

SGML/HyTime voor de ondersteuning van het gezamenlijk ontwikkelen van multimedia-applicaties

Peter A.C. Verkoulen en Henk
M. Blanken

pagina 1 van 6

In dit artikel worden twee ISO-standaarden voor het ontwerpen van multimedia-documenten gepresenteerd. SGML richt zich op gestructureerde documenten met beperkte hypertext-faciliteiten (alleen verwijzingen binnen één document); HyTime (dat in zekere zin gebruik maakt van SGML) biedt functionaliteit voor het ontwerpen van volledige hypermedia-documenten. De standaarden zorgen voor uitwisselbaarheid van documenten en voor standaardconstructies om het ontwerpen van documenten te ondersteunen. De acceptatiegraad van deze twee standaarden is in de praktijk nog laag. Deels komt dit door een gebrek aan kennis over de standaarden en aan inzicht in de manier waarop deze het omgaan met multimedia-informatie kunnen ondersteunen. Dit artikel wil bijdragen aan de oplossing van dit probleem. De mogelijkheid van het uitwisselen van multimedia-documenten is cruciaal voor het succes van multimedia in de komende jaren. In een ander artikel in dit nummer van Informatie is het World Wide Web (WWW) genoemd. Het Web bestaat uit een groot aantal knopen, verspreid over de hele wereld, die samen een enorme hoeveelheid informatie herbergen. De taal HTML wordt algemeen gebruikt om de informatie in de verschillende knopen te structureren, zodat de verschillen in hardware en software per knoop naar buiten toe niet opvallen. Hier blijkt het nut van standaardisatie-inspanningen, want zonder zo'n standaardtaal zou het WWW nooit zo'n hoge vlucht hebben genomen. Echter, HTML is een ad hoc bedachte taal. Pas later is deze taal uitgedrukt in SGML. HTML kan slechts op elementair niveau met film en geluid omgaan. Een uitbreiding van HTML op basis van HyTime zou hierin verandering kunnen brengen.

1 Introductie

De kracht van multimedia is gelegen in de enorme variatie aan soorten informatie waarmee kan worden gewerkt. Men kan steeds de beste verschijningsvorm kiezen om een bepaald stuk informatie over te dragen. Ook is het mogelijk verscheidene informatiebronnen aan elkaar te relateren. Dit alles brengt echter wel een probleem met zich mee: het is moeilijk om multimedia-applicaties op een gestructureerde en eenduidige manier te ontwikkelen. Bovendien worden dit soort applicaties vaak door grote teams ontwikkeld. Hiervoor is een mechanisme nodig dat ervoor zorgt dat een heel team inderdaad samen aan een applicatie kan werken. De ISO-standaard voor SGML en de ontwerp-standaard voor HyTime zijn bedoeld om aan de oplossing van dit probleem bij te dragen.

Deze standaarden zijn bedoeld voor het beschrijven van respectievelijk eenvoudige hypertext- en volledige hypermedia-documenten. Beide standaarden kunnen worden gebruikt om vast te leggen aan welke eisen een bepaalde klasse van documenten moet voldoen. Zo zal een artikel voor een tijdschrift bijvoorbeeld altijd bestaan uit een lijst van auteurs, een aantal hoofdstukken (op hun beurt weer opgebouwd uit tekstparagrafen en plaatjes) en een literatuurlijst. Verderop in dit artikel zullen we meer voorbeelden zien.

Zijn deze standaarden echt nodig om uitwisseling van multimedia-documenten mogelijk te maken? Om deze vraag te beantwoorden, kijken we even naar de situatie bij het uitwisselen van platte tekst. Veronderstel dat documenten elektronisch uitgewisseld moeten worden. Dit is bijvoorbeeld het geval wanneer door een internationaal team van mensen aan een bepaald rapport gewerkt wordt. Als twee teamleden elkaar elektronisch

deeldocumenten willen toesturen en ze werken allebei met een ander tekstverwerkingssysteem, dan kan dat grote problemen met zich meebrengen. Zelfs twee opeenvolgende versies van eenzelfde tekstverwerkingssysteem zijn nooit twee kanten op compatibel. En zelfs wanneer overal hetzelfde systeem gebruikt wordt, dan nog kan door het hanteren van verschillende conventies een situatie ontstaan waarbij documenten niet meer uitwisselbaar zijn. De ISO-standaarden die we in dit verhaal behandelen, hebben als doel aan de oplossing van dit probleem bij te dragen.

Overigens moet men deze uitwisselingsproblemen niet onderschatten. De ontwikkelaars van SGML zijn de documentatiespecialisten van IBM en het Amerikaanse Ministerie van Defensie; het is niet bedacht door een projectgroepje van tien man op de hei. De toepassingen worden ook steeds uitgebreider. Zo zijn er al uitgeverij bezig om alle door hen gepubliceerde wetenschappelijke artikelen op te slaan in een SGML-database.

Het probleem van uitwisseling van documenten zou op te lossen zijn door convertors te ontwerpen voor elk tweetal formaten waarin documenten kunnen voorkomen. Moderne tekstverwerkers kunnen inderdaad, naast hun eigen formaat, ook een aantal andere formaten lezen en/of schrijven. Voorbeelden hiervan zijn WordPerfect 6.0 en MS-Word 6.0. Zoals Rada & Carson (1994) opmerken, kan dit uiteindelijk natuurlijk nooit een oplossing zijn. Immers, bij n verschillende formaten zou je in de orde van n^2 conversieprogramma's nodig hebben. Bovendien: wie moet deze programma's aanleveren? Standaarden zouden hierbij een belangrijke rol kunnen spelen. Allereerst zijn bij vertaling via een standaard nog slechts in de orde van n convertors nodig! Bovendien zou bij elke tekstverwerker een conversieprogramma van en naar de standaard geleverd kunnen worden.

