

HET GEBRUIK VAN MESSAGE HANDLING SYSTEM X.400 VOOR TELEGRAPHICS

J.M. Van der Meer

Delft Augustus 1988,
Faculteit der Elektrotechniek, TU Delft

VOORWOORD

Dit verslag is tot stand gekomen in het kader van een afstudeeronderzoek, dat verricht is aan de vakgroep Telecommunicatie- en Verkeersbegeleidingssystemen van de Faculteit der Elektrotechniek van de Technische Universiteit in Delft.

Mijn dank gaat uit naar ir. J.A.M. Nijhof voor de begeleiding van dit onderzoek. Ook dank ik J.H. Bons voor zijn uitleg van het Telegraphics systeem en een demonstratie daarvan.

VOORWOORD	
SUMMARY	
SAMENVATTING	
1 INLEIDING	1
2 TELEGRAPHICS	2
2.1 Ontstaan van telegraphics	2
2.2 Coderingsmethode voor telegraphics	3
2.3 Toepassing bij Message Handling Systemen	5
3 HET MESSAGE HANDLING SYSTEM MODEL X.400	6
3.1 Het ontstaan van Message Handling systemen	6
3.2 Het Message Handling Model.	7
3.3 Beschrijving en representatie van documenten	12
4 DOCUMENT STRUCTUUR VAN MULTIMEDIA DOCUMENTEN	
IN BESTAANDE SYSTEMEN	18
4.1 Agora	19
4.2 De DARPA systemen	20
4.2.1 Het Experimentele model	20
4.2.2 Het DARPA Internet model	21
4.2.3 Diamond	23
4.3 MINOS	23
4.4 Het NEC systeem	25
4.5 Office Document Architecture	26
4.6 Keuze van een model	27
5 OFFICE DOCUMENT ARCHITECTURE	28
5.1 De basisprincipes van ODA	28
5.2 Generieke structuur	29
5.3 Attributen	30
5.4 Vormen van een document	31
5.5 Codering van een document	31
6 ARCHITECTUUR VAN EEN TELEGRAPHICS MULTIMEDIA DOCUMENT	32
6.1 Overzenden van telegraphics bij verschillende resoluties	32
6.2 Protocol voor ontvangst van Telegraphics document	34
6.3 Attributen voor Telegraphics	37
6.4 Content Architecture voor Telegraphics	38
6.4.1 Positioneren en afbeelden van een blok.	39
6.4.2 Het layout proces.	40
6.4.3 Attributen	40
6.5 Voorbeeld van een Telegraphics document	41
REFERENTIES	52
APPENDIX A	55

SUMMARY

In the Telecommunications and Traffic Control Systems Group in the department of Electrical Engineering of Delft University of Technology the Telegraphics system has been developed. With this system messages containing graphics can be transmitted over a communication network. Graphics for such a message are made by using a writing tablet. It is possible to mix text with graphics. These messages can be sent over a communication network, in which connected terminals haven the same resolution.

It is possible to transmit telegraphics documents using a Message Handling System, which is made according to the CCITT standard X.400. To make it possible to transmit documents between systems, that use terminals with different resolutions, it is necessary to contain the resolution of the sender in the document.

Many systems, that are capable of transmitting multimedia documents, use a logical structuring for the documents. This is done to be independent of the presentation device.

To structure Telegraphics documents the standard Office Document Architecture (ODA) of the ECMA, ISO and CCITT can be applied. Using this standard, documents can be coded with the rules of ASN 1, which is the same as the coding of X.400.

SAMENVATTING

Aan de vakgroep Telecommunicatie- en Verkeersbegeleidingsystemen is het Telegraphics systeem ontwikkeld. Hiermee kunnen boodschappen met graphics worden verzonden. De graphics worden met behulp van een 'writing tablet' gemaakt en het is mogelijk om de graphics door de tekst heen te zetten. De aldus gemaakte boodschappen kunnen verzonden worden over een netwerk naar terminals met gelijke resolutie.

Het is mogelijk om telegraphics boodschappen te verzenden over een Message Handling System dat voldoet aan de norm X.400. Om het mogelijk te maken om boodschappen te versturen tussen systemen met verschillende resoluties, is het nodig dat in de boodschap ook de resolutie van de zender wordt gezet.

Veel multimedia document systemen blijken een logische structuur aan de documenten te kunnen aanbrengen. Dit is nodig om niet afhankelijk van het presentatie medium te zijn.

Voor het structureren van Telegraphics documenten kan gebruik worden gemaakt van de norm Office Document Architecture (ODA) van de ECMA, ISO en CCITT. Tevens kunnen aan de hand hiervan documenten worden gecodeerd volgens ASN.1, dat gelijk is aan de codering van X.400.

1 INLEIDING

Aan de vakgroep telecommunicatie en verkeersbegeleidingssystemen is het Telegraphics systeem ontwikkeld, waarmee PC's met een writing tablet berichten kunnen verzenden met grafische informatie. Deze grafische informatie is met behulp van de writing tablet gemaakt en kan bestaan uit aparte figuren maar ook uit markeringen van de tekst bij voorbeeld om verbeteringen in teksten aan te geven. Dit systeem is zodanig ontwikkeld dat de grafische informatie op een zeer efficiënte manier in ASCII code wordt gerepresenteerd. Op deze manier kunnen berichten van tekst en figuren als een ASCII tekst verzonden worden over een netwerk dat boodschappen kan verzenden die bestaan uit tekst.

Er zijn tegenwoordig systemen ontwikkeld die zowel tekst als figuren als geluid in één bericht kunnen verzenden. Ook zijn er al standaarden hiervoor. In dit verslag zal een protocol worden beschreven waarin het Telegraphics systeem geïntegreerd wordt in het Message Handling systeem X.400 van de CCITT. Daarbij zal aan de orde komen hoe in een dergelijk Message Handling Systeem tekst en figuren gecombineerd kunnen worden tot zogenaamde multi media documenten. De standaard voor multimedia documenten, ODA, van de ISO, ECMA en CCITT zal daarbij gebruikt worden.

In hoofdstuk 2 zal een beschrijving worden gegeven van het Telegraphics systeem. In hoofdstuk 3 zal een beschrijving worden gegeven van het X.400 Message Handling systeem. In hoofdstuk 4 zal de document structuur van verschillende systemen worden uiteengezet, die in de literatuur zijn beschreven. Hoofdstuk 5 zal over de standaard ODA gaan. In hoofdstuk 6 zal uiteengezet worden wat er gebeurt als Telegraphics boodschappen verzonden worden tussen niet-homogene systemen. Verder zal worden beschreven hoe boodschappen met Telegraphics opgebouwd worden en hoe ze worden weergegeven. Hiertoe zal een toevoeging aan ODA worden gegeven.

2 TELEGRAPHICS

Telegraphics is een systeem waarbij met behulp van een PC teksten samen met figuren worden verzonden over een netwerk [1,30]. De figuren worden met behulp van een writing tablet gemaakt (een writing tablet is een vlak dat de bewegingen van een pen aftast en de coördinaten van de pen doorgeeft aan de computer). Figuren kunnen zowel apart van de tekst gemaakt worden, bij voorbeeld een handtekening onder een brief, als ook door de tekst heen, bij voorbeeld om iets te markeren of om fouten aan te geven alsmede verbeteringen van die fouten. De figuren worden net als de tekst in ASCII formaat gecodeerd en zo verzonden.

Dit hoofdstuk zal een beschrijving geven van telegraphics. Daarbij wordt allereerst de ontstaansgeschiedenis van telegraphics beschreven en vervolgens de coderingstechnieken en tot slot hoe dit in een message handling systeem gebruikt wordt.

2.1 Ontstaan van telegraphics

In het begin van de jaren 70 is het 'tele writing' ontwikkeld. Allereerst werd het elektronische schoolbord ontwikkeld. Dit is een audiovisueel educatief hulpmiddel, dat grafische informatie samen met geluid op een TV toestel weergeeft. Hierbij werd de informatie ingevoerd met een pen op een writing tablet.

Uit de ontwikkeling van het elektronische schoolbord bleek dat het verzenden, via een kabel of door uitzending, van handgeschreven tekst in 'real time' uit te voeren was. Dit gegeven leidde tot de ontwikkeling van de scribofoon. Een scribofoon is een telefoon waarbij naast het normale gebruik van de telefoon ook nog geschreven informatie doorgegeven kan worden. Hierbij wordt weer gebruik gemaakt van een writing tablet. De writing tablet genereert een digitale code waarin bewegingen van de pen worden onthouden. Dit is de zogenaamde dynamographische methode; dit staat tegenover de 'scanning digitizer' waarin steeds een sample wordt genomen van het hele plaatje, dat dan als een raster wordt onthouden. De digitale code wordt dan samen met de spraak over een analoog telefoon kanaal verzonden. Samen met een buffer zorgt de dynamografische methode voor een zeer efficiënte codering en kan het zenden gebeuren met een snelheid van 200 bit per seconde.

In de tachtiger jaren begon de internationale ontwikkeling van het 'real time' verzenden van graphics. De Europese PTT-organisatie CEPT begon een project waarbij gericht wordt naar het gebruik van digitale ISDN-kanalen. Verder begon er een ESPRIT project van de EEG, Multipoint Interactive Audiovisual Communications. Hierbij wordt gezocht naar standaardisatie op het gebied van audio conferencing met visuele grafische hulpmiddelen. In Japan heeft NTT de sketchphone op de markt gebracht. Dit is een voice/graphics terminal die vergelijkbaar is met de scribofoon. Het wordt gebruikt om kaarten door te zenden naar vrachtwagenchauffeurs, zodat die de weg makkelijk kunnen vinden.

The first part of the report deals with the general situation in the country. It is a very interesting and detailed account of the conditions prevailing at the time. The author describes the political and social situation, the state of the economy, and the general mood of the population. He also mentions the various attempts to reform the government and the role of the different political parties.

THE ECONOMIC SITUATION

The economic situation is described in great detail. The author discusses the state of the agriculture, the industry, and the commerce. He mentions the various measures taken to improve the economy and the results of these measures. He also discusses the role of the different economic sectors and the contribution of each to the national income. The author's analysis is very thorough and his conclusions are well supported by the facts.

Met de komst van interactieve videotex, waarbij een centrale computer als database fungeert, die op afstand geraadpleegd kan worden, verplaatsten de beperkingen door een beperkte transmissiecapaciteit naar beperkingen door beperkte opslagcapaciteit en verwerkingssnelheid van de computer. Per beeld was er slechts beperkte geheugenruimte aanwezig. In Prestel (videotex) bevat een beeld 24 rijen en 40 kolommen. Elke positie bevat een ASCII karakter of 2*3 dots. Door toevoegen van een writing tablet en een dynamografische codeerinrichting kan een resolutie worden bereikt van 512 * 512. Deze geometrische mode is toegevoegd aan het Nederlandse Viditel en wordt Vidibord genoemd. Hierbij kunnen lijntekeningen worden gecombineerd met ASCII tekst. Ingewikkelde figuren kunnen in meer frames worden opgeslagen.

Dit systeem staat niet open voor open gebruikers groepen door het beperkte nationale karakter en door speciaal benodigde decoders. Een internationaal netwerk met intelligente terminals die met behulp van software voor het coderen en decoderen van de lijntekeningen kunnen zorgen zou een veel breder toepassingsgebied mogelijk maken. Een IBM compatible PC voldoet prima als intelligente terminal en wordt gebruikt bij het telegraphics systeem. Een EGA compatible digitizer/writing tablet, die geschikt is voor dynamografische codering, wordt hierbij gebruikt om de tekeningen via het seriële interface de computer in te voeren. Deze configuratie kan dienen als een grafische terminal voor een Message Handling Syteem.

2.2 Coderingsmethode voor telegraphics

Er zijn verschillende manieren om een grafisch beeld te coderen. Hierin zijn twee manieren te onderscheiden : tijdcodering en ruimtecodering. Bij tijdcodering wordt op bepaalde momenten steeds een sample genomen. Bij ruimtecodering wordt alleen bij verandering van plaats een sample genomen.

Er zijn twee soorten tijd codering : area (raster) scanning en path tracing.

Bij area scanning wordt het beeldvlak verdeeld in punten (pixels). Elk punt wordt gerepresenteerd door een aantal bits in de codering, afhankelijk van het aantal mogelijke kleuren. Deze bits geven aan welke kleur een punt in het vlak heeft. Alle punten van het hele vlak worden op deze manier gecodeerd. Dit is een inefficiënte manier van coderen, want ook de punten die feitelijk geen informatie bevatten (0 zijn en gelijk aan de achtergrond) nemen geheugenruimte in. Het wordt toegepast bij standaard digitizers, facsimile en digitale TV.

Bij path tracing worden de actieve coördinaten (de plaats waar de pen is) van het writing tablet doorgegeven. Dit kan een flinke ruimte besparing opleveren omdat alleen de punten die informatie bevatten worden gecodeerd. Maar : Bij stilstaande of langzaam bewegende pen zullen steeds de zelfde coördinaten worden gecodeerd. Dit levert weer een hoop overbodige informatie. Alleen als hiertegen maatregelen worden genomen blijft het coderen efficiënt.

Bij ruimtecodering wordt alleen bij het verplaatsen van de pen gecodeerd. Dit gebeurt door de verplaatsing als een vector weer te geven. De codering kan nog efficiënter door gebruik te maken van differential chain coding. Dit is een codering waarbij niet steeds de vector zelf wordt weergegeven maar het verschil met de vorige vector. De codering is zeer zuinig met geheugen en kan dus gebruikt worden bij toepassingen waar niet veel geheugen beschikbaar is.

Door bij het verzenden van een string van verschilvectoren gebruik te maken van de Huffman codering ontstaat een codering die volgens de informatietheorie dicht bij het optimum ligt. Bij Huffman codering worden de waarden gerangschikt op kans van voorkomen. De waarde met grootste kans van voorkomen wordt als kortste gecodeerd. Zo hebben veel voorkomende waarden een kortere code dan weinig voorkomende waarden. Deze methode heeft een aantal nadelen :

- de kansverdeling van het voorkomen van vectoren moet bekend zijn.
- doordat de woordlengte wisselend is kan het decoderen makkelijk fout gaan door transmissiefouten.
- omdat fouten relatief zijn van een vorige vector zal een fout zich voortplanten.
- de scheiding van woorden hoeft niet op de bytegrens te liggen. Dit geeft problemen bij de codering en decodering.

Deze nadelen worden gedeeltelijk ondervangen als gebruik gemaakt wordt van string incremental coding, hetgeen bij telegraphics is gedaan. Het string incremental coderen gaat als volgt : Eerst wordt een selectie frame vastgesteld. Dit selectie frame geeft aan welke verplaatsingen (vectoren) er mogelijk zijn. Neem als voorbeeld een selectie frame van 16 incrementele vectoren :

7	6	5	4	3
8				2
9		P		1
10				16
11	12	13	14	15

Figuur 2.1. Selectie frame, P is het huidige punt, één van de nummers geeft het volgende punt aan.

Allereerst gaat de pen naar beneden. De coördinaten van de oorsprong worden gecodeerd : X_0 , Y_0 . Dan wordt als de pen beweegt de eerste vector increment gecodeerd. Dit is de best passende van de 16 vectoren van het selectie frame. De eerste vector is X_1 , Y_1 , dit wordt een absolute vector genoemd. Het selectie frame wordt nu verplaatst naar het volgende punt. Voor de volgende vector increment wordt nu de afwijking met schrijfrichting (dat is de vorige vector) berekend. Als de afwijking -1, 0 of 1 is dan wordt de volgende vector X_2 , Y_2 gecodeerd als één van de drie relatieve vectoren -, 0, +, anders wordt de vector gecodeerd als één van de overblijvende absolute vectoren.

Een opeenvolging van 3 relatieve vectoren wordt gegroepeerd tot een string. Er zijn 27 complete strings en 12 afgebroken strings. Een string wordt afgebroken omdat er geen drie relatieve vectoren elkaar opvolgen, ofwel omdat de afwijking te groot is, ofwel omdat de pen weer wordt opgetild. Er zijn M absolute vector increments (in het voorbeeld is M=16). Er zijn 6 bits beschikbaar, dus 64 codewoorden. In dit formaat kunnen dus $M \leq 25$ vector increments gebruikt worden. Op deze manier kan elk traject vanaf pen naar beneden tot pen omhoog gecodeerd worden als een opeenvolging van absolute vector increments en strings. Elk element hiervan past in een byte zodat de ASCII code gebruikt kan worden.

Deze codering is zeer efficiënt; dat blijkt uit de verdelingsfunctie van de verschillen tussen opvolgende vectoren. Afwijkingen van -1, 0, en 1 komen hierin het meest voor en worden op de korte manier gecodeerd.

2.3 Toepassing bij Message Handling Systemen

Omdat de graphics als ASCII karakters gecodeerd kunnen worden, kunnen computers telegraphics verzenden over een Message Handling Systeem dat ontworpen is om alleen tekst over te zenden. Dit is gedaan met het Memocom netwerk van de PTT. De graphics staan tussen de tekst. Een regel graphics wordt voorafgegaan door een speciale code.

Bij het verzenden werden alleen computers gebruikt die de zelfde grafische kaart hadden : een EGA kaart. Er zullen problemen ontstaan als hiervoor verschillende kaarten worden gebruikt. Dit wordt in hoofdstuk 6 besproken.

1. The first part of the paper is devoted to a general discussion of the problem of the existence of solutions of the system of equations (1) for arbitrary values of the parameters α and β . It is shown that the system has solutions for all values of the parameters α and β if and only if the condition $\alpha + \beta > 0$ is satisfied. In the case when $\alpha + \beta < 0$, the system has no solutions.

2. THE CASE OF $\alpha + \beta = 0$

In the case when $\alpha + \beta = 0$, the system of equations (1) can be written in the form

$$\begin{cases} x'' + \alpha x = 0 \\ y'' + \beta y = 0 \end{cases} \quad (2)$$

It is easy to see that the general solution of the system (2) is

$$\begin{cases} x = C_1 \cos \sqrt{\alpha} x + C_2 \sin \sqrt{\alpha} x \\ y = C_3 \cos \sqrt{\beta} x + C_4 \sin \sqrt{\beta} x \end{cases} \quad (3)$$

3 HET MESSAGE HANDLING SYSTEM MODEL X.400

In dit hoofdstuk wordt een beschrijving gegeven van het Message Handling System model X.400 van de CCITT. In de eerste paragraaf wordt aangegeven hoe message handling systemen zijn ontstaan. In de tweede paragraaf wordt een beschrijving van het model gegeven. Tot slot wordt in de laatste paragraaf uitvoerig uiteengezet hoe documenten in X.400 beschreven en gecodeerd moeten worden, zodat in een later hoofdstuk hiervan gebruik gemaakt kan worden om Telegraphics te beschrijven en te coderen.

Uitgebreide beschrijvingen van het Message Handling System Model zijn te vinden in twee boeken van Peter Vervest [2,3] en natuurlijk in de CCITT recommendations van X.400 [4].

3.1 Het ontstaan van Message Handling systemen.

Met de komst van computers ontstond ook de behoefte tot communicatie tussen computers. Allereerst moesten grote mainframes ook op grotere afstanden bereikbaar zijn met behulp van user terminals. Hiervoor werden vaste verbindingen gebruikt. Daarna ontstond ook de behoefte om data uit te wisselen tussen de computers zelf. Met de groei van het aantal computers en de groei van de behoefte tot communicatie werd het openbare telefoonnet gebruikt om in deze behoefte te voorzien.

Het telefoonnet bleek echter niet erg geschikt voor de communicatie tussen computers. Het opzetten van de verbinding duurt relatief lang ten opzichte de maar korte verbindingsduur. Verder kon de data niet met hoge snelheden worden verzonden.

Om sneller data transport mogelijk te maken ontstonden er datanetten. Eerst werd hiervoor circuit switching gebruikt. Bij circuit switching wordt eerst de verbinding tussen twee computers tot stand gebracht, net zoals bij het telefoonnet, waarna de communicatie over deze verbinding kan plaats vinden. Het tot stand brengen van de verbinding duurt ook hier relatief lang. Verder wordt een verbinding die langere tijd blijft bestaan een groot deel van de tijd ongebruikt gelaten omdat het dataverkeer tussen computers een burst karakter heeft. Het datanet wordt op deze manier dus niet efficiënt gebruikt.

Om het netwerk efficiënter te maken werden message switching en packet switching toegepast. Bij message switching wordt een hele boodschap op het netwerk gezet. Vervolgens zorgt het netwerk dat de boodschap op de bestemming aankomt. Bij packet switching wordt een boodschap verdeeld in pakketten en deze pakketten worden afzonderlijk over het netwerk verzonden. Er wordt niet van te voren een verbinding opgebouwd, onderweg wordt steeds bepaald over welke lijn een pakket wordt verzonden. Alle lijnen kunnen nu gelijktijdig voor verschillende gesprekken worden gebruikt.

Netwerken gingen ook gebruikt worden om boodschappen van gebruikers te versturen. Er ontstonden 'electronic mailboxes'. De verzonden berichten worden hierin opgeslagen, en de ontvanger kan wanneer hij dat wil, het bericht uit de mailbox halen. Als eerste experimenteerde ARPA hiermee met het ARPAnet. Het was het eerste netwerk waarbij boodschappen met behulp van computers werden verzonden. Dit wordt Computer Based Message System (CBMS) genoemd. Dit systeem was ontworpen om korte boodschappen die tekst bevatten over te zenden. Het was denkkelijk om hiermee ook langere boodschappen, die ook stem of samengestelde documenten kunnen bevatten, te verzenden.

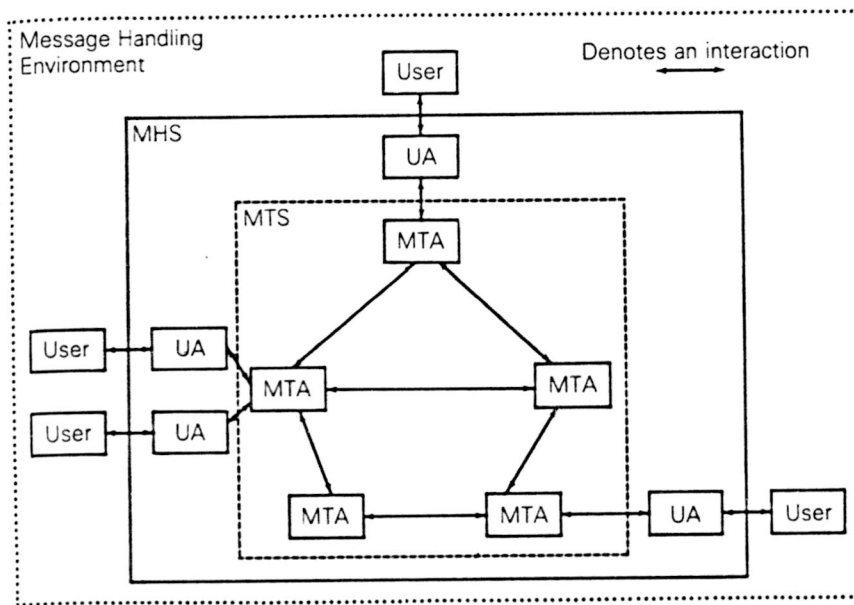
Met de groei van het aantal computers groeide ook de behoefte om te communiceren. Er ontstonden verschillende lokale netwerken. Deze werden echter op verschillende manieren ontwikkeld. Het verbinden van verschillende netwerken werd hierdoor steeds ingewikkelder. Hierom heeft de ISO een referentie model ontwikkeld om toekomstige standaarden aan te laten voldoen : Het OSI-model (Open Systems Interconnection).

