kostbaren Trank degustiert, das heisst wie die verantwortlichen Entscheidungsträger von Kommunikationsnetzen jetzt ernsthaft die Anwendbarkeit von OSI in ihrem Zusammenhang evaluieren müssen. Nachfolgend sind einige Entscheidungselemente skizziert (Fig. 14 und 15):

### Verantwortungsgrenze

Im Fall, dass zwei Informationssysteme «on line» und kontrolliert Daten austauschen müssen, ermöglicht dies die OSI-Architektur, und zwar nicht nur nach dem Motto: «Öffne jedes der Systeme soweit wie möglich», sondern eher nach dem Motto: «Öffne die Systeme gerade soweit wie nötig». OSI kann zwischen zwei Unternehmungen oder auch innerhalb eines Unternehmens eingesetzt werden, wenn die Rechner von unterschiedlichen Betriebs- oder Nutzungsorganisationen oder von verschiedenen Systemarchitekturen (IBM, DEC usw.) abhängen.

Das OSI-Konzept erlaubt unterschiedliche lokale Entwicklungen jedes dieser Systeme (neue Maschinen, neue Betriebssysteme usw.), ohne dass die OSI-Öffnung zur Umgebung gestört wird (Fig. 14).

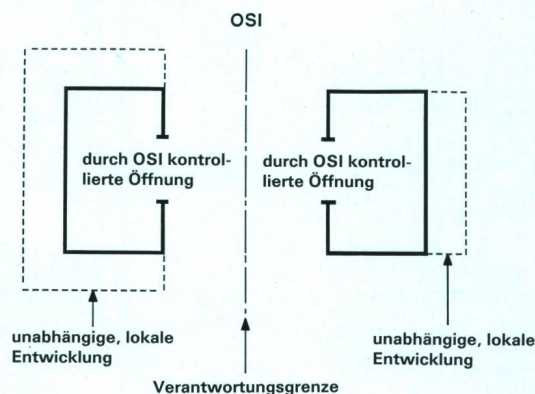


Fig. 14  
Verantwortungsgrenze und OSI

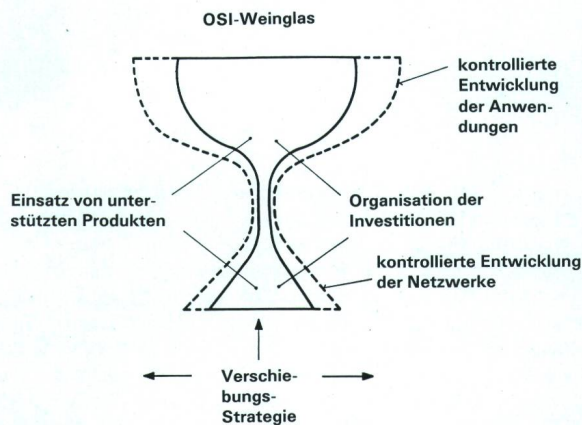


Fig. 15  
Kontrollierte Entwicklungen und OSI

#### *Kontrollierte Entwicklung der Verantwortlichkeitsgrenzen*

Die OSI-Architektur erlaubt eine kontrollierte Veränderung der Verantwortlichkeitsgrenze, unter Berücksichtigung der fortlaufenden Betriebskompatibilität. Es können Protokollsätze entfernt oder beigefügt werden, sei es in der Anwendung oder in den Transportdiensten. Dabei wird durch das Hinzufügen einer Anwendung (oben im OSI-Weinglas) die Basis nicht zerstört (Fuss des OSI-Weinglases) und umgekehrt (Fig. 15).

#### *Optimierung der Investitionen*

Die OSI-Ebenen, ausser der Schicht 1 und einem Teil der Schicht 2, sind alle in Software verwirklicht. Der Gebrauch der OSI-Architektur dient also dazu, die Entwicklungskosten und vor allem den Betrieb und den Unterhalt dieser Kommunikationssoftware durch eine kontrollierte Veränderung der Softwareschichten zu optimieren (Fig. 15).

#### *Einsatz vorhandener Produkte*

Eine Strategie in der Benutzung der OSI-Architektur ist der Einsatz von Katalogprodukten professioneller Hersteller (Computerhersteller, grosse Softwarehäuser) zum Aufbau der OSI-Säulen (Fig. 15). Diese Produkte

haben zwar einen Rückstand von zwei bis vier Jahren auf den Stand der Normen, aber die Garantie der Entwicklungskontinuität bis hin zu den künftigen Normen ist ein Sicherheitsaspekt in der Auswahl der Investitionen, dem man Rechnung tragen muss. Dies trifft um so mehr zu, weil gerade erst definierte Normen noch eine Anzahl kleinerer Mängel haben, die nur durch mehrere unterschiedliche und getestete Einführungen ausgemerzt werden können. Dieser Prozess beansprucht mehrere Jahre.

#### *Veränderungen im Griff*

Die Verantwortlichen von Telekommunikationsprojekten sollten schon heute intuitiv verspüren, dass die Zeit reif ist, von den Herstellern den Einsatz der OSI-Normen zu fordern, um sein Rechenzentrum mit der Aussenwelt zu verbinden. Sie müssen aber auch die Dynamik dieser Entwicklungen mit den dazugehörigen Softwareprodukten verstehen und zusammen mit ihrem Hersteller einen *Entwicklungs- und Änderungsplan* (Fig. 15) aufstellen. Die CCITT-Studienperioden dauern vier Jahre und könnten ein Mass für die Evolution und Entwicklung dieser offenen Tür zur Welt sein.

#### *Verdankungen*

Der Autor möchte seinen Kollegen vom CCITT danken, die ihm geholfen haben, in diese schwierige Materie einzudringen, und die ihm erlaubt haben, etwas von der Substanz dieser Normen zu verstehen. Besonders zu erwähnen sind die Herren: *J. B. Steffani* (Frankreich), *Peter Ingram* (UK), *Jim White* (USA), *Douglas Steedmann* (Kanada) und ganz besonders *Pietro Schicker* (CH).

Dank auch dem unbekanntem Übermittlungssoldaten und besonders dem unbekanntem Offizier, die uns erlaubt haben, diese komplexen Prozesse besser zu verstehen.

Dank ebenfalls meiner Kollegin, Frau *Andersen*, die sich die Mühe genommen hat, meinen ursprünglichen Text der beiden Artikel zu entziffern, zu verbessern und ins reine zu schreiben.

#### **Bibliographie**

- [1] *Pitteloud J.* Der Fuss des OSI-Weinglases. Techn. Mitt. PTT, Bern 67 (1989) 2, S. 42.