De PTT's die veel communicatie systemen bezitten zagen door incompatibiliteit tussen systemen, dat de communicatie werd geremd. In hun belang was het dan ook om dit verbeteren. De overkoepelende organisatie, de CCITT, zorgde dan ook voor een aantal standaarden op communicatie gebied. In 1984 werd een MHS model gemaakt dat een aanvulling was op het OSI model, X.400. Het model definieert een methode voor computersystemen om berichten uit te wisselen, waarbij de inhoud van de berichten van geen belang is voor het communicatieproces.

3.2 Het Message Handling Model.

Alvorens het Message Handling model van de CCITT uiteen te zetten, wordt hier eerst een definitie gegeven van electronic mail. Electronic mail is, volgens de definitie van Peter Vervest [2,3] :
het in één richting elektronisch verzenden van informatie in de vorm van een boodschap via een tussenliggend (tele-)communicatie systeem, van een geïdentificeerde verzender naar één of meer geïdentificeerde ontvangers.

Een message handling systeem zal electronic mail, zoals hierboven gedefinieerd, moeten kunnen verwerken. De CCITT heeft een model ontwikkeld voor een Message Handling Systeem. De volgende figuur geeft de funktionele relaties van het model weer.



Figuur 3.1 Funktionele relaties van het MHS model [uit [2] figuur 7.2]

Er kunnen verschillende onderdelen onderscheiden worden in deze figuur :

Allereerst de User. Dit is een persoon die achter de terminal zit. Maar het kan ook een applicatie programma zijn, dat boodschappen verzendt. De User kan zowel verzender als ontvanger van een boodschap zijn.

Vervolgens de User Agent (UA). De User gebruikt de User Agent om de boodschappen voor te bereiden. Via de User Agent wordt de boodschap op het netwerk gezet.

De User Agent geeft het bericht af aan het Message Transfer System. Het Message Transfer systeem zorgt ervoor dat boodschappen bij de User Agent van de ontvanger terecht komen. Het Message Transfer System bestaat uit een aantal Message Transfer Agents (MTA). Deze zorgen voor de overdracht van de boodschappen van en naar User Agents en ook tussen Message Transfer Agents onderling.

De User Agents vormen samen met de Message Transfer Agents het Message Handling System. De User Agents en Message Transfer Agents kunnen twee soorten interactie met elkaar hebben :

-1 Aflever interactie :

De User Agent van de zender geeft de inhoud van de boodschap samen met de enveloppe af aan de Message Transfer Agent. De enveloppe bevat informatie, die het Message Transfer System nodig heeft om bepaalde services uit te voeren en om de boodschap af te leveren.

-2 Bezorg interactie :

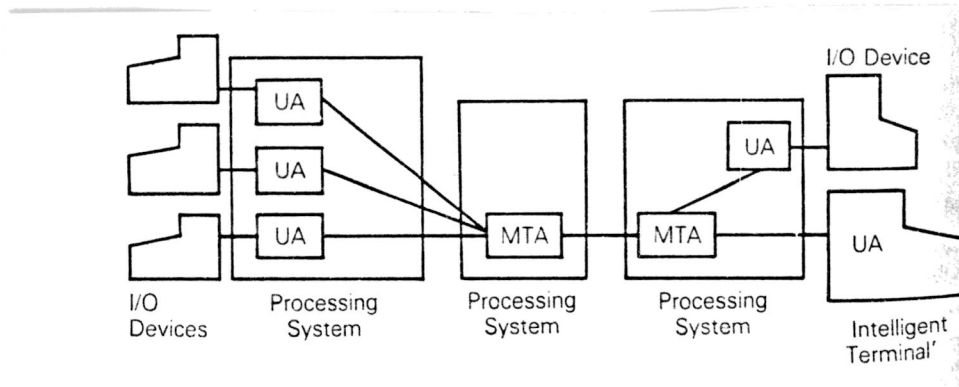
De Message Transfer Agent bezorgt de boodschap samen met de enveloppe aan de User Agent. De enveloppe bevat informatie die betrekking heeft op het bezorgen.

De fysieke uitvoering van een Message Handling Systeem.

De User Agent en de Message Transfer Agent kunnen op verschillende manieren worden geïmplementeerd.

- 1 De User Agent en de Message Transfer Agent zitten in één geheel.
- 2 De User Agent en Message Transfer Agent zitten elk in een apart systeem.
- 3 De User Agent kan ook nog, bij een intelligente terminal, in de terminal zelf worden geplaatst.

In figuur 3.2 worden de verschillende uitvoeringen weergegeven.

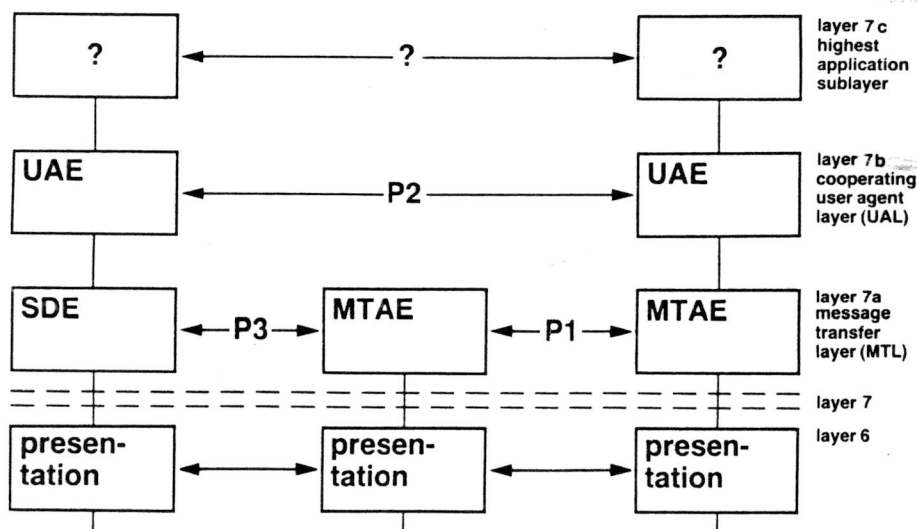


Figuur 3.2 Verschillende implementaties van de functies van het MHS model [uit [2] figuur 7.3]

De verschillende lagen in het MHS model.

Het X.400 MHS model is opgezet volgens het OSI lagen model. Het MHS model bevindt zich in de applicatielaag en bestaat zelf uit twee lagen :

- 1 De User Agent laag
- 2 De Message Transfer laag



Figuur 3.3 Message Handling protocol structuur in het lagen model [uit [3] figuur A.3]

De lagen zijn in drie functionele eenheden te verdelen (zie figuur 3.3). In de User Agent laag :

- a De User Agent Entity (UAE). De User Agent Entity houdt zich bezig met de representatie van de inhoud van de boodschap. Ook zorgt de User Agent Entity ervoor dat de zich boven hem bevindende gebruiker de functies van het Message Transfer systeem kan gebruiken. Het P2 protocol definieert de boodschappen service tussen twee User Agent Entities.

In de Message Transfer laag:

- b De Message Transfer Agent Entity (MTAE). De Message Transfer Agent Entity verzorgt de services van de Message Transfer laag samen met andere Message Transfer Agent Entities. Het P1 protocol definieert de interacties tussen twee Message Transfer Agent Entities.
- c De Submission and Delivery Entity (SDE). De Submission and Delivery Entity maakt de services van de Message Transfer laag beschikbaar voor een stand-alone User Agent. Het P3 protocol zorgt voor de interactie tussen Submission and Delivery Entity en Message Transfer Agent.

Namen en adressen

X.400 onderscheid twee soorten namen :

- 1 Primitief - Elke naam is uniek. Hij wordt uitgegeven door een zogenaamde 'Naming Authority'.
- 2 Beschrijvend - Meestal wordt een functie van een persoon beschreven. Bij voorbeeld : Vice-president, marketing, ABC- company.

Primitieve namen kunnen samen een beschrijvende naam vormen.

X.400 gebruikt het begrip O/R naam (Originator/Recipient) als aanvulling op boven genoemde namen, omdat de bovengenoemde namen meestal in een beperkt gebied betekenis hebben. De O/R namen kunnen de volgende attributen bevatten :

- personal name
- country name
- organization name
- organization unit name
- adress according to CCITT Recommendations X.121
- unique User Agent identifier
- administration management domain name
- private management domain name

Een O/R naam kan twee vormen aannemen, een vorm gebaseerd op architectuur of gebaseerd op terminal.

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

... ..

De eerste vorm bestaat uit een country name en een administration management domain name, verder worden er nog meer attributen gegeven. Dit kan op drie verschillende manieren:

- 1 Een aantal van de volgende attributen wordt toegevoegd :
 - private management domain name
 - personal name
 - organization name
 - organization unit name
 - domain defined attributes
- 2 Alleen het attribuut unique User Agent identifier wordt toegevoegd. Het is mogelijk om hier nog domain defined attributen bij te doen.
- 3 Alleen het X.121 adres wordt toegevoegd. Ook hier kunnen domain defined attributen bij gedaan worden.

De tweede vorm kan gebruikt worden om teletex terminals te adresseren. Hiervoor wordt het (internationale) telematic terminal nummer toegevoegd. Als optie kan hier nog het telematic terminal identifier toegevoegd worden.

Services

Een Message Handling System zal bepaalde diensten moeten kunnen leveren. Er worden twee soorten onderscheiden :

- 1 Message Transfer Service
- 2 Interpersonal Messaging Service

Message Transfer Service

De basis service, die een Message Handling System levert is, dat alle typen User Agent toegang hebben om boodschappen te verzenden of benaderd kunnen worden om boodschappen te ontvangen. Een andere service is : De zender wordt op de hoogte gebracht als de boodschap niet kan worden afgeleverd. Een User Agent kan aangeven welke typen berichten hij kan verwerken. In het bericht worden verder aangegeven: type inhoud, original encoded information types, time of submission and delivery, conversion occurred.

De service elementen kunnen op twee manieren worden onderscheiden : Allereerst wordt onderscheid gemaakt in basis, essential optional en in additional optional service elements. Het tweede onderscheid wordt gemaakt bij de optionele service elementen : Per Message (PM) of Contractual Period of Time (CPT).

De basis service elementen zijn inherent aan het message transfer systeem. De optionele essentiële services zijn altijd beschikbaar voor gebruikers van de internationale message transfer service. De toegevoegde optionele services kunnen door de beheerder worden toegevoegd. Optionele services kunnen ofwel per boodschap worden beschikbaar gesteld danwel voor een bepaalde contract periode.

Interpersonal Messaging Service

De eerste klasse van de user agent laag service in X.400 is de Interpersonal Messaging Service. Het doel van de service is het mogelijk maken dat gebruikers Interpersonal boodschappen verzenden naar één of meer ontvangers. Deze ontvangers zijn ofwel met een User Agent verbonden danwel met telex of telematic diensten. Hiervoor maakt de Interpersonal Messaging service gebruik van message transfer service elementen.

Op een zelfde manier als bij de message transfer service elementen wordt een onderscheid gemaakt tussen basic, essential optional en additional optional service elementen. Er wordt nu echter ook onderscheid gemaakt of de service aan ontvangende danwel zendende kant moet worden geleverd. Dit is nodig omdat bepaalde services, die additional zijn voor de zender, essential zijn voor de ontvanger. Als de zender gebruik maakt van een bepaalde additionele dienst, die informatie van de ontvanger nodig heeft, dan moet hij er ook zeker van zijn dat deze service ook aan ontvangende zijde wordt geboden. Een voorbeeld is Expiring Date indication. Als de ontvanger na een bepaalde datum de boodschap nog niet heeft bekeken dan moet de zender hiervan op de hoogte worden gebracht.

3.3 Beschrijving en representatie van documenten.

Voor dit onderzoek is de beschrijving en codering van boodschappen van belang, want bekeken moet worden of en hoe binnen X.400 telegraphics gecodeerd kan worden. Daarom zal in deze paragraaf uitvoerig worden uiteengezet op welke manier een document wordt beschreven en hoe dit gecodeerd wordt. In het hoofdstuk 6 zal hiervan dan gebruik worden gemaakt om een telegraphics document te beschrijven en coderen.

Voor de beschrijving en representatie heeft de CCITT een aparte Recommendation gemaakt, X.409: 'Message Handling Systems' presentation transfer syntax and notation. Van deze standaard is afgeleid 'Abstract Syntax Notation 1' (ASN 1), die door de ISO is vastgelegd in de normen ISO 8824 en 8825. X.409 en ASN 1 komen op de wezenlijke onderdelen met elkaar overeen [5]. ISO 8613 (Office Document Architecture), dat in een later hoofdstuk zal worden besproken, gebruikt deze standaard.

X.409

Een document bestaat uit een aantal elementen (ook wel data element genoemd). Een element is bij voorbeeld een hoofdstuk; deze is weer samengesteld uit kleinere elementen, paragrafen. Elk element wordt weergegeven door drie onderdelen : identifier, length en content (in deze volgorde).

De identifier geeft de functie van het element weer en ook het type van de inhoud. De lengte geeft aan uit hoeveel bytes de inhoud bestaat. De inhoud bevat de informatie van het element. Als een element uit meerdere elementen is samengesteld dan bevat de inhoud deze elementen.

Identifier

Er worden vier klassen identifiers onderscheiden :

- 1 Universal type : Deze zijn algemeen te gebruiken en onafhankelijk van de toepassing. Voorbeelden zijn : Boolean, Integer, Sequence en Set. Deze zijn samen met nog een aantal andere reeds gedefinieerd in X.409.
- 2 Application-wide type : Deze zijn alleen door een bepaalde toepassing te gebruiken. Elke toepassing kan hiermee zijn eigen typen definieren.
- 3 Context-specific type : Deze hebben alleen betrekking op een nog beperktere omgeving, bij voorbeeld een Set. Het type is dan alleen binnen de Set gedefinieerd.
- 4 Private type : Deze moeten zelf worden gedefinieerd en zijn alleen voor privé gebruik bedoeld.

Het gebruik van de verschillende klassen identifiers zal aan het eind van de paragraaf aan de hand van een voorbeeld worden uiteengezet. Eerst wordt echter aangegeven hoe een data element gecodeerd moet worden.

Codering

Zoals reeds is vermeld bestaat een data element uit drie onderdelen : identifier, length en content. De codering gebeurt in octets (bytes).

Een identifier wordt als volgt gecodeerd :

de twee hoogstwaardige bits geven de klasse aan. Dit gebeurt volgens onderstaande tabel :

Class	Bit assignment (Bit 8,7)
Universal	00
Application-wide	01
Context-specific	10
Private use	11

Tabel 3.1

Het volgende bit (bit 6) geeft aan of het een samengesteld element is of een primitief element.

Form	Bit assignment (Bit 6)
Primitive	0
Constructor	1

Tabel 3.2

Er blijven dan vijf bits over van het eerste octet. Deze geven een ID-code waarmee de verschillende typen van een klasse worden onderscheiden. De typen van de universele klasse zijn reeds gedefinieerd door X.409. Van deze typen staan de ID-codes in tabel 3.3.

ID-Code	Data Type
0	End of contents
1	Boolean
2	Integer
3	Bit String
4	Octet String
5	Null
6-15	Unassigned
16	Sequence
17	Set
18	Numeric String
19	Printable String
20	T.61 String
21	Videotex String
22	IA5String
23	UTC Time
24	Generalized Time

Tabel 3.3

In de vijf bits kunnen 31 verschillende ID-codes worden gecodeerd. Als er meer typen zijn dan kan er gebruik worden gemaakt van één of meer extension octets. In dat geval hebben de laatste vijf bits de waarde 11111. Bit 8 van elk extension octet geeft aan of het al dan niet het laatste octet is. Dit bit heeft de waarde 0 als het tot het laatste octet behoort. Bij het gebruik van extension octets vormen bits 7-1 van de extension octets de ID-code. Bit 7 van het eerste extension octet is dan het meest significante bit en bit 1 van het laatste octet het minst significante bit. Een ID-code moet in zo min mogelijk octets worden gecodeerd. Dus mogen er geen extension octets voorkomen met bit 7 tot 1 allemaal de waarde 0.

Tot slot wordt schematisch aangegeven hoe een identifier gecodeerd wordt. Elke letter stelt één bit voor. Een c staat voor klasse, een f voor vorm en een i voor ID-code.

```
ccfiiiiii            zonder extension octets
ccf11111 11111111 ... 01111111 met extension octets
```

De lengte L geeft het aantal octets van de inhoud van het element weer. Er zijn drie vormen van coderen mogelijk : kort, lang en ongedefinieerd.

De korte vorm bestaat uit één octet. Hij wordt gebruikt als de lengte kleiner is dan 128. Bit 8 heeft de waarde 0. Bits 7 tot 1 geven de lengte aan. Bij voorbeeld : L = 38 wordt gecodeerd als 00100110.

De lange vorm bestaat uit 2 tot 127 octets. Hij wordt gebruikt als de lengte groter of gelijk is aan 128 en kleiner dan 2^{1008} , tenzij de ongedefinieerde vorm wordt gebruikt. Bit 8 van het eerste octet heeft de waarde 1. Bits 7 tot 1 van het eerste octet geven aan hoeveel octets er verder nog gebruikt worden om de lengte aan te geven. Dit zijn dus het totale aantal octets om de lengte aan te geven - 1, het eerste octet. De resterende octets geven de lengte van de inhoud weer. Ook dit gebeurt in zo min mogelijk octets. Bij voorbeeld :
 $L = 201$ wordt gecodeerd als 10000001 11001001.

De ongedefinieerde vorm is één octet lang. Hij kan gebruikt worden als een element bestaat uit andere elementen (constructor), maar dit hoeft niet. De waarde is dan 10000000. Een speciaal element geeft het einde van de inhoud aan als deze vorm wordt gebruikt (EOC, End Of Contents).

De inhoud van het element bevat de werkelijke informatie. De codering is afhankelijk van het type. De lengte is variabel, maar is altijd een geheel aantal octetten.

Het definiëren van een type

Het is mogelijk om zelf typen te definiëren. Deze worden gevormd met de reeds door X.409 gedefinieerde typen (Zie tabel 3.3). De type definitie wordt volgens een standaard notatie weergegeven :

identifiser ::= Type

Op de plaats van de identifiser staat de naam van het type. De definitie van het type zelf wordt achter het '::=' teken gezet. Dit gebeurt met een zogenaamd 'Tagged Type'.

Een Tagged type bestaat uit een getal (numeric tag) gevolgd door het type. De tag moet verschillen van de tags, die aan andere typen van de zelfde klasse zijn toegekend. De woorden UNIVERSAL, APPLICATION of PRIVATE worden voor de tag gezet als het type respectievelijk van de universele, application of private klasse is. Staats er niets voor de tag dan is het type van de context-specific klasse. Een impliciete type definitie wordt voorafgegaan door het woord IMPLICIT. Staat er niets voor dan is het type expliciet.

Een voorbeeld van een type definitie :

```
Username ::= [APPLICATION 3] IA5String
      ▲           ▲           ▲           ▲
      identifieer  class      tag      type
```

Op deze manier heeft de ISO in haar norm voor Office Document Architecture (ISO 8613) verschillende typen gedefinieerd om Multi media documenten te kunnen samenstellen (zie Appendix A).

Voorbeeld van een beschrijving en de codering daarvan volgens X.409

Om aan te geven hoe het bovenstaande moet worden toegepast wordt er hier een voorbeeld gegeven. Dit voorbeeld is een verkorte versie van het voorbeeld dat in appendix II van X.409 wordt gegeven. Het voorbeeld geeft een beschrijving van persoonsgegevens.

THE
FEDERAL
BUREAU OF
INVESTIGATION
OF THE
DEPARTMENT OF JUSTICE
WASHINGTON, D. C.
20535

REPORT OF THE

INVESTIGATION OF THE
FEDERAL BUREAU OF
INVESTIGATION
OF THE
DEPARTMENT OF JUSTICE
WASHINGTON, D. C.
20535

REPORT OF THE
INVESTIGATION OF THE
FEDERAL BUREAU OF
INVESTIGATION
OF THE
DEPARTMENT OF JUSTICE
WASHINGTON, D. C.
20535

Allereerst wordt de structuur van de informatie informeel gegeven.
Ook de waarden zelf worden aangegeven :

```
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
```

Vervolgens wordt een formele beschrijving gegeven van de gebruikte typen zoals dat in de vorige paragraaf is uiteengezet :

```
PersonnelRecord ::= [APPLICATION 0] IMPLICIT Set {
    Name,
    title [0] IA5String,
    EmployeeNumber,
    dateOfHire [1] Date,
    nameOfSpouse [2] Name,
    numberOfChildren [3] Integer}
```

```
Name ::= [APPLICATION 1] IMPLICIT Sequence {
    givenName IA5String,
    initial IA5String,
    familyName IA5String}
```

```
EmployeeNumber ::= [APPLICATION 2] IMPLICIT Integer
```

```
Date ::= [APPLICATION 3] IMPLICIT IA5String --- YYYYMMDD
```

Er worden hier vier typen gedefinieerd voor deze toepassing :
PersonnelRecord, Name, EmployeeNumber en Date. Ze worden van elkaar onderscheiden door de volgnummers 0 tot 3.

Binnen de Set van het PersonnelRecord worden ook nog 4 typen gedefinieerd. Deze zijn context specifiek, en hebben alleen binnen de Set betekenis. Ook deze zijn genummerd van 0 tot 3.

Hieronder volgt een formele beschrijving van de waarde van het record:

```
{
    {givenName "John", initial "P", familyName "Smith"},
    title "Director",
    51,
    dateOfHire "19710917",
    nameOfSpouse{givenName "Mary", initial "T", familyName "Smith"},
    numberOfChildren 2}
```

De codering wordt als volgt gedaan:

Een PersonnelRecord is van de klasse application, dus bit 8,7 zijn 0,1. Het bestaat uit andere elementen en is dus constructor, bit 6 is dus 1. De ID-code (volgnummer) is hierboven gegeven en is 0. Het identifier octet is dus 01100000 of hexadecimaal 60. Op de zelfde manier volgt de codering van Name : 61, alleen de ID-code is anders. EmployeeNumber en Date zijn beide primitief. Bit 6 is bij deze typen dus 0. De codering wordt dus 42 voor EmployeeNumber en 43 voor Date.

Title is van de Klasse context-specific, en hier zijn bit 8 en 7 10.
 Het is samengesteld uit een ander element, namelijk een IA5String.
 Bit 6 is dus 1. Title wordt dus gecodeerd als 10100000 : A0
 Hexadecimaal. DateOfhire, nameOfSpouse en numberOfChildren worden op
 de zelfde manier gecodeerd als respectievelijk A1, A2 en A3.

Het hele record wordt als volgt gecodeerd :

Personnel

record	Length	Contents
60	46	
	Name	Length Contents
	61	10
		IA5String Length Contents
	16	04 "John"
		IA5String Length Contents
	16	01 "P"
		IA5String Length Contents
	16	05 "Smith"
	Title	Length Contents
	A0	0A
		IA5String Length Contents
	16	08 "Director"
	Employee	Length Contents
	Number	
	42	01 33
	Date of	Length Contents
	Hire	
	A1	0A
		Date Length Contents
	43	08 "19710917"
	Name of	Length Contents
	Spouse	
	A2	12
		Name Length Contents
	61	10
		IA5String Length
	16	04
	Contents	"Mary"
		IA5String Length
	16	01
	Contents	"T"
		IA5String Length
	16	05
	Contents	"Smith"
	Number of	Length Contents
	Children	
	A3	03
		Integer Length Contents
	02	01 03

4 DOCUMENT STRUCTUUR VAN MULTIMEDIA DOCUMENTEN IN BESTAANDE SYSTEMEN

Om telegraphics samen met tekst in één document over een Message Handling Systeem te kunnen versturen, moet het Message Handling Systeem de mogelijkheid hebben om tekst en telegraphics te combineren in één document. Er zijn verschillende manieren om verschillende soorten inhoud samen te voegen in één document, dat dan een multimedia document wordt genoemd (in de literatuur komen ook de termen 'mixed mode document', 'multi type document' en 'compound document' voor). In dit hoofdstuk zullen een aantal systemen worden beschreven waarmee multimedia documenten verstuurd kunnen worden.

Aan het 'Norwegian Computing Centre' (NCC) is het UNINETT Multe project uitgevoerd, wat als doel had een model te maken van een multimedia document [6]. Hiermee moest het mogelijk zijn om multimedia documenten uit te wisselen tussen verschillende computers met verschillende presentatievormen. Om dit mogelijk te maken zijn er eisen opgesteld voor een multimedia document systeem. De volgende hebben betrekking op de document structuur :

- 1- Het model moet bruikbare logische concepten bevatten.
Omdat het model gebruikt moet worden tussen verschillende configuraties van systemen, moet het model onafhankelijk zijn van een bepaalde configuratie. Bij voorbeeld: een bladzijde, die door een printer is geproduceerd, kan niet op dezelfde manier op een beeldscherm worden getoond. Het model moet dus zo min mogelijk dit soort fysieke concepten (pagina) gebruiken, maar logische zoals hoofdstuk, paragraaf en alinea.
- 2- Er moeten niet te veel logische concepten worden gebruikt.
Hierdoor kan het model voor verschillende behoeften worden gebruikt. Zo ontstaat een flexibel systeem, waarmee eenvoudige berichten kunnen worden verzonden maar ook ingewikkelde.

Bovenstaande eisen zijn ook van toepassing als tekst en telegraphics worden samengevoegd in een multimedia document. Deze eisen zullen dan ook gebruikt worden om de modellen te toetsen, die in dit hoofdstuk worden beschreven.

De volgende systemen komen aan de orde :

- Agora van INRIA (Frankrijk)
- van DARPA (USA): - het experimentele model
 - het Internet model, dat door verschillende systemen wordt gebruikt
 - DIAMOND
- MINOS van het CSRI van de universiteit van Toronto (Canada),
- het systeem van NEC corporation (Japan),
- twee ESPRIT projecten van de EEG, die gebaseerd zijn op de Office Document Architecture standaard van de ECMA, ISO en CCITT.

Steeds zal eerst een beschrijving van de document structuur gegeven worden. Deze zal daarna getoetst worden aan bovenstaande eisen. Tot slot van dit hoofdstuk zal een keuze gemaakt worden voor een model waarin Telegraphics gebruikt kan worden.

4.1 Agora

In deze paragraaf zal de structuur van een document in Agora worden behandeld [7,8,9,10,11]. Agora is een message system dat is ontwikkeld als een onderdeel van het Kayak project aan het INRIA in Frankrijk. Kayak is een project met office systems, waarvan één van de doelen was om nieuwe oplossingen te vinden voor informatie representatie en manipulatie.

In Agora wordt een document opgeslagen in een standaard vorm. Dit wordt een virtueel document genoemd [11]. Hier kunnen nog een aantal attributen aan toegevoegd worden. Deze attributen bepalen zaken als presentatie, opslag en distributie. Het voordeel van het opslaan van een document in virtuele vorm is dat het onafhankelijk is van het uitvoer orgaan. Het kan bij voorbeeld in verschillende formaten worden weergegeven.

Een document bestaat uit : Front Information, Body en eventueel Back-Information. In formele notatie :

Document ::= Front-Information, Body, [Back-Information]

Hierbij kan de Front-Information een titel, een logo, de schrijver, het voorwoord of een samenvatting bevatten. In formele notatie :

Front-Information ::= Document-Title, [Logo], [Author],
[Preface], [Summary], [Abstract]

Een Body bestaat uit een aantal segmenten :

Body ::= Segment, { Segment }

Een segment kan uit weer andere segmenten bestaan. Een segment kan ook bestaan uit units en atoms. Bij voorbeeld in een text document zijn de letters atoms, de woorden de units en de zinnen, paragrafen en hoofdstukken de segmenten.

Segment n ::= Name, [attributes], { Segment n-1 } / { data-unit }

In een data unit wordt eerst aangegeven van welk type het is, daarachter wordt de data van de unit gegeven in atoms.

Er zijn vier verschillende typen :

- 1- Alfnumeriek
- 2- Raster-graphics
- 3- Vector-graphics
- 4- Photographic-Image

De Back-Information kan een appendix en een index bevatten.

Back-Information ::= [Appendix], [Index]

Er kunnen verschillende soorten attributen aan het document toegevoegd worden. Ik noem hier alleen de zogenaamde Viewing Attributes. Deze kunnen op drie niveaus worden toegevoegd :

- 1- Atomic: Bij text geven ze bijvoorbeeld aan welke karakterset er gebruikt moet worden, of karakters onderstreept of dik gedrukt moeten worden.
- 2- Global: Deze geven aan hoe de rechthoek, waarin de inhoud van het segment komt, wordt weergegeven. Bij voorbeeld de X,Y dimensies van de rechthoek, justification links of rechts.
- 3- Page: Geven aan hoe de rechthoek op het papier gezet zal worden. Bij voorbeeld de marges, nummering van bladzijden.

De virtuele vorm zorgt ervoor dat een document niet afhankelijk is van een bepaald uitvoerorgaan. De logische structuur wordt in een document gebracht met behulp van segmenten. Elk segment kan bestaan uit andere segmenten en op deze manier wordt een document hiërarchisch gestructureerd in hoofdstukken, paragrafen, alinea's, figuren enzovoorts. Ook eenvoudige documenten kunnen worden gemaakt. De eenvoudigste vorm is een document, dat slechts uit één segment bestaat.

4.2 De DARPA systemen

Deze paragraaf behandelt systemen die achtereenvolgens in Amerika bij verschillende instituten onder DARPA (Defense Advanced Research Projects Agency) zijn ontwikkeld, als toepassing van de DARPA Internet Protocol Architecture. Allereerst in paragraaf 4.2.1 het experimentele model, dat is ontwikkeld om technieken uit te vinden om verschillende typen objecten in één document bij elkaar te voegen [12,18].

Paragraaf 4.2.2 zal het DARPA internet model behandelen [13,14]. Het DARPA internet model is voortgekomen uit het experimentele model en is ontwikkeld door verschillende groepen in de DARPA onderzoeksgemeenschap. Hierbij werd een oplossing gezocht naar het probleem van het verzenden van multimedia documenten tussen verschillende computersystemen. Het DARPA internet model gebruikt de multimedia content (MMCP) structuur. Deze structuur is bij SRI-international geïmplementeerd in het CCWS (Command and Control WorkStation) project [15]. Het Information Sciences Institute (ISI) van de University of Southern California gebruikt dezelfde structuur, echter met een kleine toevoeging, in het MMM systeem [16].

Paragraaf 4.2.3 zal het Diamond model behandelen, dat is ontwikkeld bij Bolt Beranek and Newman [17,18,19]. Dit model is ontstaan uit de ervaringen van zowel het experimentele model als het DARPA internet model. Hierbij zijn verbeteringen aangebracht ten opzichte van de vorige twee modellen.

4.2.1 Het Experimentele model

Een document bestaat uit objecten. Een object kan worden weergegeven door een rechthoek. De positie van het object wordt aangegeven in pixel coördinaten. De hoogte en breedte worden in pixels aangegeven.

1. The first part of the paper discusses the importance of the study of the history of the United States. It is argued that a knowledge of the past is essential for a full understanding of the present and for the development of a sound policy for the future. The author points out that the study of history is not only a means of acquiring knowledge, but also a means of developing the ability to think critically and to make sound judgments.

2. The second part of the paper discusses the importance of the study of the history of the United States.

It is argued that a knowledge of the past is essential for a full understanding of the present and for the development of a sound policy for the future. The author points out that the study of history is not only a means of acquiring knowledge, but also a means of developing the ability to think critically and to make sound judgments. The author also discusses the importance of the study of the history of the United States in the context of the current political and social climate. It is argued that a knowledge of the past is essential for a full understanding of the present and for the development of a sound policy for the future. The author points out that the study of history is not only a means of acquiring knowledge, but also a means of developing the ability to think critically and to make sound judgments.

3. The third part of the paper discusses the importance of the study of the history of the United States.

It is argued that a knowledge of the past is essential for a full understanding of the present and for the development of a sound policy for the future. The author points out that the study of history is not only a means of acquiring knowledge, but also a means of developing the ability to think critically and to make sound judgments.

Er zijn vier soorten objecten :

- 1-Text Het object is een regel text. Elke regel is slechts één object. Er kunnen dus niet meerdere regels in één object.
- 2-Scanned Image Een figuur of tekening die door een facsimile scanner is gedigitaliseerd.
- 3-Voice Een gesproken passage dat gecodeerd is door een vocoder. Een Icon (merkteken) geeft op het scherm de aanwezigheid aan van een stuk gesproken tekst.
- 4-General Object Een verzameling data, waarvan de presentatie door een 'companion program' wordt uitgevoerd. Het programma maakt over het document een window waarin het general object wordt getoond. Hiermee kan steeds meer detail worden getoond. De volgende data typen kunnen ingeneral objects worden gebruikt :
Spread-sheets, graphical line drawings en graphs.

Het voordeel van het gebruik van general objects is dat het systeem niets hoeft af te weten van de details van een general object maar alleen het data type en de naam van companion program.

Het nadeel is dat het document niet in één keer visueel is weer te geven. Er komen nieuwe windows over de oorspronkelijke tekst heen.

De structuur wordt gevormd door middel van objecten. Omdat elk object steeds één onderdeel bevat, bij tekst één regel, bij scanned image een figuur, bij voice een stuk gesproken woord, is er geen logische structuur op te bouwen.

4.2.2 Het DARPA Internet model

Een document bestaat uit velden. Een veld is een zogenaamd 'name-value' paar. Daarin staan een keyword dat het veld type aangeeft en een waarde van het veld. Beide zijn van een bepaald data type en bestaan uit data elementen. Er moeten minstens vier velden aanwezig zijn :

- 1- 'date : value'
- 2- 'from : value'
- 3- 'subject : value'
- 4- 'body : value'

Er zijn twee soorten relaties tussen onderdelen van een document :

- 1- logische:
deze bepalen de logische structuur. Bij voorbeeld hoofdstukken, paragrafen, voetnoten, enzovoorts.
- 2- presentatie :
deze bepalen de vorm waarin een document wordt weergegeven in tijd en in ruimte.

Elk gedeelte van een document dat één medium bevat is gestructureerd in het 'name-value' paar presentation element (pe). De naam geeft het type aan. De volgende typen zijn mogelijk : text, graphics, Image, Voice en Facsimile. De value geeft de data samen met control informatie.

Appendix A: Sample Financial Statement

Income Statement (Sample)			
Item	Amount	Unit	Description
Revenue	1000	Dollars	From Sales
Cost of Goods Sold	(600)	Dollars	Materials and Labor
Gross Profit	400	Dollars	
Operating Expenses	(200)	Dollars	Salaries, Rent, etc.
Operating Income	200	Dollars	
Other Income	50	Dollars	Interest Income
Other Expenses	(20)	Dollars	Depreciation
Net Income	230	Dollars	

De presentation elements worden samengevoegd in een zogenaamde presentation descriptor (pd). Er zijn vijf verschillende soorten presentation descriptors :

- Sequential
- Simultaneous
- Independent
- Single Medium
- Single Document

De eerste drie geven aan in welke volgorde de presentation elements moeten worden weergegeven. Bij voorbeeld : Verdeel de display in horizontale stroken. Sequential wil dan zeggen dat er steeds één presentation element in een strook staat. Simultaneous betekent dan dat er twee of meer presentation elements naast elkaar in één strook staan. Independent wordt gebruikt bij een presentation element van het type Voice. De volgorde van weergave maakt dan niet uit.

Bij het MMM syteem is een kleine toevoeging aangebracht. Omdat een boodschap zeer lang kan zijn, vanwege een bit map of voice data, is het mogelijk dat deze niet in zijn geheel in het geheugen van de gebruikte PERQ personal computer past. Daarom worden de verschillende delen van een document in verschillende files opgeslagen. De structuur van het dokument wordt dan in een aparte file gezet. Normaal gesproken zou alles in één document worden gezet. Voordat een boodschap van MMM naar een ander systeem kan worden verzonden moet het dus eerst worden overgezet naar één document.

De body van een document is volgens een boomstructuur opgebouwd en bestaat uit pd's en pe's. Hiermee kan de logische structuur worden gevormd. Het meest eenvoudige document bestaat uit één pe en bestaat dan ook slechts uit één medium.

4.2.3 Diamond

Een document bestaat uit een aantal niet overlappende 'boxes'.
Een box kan bevatten :

- 1- Text
De text kan op verschillende manieren worden gestructureerd.
Paragraph, Itemization, Enumeration en verbatim (woord voor woord).
- 2- Graphics
Er zijn de volgende soorten van graphics mogelijk :
Line, geometric, text en bit map image.
- 3- Images
Dit is een bit map representatie van een tekening, kaart of foto.
- 4- Voices
- 5- Electronic Spread Sheets.

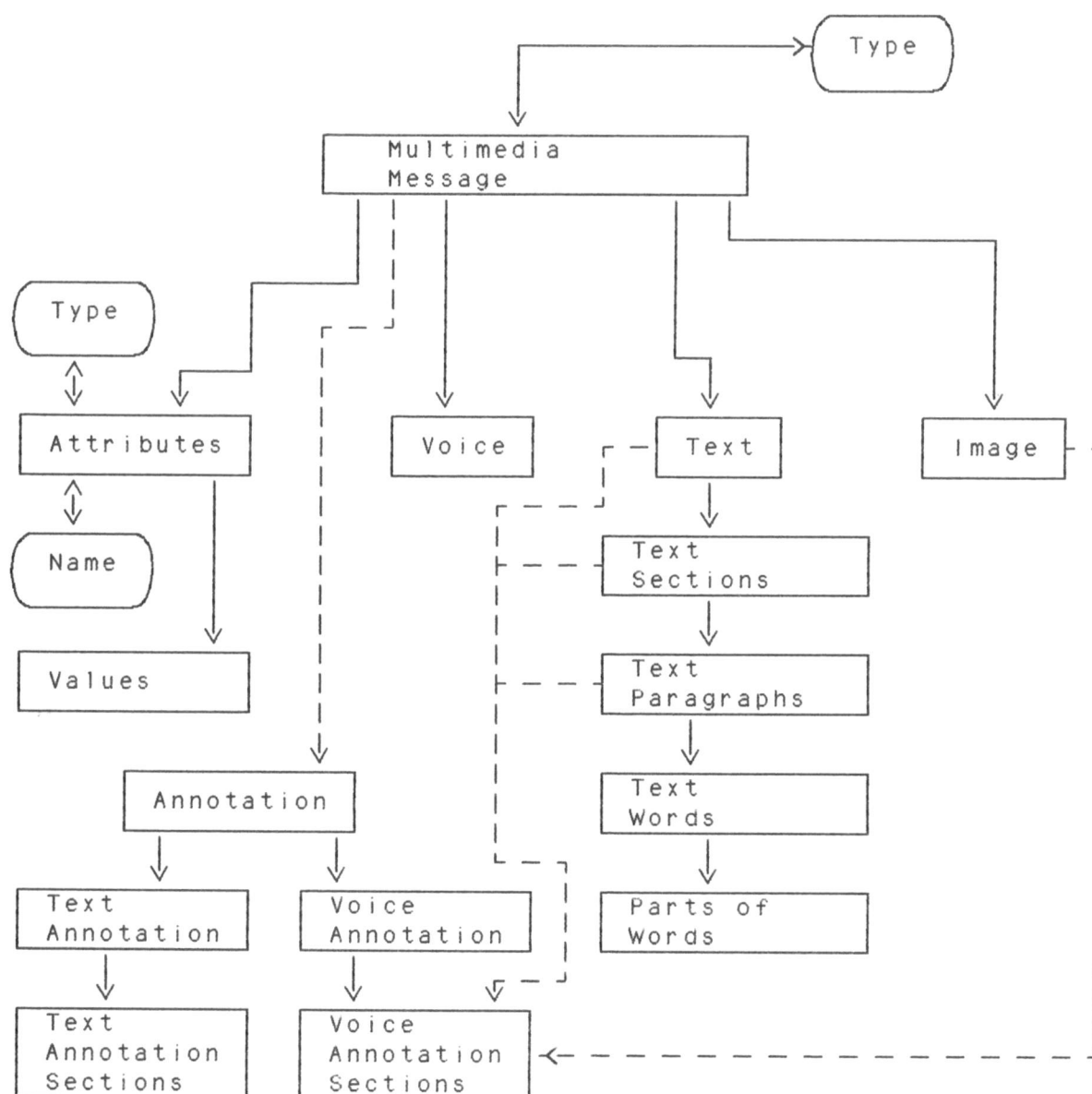
De vorm en plaats van een box wordt weergegeven door de hoogte, breedte en positie. Een positie kan absoluut worden weergegeven door het aangeven van de coördinaten van de linkerbovenhoek. De positie kan ook relatief worden aangegeven. Er wordt dan een relatie met een andere box aangegeven. Bij voorbeeld Above, Below, Right of, Left of etc. Als de positie van een box op deze manier wordt aangegeven dan is het document makkelijk op verschillende soorten displays weer te geven.

Een box wordt ook gebruikt om objecten van verschillende media te groeperen. Het doel hiervan is dat de delen van een document beter zijn te onderscheiden als ook om twee objecten te bundelen tot één object zodat ze zich bij het positioneren en editen als één object gedragen.

Een logische structuur wordt in een document gebracht door boxes te bundelen in een andere box. Op deze manier ontstaat een boomstructuur.

4.3 MINOS

Aan het Computer Systems Research Institute van de Universiteit van Toronto (Canada), is een database systeem ontwikkeld, waarin multimedia documenten opgeslagen kunnen worden [20]. Ze kunnen op verschillende manieren worden opgevraagd, waaronder adressering van de inhoud. Hierdoor moeten extra eisen aan de document structuur worden gesteld.



figuur 4.1. Structuur van een document in MINOS

In figuur 4.1 is de logische structuur van een document weergegeven. Een document heeft een type, waarin de minimale gemeenschappelijke informatie in een groot aantal documenten staan beschreven als een verzameling attributen. Eén of meer van de volgende vier onderdelen kunnen verder nog aanwezig zijn : attributen, voice, text en images. Er mogen nog aantekeningen worden toegevoegd.

Attributen hebben een naam een type en een waarde.

Het text gedeelte bestaat uit sections; deze bestaan weer uit paragrafen; paragrafen bestaan uit woorden; tot slot bestaan woorden uit delen van woorden. Deze manier van structureren is gedaan om het terugvinden van stukken tekst te vereenvoudigen.

Het image gedeelte bestaat uit de volgende delen : type, vector form, raster form, image text en statistical parts.

Er zijn zes verschillende typen mogelijk :

- 1- Graph : de afbeelding bevat minimaal een grafiek
- 2- Pie Chart : de afbeelding bevat minmaal een pie chart
- 3- Histogram : de afbeelding bevat minmaal een histogram
- 4- Table : De afbeelding bevat minmaal een tabel
- 5- Statistical : één van bovengenoemde
- 6- Picture : geen van bovengenoemde

In de vector vorm wordt de afbeelding afgebeeld met behulp van regions, polylines en lines.

In de raster vorm wordt de afbeelding afgebeeld door een raster van punten.

Het statistische gedeelte bestaat uit tabellen, waarin statistische informatie staat voor afbeeldingen van het type graph, pie chart, histogram, tabel of statistical.

Het image text gedeelte bevat tekst, die betrekking heeft op de afbeelding.

Het voice gedeelte bestaat uit gesproken woord.

Aantekeningen bevatten informele informatie over de inhoud van een document en zijn ofwel als tekst danwel als gesproken woord opgenomen. Ze kunnen betrekking hebben op text sections, paragraphs, words en op afbeeldingen.

Een document wordt verdeeld in de drie mogelijke media. Een logische structuur is alleen in de tekst aan te brengen door onderverdeling in hoofdstukken, paragrafen en woorden.

4.4 Het NEC systeem

Aan het C&C Systems research laboratory van NEC corporation in Japan is een experimenteel Mail systeem ontwikkeld, waarmee multimedia documenten kunnen worden verzonden [21].

Een document is hiërarchisch gestructureerd in vijf onderdelen : document, pagina, gebied in een pagina, regels en letters. Er zijn twee typen documenten : stream control en area control.

Stream control documenten zijn bedoeld voor bestaande werkstations, die zijn gebaseerd op tekstverwerkers. Hierin wordt een gebied, dat niet uit tekst bestaat, tussen de tekst stroom gezet.

Area control documenten zijn bedoeld voor geïntegreerde multimedia document editors. Hierin worden de gebieden op de pagina gezet, zoals dat gewenst wordt.

Attributen worden bij elk structureel element (pagina, area, document) gezet. Er zijn twee soorten parameters voor een gebied. De eerste heeft betrekking op de plaatsing, grootte enzovoorts, van het gebied zelf. De tweede heeft betrekking op het medium. Bij tekst bij voorbeeld regelafstand en lettergrootte. Bij figuren en geluid bij voorbeeld codering en lengte.

the first of these is the fact that the
the second is the fact that the
the third is the fact that the
the fourth is the fact that the
the fifth is the fact that the
the sixth is the fact that the
the seventh is the fact that the
the eighth is the fact that the
the ninth is the fact that the
the tenth is the fact that the
the eleventh is the fact that the
the twelfth is the fact that the
the thirteenth is the fact that the
the fourteenth is the fact that the
the fifteenth is the fact that the
the sixteenth is the fact that the
the seventeenth is the fact that the
the eighteenth is the fact that the
the nineteenth is the fact that the
the twentieth is the fact that the
the twenty-first is the fact that the
the twenty-second is the fact that the
the twenty-third is the fact that the
the twenty-fourth is the fact that the
the twenty-fifth is the fact that the
the twenty-sixth is the fact that the
the twenty-seventh is the fact that the
the twenty-eighth is the fact that the
the twenty-ninth is the fact that the
the thirtieth is the fact that the
the thirty-first is the fact that the
the thirty-second is the fact that the
the thirty-third is the fact that the
the thirty-fourth is the fact that the
the thirty-fifth is the fact that the
the thirty-sixth is the fact that the
the thirty-seventh is the fact that the
the thirty-eighth is the fact that the
the thirty-ninth is the fact that the
the fortieth is the fact that the
the forty-first is the fact that the
the forty-second is the fact that the
the forty-third is the fact that the
the forty-fourth is the fact that the
the forty-fifth is the fact that the
the forty-sixth is the fact that the
the forty-seventh is the fact that the
the forty-eighth is the fact that the
the forty-ninth is the fact that the
the fiftieth is the fact that the
the fifty-first is the fact that the
the fifty-second is the fact that the
the fifty-third is the fact that the
the fifty-fourth is the fact that the
the fifty-fifth is the fact that the
the fifty-sixth is the fact that the
the fifty-seventh is the fact that the
the fifty-eighth is the fact that the
the fifty-ninth is the fact that the
the sixtieth is the fact that the
the sixty-first is the fact that the
the sixty-second is the fact that the
the sixty-third is the fact that the
the sixty-fourth is the fact that the
the sixty-fifth is the fact that the
the sixty-sixth is the fact that the
the sixty-seventh is the fact that the
the sixty-eighth is the fact that the
the sixty-ninth is the fact that the
the seventieth is the fact that the
the seventy-first is the fact that the
the seventy-second is the fact that the
the seventy-third is the fact that the
the seventy-fourth is the fact that the
the seventy-fifth is the fact that the
the seventy-sixth is the fact that the
the seventy-seventh is the fact that the
the seventy-eighth is the fact that the
the seventy-ninth is the fact that the
the eightieth is the fact that the
the eighty-first is the fact that the
the eighty-second is the fact that the
the eighty-third is the fact that the
the eighty-fourth is the fact that the
the eighty-fifth is the fact that the
the eighty-sixth is the fact that the
the eighty-seventh is the fact that the
the eighty-eighth is the fact that the
the eighty-ninth is the fact that the
the ninetieth is the fact that the
the ninety-first is the fact that the
the ninety-second is the fact that the
the ninety-third is the fact that the
the ninety-fourth is the fact that the
the ninety-fifth is the fact that the
the ninety-sixth is the fact that the
the ninety-seventh is the fact that the
the ninety-eighth is the fact that the
the ninety-ninth is the fact that the
the hundredth is the fact that the

Een document is gestructureerd in een pagina, regels en letters. Dit zijn presentatie onderdelen en ze bieden geen mogelijkheid tot logische structuren.

4.5 Office Document Architecture

Door de ECMA, CCITT en ISO is een norm ontwikkeld voor 'Office Document Architecture' [22,23,24,25,26,27,28]. Met deze architectuur kunnen multimedia documenten worden gemaakt. Een project van ESPRIT (European Strategic Program of Research and Development in Information Technology) van de EEG, genaamd HERODE, heeft zich bezig gehouden met de implementatie van een prototype, waarmee gebruikers documenten kunnen maken, die aan ODA voldoen [23]. Een vervolg op dit project was een ander ESPRIT project : PODA, Piloting the Office Document Architecture [5].

Structuur kan op twee manieren aan een document worden gebracht. Door een logische en een layout structuur. De logische structuur is een onderverdeling in hoofdstukken, paragrafen, alinea's, enzovoorts. De layout structuur is een onderverdeling die betrekking heeft op layout onderdelen : page set, page, frame en block.

Een document kan drie vormen aannemen, afhankelijk van de aanwezigheid van bovenstaande structuren :

- formatted : alleen de layout structuur is aangegeven. het document wordt weergegeven zoals de zender dat bedoeld heeft.
- processable : de logische structuur is aangegeven. Het document kan nog veranderd worden.
- formatted/processable : Beide structuren zijn aanwezig. Het document kan bekeken worden maar er kunnen ook veranderingen worden aangebracht.

Naast bovenstaande specifieke structuren kunnen er ook generieke structuren worden gedefinieerd. Documenten met dezelfde generieke structuur vormen een document klasse, waarin de documenten bepaalde eigenschappen gemeen hebben. Een voorbeeld hiervan is een brief. De vaste structuur van een brief wordt in een generieke structuur vastgelegd, die steeds gebruikt wordt als er een nieuwe brief wordt gemaakt.

Er zijn drie zogenaamde 'Content Architectures' gedefinieerd : Character, Raster graphics en geometric graphics. In een content architecture staat aangegeven hoe informatie van een bepaald type gecodeerd moet worden, welke parameters er moeten worden opgegeven en hoe dit afgebeeld wordt. Er is ruimte in ODA om nog meer content architectures te definiëren.

Bij ODA kan zowel een logische structuur als een layout structuur worden aangebracht aan een document. De logische structuur wordt aangegeven door het document onder te delen in objecten. Er zijn twee soorten objecten, elementaire objecten en samengestelde objecten. Hiermee kan een hiërarchische structuur worden gemaakt.

4.6 Keuze van een model

Uit het bovenstaande blijkt dat het experimentele model van DARPA, MINOS en het NEC systeem geen logische structuur bezitten. De andere systemen hebben allemaal de mogelijkheid om een document volgens een boomstructuur op te bouwen.

De ODA structuur zal worden gebruikt voor het maken van multimedia documenten met Telegraphics in X.400, omdat dit de internationale norm is en bovendien aansluit bij de presentatie vorm van X.409. In het volgende hoofdstuk zal een uitgebreidere beschrijving van ODA worden gegeven.

5 OFFICE DOCUMENT ARCHITECTURE

Vanaf 1980 hebben verschillende standaardisatie-instituten zich gezamenlijk beziggehouden met het maken van een norm voor kantoor document architectuur, ECMA, CCITT en ISO [22,24,25,26,27,28]. In 1984 kwam de CCITT met de recommendation T.73 : "Document interchange protocol for telematic services", een norm voor het verzenden van gemengde tekst en facsimile documenten. In 1985 kwam ECMA met de norm ECMA-101 "Office Document Architecture"(ODA), die opgevolgd werd in 1986 door de ISO norm ISO DIS 8613 "Document Structures". Deze norm is een verbetering en uitbreiding van ECMA-101 en zal in 1988 een internationale standaard worden. Gelijkzeitig heeft de CCITT de norm van de T.400 serie (Document Transfer, Access and Manipulation) ontwikkeld. In deze serie wordt T.73 uitgebreid met ISO 8613. Daarbij zijn T.411 tot T.418 gelijk aan ISO 8613 deel 1 tot en met 8. In deze standaard kunnen zowel tekst als geometrische en bitmap graphics in een document voorkomen. In dit hoofdstuk zal een uiteenzetting worden gegeven van de standaard ISO 8613 (ODA).

5.1 De basisprincipes van ODA

Een document kan op twee manieren hiërarchisch gestructureerd worden: met een logische structuur en met een layout structuur.

De logische structuur geeft aan hoe de inhoud in logisch verband met elkaar staat. Hiertoe is het document verdeeld in een aantal logische objecten. Bij voorbeeld hoofdstuk, paragraaf, alinea of figuren. Er zijn samengestelde en elementair logische objecten. Een hoofdstuk is bij voorbeeld een samengesteld object, dat kan bestaan uit paragrafen, alinea's en figuren. De laatste twee zijn elementaire objecten.

De layout structuur geeft aan hoe het document moet worden weergegeven. Bij voorbeeld de plaats en de dimensies van een stuk tekst of van een tekening. Er zijn vier soorten layout objecten :

- page set - een groep verbonden pagina's bij voorbeeld een hoofdstuk
- page - pagina, om de inhoud te positioneren op het weergave oppervlak
- frame - om gebieden met een bepaalde layout aan te geven
- block - om aparte stukken van een document te positioneren

Voor de inhoud zijn zogenaamde 'content architectures' gedefinieerd. Daarin worden de regels over structuur en presentatie van het inhoud type gedefinieerd. Er zijn drie 'content architectures' gedefinieerd: tekst, raster graphics en geometric graphics. Hieraan kunnen andere 'content architectures' worden toegevoegd zonder dat daarvoor de document architectuur veranderd hoeft worden.

In figuur 5.1 wordt aangegeven hoe de logische structuur en layout structuur en de inhoud met elkaar in verband staan.

THE EFFECT OF TEMPERATURE ON THE

DETERMINATION OF THE RATE OF
REACTION

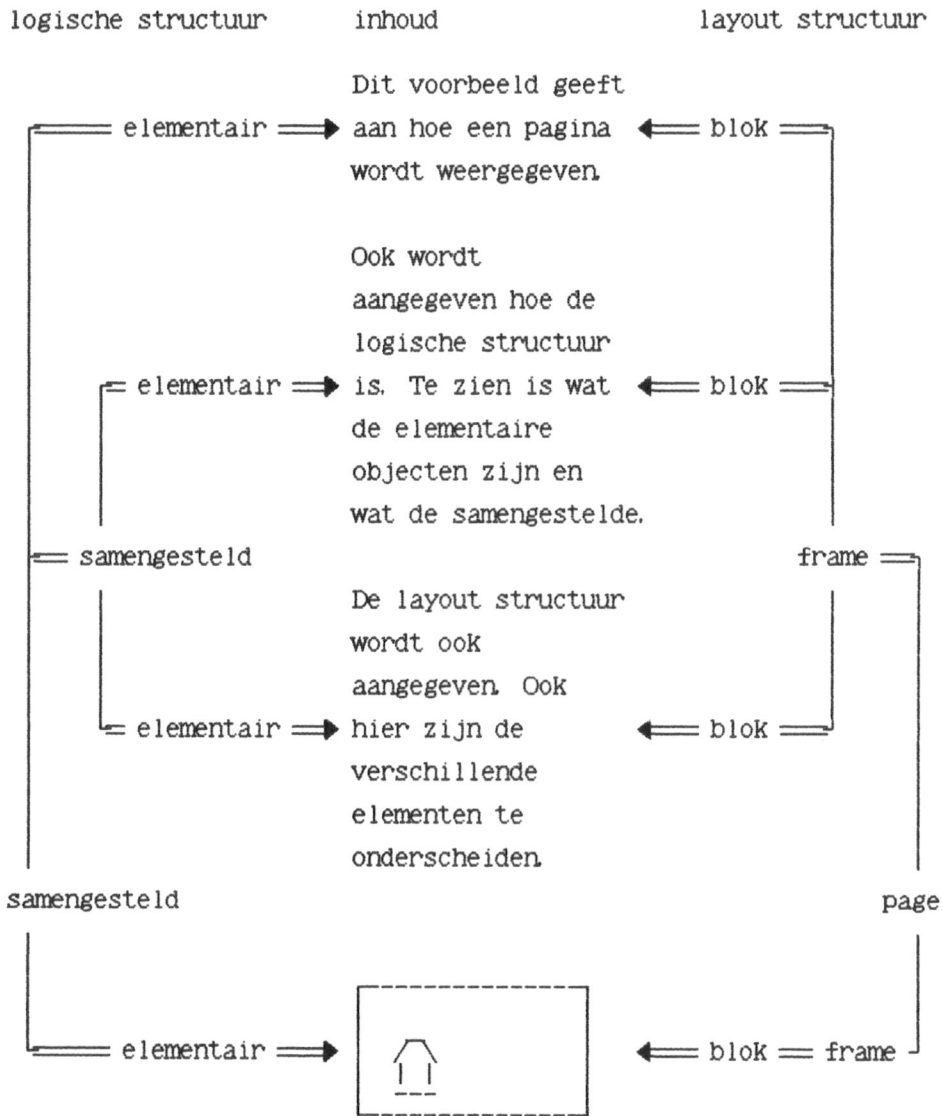
The rate of reaction is affected by temperature. The rate of reaction increases as the temperature increases. This is because the molecules have more kinetic energy and are moving faster. They are more likely to collide with each other and react. The rate of reaction also increases as the concentration of the reactants increases. This is because there are more molecules in the same volume, so they are more likely to collide and react.

THE EFFECT OF TEMPERATURE ON THE

DETERMINATION OF THE RATE OF
REACTION

The rate of reaction is affected by temperature. The rate of reaction increases as the temperature increases. This is because the molecules have more kinetic energy and are moving faster. They are more likely to collide with each other and react. The rate of reaction also increases as the concentration of the reactants increases. This is because there are more molecules in the same volume, so they are more likely to collide and react.

The rate of reaction is affected by temperature. The rate of reaction increases as the temperature increases. This is because the molecules have more kinetic energy and are moving faster. They are more likely to collide with each other and react. The rate of reaction also increases as the concentration of the reactants increases. This is because there are more molecules in the same volume, so they are more likely to collide and react.



figuur 5.1

5.2 Generieke structuur

Een belangrijk onderdeel van ODA is de generieke structuur. Met een generieke structuur kan een klasse van documenten gecreëerd worden. Een klasse is een verzameling met dezelfde eigenschappen. Bij voorbeeld een memo, een brief of een verslag. Er kan zowel een logische generieke structuur als een layout generieke structuur bestaan.

Het gebruik van generieke structuren biedt twee voordelen. Eén voordeel van generieke structuren is dat daarmee documenten van een klasse op een consistente manier kunnen worden gemaakt. Een ander voordeel is dat de beschrijving van een klasse slechts één keer hoeft te worden verzonden. Als de ontvanger de beschrijving van een klasse al heeft dan kan in het verzonden document volstaan worden hiernaar te verwijzen en dan hoeft de generieke structuur niet verzonden te worden.

5.3 Attributen

Een attribuut is een parameter van een document, of een gedeelte van een document, die bepaalde eigenschappen aangeeft. Er zijn vier soorten attributen :

- 1- document architectuur attributen
- 2- presentatie attributen
- 3- coderingsattributen
- 4- attributen voor de verwerking van documenten

Voorbeelden van document architectuur attributen zijn 'object identifier', positie, afmetingen maar ook layout aanwijzingen. Layout aanwijzingen (layout directives) zijn attributen, die bij logische objecten horen. Deze attributen geven aan hoe bepaalde layout aanwijzingen van toepassing zijn op deze objecten. Voorbeelden hiervan zijn : paragraaf op dezelfde bladzijde, eerste regel inspringen.

Presentatie attributen zijn alleen van toepassing op elementaire logische en layout objecten. Voorbeelden zijn : regelafstand, tekenafstand of het zogenaamde 'clipping' bij figuren.

Coderingsattributen zijn alleen op de inhoud van toepassing. Een voorbeeld is het aantal pels per rij bij een bitmap tekening.

Attributen voor verwerking hebben betrekking op het opslaan en weer ophalen van documenten en op het formatten en editen. Deze attributen worden gezamenlijk het document profiel genoemd. Voorbeelden zijn : titel, schrijver, onderwerp, datum, etc.

De volgende attributen moeten in elk document staan : title, length, character set en content type.

5.4 Vormen van een document

Een document kan één van de volgende drie vormen aannemen :

'formatted' - Zoals bedoeld is door de maker van het document. Een formatted document bestaat minimaal uit het document profiel en de specifieke layout structuur (en de inhoud). Daar kan dan eventueel de generieke layout structuur aan toegevoegd worden.

'processable' - De ontvanger kan veranderingen en toevoegingen aanbrengen aan het document. Een processable document bestaat minimaal uit het document profiel en de specifieke logische structuur (en de inhoud). Daar kan nog eventueel de generiek logische en layout structuur aan toegevoegd worden.

'formatted processable' - Combineert de mogelijkheden van de eerste twee vormen; sommige gebruikers kunnen alleen de inhoud willen bekijken, terwijl andere ook wijzigingen zullen kunnen aanbrengen. Een formatted processable document bestaat minimaal uit het document profiel en uit de specifieke logische en layout structuur (en de inhoud). Daar kan nog de generiek logische en layout structuur aan worden toegevoegd.

5.5 Codering van een document

Deel 5 van ISO 8613 geeft een beschrijving van Office Document Interchange Format (ODIF). Hiermee wordt het formaat gedefiniëerd van documenten, die volgens ODA zijn gestructureerd. In Appendix A wordt een lijst gegeven van de data formaten. Deze worden weergegeven in de abstracte notatie van ASN.1, dat ook door X.409 wordt gebruikt. De codering van het ASN.1 formaat is in paragraaf 3.3 besproken.

6 ARCHITECTUUR VAN EEN TELEGRAPHICS MULTIMEDIA DOCUMENT

In vorige hoofdstukken is duidelijk gemaakt, dat voor het verzenden van Telegraphics de norm ODA voor document architectuur gebruikt kan worden. Er is echter nog niet duidelijk gemaakt met welke problemen rekening gehouden moet worden als Telegraphics wordt verzonden tussen niet-homogene systemen, dat wil zeggen systemen met verschillende presentatie apparatuur zoals terminals en printers. Dit hoofdstuk zal hier aandacht aan besteden. Tevens zal worden aangegeven hoe deze problemen zijn op te lossen. Daarna zal een beschrijving worden gegeven van een Content Architecture van Telegraphics. Tot slot wordt een voorbeeld gegeven van een document, dat van deze content architecture gebruik maakt.

6.1 Overzenden van telegraphics bij verschillende resoluties

Als er boodschappen met X.400 worden verzonden dan moet het mogelijk zijn om hierbij terminals van verschillende resoluties te gebruiken. In deze paragraaf worden de problemen uiteengezet, die hierbij ontstaan. Losse tekst of losse figuren leveren geen problemen op maar als tekst met figuren gecombineerd worden, zoals bij telegraphics, ontstaan er problemen. Alleen door communicatie tussen terminals met gelijke resolutie toe te staan of door 'sideways scrolling' mogelijk te maken kan dit opgelost worden.

Als er alleen teksten worden verzonden dan ontstaan er geen problemen bij verschillende resoluties. Neem bij voorbeeld een terminal met een regelbreedte van 40 tekens. Als deze een tekst verzendt naar een terminal met een regelbreedte van 80 tekens dan komen op deze terminal twee regels van de eerste terminal op één regel te staan. Wordt een tekst teruggezonden dan kan er slechts een halve regel op één regel worden weergegeven. In beide gevallen verandert er aan de informatie van de tekst niets.

Wanneer er alleen figuren worden overgezonden, ontstaan er nog steeds geen problemen als dit gebeurt van één resolutie naar een andere resolutie. Bij verkleining van de resolutie wordt het figuur groter. Andersom wordt het figuur kleiner. Alleen als het oppervlak van de afbeelding vaste afmetingen heeft zal er bij verkleining van de resolutie informatie verloren gaan, de afbeelding wordt vager.

In de systemen, die in hoofdstuk 4 zijn besproken, zijn tekst en figuren gecombineerd. Hierbij werd de tekst in één blok gezet en de figuren in een ander blok. Als er dan verzonden wordt van de ene naar de andere resolutie dan veranderen de maten van elk blok en daardoor ook de tekst of het figuur in het blok. Dit gebeurt voor elk blok afzonderlijk, zodat de verandering binnen het blok het zelfde is als de verandering van tekst alleen of van figuren alleen.

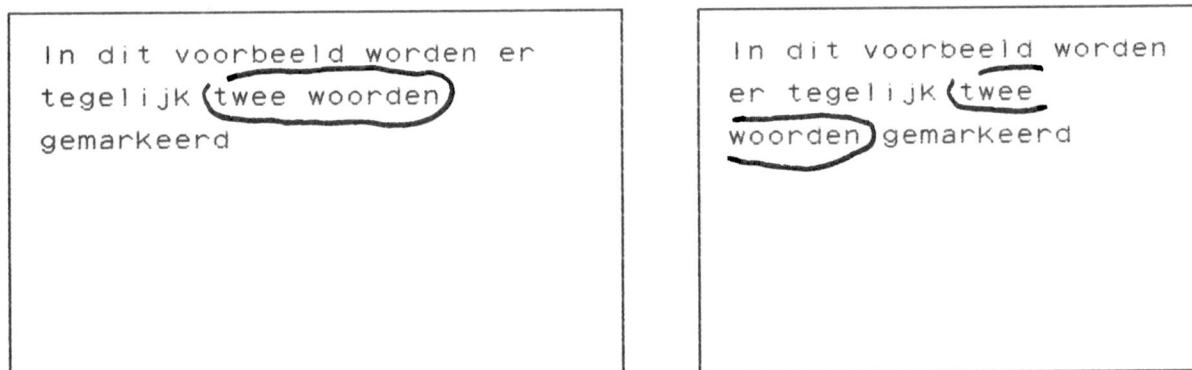
In telegraphics ontstaan er wel problemen met het verzenden van één resolutie naar een andere resolutie. In telegraphics worden ook tekst en figuren gecombineerd. Hier worden de tekst en de figuren in het zelfde blok gezet, omdat de figuren een directe relatie met de tekst kunnen hebben en niet als een afzonderlijk geheel gezien kunnen worden.

Aan de hand van een aantal voorbeelden zal uiteengezet worden welke problemen er kunnen ontstaan. Bij de voorbeelden wordt telkens van een hogere naar een lagere resolutie overgegaan. De problemen zijn hetzelfde als van een lagere naar een hogere resolutie wordt gegaan. In de rechter plaatjes staat de markering aangegeven zoals hij gewenst wordt.



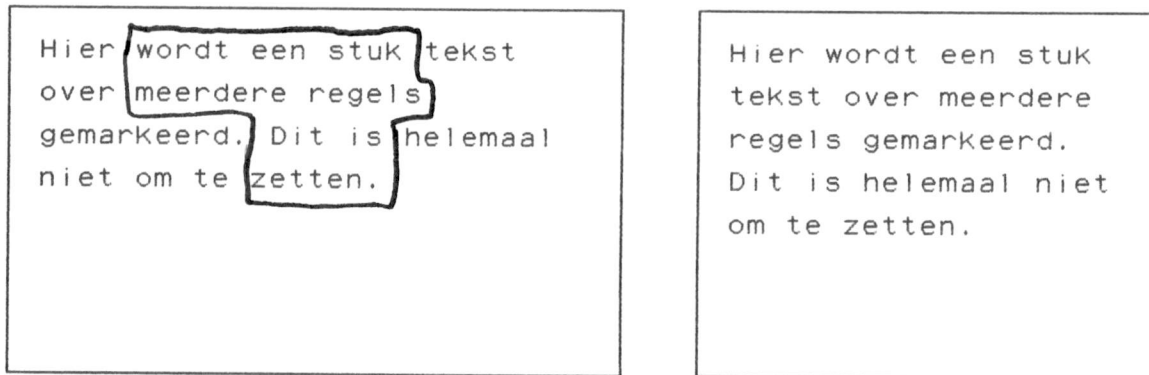
Figuur 6.1 Markering van een woord

In figuur 6.1 is aangegeven wat er gebeurt als een woord is gemarkeerd. Omdat er minder woorden passen op een regel, komen de meeste woorden van een tekst op een andere plaats te staan. Alleen het eerste stuk van de eerste regel blijft op zijn plaats. Omdat de markering is gedefinieerd ten opzichte van de linkerbovenhoek van het scherm, blijft deze op zijn plaats staan en markeert een verkeerd woord. Dit zou verholpen kunnen worden door de markering ten opzichte van het woord dat gemarkeerd moet worden te definiëren.



Figuur 6.2 Markering van twee woorden

Een volgend probleem zien we in figuur 6.2. Nu zijn er twee woorden op de zelfde regel achter elkaar gemarkeerd. Het kan dan, zoals in het voorbeeld, zo uitkomen dat het eerste woord op één regel staat terwijl het volgende woord op de volgende regel staat. Om dit goed te blijven te markeren moet de markering in tweeën worden gesplitst.



Figuur 6.3 Markering van een blok

In het figuur 6.3 zien we waar het niet meer bevredigend is om te zetten. Nu wordt een blok tekst gemarkeerd. De woorden binnen het blok komen in het tweede plaatje geheel verspreid over de regels terecht en hebben geen samenhang meer. De markering kan dan niet worden overgezet.

Het blijkt dus dat het verzenden van telegraphics bij terminals van verschillende resoluties gepaard kan gaan met verminking van de boodschap. Daarom kan alleen het verzenden van berichten van terminals met gelijke resolutie worden uitgevoerd. Hiertoe moet in de boodschap worden aangegeven welke resolutie de zender gebruikt. De ontvanger kan dan alleen de boodschap ontvangen als zijn terminal ook in die resolutie kan werken. Bij terminals, die verschillende resoluties aankunnen, moet de terminal eerst in de juiste resolutie worden gezet alvorens het bericht kan worden weergegeven.

Om het toch mogelijk te maken met verschillende resoluties te werken kan gebruik gemaakt worden van 'sideways scrolling'. Hierbij wordt aanvankelijk alleen de linkerhelft van de pagina op het scherm getoond. Door de pagina op het scherm naar links te schuiven kan dan ook de rest van de pagina bekeken worden. Ook hier moet ergens in de boodschap de resolutie vermeld staan van de zender, omdat dan de breedte van de pagina bepaald kan worden.

6.2 Protocol voor ontvangst van Telegraphics document

Een document bevat de volgende attributen : graphics resolution en text resolution.

De text-resolutie en de graphics-resolutie worden apart opgegeven, omdat deze niet altijd de zelfde verhouding hebben. Vergelijk bij voorbeeld CGA met Hercules :

CGA heeft een grafische resolutie van 640*200 pixels en een tekst resolutie van 80*25.

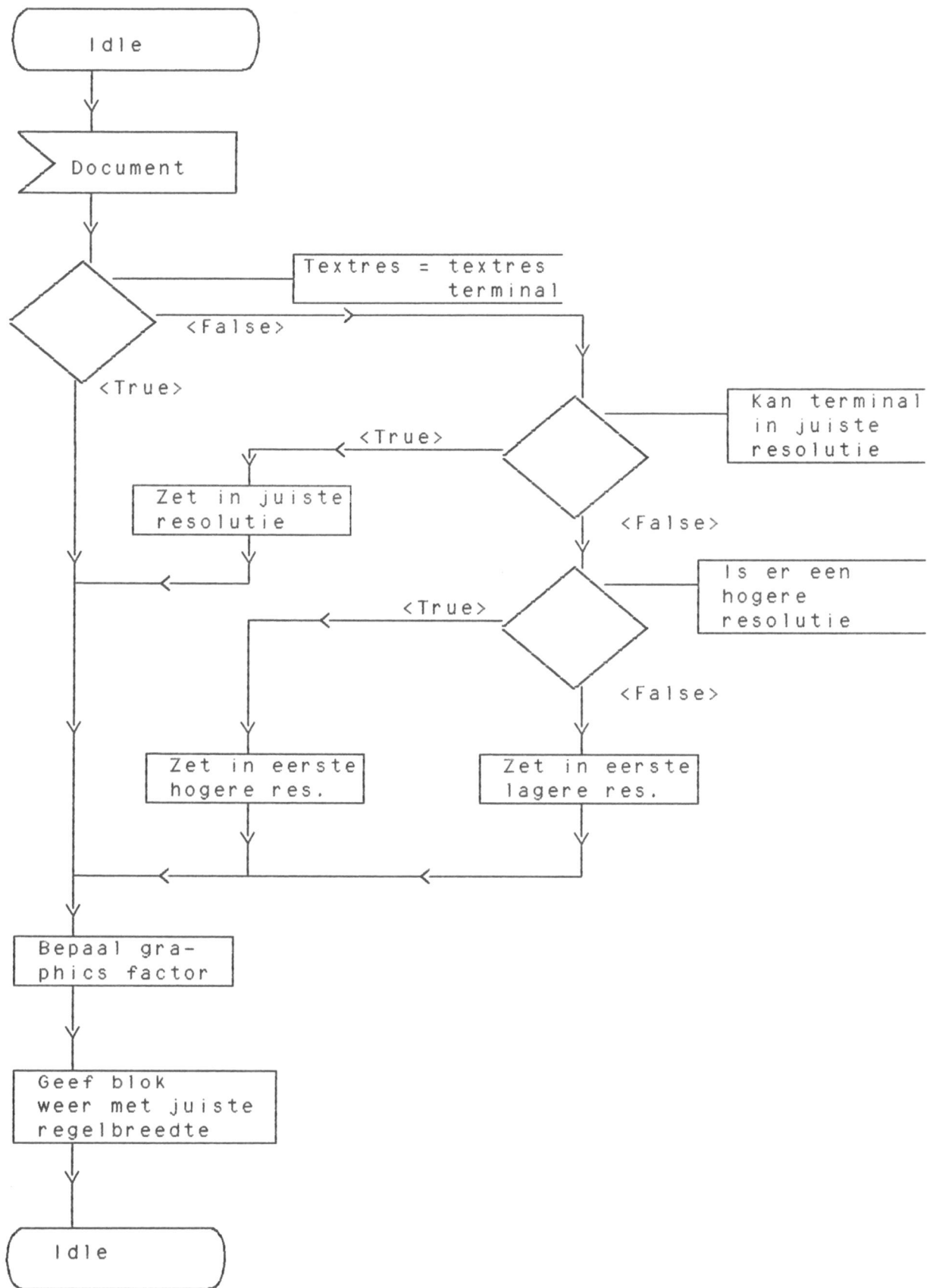
Hercules heeft een grafische resolutie van 720*350 pixels maar ook een tekst resolutie van 80*25.

Als van Hercules naar CGA wordt verzonden dan zal de tekst op de zelfde plaats blijven hoewel de (grafische) resoluties verschillend zijn.

Op de volgende bladzijde staat het protokol voor ontvangst van een document. Hierbij is gebruik gemaakt van SDL symbolen [29]. Als een document wordt ontvangen, dan wordt eerst gekeken of de tekst resolutie van het document overeenkomt met de tekst resolutie van de terminal. Als dit het geval is dan hoeft alleen een schaalfaktor voor de graphics te worden berekend. Neem als voorbeeld weer een boodschap van Hercules naar CGA : het middelste punt bij voorbeeld is bij Hercules 360,175 maar bij CGA 320,100. Om dit dus op de goede plaats te krijgen moeten de horizontale coördinaten met 8/9 worden vermenigvuldigd en de vertikale met 4/7.

Als de tekst resoluties niet gelijk zijn, dan wordt eerst gekeken of de terminal in de juiste resolutie gezet kan worden. Is dit niet mogelijk dan wordt de terminal in de eerste hogere resolutie gezet. Kan ook dat niet dan wordt hij in de eerste lagere resolutie gezet. De regelbreedte wordt gelijk aan de regelbreedte van het document, zodat bij grotere resolutie de regels op de juiste plaats kunnen worden afgebroken en bij kleinere resolutie er gebruik gemaakt kan worden van scrolling.

Tot slot wordt de graphics faktor bepaald en kan het document worden weergegeven.



Figuur 6.4 Protocol voor ontvangst van een Telegraphics document.

6.3 Attributen voor Telegraphics

In de vorige paragraaf bleek dat er twee attributen nodig waren voor het afbeelden van Telegraphics. In de norm ISO 8613 wordt geen afbeeldingsproces gedefinieerd. In deel 2 van de norm staat het volgende over het afbeeldingsproces :

"The imaging process.

The imaging process is a locally defined process. Hence, apart from defining the input information required by this process, which is the format of the specific and generic structures, the formatted content and associated presentation styles, this process is not formally defined in this International standard. "

Dit betekent dat de structuur van een telegraphics document wel volgens deze standaard bepaald kan worden, maar niet hoe dit wordt afgebeeld.

Echter het doel van de norm wordt als volgt beschreven :

Scope

ISO 8613 provides for the interchange of documents for either or both of the following purposes :

- 1- to allow presentation as intended by the originator;
- 2- to allow processing, such as editing and reformatting;

Omdat voor het weergeven van telegraphics documenten zoals bedoeld door de zender het nodig is om wel een afbeeldingsproces te definiëren, moet de norm hiermee worden uitgebreid.

De norm moest in ieder geval uitgebreid worden met een beschrijving van de telegraphics codering. Dit wordt een nieuwe content architecture. In deze content architecture kan het meezenden van de beeldresolutie ook gedefinieerd worden als representatie attributen.

De volgende twee attributen worden gedefinieerd.
De manier van beschrijven van een attribuut in ODA staat op een volgende pagina.

Grafische resolutie :

Constituents : basic component descriptions

classification : non-mandatory, defaultable

Structure : two parameters , "horizontal resolution", "vertical resolution"

Permissible values : positive integer

default values : EGA waarden

Definition : Deze attributen geven de grafische resolutie aan van de display van de zender. Hiermee kan samen met de resolutie van de ontvangstdisplay een schalingsfactor worden berekend. De resolutie wordt in scherm pixels gegeven.

THEORY OF THE EARTH

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

The theory of the earth is a branch of geology which deals with the origin and development of the earth and its various parts. It is a science which seeks to explain the processes which have shaped the earth and its various parts.

Tekst resolutie

Constituents : basic component descriptions

classification : non-mandatory, defaultable

Structure : two parameters , "horizontal resolution", "vertical resolution"

Permissible values : positive integer

default values : EGA waarden

Definition : Deze attributen geven de tekst resolutie, horizontaal het aantal karakters per regel en vertikaal het aantal regels per scherm. De display aan ontvangstzijde kan in deze resolutie gezet worden.

6.4 Content Architecture voor Telegraphics

In de vorige paragrafen is duidelijk geworden, dat een document, dat Telegraphics bevat, ook voorzien moet zijn van attributen waarin de resolutie van de zender is opgenomen. In deze paragraaf zal de opzet van een content architecture (volgens ODA) worden gegeven van Telegraphics. Hierin zullen de attributen, die daarvoor nodig zijn worden opgegeven.

Een content architecture bestaat uit de volgende vier onderdelen :

- 1- Een beschrijving van de codering van de content portions. Voor Telegraphics is in hoofdstuk 2 een beschrijving gegeven van de codering.
- 2- Aspecten van het positioneren en afbeelden van een blok. Dat wil zeggen op welke manier en op welke plaats wordt een blok neergezet.
- 3- Het layout proces van de inhoud. Hoe wordt de inhoud in het blok gezet.
- 4- Presentatie en content portion attributen.

In ISO 8613 zijn twee grafische content architectures gedefinieerd : raster graphics en geometric graphics. Het layout proces en het positioneren en afbeelden worden in deze twee content architectures gegeven. Dit kan als uitgangspunt dienen voor de content architecture van telegraphics.

Het positioneren en afbeelden wordt in de Content Architecture van raster graphics op een andere manier gedaan dan in die van geometric graphics. Er zal een vergelijking tussen de twee worden gemaakt zodat de voor telegraphics meest geschikte methode kan worden gekozen. Dit wordt in paragraaf 6.4.1 behandeld.

Het layout proces van beide architectures is gelijk en kan dus ook voor telegraphics worden gebruikt, als er rekening wordt gehouden met de problemen die telegraphics met zich meebrengt.

De meeste attributen komen voort uit het positioneren en afbeelden en uit het layout proces. Verder zijn er voor telegraphics nog de attributen nodig voor de resolutie, die in de vorige paragraaf zijn beschreven. Alle attributen worden in paragraaf 6.4.3 op een rij gezet.

6.4.1 Positioneren en afbeelden van een blok.

In beide content architectures wordt een methode beschreven om de richting van een afbeelding te bepalen.

Raster graphics : In twee attributen (pel path en line progression) wordt aangegeven welke richting de x- respectievelijk de y-as hebben. Het attribuut pel path geeft de richting van de opeenvolgende pels (picture elements, punten) aan en wordt uitgedrukt in een hoek ten opzichte van de horizontale as van de pagina. Het attribuut line progression is de richting van opeenvolgende regels en wordt uitgedrukt als een hoek ten opzichte van pel path.

Een pel path van 0 graden en een line progression van 270 graden geeft de afbeelding op de normale manier. Punten worden van links naar rechts en van boven naar beneden ingevuld. Met behulp van deze attributen kan een afbeelding op acht verschillende manieren gedraaid worden afgebeeld.

Geometric graphics : Ook hier worden twee attributen gebruikt voor het bepalen van de draaiing. Region of interest en picture orientation. In region of interest wordt met twee coördinaten paren de rechthoek van de afbeelding aangegeven. De eerste geeft de linkeronderhoek, de tweede de rechterbovenhoek. Picture orientation geeft de draaiingshoek van de afbeelding aan. Er zijn vier mogelijkheden : 0, 90, 180 en 270 graden.

De resolutie van de afbeelding wordt ook op verschillende manieren bepaald.

Bij geometrische afbeeldingen wordt de 'fundamental measurement unit' bepaald, aan de hand van het verband tussen de region of interest en de dimensie van het basis layout object. De afbeelding wordt zo groot mogelijk in de beschikbare ruimte gezet.

Bij raster graphics wordt de resolutie bepaald door twee parameters, pel spacing en line spacing; deze geven aan hoe ver de punten (pels) uit elkaar liggen, horizontaal respectievelijk vertikaal.

De methode voor het aangeven van de richting van een afbeelding voor geometrische graphics sluit het beste aan bij Telegraphics. De resolutie wordt in aparte attributen opgegeven.

6.4.2 Het layout proces.

Voor het layout proces zijn de volgende attributen van belang:

image dimensions :

- minimum width, preferred width.** Deze geven minimum respektievelijk de gewenste breedte aan.
- minimum height, preferred height.** Deze geven de minimum respektievelijk de gewenste hoogte aan.
- aspect ratio flag :** kan ofwel fixed danwel variable zijn.
- automatic :** voor het automatisch bepalen van breedte en hoogte.

Er zijn nu vier mogelijkheden voor de layout :

- 1- Er wordt uitgegaan van de breedte. Alleen de minimum en gewenste breedte zijn opgegeven. De breedte wordt tussen deze twee waarden genomen. De hoogte is dan afhankelijk van de beschikbare ruimte en van de verhouding tussen hoogte en breedte van de figuur.
- 2- Er wordt uitgegaan van de hoogte. Alleen de minimum en de gewenste hoogte zijn opgegeven. De hoogte ligt tussen het minimum en de gewenste waarde. De breedte wordt bepaald door de beschikbare ruimte en de de verhouding tussen hoogte en breedte van de figuur.
- 3- Er wordt uitgegaan van de hoogte en de breedte. De image dimensions en de aspect ratio flag zijn opgegeven. De hoogte en de breedte moeten liggen tussen de opgegeven waarden. Als de aspect ratio flag 'fixed' is dan moet de aspect ratio (de verhouding tussen hoogte en breedte van de figuur) gelijk zijn aan de aspect ratio van de figuur. Verder moet het geheel natuurlijk passen in de beschikbare ruimte.
- 4- Het attribuut picture dimensions heeft de waarde automatic. De hoogte en de breedte worden automatisch bepaald. Eerst wordt de breedte gelijk gemaakt aan de beschikbare breedte. Dan wordt de hoogte bepaald door de verhouding tussen hoogte en breedte van de figuur.

Bovenstaande methode kan gebruikt worden als Telegraphics zonder tekst wordt gebruikt. Als er wel tekst wordt gebruikt dan moet de weergave geschieden zoals in paragraaf 6.2 is beschreven.

6.4.3 Attributen

Hier volgt een lijst met attributen, die nodig zijn in de content architecture van Telegraphics.

Region of interest : automatic
rectangle {VDC pair, VDC pair }

Picture orientation : 0, 90, 180, 270

Picture dimensions :

```

picture width
    minimum
    preferred
picture height
    minimum
    preferred
picture size
    minimum width
    preferred width
    minimum height
    preferred height
    aspect ratio flag
automatic
terminal resolution originator
    text resolution
        width
        height
    graphics resolution
        width
        height

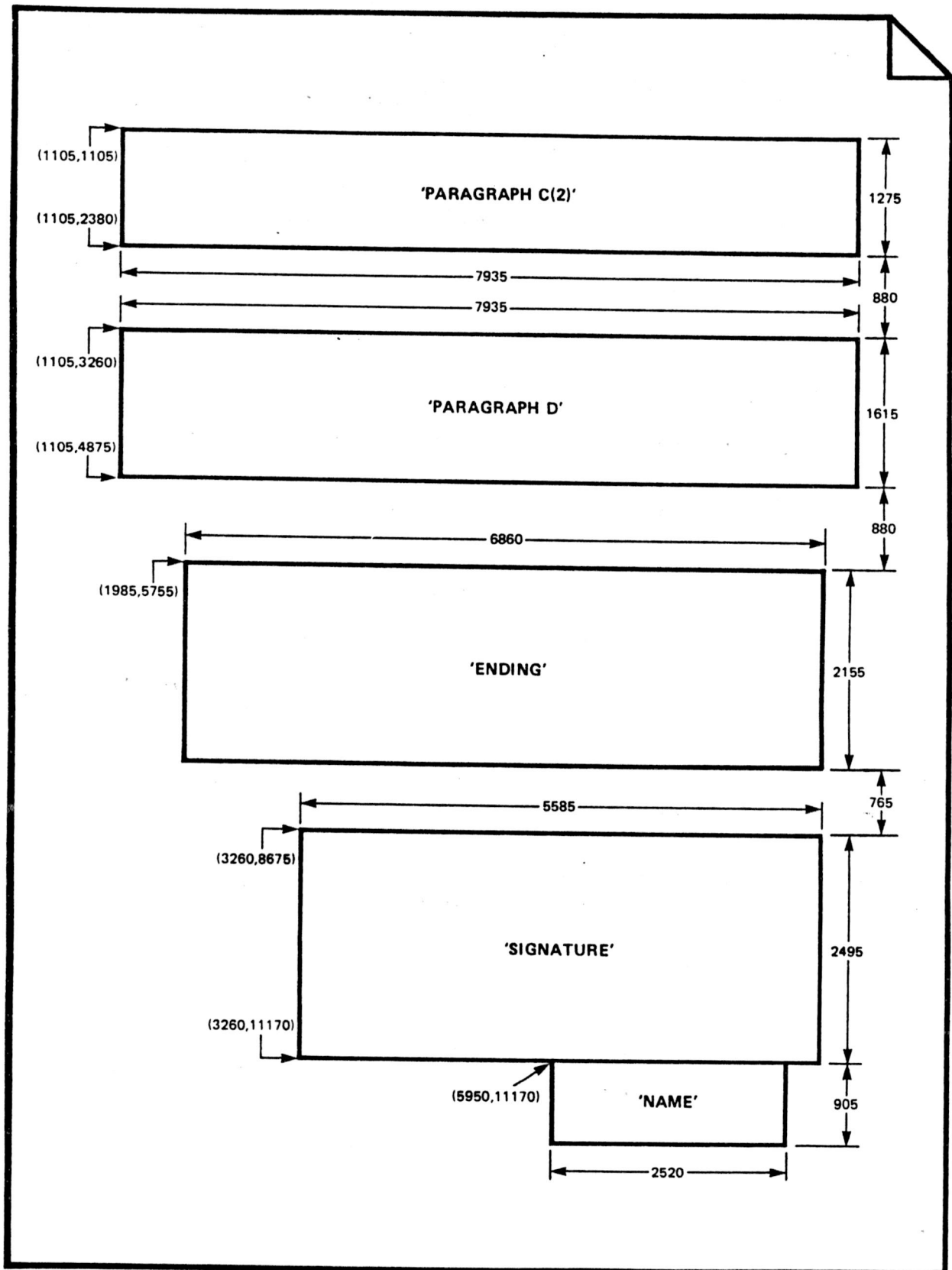
```

6.5 Voorbeeld van een Telegraphics document

Tot slot van dit hoofdstuk een voorbeeld van een document met Telegraphics. Dit voorbeeld is een aangepaste versie van het voorbeeld dat in de norm ISO 8613 (Deel 2 en 5) wordt gegeven. Het oorspronkelijke voorbeeld bestaat uit een brief van drie bladzijden. In het voorbeeld hier wordt daarvan de laatste bladzijde gebruikt met een paar wijzigingen. Een stuk tekst wordt gemengd met telegraphics en verder wordt een stuk raster graphics telegraphics.

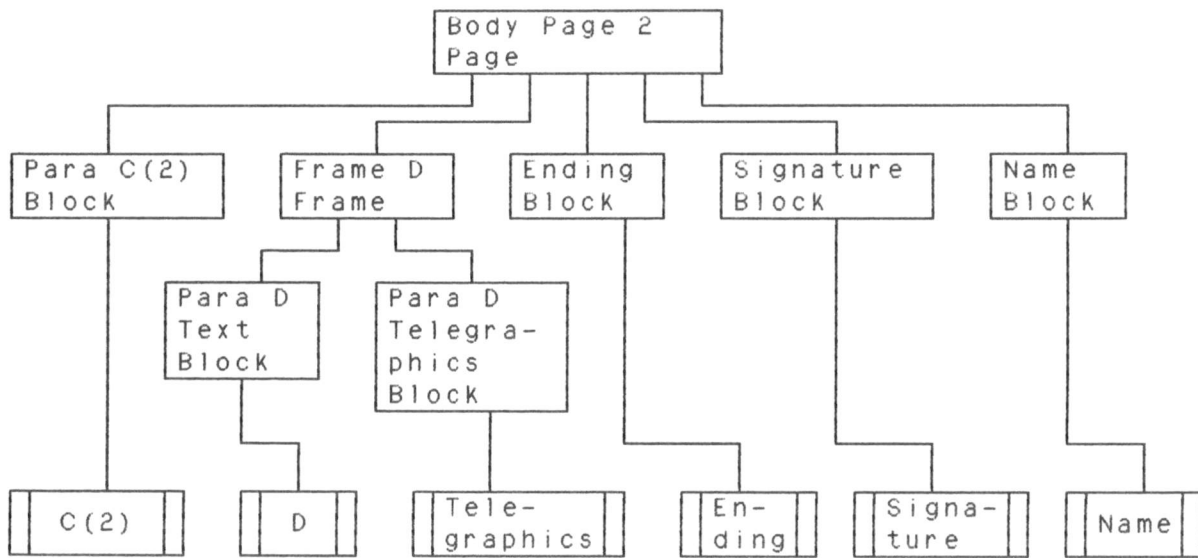
In figuur 6.5 op de volgende bladzijde staat de brief.

In de figuur hieronder staat de layout.



SECOND BODY PAGE

Figuur 6.6. De layout van het document.



Figuur 6.7. De specifieke layout structuur.

Figuur 6.7 geeft aan hoe de specifieke layout structuur is. Hij bestaat uit een pagina, uit vier blokken en een frame. Het frame bevat de paragraaf tekst met Telegraphics, en bestaat uit twee blokken. Onderaan de blokken zijn de content portions aangegeven.

Hieronder wordt de specifieke layout structuur verder uitgewerkt. Behalve de namen en type van de objecten wordt ook de plaats aangegeven. Verder worden de attributen aangegeven, die bij een object horen.

Object Type	PAGE
Object Identifier	1 2
User Visible Name	"Body Page 2"
Dimensions	W=9920, H=14030
Subordinates	0, 1, 2, 3, 4

Object Type	BLOCK
Object Identifier	1 2 0
User Visible Name	"Para C(2)"
Position	X=1105, Y=1105
Dimensions	W=7935, H=1275
Line Spacing	300
Alignment	JUSTIFIED
Content Portions	0

Content Identifier - Layout	1 2 0 0
Content Information	/* Formatted string of C's */

Object Type	FRAME
Object Identifier	1 2 1
User Visible Name	"Frame D"
Position	X=1105, Y=3260
Dimensions	W=7935, H=1615
Subordinates	0, 1

Object Type	BLOCK
Object Identifier	1 2 1 0
User Visible Name	"Para D text"
Position	X=1105, Y=3260
Dimensions	W=7935, H=1615
Line Spacing	300
Alignment	JUSTIFIED
Content Portions	0

Content Identifier - Layout 1 2 1 0 0
 Content Information /* Formatted string of D's */

Object Type	BLOCK
Object Identifier	1 2 1 1
User Visible Name	"Para D telegraphics"
Position	X=1105, Y=3260
Dimensions	W=7935, H=1615
Text Resolution	W=80, H=25
Graphics Resolution	W=512, H=200
Content Portions	0

Content Identifier - Layout 1 2 1 1 0
 Content Information /* Telegraphics */

Object Type	BLOCK
Object Identifier	1 2 2
User Visible Name	"Para Ending"
Position	X=1985, Y=5755
Dimensions	W=6860, H=2155
Line Spacing	300
Alignment	JUSTIFIED
Content Portions	0

Content Identifier - Layout 1 2 2 0
 Content Information /* Formatted string of
 FORMAL ENDING */

Object Type	BLOCK
Object Identifier	1 2 3
User Visible Name	"Signature"
Position	X=3260, Y=8675
Dimensions	W=5585, H=2495
Content Architecture Class	TELEGRAPHICS
Content Portions	0

Content Identifier - Layout 1 2 3 0
Content Information /* Telegraphics */

Object Type	BLOCK
Object Identifier	1 2 4
User Visible Name	"Name"
Position	X=5950, Y=11170
Dimensions	W=2520, H=905
Line Spacing	300
Content Portions	0

Content Identifier - Layout 1 2 4 0
Content Information "Miss Aude HEA\nDocument Architect"

Tot slot wordt in formele notatie volgens ASN.1 het document gegeven. Erachter staat de codering hexadecimaal weergegeven. De lengte is steeds dikgedrukt weergegeven. De bepaling van de code van de identifiers gaat aan de hand van Appendix A.

Voor Telegraphics zijn de volgende dataformaten gebruikt :

Aan de presentation attributes is toegevoegd :

telegraphics [6] IMPLICIT Telegraphics-Attribute OPTIONAL.

Telegraphics-Attributes zijn als volgt gedefinieerd:

```
Telegraphics-Attributes ::= SET {
    region of interest      [0] IMPLICIT Region of Interest
                             OPTIONAL
    picture orientation     [1] IMPLICIT Picture Orientation
                             OPTIONAL
    picture dimensions      [2] IMPLICIT Picture Dimensions
                             OPTIONAL
    terminal resolution     [3] IMPLICIT Terminal Resolution
    originator              [3] IMPLICIT Terminal Resolution
                             Originator OPTIONAL
}
```

Hiervan zijn Region of Interest, Picture Orientation en Picture Dimensions in Appendix A gegeven. Terminal Resolution Originator wordt als gedefiniëerd:

```
Terminal Resolution Originator ::= IMPLICIT SET OF {  
  text resolution width      [0] IMPLICIT INTEGER  
  text resolution height     [1] IMPLICIT INTEGER  
  graphics resolution width  [2] IMPLICIT INTEGER  
  graphics resolution height [3] IMPLICIT INTEGER}
```

```

layout-object{
  object-type page,
  descriptor-body{
    object-identifier " 1 2",
    subordinates{
      "0","1","2","3","4"},

    dimensions {
      horizontal fixed 9920,
      vertical fixed 14030},
    user-visible-name "Body Page 2"}},

layout-object{
  object-type block,
  descriptor-body {
    object-identifier "1 2 0",

    content-portions {"0"},
    position {
      horizontal fixed 1105,
      vertical fixed 1105},
    dimensions {
      horizontal fixed 7935,
      vertical fixed 1275},
    presentation-attributes {
      character-attributes {
        line-spacing 300,
        alignment justified}},
    user-visible-name "Para C(2)"}},

content-portion {
  content-portion-attributes {
    content-identifier-layout "1 2 0 0"},

  content-information "CCCCC...."},

layout-object {
  object-type frame ,
  descriptor-body {
    object-identifier "1 2 1",

    subordinates { "0", "1"},

    position {
      horizontal fixed 1105,
      vertical fixed 3260},
    dimensions {
      horizontal fixed 7935,
      vertical fixed 1615},
    user-visible-name "Frame D"}},

layout-object {
  object-type block,
  descriptor-body {
    object-identifier "1 2 1 0",

    content-portions {"0"},
    position {

```

```

A2 32
02 01 02
31 2D
41 03 31 20 31
A0 0F
12 01 30 12 01
31 12 01 32 12
01 33 12 01 34
A4 08
80 02 26 C0
80 02 36 CE
8E 0B 42 6F 64
79 20 50 61 67
65 20 32
A2 3B
02 01 04
31 36
41 05 31 20 32
20 30
A1 03 12 01 30
A3 08
80 02 04 51
80 02 04 51
A4 08
80 02 1E FF
80 02 04 FB
A6 09
A0 07
87 02 01 2C
88 01 03
8E 09 50 61 72
61 20 43 28 32
29
A3 LL
31 09
40 07 31 20 32
20 30 20 30
14 LL 43 43 43
43 43 ....
A2 2E
02 01 03
31 29
41 05 31 20 32
20 31
A0 06 12 01 30
12 01 31
A3 08
80 02 04 51
80 02 0C BC
A4 08
80 02 1E FF
80 02 06 4F
8E 07 46 72 61
6D 65 20 44
A2 3F
02 01 04
31 3A
41 07 31 20 32
20 31 20 30
A1 03 12 01 30
A3 08

```



```

horizontal fixed 1105,
vertical fixed 3260},
dimensions {
horizontal fixed 7935,
vertical fixed 1615},
presentation-attributes {
character-attributes {
line-spacing 300,
alignment justified}
user-visible-name "Para D text"}}},
80 02 04 51
80 02 0C BC
A4 08
80 02 1E FF
80 02 06 4F
A6 09
A0 07
87 02 01 2C
88 01 03
8E 0B 50 31 72
61 20 44 20 74
65 78 74
content-portion {
content-portion-attributes {
content-identifier-layout "1 2 1 0 0"},
40 09 31 20 32
20 31 20 30 20
30
content-information "DDDDD...."},
14 LL 44 44 44
44 44 ....
layout-object {
object-type block,
descriptor-body {
object-identifier "1 2 1 1",
41 07 31 20 32
20 31 20 31
content-portions {"0"},
A1 03 12 01 30
position {
A3 08
horizontal fixed 1105,
80 02 04 51
vertical fixed 3260},
80 02 0C BC
dimensions {
A4 08
horizontal fixed 7935,
80 02 1E FF
vertical fixed 1615},
80 02 06 4F
presentation-attributes {
A6 11
telegraphics-attributes {
A6 0F
terminal-resolution-originator {
A3 OD
text-resolution width 80
80 01 50
text-resolution height 25,
81 01 19
graphics-resolution width 512
82 02 02 00
graphics-resolution height 200 }}
83 01 C8
user-visible-name "Para D Telegraphics"},
8E 13 50 61 72
61 20 44 20 54
65 6C 65 67 72
61 70 68 6A 63
73
content-portion {
A3 LL
content-portion-attributes {
31 0B
content-identifier-layout "1 2 1 1 0"},
40 09 31 20 32
20 31 20 31 20
30
content-information "Telegraphics"},
04 LL ....
layout-object{
A2 3D
object-type block,
02 01 04
descriptor-body {
31 38
object-identifier "1 2 2",
41 05 31 20 32
20 32
content-portions {"0"},
A1 03 12 01 30
position {
A3 08
horizontal fixed 1985,
80 02 07 C1
vertical fixed 5755},
80 02 16 7B
dimensions {
A4 08
horizontal fixed 6860,
80 02 1A CC
vertical fixed 2155},
80 02 08 6B

```

```

presentation-attributes {
  character-attributes {
    line-spacing 300,
    alignment justified}},
user-visible-name "Para Ending"}},

content-portion {
  content-portion-attributes {
    content-identifier-layout "1 2 2 0"},

  content-information "ENDING ..."},

layout-object{
  object-type block,
  descriptor-body {
    object-identifier "1 2 3",

    content-portions {"0"},
    position {
      horizontal fixed 3260,
      vertical fixed 8675},
    dimensions {
      horizontal fixed 5585,
      vertical fixed 2495},
    presentation-attributes {
      content-architecture-class
      {oda-cont-arch-class telegraphics}},
    user-visible-name "Signature"}},

content-portion {
  content-portion-attributes {
    content-identifier-layout "1 2 3 0",

    content-information 'ZZZZ ...ZZZ'H},

layout-object{
  object-type block,
  descriptor-body {
    object-identifier "1 2 4",

    content-portions {"0"},
    position {
      horizontal fixed 5950,
      vertical fixed 11170},
    dimensions {
      horizontal fixed 2520,
      vertical fixed 905},
    presentation-attributes {
      character-attributes {
        line-spacing 300}},
    user-visible-name "Name"}},

content-portion {
  content-portion-attributes {
    content-identifier-layout "1 2 4 0"},

  content-information "Miss AudeHEA
  Document Architect"}} 73 ...

```

```

A6 09
A0 07
87 02 01 2C
88 01 03
8E 0B 50 61 72
61 20 45 6E 64
69 6E 67
A3 LL
31 09
40 07 31 20 32
20 32 20 30
14 LL 45 4E 44
49 4E 47...
A2 37
02 01 04
31 32
41 05 31 20 32
20 33
A1 03 12 01 30
A3 08
80 02 0C BC
80 02 21 E3
A4 08
80 02 15 D1
80 02 09 BF
A6 05
06 03
07 00 1E
8E 09 53 69 67
6E 61 74 75 72
65
A3 LL
31 0F
40 07 31 20 32
20 33 20 30
04 LL ...
A2 33
02 01 04
31 2E
41 05 31 20 32
20 34
A1 03 12 01 30
A3 08
80 02 17 3E
80 02 2B A2
A4 08
80 02 09 D8
80 02 03 89
A6 06
A0 04
87 02 01 2C
8E 04 4E 61 6D
65
A3 LL
31 09
40 07 31 20 32
20 34 20 30
14 LL 4D 69 73

```

7 CONCLUSIES

Voor het verzenden van documenten tussen niet homogene systemen, moeten de documenten een logische structuur bezitten om goed weergegeven te kunnen worden. Als een document opgebouwd is volgens logische structuren, dan maakt het niet uit op welk presentatie systeem het document wordt afgebeeld.

Daarentegen moet er voor Telegraphics vastgelegd worden met welke resolutie een document is gemaakt zodat de ontvanger de zelfde resolutie gebruikt voor het afbeelden van het document. Als dit niet mogelijk is dan moet er van 'sideways scrolling' gebruik worden gemaakt.

De ODA document structuur kan gebruikt worden voor het maken van Telegraphics documenten. De codering gaat dan volgens ODIF. Aldus gemaakte documenten kunnen over een Message Handling System worden verzonden, dat voldoet aan X.400.

REFERENTIES

- 1- Arnbak, J.C.; Bons, J.H.; Vieveen, J.W. "Graphical correspondence in electronic-mail networks using personal computers", july 1988, to be published in IEEE Journal on Selected Areas in Communications (JSAC).
(TU-Bibliotheek : no 5935)
- 2- Vervest, Peter. *Electronic Mail and Message Handling*. 1985, Francis Pinter (Publishers), London.
(TU-Bibliotheek : no 2136 5292)
- 3- Vervest, Peter. *Innovation in Electronic Mail, towards open information networks - perspectives on innovation policy*. Proefschrift, 1986, Eburon.
(TU-Bibliotheek : no 2167 6022)
- 4- Recommendations of the CCITT, Series X.400, (Message Handling Systems), Red Books, ITU, Geneva, 1984
- 5- Campbell-Grant, I.R. "Piloting the Office Document Architecture". *IEEE International Conference on Communications '86. ICC '86: 'Integrating the World Through Communications'*. Conference Record. Toronto, Canada. 22-25 June 1986. Vol 3, P 1419-1423.
(TU-Bibliotheek : no 1202)
- 6- Skogseth, Gunn; Verne, Guri. "Requirements in a multimedia document system". *Computer Message Systems -85* R. Uhlig (editor), Elsevier Science Publishers B.V. (North Holland), pg 221-233.
- 7- Naffah, Najah. "Design Issues of presentation protocols in office systems". *National Telecommunications Conference, NTC '80, Houston, Texas*. 1980, IEEE, pg 16.4.1 - 16.4.6.
(TU-Bibliotheek : no 1010)
- 8- Naffah, Najah. "Editing multitype document". *Office Information Systems, Proceedings of the second international workshop on Office Information Systems, Couvent Royale de Saint-Maximin, France, 13-15 October 1981, edited by N.Naffah*, North Holland Publishing Company, 1982, pg 125 - 148.
(TU-Bibliotheek : no 1713 6389)
- 9- Frantz, G.; Karmouch, A.; Naffah, N. "The Agora message system architecture". *Networks from the user's point of view. Proceedings of the IFIP TC-6 working conference COMNET '81, edited by L. Csaba, T.Szentivanyi, K. Tamay*, North Holland Publishing Company, 1981, pg 575 - 589.
(TU-Bibliotheek : no 1690 6298)
- 10- Naffah, Najah; Karmouch, Ahmed. "Agora- An experiment in multimedia message systems". *Computer, May 1986*, pg 56 - 66.
(TU-Bibliotheek : no 208E)
- 11- Naffah, Najah; "High level protocol for alphanumeric data-entry terminals". *Computer Networks, vol 2, 1978*, pg 84 - 94.
(TU-Bibliotheek : no 3539)

- 12- Finn, G.; Postel, J. "Data structures and presentaiton control for multimedia computer mail". *EASCON 81: Electronics and aerospace systems conference, Washington, Nov 16-18, 1981*, IEEE, New York, 1981, pg 358 - 364.
- 13- Reynolds, Joyce K.; Postel, Jonathan B.; Katz, Alan R.; Finn, Gregg G.; DeSchon, Annette L. "The DARPA Experimental Multimedia mail system". *Computer, October, 1985* pg 82 - 89.
(TU-Bibliotheek : no 208E)
- 14- Garcia Luna Aceves, J.J.; Poggio, A. "Multimedia message content protocols for computer mail". *Computer-Based Message Services, Proceedings of the IFIP WG-6.5 working conference on computer-based message services, Nottingham, England, May 1984, edited by H.T. Smith*, Elsevier Science Publishers B.V. (North Holland), 1984, pg 87 - 98.
(TU-Bibliotheek : no 2138 6121)
- 15- Poggio, A.; Garcia Luna Aceves, J.J.; Craighill, E.J.; Moran, D.; Aguilar, L.; Worthington, D.; Hight, J. "CCWS: a computer-based, multimedia information system". *Computer, October, 1985*, pg 92 - 103.
(TU-Bibliotheek : no 208E)
- 16- Katz, Alan R. "An experimental internetwork multimedia mail system". *Computer-Based Message Services, Proceedings of the IFIP WG-6.5 working conference on computer-based message services, Nottingham, England, May 1984, edited by H.T. Smith*, Elsevier Science Publishers B.V. (North Holland), 1984, pg 115 - 123.
(TU-Bibliotheek : no 2138 6121)
- 17- Thomas, Robert H.; Forsdick, Harry C.; Crowley, Terrence R.; Schaaf, Richard W.; Tomlinson, Raymond S.; Travers, Virginia M. "Diamond : A multimedia message system built on a distributed architecture". *Computer, december, 1985* pg 65 - 77.
(TU-Bibliotheek : no 208E)
- 18- Forsdick, Harry; Thomas, Robert; Robertson, George; Travers, Virginia. "Initial experience with multimedia documents in Diamond". *Computer-Based Message Services, Proceedings of the IFIP WG-6.5 working conference on computer-based message services, Nottingham, England, May 1984, edited by H.T. Smith*, Elsevier Science Publishers B.V. (North Holland), 1984, pg 99 - 113.
(TU-Bibliotheek : no 2138 6121)
- 19- Forsdick, Harry. "Explorations into real-time multimedia conferencing". *Computer Message Systems -85* R. Uhlig (editor), Elsevier Science Publishers B.V. (North Holland), pg 331-347.
- 20- Christodoulakis, S.; Faloutsos, C. "A multimedia document server". *IEEE AES Magazine, November 1986* pg 2 - 9,
(TU-Bibliotheek no : 6812)
en ."Design and performance considerations for a multimedia document server". *First international conference on supercomputing systems SCS 85, St. Petersburg, Florida, USA, December 16-20, 1985* pg 441 - 449.
(TU-Bibliotheek : no 2186 2106)

- 21- Sakata, Shiro; Ueda, Tetsuo. "A distributed interoffice mail system". *Computer, October, 1985*, pg 106 - 116.
(TU-Bibliotheek : no 208E)
- 22- Campbell-Grant, I.R. "Open systems application layer standards for text and office systems". *International conference on networks and electronic office systems NEOS, London, 17-19 Sept 1985*, The Institute of Electronic and Radio Engineers (IERE), pg 129 - 135.
(TU-Bibliotheek : no 2196 1490)
- 23- Horak, W.; Krönert, G. "Document editing and entry based on the office document architecture standard ECMA-101". *IEEE International Conference on Communications '86. ICC '86: 'Integrating the world through communications'. Conference record. Toronto, Canada, 22-25 June 1986* Vol 3 pg 1520 - 1524.
(TU-Bibliotheek : no 1202)
- 24- Smith, Joan. "Formatting standards for document interchange". *Computer Communications (GB, Guildford)*. Vol 7, no 4, Aug 1984 : 171 - 180
(TU-Bibliotheek : no 4201)
- 25- Zeckendorf, L.J. "Kantoorautomatisering en verder ...7, De ISO/ECMA/CCITT- documentarchitectuur". *Informatie (Nederland)*.
Vol 29, no 1, 1987 : 4 - 9
(TU-Bibliotheek : no 398A)
- 26- Horak, Wolfgang; Lange, Kerstin. "Interchange of mixed-mode documents over ISDN". *Telcom Report (West Germany, Berlin)*.
Vol 10, no 5, 1987 : 286 - 292
(TU-Bibliotheek : no 4473, engels; no 4225, duits)
- 27- Campbell-Grant, Ian R.; Robinson, Peter J. "An introduction to ISO DIS 8613, "Office document architecture", and its application to computer graphics". *Computers & Graphics (GB, Oxford)*. Vol 11, no 4, 1987 : 325 - 341
(TU-Bibliotheek : no 3513)
- 28- ISO DIS 8613. Information Processing - Text and office systems - Office Document Architecture (ODA) and Interchange format - part 1 - 8, 1987.
- 29- Van der Meer, J.M.; Oude Lohuis, H.G. *SDL, Specification and Description Language*, Vakgroep Telecommunicatie- en Verkeersbegeleidingssystemen, Laboratorium voor Automatische Verkeerssystemen, Faculteit der Elektrotechniek, TU Delft, 1987
- 30- CCITT Draft Recommendations T.150 "Telewriting Terminal Equipment" , Version 8, Study group VIII, Febr 1988

APPENDIX A

Een document, dat gestructureerd is volgens ODA, zoals in hoofdstuk 5 is behandeld, wordt gerepresenteerd als een data stroom, die uit één of meer data structuren van de volgende typen bestaat:

- document profile descriptor;
- layout object dscriptor;
- layout object class descriptor;
- logical object descriptor;
- logical object class descriptor;
- presentation style descriptor;
- layout style descriptor;
- text unit.

In deze data stroom worden de data elementen geordend volgens bepaalde regels. Deze regels worden hier in de appendix gegeven. De notatie komt overeen met de notatie, die in X.409 wordt gebruikt voor het definiëren van een type (zie paragraaf 3.3). Met behulp van deze lijst kan de codering gemaakt worden van een ODA document. Hiervoor worden de coderingsregels gebruikt van ASN 1, hetgeen ook in paragraaf 3.3 is beschreven.

Deze zogenaamde data formaten zijn afkomstig uit de norm ISO DIS 8613. Het eerste gedeelte (5.1 t/m 5.8, blz. 56 t/m blz 73) heeft betrekking op algemene elementen en op logische en layout structuur elementen. Deze komen uit deel 5, hoofdstuk 5 van de norm.

Het tweede gedeelte komt uit deel 6, hoofdstuk 9 van de norm en geeft die elementen, die in de character content architecture zijn gedefiniëerd (blz. 74 t/m 77).

Het derde gedeelte komt uit deel 7, hoofdstuk 8 van de norm en geeft de elementen, die in de raster graphics content architecture zijn gedefiniëerd (blz. 78 t/m 80).

Het laatste gedeelte komt uit deel 8, hoofdstuk 9 van de norm en geeft de elementen, die in de geometric graphics content architecture zijn gedefiniëerd (blz. 81 t/m 85).

5 Data formats

5.1 Interchange data units

```
Interchange-Data-Unit ::= CHOICE {
    document-profile           [0] IMPLICIT Document-Profile-Descriptor,
    layout-object-class        [1] IMPLICIT Layout-Class-Descriptor,
    layout-object              [2] IMPLICIT Layout-Object-Descriptor,
    content-portion            [3] IMPLICIT Text-Unit,
    logical-object-class        [5] IMPLICIT Logical-Class Descriptor,
    logical-object              [6] IMPLICIT Logical-Object-Descriptor,
    presentation-style         [7] IMPLICIT Presentation-Style-Descriptor,
    layout-style               [8] IMPLICIT Layout-Style-Descriptor}
X
```

5.2 Document profile descriptor

```
Document-Profile-Descriptor ::= SET {
    generic-layout-structure    [0] IMPLICIT NumericString OPTIONAL,
    specific-layout-structure   [1] IMPLICIT NumericString OPTIONAL,
    generic-logical-structure   [4] IMPLICIT NumericString OPTIONAL,
    specific-logical-structure  [5] IMPLICIT NumericString OPTIONAL,
    presentation-styles         [6] IMPLICIT NumericString OPTIONAL,
    layout-styles               [7] IMPLICIT NumericString OPTIONAL,

    -- each of these numeric strings consists of a single character;
    -- 'partial' is represented by "0", and 'complete' or 'present'
    -- is represented by "1".

    external-document-class     [9] IMPLICIT OBJECT IDENTIFIER OPTIONAL,
    resource-document           [10] IMPLICIT SET {
        title                    [0] IMPLICIT Character-Data,
        owner                    [1] IMPLICIT Character-Data} OPTIONAL,
    resources                   [11] IMPLICIT SET OF SET {
        resource-identifier      INTEGER,
        object-class-identifier Object-or-Class-Identifier} OPTIONAL,
    document-characteristics    [2] IMPLICIT Document-Characteristics
                                OPTIONAL,
    document-management-attributes [3] IMPLICIT Document-Management-Attributes
                                OPTIONAL}

```

```

Document-Characteristics ::= SET {
  application-profile          CHOICE {
                                [0] IMPLICIT INTEGER {
                                      group-4-facsimile (2),
                                      mixed-mode (3)},
                                [4] IMPLICIT OBJECT IDENTIFIER} OPTIONAL,
  application-profile-defaults [10] IMPLICIT Application-Profile-Defaults
                                OPTIONAL,
  document-architecture-class [1] IMPLICIT INTEGER {
                                formatted (1),
                                processable (2),
                                formatted-processable (3)} OPTIONAL,
  content-architecture-classes [5] IMPLICIT SET OF OBJECT IDENTIFIER
                                OPTIONAL,
  interchange-format-class     [6] IMPLICIT INTEGER {
                                if-a (0),
                                if-b (1)} OPTIONAL,
  oda-version                  [8] IMPLICIT Character-Data OPTIONAL,
  non-basic-doc-characteristics [2] IMPLICIT Non-Basic-Doc-Characteristics
                                OPTIONAL,
  non-basic-struc-characteristics [3] IMPLICIT Non-Basic-Struc-Characteristics
                                OPTIONAL,
  additional-doc-characteristics [9] IMPLICIT Additional-Doc-Characteristics
                                OPTIONAL}

```

```

Application-Profile-Defaults ::= SET {
  document-architecture-defaults [0] IMPLICIT Document-Architecture-Defaults
                                OPTIONAL,
  character-content-defaults      [1] IMPLICIT Character-Content-Defaults
                                OPTIONAL,
  raster-gr-content-defaults      [2] IMPLICIT Raster-Gr-Content-Defaults
                                OPTIONAL,
  geo-gr-content-defaults         [3] IMPLICIT Geo-Gr-Content-Defaults
                                OPTIONAL}

```

```

-- Character-Content-Defaults is defined in part 6
-- Raster-Gr-Content-Defaults is defined in part 7
-- Geo-Gr-Content-Defaults is defined in part 8
-- the following tags are reserved for additional types
-- of content defaults:
-- [4] composite-graphics
-- [5] audio
-- [6] dynamic-graphics

```

```

Document-Architecture-Defaults ::= SET {
  content-architecture-class [0] IMPLICIT Content-Architecture-Class
                                OPTIONAL,
  content-type               [1] IMPLICIT Content-Type OPTIONAL,
  page-dimensions            [2] IMPLICIT Measure-Pair OPTIONAL,
  transparency                [3] IMPLICIT Transparency OPTIONAL,
  colour                      [4] IMPLICIT Colour OPTIONAL,
  layout-path                 [5] IMPLICIT One-Of-Four-Angles OPTIONAL,
  medium-type                 [6] IMPLICIT Medium-Type OPTIONAL,
  block-alignment             [7] IMPLICIT Block-Alignment OPTIONAL}

```

Non-Basic-Doc-Characteristics ::= SET (
 profile-character-sets [5] IMPLICIT OCTET STRING OPTIONAL,
 comments-character-sets [1] IMPLICIT OCTET STRING OPTIONAL,
 alternative-character-sets [6] IMPLICIT OCTET STRING OPTIONAL,

X

— each of these octet strings represents a string of escape sequences

page-dimensions [2] IMPLICIT SET OF Measure-Pair OPTIONAL,
 medium-types [8] IMPLICIT SET OF Medium-Type OPTIONAL,
 layout-paths [21] IMPLICIT SET OF One-Of-Four-Angles
 OPTIONAL,
 transparency [22] IMPLICIT SET OF Transparency OPTIONAL,
 protection [23] IMPLICIT SET OF INTEGER OPTIONAL,
 block-alignment [24] IMPLICIT SET OF Block-Alignment
 OPTIONAL,
 fill-order [25] IMPLICIT SET OF Fill-Order OPTIONAL,
 mixed-presentation-attributes [4] IMPLICIT SET OF
 Mixed-Presentation-Attribute OPTIONAL,
 char-presentation-attributes [9] IMPLICIT SET OF
 Char-Presentation-Attribute OPTIONAL,
 ra-gr-presentation-attributes [10] IMPLICIT SET OF
 Ra-Gr-Presentation-Attribute OPTIONAL,
 geo-gr-presentation-attributes [12] IMPLICIT SET OF
 Geo-Gr-Presentation-Attribute OPTIONAL,

-- Char-Presentation-Attribute is defined in part 6
 -- Ra-Gr-Presentation-Attribute is defined in part 7
 -- Geo-Gr-Presentation-Attribute is defined in part 8
 -- the following tags are reserved for additional types
 -- of presentation attributes:
 — [13] composite-graphics
 -- [14] audio
 — [15] dynamic-graphics

character-coding-attributes [16] IMPLICIT SET OF
 Character-Coding-Attribute OPTIONAL,
 ra-gr-coding-attributes [3] IMPLICIT SET OF
 Ra-Gr-Coding-Attribute OPTIONAL,
 geo-gr-coding-attributes [17] IMPLICIT SET OF
 Geo-Gr-Coding-Attribute OPTIONAL)

-- Character-Coding-Attribute is defined in part 6
 -- Ra-Gr-Coding-Attribute is defined in part 7
 -- Geo-Gr-Coding-Attribute is defined in part 8
 -- the following tags are reserved for additional types
 -- of coding attributes:
 — [18] composite-graphics
 -- [19] audio
 — [20] dynamic-graphics

```

Mixed-Presentation-Attribute ::= CHOICE {
  character-path           [0] IMPLICIT One-Of-Four-Angles,
  character-line-progression [1] IMPLICIT One-Of-Two-Angles,
  character-orientation    [2] IMPLICIT One-Of-Four-Angles,
  character-spacing        [5] IMPLICIT INTEGER,
  line-spacing             [6] IMPLICIT INTEGER,
  alignment                [7] IMPLICIT Alignment,
  graphic-rendition        [8] IMPLICIT Graphic-Rendition,
  pel-path                 [9] IMPLICIT One-Of-Four-Angles,
  raster-gr-line-progression [10] IMPLICIT One-Of-Two-Angles,
  pel-transmission-density [11] IMPLICIT Pel-Transmission-Density}

```

```

-- Alignment and Graphic-Rendition are defined in part 6
-- Pel-Transmission-Density is defined in part 7
-- One-Of-Two-Angles and One-Of-Four-Angles are defined
-- both in part 6 and in part 7

```

```

Non-Basic-Struc-Characteristics ::= SET {
  number-of-objects-per-page [0] IMPLICIT INTEGER OPTIONAL}

```

```

Additional-Doc-Characteristics ::= SET {
  unit-scaling [3] IMPLICIT SEQUENCE {INTEGER,INTEGER}
                                     OPTIONAL,
  fonts-list [2] IMPLICIT Fonts-List OPTIONAL}

```

```

Fonts-List ::= SET OF SET {
  font-identifier INTEGER,
  font-description Font-Description}

```

```

-- Font-Description is defined in ISO 9541-5

```

```

Document-Management-Attributes ::= SET {
  document-description [7] IMPLICIT Document-Description OPTIONAL,
  dates-and-times [0] IMPLICIT Dates-and-Times OPTIONAL,
  originators [1] IMPLICIT Originators OPTIONAL,
  other-user-information [2] IMPLICIT Other-User-Information
                                     OPTIONAL,
  external-references [3] IMPLICIT External-References OPTIONAL,
  filing-and-retrieval [4] IMPLICIT Filing-and Retrieval OPTIONAL,
  content-attributes [5] IMPLICIT Content-Attributes OPTIONAL,
  security-attributes [6] IMPLICIT Security-Attributes OPTIONAL}

```

```

Document-Description ::= SET {
  title [0] IMPLICIT Character-Data OPTIONAL,
  subject [1] IMPLICIT Character-Data OPTIONAL,
  document-type [2] IMPLICIT Character-Data OPTIONAL,
  abstract [3] IMPLICIT Character-Data OPTIONAL}

```

```

X Character-Data ::= [APPLICATION 3] IMPLICIT OCTET STRING

```

```

-- string of characters from the sets designated by the attribute
-- "profile character sets", plus carriage return and line feed

```

```

Dates-and-Times      ::= SET {
    document-date-and-time      [0] IMPLICIT Date-and-Time OPTIONAL,
    creation-date-and-time      [1] IMPLICIT Date-and-Time OPTIONAL,
    local-filing-date-and-time  [2] IMPLICIT SEQUENCE OF Date-and-Time
                                   OPTIONAL,
    expiry-date-and-time        [3] IMPLICIT Date-and-Time OPTIONAL,
    start-date-and-time         [4] IMPLICIT Date-and-Time OPTIONAL,
    purge-date-and-time         [5] IMPLICIT Date-and-Time OPTIONAL,
    release-date-and-time       [6] IMPLICIT Date-and-Time OPTIONAL,
    revision-history            [7] IMPLICIT SEQUENCE OF SET {
        revision-date-and-time  [0] IMPLICIT Date-and-Time OPTIONAL,
        version-number          [1] IMPLICIT NumericString OPTIONAL,
        revisors                [2] IMPLICIT SET OF SET {
            names                [0] IMPLICIT SET OF Personal-Name
                                   OPTIONAL,
            position             [1] IMPLICIT Character-Data
                                   OPTIONAL,
            organization          [2] IMPLICIT Character-Data
                                   OPTIONAL)
        OPTIONAL,
    local-filing-reference      [3] IMPLICIT Character-Data OPTIONAL,
    user-comments               [4] IMPLICIT Character-Data OPTIONAL)
    OPTIONAL)

```

```

Originators      ::= SET {
    organizations      [0] IMPLICIT SET OF Character-Data OPTIONAL,
    preparers          [1] IMPLICIT SEQUENCE OF SET {
        personal-name    [0] IMPLICIT Personal-Name OPTIONAL,
        organization      [1] IMPLICIT Character-Data OPTIONAL}
                        OPTIONAL,

    owners              [2] IMPLICIT SEQUENCE OF SET {
        personal-name    [0] IMPLICIT Personal-Name OPTIONAL,
        organization      [1] IMPLICIT Character-Data OPTIONAL}
                        OPTIONAL,

    authors             [3] IMPLICIT SEQUENCE OF SET {
        personal-name    [0] IMPLICIT Personal-Name OPTIONAL,
        organization      [1] IMPLICIT Character-Data OPTIONAL}
                        OPTIONAL)

```

```

Personal-Name ::= [APPLICATION 6] IMPLICIT SET {
    surname      [0] IMPLICIT Character-Data,
    givenname    [1] IMPLICIT Character-Data OPTIONAL,
    initials     [2] IMPLICIT Character-Data OPTIONAL,
    title        [3] IMPLICIT Character-Data OPTIONAL}

```

```

Other-User-Information      ::= SET {
    copyright-information    [0] IMPLICIT SET OF Character-Data OPTIONAL,
    copyright-dates          [4] IMPLICIT SET OF Character-Data OPTIONAL,
    status                   [1] IMPLICIT Character-Data OPTIONAL,
    user-specific-codes      [2] IMPLICIT SET OF Character-Data OPTIONAL,
    distribution-list        [3] IMPLICIT SET OF Character-Data OPTIONAL,
    additional-information   [5] ANY OPTIONAL}

External-References         ::= SET {
    references-to-other-documents [0] IMPLICIT SET OF Filing-and-Retrieval
                                         OPTIONAL,
    superseded-documents        [1] IMPLICIT SET OF Filing-and-Retrieval
                                         OPTIONAL}

Filing-and-Retrieval       ::= SET {
    keywords                 [3] IMPLICIT SET OF Character-Data OPTIONAL,
    document-reference       [4] IMPLICIT Character-Data OPTIONAL,
    local-file-reference     [5] IMPLICIT SET OF SET {
        filename            [0] IMPLICIT Character-Data OPTIONAL,
        location            [1] IMPLICIT Character-Data OPTIONAL,
        user-comments       [2] IMPLICIT Character-Data OPTIONAL}
        OPTIONAL}

Content-Attributes          ::= SET {
    document-size            [1] IMPLICIT INTEGER OPTIONAL,
    number-of-pages         [2] IMPLICIT INTEGER OPTIONAL,
    languages                [4] IMPLICIT SET OF Character-Data OPTIONAL}

Security-Attributes        ::= SET {
    CHOICE {
        authorization       [0] IMPLICIT Personal-Name,
        person              [4] IMPLICIT Character-Data} OPTIONAL,
        organization        [1] IMPLICIT Character-Data OPTIONAL,
        security-classification [2] IMPLICIT SET OF Character-Data OPTIONAL,
        access-rights       [3] IMPLICIT SET OF Character-Data OPTIONAL,
        encryption-indicator [5] IMPLICIT Character-Data OPTIONAL}
    password

```

5.3 Identifiers and expressions

Content-Portion-Identifier ::= [APPLICATION 0] IMPLICIT PrintableString

-- only digits and space are used in the present version
 -- of the standard; other characters are reserved for extensions

Object-or-Class-Identifier ::= [APPLICATION 1] IMPLICIT PrintableString

-- only digits and space are used in the present version
 -- of the standard; other characters are reserved for extensions;
 -- a "null" value is represented by an empty string

Style-Identifier ::= [APPLICATION 5] IMPLICIT PrintableString

-- only digits and space are used in the present version
 -- of the standard; other characters are reserved for extensions
 -- a "null" value is represented by an empty string

Layout-Category-Name ::= INTEGER

-- a "null" value is represented by zero

Construction-Expression ::= CHOICE {
 sequence-construction [0] IMPLICIT Term-Sequence,
 aggregate-construction [1] IMPLICIT Term-Sequence,
 choice-construction [2] IMPLICIT Term-Sequence,
 single-term-construction [3] Construction-Term}

Term-Sequence ::= SEQUENCE OF Construction-Term

Construction-Term ::= CHOICE {
 required-construction-factor [0] Construction-Factor,
 optional-construction-factor [1] Construction-Factor,
 repetitive-construction-factor [2] Construction-Factor,
 optional-repetitive-factor [3] Construction-Factor}

Construction-Factor ::= CHOICE {
 object-class-identifier Object-or-Class-Identifier,
 construction-expression Construction-Expression}

Object-Id-Expression ::= CHOICE {
 current-object-function [0] IMPLICIT NULL,
 preceding-object-function [1] Object-Id-Expression,
 superior-object-function [3] Object-Id-Expression,
 current-instance-function [4] Current-Instance-Function}

Numeric-Expression	::= CHOICE {
numeric-literal	[0] IMPLICIT INTEGER,
increment-application	[1] Numeric-Expression,
decrement-application	[2] Numeric-Expression,
ordinal-application	[3] CHOICE {
identifier	Object-or-Class-Identifier,
expression	Object-Id-Expression),
binding-reference	[4] IMPLICIT Binding-Reference}
Binding-Reference	::= SET {
object-reference	CHOICE {
identifier	Object-or-Class-Identifier,
expression	Binding-Selection-Function),
binding-identifier	INTEGER}
Binding-Selection-Function	::= CHOICE {
current-object-function	[0] IMPLICIT NULL,
preceding-function	[1] Object-Id-Expression,
superior-function	[3] Object-Id-Expression,
current-instance-function	[4] Current-Instance-Function}
Current-Instance-Function	::= SET {
first-parameter	CHOICE {
identifier	Object-or-Class-Identifier,
object-type	CHOICE {
layout-object-type	[0] IMPLICIT Layout-Object-Type,
logical-object-type	[1] IMPLICIT Logical-Object-Type),
second-parameter	CHOICE {
identifier	Object-or-Class-Identifier,
expression	Object-Id-Expression}}
String-Expression	::= SEQUENCE OF Atomic-String-Expression
Atomic-String-Expression	::= CHOICE {
string-literal	[0] IMPLICIT OCTET STRING,
binding-reference	[2] IMPLICIT Binding-Reference,
make-string-application	[3] Numeric-Expression,
upper-alpha-application	[4] Numeric-Expression,
lower-alpha-application	[5] Numeric-Expression,
upper-roman-application	[6] Numeric-Expression,
lower-roman-application	[7] Numeric-Expression}

5.4 Layout descriptors

```

Layout-Object-Descriptor      ::= SEQUENCE {
    object-type                Layout-Object-Type OPTIONAL,
    descriptor-body            Layout-Object-Descriptor-Body OPTIONAL}

Layout-Object-Type            ::= INTEGER {document-layout-root (0),
                                           page-set (1), page (2),
                                           frame (3), block (4)}

Layout-Object-Descriptor-Body ::= SET {
    object-identifier          Object-or-Class-Identifier OPTIONAL,
    subordinates                [0] IMPLICIT SEQUENCE OF NumericString
                                           OPTIONAL,
    content-portions           [1] IMPLICIT SEQUENCE OF NumericString
                                           OPTIONAL,
    object-class                [2] IMPLICIT Object-or-Class-Identifier
                                           OPTIONAL,
    position                    [3] IMPLICIT Position-Pair OPTIONAL,
    dimensions                  [4] IMPLICIT Dimension-Pair OPTIONAL,
    transparency                [5] IMPLICIT Transparency OPTIONAL,
    presentation-attributes     [6] IMPLICIT Presentation-Attributes
                                           OPTIONAL,
    default-value-lists        [7] IMPLICIT Default-Value-Lists-Layout
                                           OPTIONAL,
    user-readable-comments      [8] IMPLICIT Comment-String OPTIONAL,
    bindings                    [9] IMPLICIT SET OF Binding-Pair OPTIONAL,
    layout-path                 [11] IMPLICIT One-Of-Four-Angles OPTIONAL,
    imaging-order               [12] IMPLICIT SEQUENCE OF NumericString
                                           OPTIONAL,
    permitted-categories        [13] IMPLICIT SET OF Layout-Category-Name
                                           OPTIONAL,

    -- a "null" value is represented by an empty set

    user-visible-name           [14] IMPLICIT Comment-String OPTIONAL,
    page-position               [15] IMPLICIT Measure-Pair OPTIONAL,
    medium-type                 [16] IMPLICIT Medium-Type OPTIONAL,
    presentation-style          [17] IMPLICIT Style-Identifier OPTIONAL,
    balance                     [21] IMPLICIT SET OF
                                Object-or-Class-Identifier OPTIONAL,

    -- a "null" value is represented by an empty set

    colour                      [22] IMPLICIT Colour OPTIONAL,
    border                      [23] IMPLICIT Border OPTIONAL}

```



```

Measure-Pair                                ::= SEQUENCE {
    horizontal                               Measure,
    vertical                                Measure}

Measure                                     ::= CHOICE {
    fixed                                    [0] IMPLICIT INTEGER}

    -- other measures for further study

Medium-Type                                ::= SEQUENCE {
    nominal-page-size                       Measure-Pair OPTIONAL,
    side-of-sheet                           INTEGER {unspecified (0), recto (1),
    verso (2)} OPTIONAL}

X
| Colour                                    ::= INTEGER {colourless (0), white (1)}

| Border                                   ::= SET {
    line-width                             [0] IMPLICIT INTEGER OPTIONAL,
    line-type                             [1] IMPLICIT INTEGER {
        solid (0), dashed (1), dot (2),
        dash-dot (3), dash-dot-dot (4)}
                                           OPTIONAL,

    freespace-width                       [2] IMPLICIT INTEGER OPTIONAL,
    edges                                [3] IMPLICIT SET OF INTEGER {
        left-hand (0), right-hand (1),
        leading (2), trailing (3)} OPTIONAL}

    -- a "null" value is represented by an empty set

```

```

Layout-Class-Descriptor      ::= SEQUENCE {
    object-type               Layout-Object-Type,
    descriptor-body           Layout-Class-Descriptor-Body}

Layout-Class-Descriptor-Body ::= SET {
    object-class-identifier   Object-or-Class-Identifier,
    generator-for-subordinates Generator OPTIONAL,
    content-portions          [1] IMPLICIT SEQUENCE OF NumericString
                                OPTIONAL,
    position                  [3] IMPLICIT Position-Pair OPTIONAL,
    dimensions                 [4] IMPLICIT Dimension-Pair OPTIONAL,
    transparency               [5] IMPLICIT Transparency OPTIONAL,
    presentation-attributes   [6] IMPLICIT Presentation-Attributes
                                OPTIONAL,
    default-value-lists       [7] IMPLICIT Default-Value-Lists-Layout
                                OPTIONAL,
    user-readable-comments    [8] IMPLICIT Comment-String OPTIONAL,
    bindings                   [9] IMPLICIT SET OF Binding-Pair OPTIONAL,
    content-generator         [10] IMPLICIT String-Expression OPTIONAL,
    layout-path                [11] IMPLICIT One-Of-Four-Angles OPTIONAL,
    permitted-categories      [13] IMPLICIT SET OF Layout-Category-Name
                                OPTIONAL,

```

-- a "null" value is represented by an empty set

```

user-visible-name           [14] IMPLICIT Comment-String OPTIONAL,
page-position                [15] IMPLICIT Measure-Pair OPTIONAL,
medium-type                  [16] IMPLICIT Medium-Type OPTIONAL,
presentation-style           [17] IMPLICIT Style-Identifier OPTIONAL,
logical-source                [18] IMPLICIT Object-or-Class-Identifier
                                OPTIONAL,
balance                      [21] IMPLICIT SET OF
                                Object-Id-Expression OPTIONAL,

```

-- a "null" value is represented by an empty set

```

colour                       [22] IMPLICIT Colour OPTIONAL,
border                       [23] IMPLICIT Border OPTIONAL,
resource                      [24] IMPLICIT INTEGER OPTIONAL}

```

```

Generator                    ::= CHOICE {
    simple-construction       [0] IMPLICIT Simple-Construction,
    construction-expression   [2] Construction-Expression}

```

```

Simple-Construction          ::= SEQUENCE OF NumericString

```



```

Property-Attributes      ::= SET {
  protection              [0] IMPLICIT INTEGER {unprotected (0),
                           protected (1)} OPTIONAL}

```

-- other property attributes to be defined

5.6 Style descriptors

```

Presentation-Style-Descriptor ::= SEQUENCE {
  style-identifier          Style-Identifier,
  user-readable-comments    [0] IMPLICIT Comment-String OPTIONAL,
  user-visible-name         [1] IMPLICIT Comment-String OPTIONAL,
  transparency              [2] IMPLICIT Transparency OPTIONAL,
  presentation-attributes   [3] IMPLICIT Presentation-Attributes
                              OPTIONAL,
  colour                    [4] IMPLICIT Colour OPTIONAL,
  border                    [5] IMPLICIT Border OPTIONAL}

```

```

Presentation-Attributes    ::= SET {
  content-type              Content-Type OPTIONAL,
  content-architecture-class Content-Architecture-Class OPTIONAL,
  character-attributes      [0] IMPLICIT Character-Attributes OPTIONAL,
  raster-graphics-attributes [1] IMPLICIT Raster-Graphics-Attributes
                              OPTIONAL,
  geometric-graphics-attributes [2] IMPLICIT Geometric-Graphics-Attributes
                              OPTIONAL}

```

-- Character-Attributes is defined in part 6
 -- Raster-Graphics-Attributes is defined in part 7
 -- Geometric-Graphics-Attributes is defined in part 8
 -- the following tags are reserved for additional types
 -- of presentation attributes:
 -- [3] composite-graphics
 -- [4] audio
 -- [5] dynamic-graphics

```

Content-Type              ::= [APPLICATION 2] IMPLICIT INTEGER {
  formatted-characters (0),
  formatted-raster-graphics (1)}

```

X

```

Content-Architecture-Class ::= OBJECT IDENTIFIER

```


Block-Attributes

position
 dimensions
 transparency
 presentation-attributes
 presentation-style
 colour
 border

```

::= SET {
    < Attribute OPTIONAL,
    < Attribute OPTIONAL,
    < Attribute OPTIONAL,
    < Attribute OPTIONAL,
    < Attribute OPTIONAL,
    < Attribute OPTIONAL,
    < Attribute OPTIONAL}

```

Composite-Logical-Attributes

property-attributes
 layout-style

```

::= SET {
    < Attribute OPTIONAL,
    < Attribute OPTIONAL}

```

Basic-Logical-Attributes

presentation-attributes
 property-attributes
 presentation-style
 layout-style

```

::= SET {
    < Attribute OPTIONAL,
    < Attribute OPTIONAL,
    < Attribute OPTIONAL,
    < Attribute OPTIONAL}

```

Attribute

position
 dimensions
 transparency
 presentation-attributes
 layout-path
 page-position
 medium-type
 permitted-categories
 property-attributes
 presentation-style
 layout-style
 colour
 border

```

::= CHOICE {
    [0] IMPLICIT Measure-Pair,
    [1] IMPLICIT Measure-Pair,
    [2] IMPLICIT Transparency,
    [3] IMPLICIT Presentation-Attributes,
    [4] IMPLICIT One-Of-Four-Angles,
    [5] IMPLICIT Measure-Pair,
    [6] IMPLICIT Medium-Type,
    [7] IMPLICIT SET OF Layout-Category-Name,
    [8] IMPLICIT Property-Attributes,
    [9] IMPLICIT Style-Identifier,
    [10] IMPLICIT Style-Identifier,
    [11] IMPLICIT Colour,
    [12] IMPLICIT Border}

```

5.8 Text units

```

Text-Unit ::= SEQUENCE {
    content-portion-attributes Content-Portion-Attributes OPTIONAL,
    content-information       Content-Information)

Content-Portion-Attributes ::= SET {
    content-identifier-layout Content-Portion-Identifier OPTIONAL,
    content-identifier-logical [4] IMPLICIT Content-Portion-Identifier
                                OPTIONAL,
    type-of-coding            Type-Of-Coding OPTIONAL,
    coding-attributes         CHOICE {
        character-coding-attributes [1] IMPLICIT Character-Coding-Attributes,
        raster-gr-coding-attributes [2] IMPLICIT Raster-Gr-Coding-Attributes,
        geo-gr-coding-attributes    [7] IMPLICIT Geo-Gr-Coding-Attributes)
                                OPTIONAL,

    -- Character-Coding-Attributes is defined in part 6
    -- Raster-Gr-Coding-Attributes is defined in part 7
    -- Geo-Gr-Coding-Attributes is defined in part 8
    -- the following tags are reserved for additional types
    -- coding attributes:
    --   [8] composite-graphics
    --   [9] audio
    --   [10] dynamic-graphics

    alternative-representation [3] IMPLICIT Alternative-Representation
                                OPTIONAL)

Content-Information ::= CHOICE {
    teletex-information TeletexString,
    other-information   OCTET STRING}

Type-Of-Coding ::= CHOICE {
    [0] IMPLICIT INTEGER {
        iso2022 (0), t6 (1)},
    [6] IMPLICIT OBJECT IDENTIFIER}

Alternative-Representation ::= OCTET STRING

    -- string of characters from the sets designated by
    -- the document profile attribute "alternative character sets",
    -- plus carriage return and line feed

```

Character Content Architecture

```

Character-Attributes ::= SET (
    character-path      [0] IMPLICIT One-Of-Four-Angles
                        OPTIONAL,
    line-progression    [1] IMPLICIT One-Of-Two-Angles
                        OPTIONAL,
    character-orientation [2] IMPLICIT One-Of-Four-Angles
                        OPTIONAL,
    initial-offset      [3] IMPLICIT Measure-Pair
                        OPTIONAL,
    character-spacing    [6] IMPLICIT INTEGER OPTIONAL,
    line-spacing        [7] IMPLICIT INTEGER OPTIONAL,
    alignment           [8] IMPLICIT Alignment OPTIONAL,
    line-layout-table    [9] IMPLICIT Layout-Table
                        OPTIONAL,
    graphic-rendition    [10] IMPLICIT Graphic-Rendition
                        OPTIONAL,
    alignment-indicator [11] IMPLICIT Alignment-Indicator
                        OPTIONAL,
    character-fonts      [12] IMPLICIT Character-Fonts
                        OPTIONAL,
    graphic-char-subrepertoire [13] IMPLICIT INTEGER OPTIONAL,
    first-line-format    [14] IMPLICIT First-Line-Format
                        OPTIONAL,
    widow-size          [15] IMPLICIT INTEGER OPTIONAL,
    orphan-size         [16] IMPLICIT INTEGER OPTIONAL,
    graphic-character-sets [17] IMPLICIT OCTET STRING OPTIONAL,
    precision           [18] IMPLICIT Precision OPTIONAL,
    indentation         [19] IMPLICIT INTEGER OPTIONAL,
    kerning-offset      [20] IMPLICIT Kerning-Offset
                        OPTIONAL,
    leading             [21] IMPLICIT Leading OPTIONAL,
    pairwise-kerning     [22] IMPLICIT Pairwise-Kerning
                        OPTIONAL )

```

```

One-Of-Four-Angles ::= INTEGER ( d0   (0),
                                d90  (1),
                                d180 (2),
                                d270 (3) )

```

```

One-Of-Two-Angles ::= INTEGER ( d90  (1),
                                d270 (3) )

```

```

Measure-Pair ::= SEQUENCE (
    horizontal [0] IMPLICIT INTEGER,
    vertical   [0] IMPLICIT INTEGER )

```

```

Alignment ::= INTEGER ( start-aligned (0),
                        end-aligned   (1),
                        centred       (2),
                        justified      (3) )

Layout-Table ::= SET OF Tabulation-Stop

Tabulation-Stop ::= SET (
  tabulation-reference [0] IMPLICIT NumericString,
  tabulation-position  [1] IMPLICIT INTEGER,
  alignment             [2] IMPLICIT INTEGER (
    start-aligned (0),
    end-aligned   (1),
    centred       (2),
    aligned-around (3) )
    OPTIONAL,
  alignment-character-string [3] IMPLICIT OCTET STRING
    OPTIONAL)

-- string of graphic characters
-- from the set of graphic elements
-- specified by the presentation
-- attributes "graphic character
-- sets" and "graphic character
-- repertoire"

Graphic-Rendition ::= SET OF Graphic-Rendition-Aspect

Character-Fonts ::= SET (
  primary-font [0] IMPLICIT Font-Type OPTIONAL,
  first-alternative-font [1] IMPLICIT Font-Type OPTIONAL,
  second-alternative-font [2] IMPLICIT Font-Type OPTIONAL,
  third-alternative-font [3] IMPLICIT Font-Type OPTIONAL,
  fourth-alternative-font [4] IMPLICIT Font-Type OPTIONAL,
  fifth-alternative-font [5] IMPLICIT Font-Type OPTIONAL,
  sixth-alternative-font [6] IMPLICIT Font-Type OPTIONAL,
  seventh-alternative-font [7] IMPLICIT Font-Type OPTIONAL,
  eighth-alternative-font [8] IMPLICIT Font-Type OPTIONAL,
  ninth-alternative-font [9] IMPLICIT Font-Type OPTIONAL)

Font-Type ::= SET (
  font-size [0] IMPLICIT INTEGER,
  font-identifier [1] IMPLICIT INTEGER )

```

```

Graphic-Rendition-Aspect ::= INTEGER (
    cancel (0),
    increased-intensity (1),
    decreased-intensity (2),
    italicized (3),
    underlined (4),
    slowly-blinking (5),
    rapidly-blinking (6),
    negative-image (7),
    crossed-out (9),
    primary-font (10),
    first-alternative-font (11),
    second-alternative-font (12),
    third-alternative-font (13),
    fourth-alternative-font (14),
    fifth-alternative-font (15),
    sixth-alternative-font (16),
    seventh-alternative-font (17),
    eighth-alternative-font (18),
    ninth-alternative-font (19),
    doubly-underlined (21),
    normal-intensity (22),
    not-italicized (23),
    not-underlined (24),
    steady (25),
    variable-spacing (26),
    positive-image (27),
    not-crossed-out (29) )

```

```

Alignment-Indicator ::= INTEGER ( not-aligned (0),
                                   aligned (1) )

```

```

First-Line-Format
  itemization ::= SET (
    [0] IMPLICIT INTEGER
    (indentation (0),
     overhang-without-item-id (1),
     overhang-with-s-a-item-id (2),
     overhang-with-e-a-item-id (3))
    OPTIONAL,
    first-line-offset [1] IMPLICIT INTEGER OPTIONAL )

```

```

Precision ::= SEQUENCE ( INTEGER , INTEGER )

```

```

Kerning-Offset
  start-offset ::= SET (
    [0] IMPLICIT INTEGER,
    [1] IMPLICIT INTEGER )
  end-offset

```

```

Leading                               ::= SET (
    leading-indicator                [0] IMPLICIT INTEGER
                                     ( no (0) , yes (1) ) OPTIONAL,
    inter-line-space                  [1] IMPLICIT INTEGER OPTIONAL )
Pairwise-Kerning                      ::= INTEGER ( no (0) , yes (1) )

```

9.3 Representation of coding attributes

```

Character-Coding-Attributes ::= SET ( )

-- no character coding attributes
-- are defined in this version of
-- the standard.

```

9.4 Representation of non-basic presentation attribute values

```

Char-Presentation-Attribute ::= CHOICE (
    character-path                [0] IMPLICIT One-Of-Four-Angles,
    line-progression              [1] IMPLICIT One-Of-Two-Angles,
    character-orientation         [2] IMPLICIT One-Of-Four-Angles,
    character-spacing             [6] IMPLICIT INTEGER,
    line-spacing                  [7] IMPLICIT INTEGER,
    alignment                     [8] IMPLICIT Alignment,
    graphic-rendition             [10] IMPLICIT Graphic-Rendition,
    graphic-char-subrepertoire    [13] IMPLICIT INTEGER,
    graphic-character-sets        [17] IMPLICIT OCTET STRING )

```

9.5 Representation of non-basic coding attribute values

```

Character-Coding-Attribute ::= CHOICE ( )

-- no character coding attributes
-- are defined in this version of
-- the standard.

```

Raster Graphics Content Architecture

```

Raster-Graphics-Attributes ::= SET (
    pel-path                [0] IMPLICIT One-Of-Four-Angles
                                OPTIONAL,
    line-progression         [1] IMPLICIT One-Of-Two-Angles
                                OPTIONAL,
    pel-transmission-density [2] IMPLICIT Pel-Transmission-
                                Density OPTIONAL,
    initial-offset           [3] IMPLICIT Measure-Pair
                                OPTIONAL,
    clipping                 [4] IMPLICIT Clipping OPTIONAL,
    pel-spacing              [5] IMPLICIT Pel-Spacing
                                OPTIONAL,
    spacing-ratio            [6] IMPLICIT Spacing-Ratio
                                OPTIONAL,
    image-dimensions         [7] Image-Dimensions OPTIONAL )

One-Of-Four-Angles ::= INTEGER (
    d0 (0), -- 0 degrees
    d90 (1), -- 90 degrees
    d180 (2), -- 180 degrees
    d270 (3) -- 270 degrees
)

One-Of-Two-Angles ::= INTEGER (
    d90 (1), -- 90 degrees
    d270 (3) -- 270 degrees
)

Pel-Transmission-Density ::= INTEGER (
    p7 (0), -- 180 pels/1200 SMU
    p6 (1), -- 200 pels/1200 SMU
    p5 (2), -- 240 pels/1200 SMU
    p4 (3), -- 300 pels/1200 SMU
    p3 (4), -- 400 pels/1200 SMU
    p2 (5), -- 600 pels/1200 SMU
    p1 (6) -- 1200 pels/1200 SMU
)

Measure-Pair ::= SEQUENCE (
    horizontal [0] IMPLICIT INTEGER,
    vertical   [0] IMPLICIT INTEGER )

Clipping ::= SEQUENCE (
    first-coordinate-pair [0] IMPLICIT Coordinate-Pair
                                OPTIONAL,
    second-coordinate-pair [1] IMPLICIT Coordinate-Pair
                                OPTIONAL )

```

Coordinate-Pair x-coordinate y-coordinate	::= SEQUENCE (INTEGER, INTEGER)
Pel-Spacing line-length pel-spaces	::= SEQUENCE (INTEGER, INTEGER)
Spacing-Ratio line-spacing-value pel-spacing-value	::= SEQUENCE (INTEGER, INTEGER)
Image-Dimensions	::= CHOICE (image-width minimum-width preferred_width image-height minimum-height preferred_height image-size minimum-width preferred_width minimum-height preferred_height aspect-ratio-flag automatic [0] IMPLICIT SET ([0] IMPLICIT INTEGER, [1] IMPLICIT INTEGER), [1] IMPLICIT SET ([2] IMPLICIT INTEGER, [3] IMPLICIT INTEGER), [2] IMPLICIT SET ([0] IMPLICIT INTEGER, [1] IMPLICIT INTEGER, [2] IMPLICIT INTEGER, [3] IMPLICIT INTEGER, [4] IMPLICIT INTEGER (fixed (0), variable (1))), [3] IMPLICIT NULL)

8.3 Representation of coding attributes

The data type "Raster-Gr-Coding-Attributes" contains a set of subordinate data types that specify the raster graphics coding attributes. The format of these data types is given below.

```
Raster-Gr-Coding-Attributes ::= SET {
    number-of-pels-per-line      [0] IMPLICIT INTEGER OPTIONAL,
    number-of-lines              [1] IMPLICIT INTEGER OPTIONAL,
    number-of-discarded-pels     [3] IMPLICIT INTEGER OPTIONAL,
    pel-array-order              [4] IMPLICIT INTEGER (
                                   down (0),
                                   up   (1) ) OPTIONAL }
```

8.4 Representation of non-basic presentation attribute values

```
Ra-Gr-Presentation-Attribute ::= CHOICE (
    pel-path                    [0] IMPLICIT One-Of-Four-Angles,
    line-progression            [1] IMPLICIT One-Of-Two-Angles,
    pel-transmission-density    [2] IMPLICIT Pel-Transmission
                                   Density )
```

8.5 Representation of non-basic coding attribute values

```
Ra-Gr-Coding-Attribute ::= CHOICE ( )
-- no non-basic values are expected to
-- be defined for the raster graphics
-- coding attributes defined in this
-- part of ISO 8613.
```

9.2 Coded representation of geometric graphics presentation attributes

```

Geometric-Graphics-Attributes ::= SET {
    encoding-announcer           [0] IMPLICIT Encoding-Announcer OPTIONAL,
    line-rendition               [1] IMPLICIT Line-Rendition OPTIONAL,
    marker-rendition             [2] IMPLICIT Marker-Rendition OPTIONAL,
    text-rendition               [3] IMPLICIT Text-Rendition OPTIONAL,
    filled-area-rendition        [4] IMPLICIT Filled-Area-Rendition OPTIONAL,
    edge-rendition               [5] IMPLICIT Edge-Rendition OPTIONAL,
    colour-representations       [6] IMPLICIT Colour-Representations OPTIONAL,
    transparency-specification   [7] IMPLICIT Transparency-Specification OPTIONAL,
    transformation-specification [8] IMPLICIT Transformation-Specification OPTIONAL,
    region-of-interest           [9] IMPLICIT Region-Of-Interest OPTIONAL,
    picture-orientation          [10] IMPLICIT Picture-Orientation OPTIONAL,
    picture-dimensions           [11] IMPLICIT Picture-Dimensions OPTIONAL }

Encoding-Announcer ::= OCTET STRING
    -- octet string representing the binary encoding of any
    -- ordered set of metafile elements identified in 7.1.1.1

Line-Rendition ::= SEQUENCE {
    individual-part              [0] IMPLICIT OCTET STRING OPTIONAL,
    -- octet string representing the binary encoding of any
    -- ordered set of metafile elements identified in 7.1.1.2,
    -- up to parameter "line aspect source flags"

```

asf-part	[1] IMPLICIT SEQUENCE {
line-type-asf	INTEGER { bundled(0), individual(1) },
line-width-asf	INTEGER { bundled(0), individual(1) },
line-colour-asf	INTEGER { bundled(0), individual(1) }
	} OPTIONAL,
bundle-part	[2] IMPLICIT SEQUENCE OF {
	SEQUENCE {
bundle-index	INTEGER,
bundle-representation	OCTET STRING }} OPTIONAL
	-- octet string representing the binary
	-- encoding of the metafile elements
	-- LINE TYPE, LINE WIDTH and LINE
	-- COLOUR, as identified in 7.1.1.2
	}
Marker-Rendition ::=	SEQUENCE {
individual-part	[0] IMPLICIT OCTET STRING OPTIONAL,
	-- octet string representing the binary encoding of any
	-- ordered set of metafile elements identified in 7.1.1.3,
	-- up to parameter "marker aspect source flags"
asf-part	[1] IMPLICIT SEQUENCE {
marker-type-asf	INTEGER { bundled(0), individual(1) },
marker-size-asf	INTEGER { bundled(0), individual(1) },
marker-colour-asf	INTEGER { bundled(0), individual(1) }
	} OPTIONAL,
bundle-part	[2] IMPLICIT SEQUENCE OF {
	SEQUENCE {
bundle-index	INTEGER,
bundle-representation	OCTET STRING }} OPTIONAL
	-- octet string representing the binary
	-- encoding of the metafile elements
	-- MARKER TYPE, MARKER SIZE and
	-- MARKER COLOUR, as identified
	-- in 7.1.1.3
	}
Text-Rendition ::=	SEQUENCE {
individual-part	[0] IMPLICIT OCTET STRING OPTIONAL,
	-- octet string representing the binary encoding of any
	-- ordered set of metafile elements identified in 7.1.1.4,
	-- up to parameter "text aspect source flags"
asf-part	[1] IMPLICIT SEQUENCE {
text-font-asf	INTEGER { bundled(0), individual(1) },
text-precision-asf	INTEGER { bundled(0), individual(1) },
character-expansion-factor-asf	INTEGER { bundled(0), individual(1) },
character-spacing-asf	INTEGER { bundled(0), individual(1) },
text-colour-asf	INTEGER { bundled(0), individual(1) }
	} OPTIONAL,

bundle-part	[2] IMPLICIT SEQUENCE OF {
bundle-index	SEQUENCE {
bundle-representation	INTEGER,
	OCTET STRING }} OPTIONAL
	-- octet string representing the binary
	-- encoding of the metafile elements
	-- TEXT FONT INDEX, TEXT PRECISION,
	-- CHARACTER EXPANSION FACTOR,
	-- CHARACTER SPACING and TEXT CO-
	--LOUR, as identified in 7.1.1.4
	}
Filled-Area-Rendition ::=	SEQUENCE {
individual-part	[0] IMPLICIT OCTET STRING OPTIONAL,
	-- octet string representing the binary encoding of any
	-- ordered set of metafile elements identified in 7.1.1.5,
	-- up to parameter "pattern table specifications"
pattern-table-part	[1] IMPLICIT SEQUENCE OF {
pattern-table-element	OCTET STRING } OPTIONAL
	-- octet string representing the binary
	-- encoding of the metafile element
	-- PATTERN TABLE, as identified in 7.1.1.5
asf-part	[2] IMPLICIT SEQUENCE {
interior-style-asf	INTEGER { bundled(0), individual(1) },
fill-colour-asf	INTEGER { bundled(0), individual(1) },
hatch-index-asf	INTEGER { bundled(0), individual(1) },
pattern-index-asf	INTEGER { bundled(0), individual(1) }
	} OPTIONAL,
bundle-part	[3] IMPLICIT SEQUENCE OF {
bundle-index	SEQUENCE {
bundle-representation	INTEGER,
	OCTET STRING }} OPTIONAL
	-- octet string representing the binary
	-- encoding of the metafile elements
	-- INTERIOR STYLE, FILL COLOUR, HATCH
	-- INDEX and PATTERN INDEX as
	-- identified in 7.1.1.5
	}
Edge-Rendition ::=	SEQUENCE {
individual-part	[0] IMPLICIT OCTET STRING OPTIONAL,
	-- octet string representing the binary encoding of any
	-- ordered set of metafile elements identified in 7.1.1.6,
	-- up to parameter "edge aspect source flags"
asf-part	[1] IMPLICIT SEQUENCE {
edge-type-asf	INTEGER { bundled(0), individual(1) },
edge-width-asf	INTEGER { bundled(0), individual(1) },
edge-colour-asf	INTEGER { bundled(0), individual(1) }
	} OPTIONAL,

bundle-part	[2] IMPLICIT SEQUENCE OF {
bundle-index	SEQUENCE {
bundle-representation	INTEGER,
	OCTET STRING }} OPTIONAL
	-- octet string representing the binary
	-- encoding of the metafile elements
	-- EDGE TYPE, EDGE WIDTH and EDGE
	-- COLOUR, as identified in 7.1.1.6
	}
Colour-Representations ::=	SEQUENCE {
background-colour	[0] IMPLICIT OCTET STRING OPTIONAL,
	-- octet string representing the binary encoding of the
	-- metafile element BACKGROUND COLOUR as identified
	-- in 7.1.1.7
colour-table-part	[1] IMPLICIT SEQUENCE OF {
colour-table-element	OCTET STRING } OPTIONAL
	-- octet string representing the binary
	-- encoding of the metafile element
	-- COLOUR TABLE, as identified in 7.1.1.7
	}
Transparency-Specification ::=	OCTET STRING
	-- octet string representing the binary encoding of any
	-- ordered set of the metafile elements identified in 7.1.1.8
Transformation-Specification ::=	OCTET STRING
	-- octet string representing the binary encoding of any
	-- ordered set of the metafile elements identified in 7.1.1.9
Region-Of-Interest ::=	CHOICE {
automatic	[0] IMPLICIT NULL,
rectangle	[1] IMPLICIT SEQUENCE { VDC-Pair, VDC-Pair }
	}
VDC-Pair ::=	OCTET STRING
	-- octet string representing the binary encoding of a CGM
	-- Virtual Device Coordinate pair
Picture-Orientation ::=	One-Of-Four-Angles
One-Of-Four-Angles ::=	INTEGER { d0(0),
	d90(1),
	d180(2),
	d270(3) }

```

Picture-Dimensions ::=
    picture-width
        minimum-width
        preferred-width
    picture-height
        minimum-height
        preferred-height
    picture-size
        minimum-width
        preferred-width
        minimum-height
        preferred-height
        aspect-ratio-flag
    automatic

CHOICE {
    [0] IMPLICIT SET {
        [0] IMPLICIT INTEGER,
        [1] IMPLICIT INTEGER },
    [1] IMPLICIT SET {
        [0] IMPLICIT INTEGER,
        [1] IMPLICIT INTEGER },
    [2] IMPLICIT SET {
        [0] IMPLICIT INTEGER,
        [1] IMPLICIT INTEGER,
        [2] IMPLICIT INTEGER,
        [3] IMPLICIT INTEGER,
        [4] IMPLICIT INTEGER {
            fixed(0),
            variable(1) }},
    [3] IMPLICIT NULL
}

```

9.3 Coded representation of coding attributes

```

Geo-Gr-Coding-Attributes ::= SET { }
    -- no geometric graphics coding attributes are defined
    -- in this version of ISO 8613

```

9.4 Representation of non-basic presentation attribute values

```

Geo-Gr-Presentation-Attribute ::= CHOICE { }
    -- no non-basic values are expected to be defined for the
    -- geometric graphics presentation attributes in this version
    -- of ISO 8613

```

9.5 Representation of non-basic coding attribute values

```

Geo-Gr-Coding-Attribute ::= CHOICE { }
    -- no geometric graphics coding attributes are defined
    -- in this version of ISO 8613

```