Commerciële pakketten hebben dit onderkend. De twee marktleiders, WordPerfect en Microsoft Word, zijn beide bezig met de ontwikkeling van vertalingen van en naar SGML. Dit loopt echter minder soepel dan men tevoren had ingeschat. Rada & Carson (1994) geven als verklaring dat het combineren van het WYSIWYG (What You See Is What You Get)-principe van deze tekstverwerkers met het idee van SGML om logische inhoud van een document en lay-out-aspecten te scheiden, moeilijker blijkt te zijn dan men van tevoren wellicht had gehoopt.

SGML/HyTime is bedoeld om bovengenoemde problemen het hoofd te bieden. Als het alleen maar zou gaan om een universeel standaardformaat voor documentuitwisseling, dan was SGML/HyTime niet nodig: neem dan bijvoorbeeld het WP-6.0-formaat. Het hebben van één formaat lost echter niet alle problemen op; door verschillende conventies kan het effect optreden dat mensen met niet-uitwisselbare subformaten aan de gang gaan. Wat SGML onderscheidt, is dat het de logische opbouw van documenten expliciet gescheiden houdt van aspecten van opmaak en semantiek (dus van wat de applicatie ermee doet). Het idee van een datamodel dus. Daarmee probeer je de applicatiespecifieke van de algemene aspecten te scheiden. HyTime voegt daaraan nog een aantal standaardconcepten voor het ontwerp van multimedia-documenten toe.

In de rest van dit artikel zullen we eerst de SGML-standaard toelichten. Met enige voorbeelden zullen we proberen onze beschrijving te verlevendigen. Vervolgens wordt

ingegaan op HyTime en het verband tussen SGML en HyTime. Zoals zal blijken, kan HyTime in feite worden gezien als een extra laag bovenop SGML. We besluiten dit artikel met enige opmerkingen over de toepasbaarheid van SGML en HyTime.

2 SGML

De afkorting SGML staat voor Standard Generalized Markup Language. Het is een ISO-standaard uit 1986 (ISO, 1986). SGML is een zogenaamde mark-up-taal. Dit begrip komt uit de uitgeverwereld, waar manuscripten met de pen geannoteerd worden om de lay-out (en soms ook de structuur) van de tekst aan te geven. Dit gebeurt door gebruik te maken van speciale tekens en karakters. Zulke markers zijn overigens al eerder gebruikt in tekstverwerkingspakketten. Zo betekent de aanwijzing .ce aan het begin van een regel in een troff-document dat de volgende regel gecentreerd moet worden. Ook de bekende editor WordStar werkte op zo'n manier. Probleem bij de manier van het gebruik van markers zoals bij troff en WordStar is wel dat de inhoud van de tekst vervuild wordt met aanwijzingen voor de structuur en de lay-out.

SGML werkt ook met markers, maar de hoofddoelstelling van SGML is (zoals gezegd) documenten zodanig te kunnen beschrijven dat de logische inhoud en de bijbehorende lay-out-informatie gescheiden blijven. Documenten worden opgebouwd uit voorgedefinieerde bouwstenen, elementen genoemd. De volgorde waarin deze elementen in een bepaald type document mogen voorkomen, en de mogelijke verbanden ertussen, vormen de structuur van dat documenttype. In zogenaamde DTD's (Document Type Definitions) worden dergelijke documentstructuren gedefinieerd, door te specificeren hoe elementtypen mogen worden opgebouwd uit andere elementtypen en/of atomaire gegevens. Zo kan men vastleggen dat een elektronisch bericht altijd een geadresseerde, een afzender, een onderwerp en een inhoud heeft. Door gebruik te maken van een DTD voor elektronische berichten zorg je enerzijds dat de auteur niet meer werk hoeft te doen dan nodig, omdat hem een aantal standaardzaken uit handen wordt genomen. Daartoe behoort alles wat te maken heeft met de applicatie en de opmaak. Aan de andere kant dwing je zo een eenduidige structuur van elektronische berichten af. Doordat deze structuur formeel in een DTD is vastgelegd, kunnen automatische checkers en syntax-gestuurde editors worden gebruikt bij het opstellen van dergelijke berichten. Bovendien zijn deze hulpmiddelen generiek voor alle SGML-documenten. Dus als je bijvoorbeeld orderbriefjes wilt gaan maken, zijn deze hulpmiddelen nog steeds bruikbaar, omdat je alleen maar met een andere DTD gaat werken. Dit alles komt de uitwisselbaarheid van documenten uiteraard ten goede.

Voor de beschrijving van de lay-out van documenten die met behulp van SGML worden gedefinieerd, is de taal DSSSL ontwikkeld. DSSSL staat voor Document Style Semantics and Specification Language (ISO, 1991). Met behulp van DSSSL is een style sheet te definiëren, waarin de lay-out van een bepaalde klasse documenten vastgelegd wordt. Zo kan men bijvoorbeeld vastleggen dat hoofdstuktitels in een document vet gedrukt weergegeven moeten worden. Net zoals eenmalig een DTD moet worden gedefinieerd die de logische structuur van een documenttype vastlegt, moet men ook eenmalig een DSSSL style sheet definiëren ten behoeve van de lay-out.

Veronderstel bijvoorbeeld dat de betrokkenen bij het Betuwelijn-project met elkaar willen communiceren via electronic mail. Besloten wordt met behulp van SGML een vaste structuur voor dergelijke berichten vast te leggen. Hiervoor wordt een specifieke DTD gecreëerd. Deze DTD vormt in feite een skelet voor deze elektronische berichten, dat wordt opgebouwd uit zogenaamde elementtype-declaraties. Een elementtype-declaratie geeft de naam van het elementtype en (onder andere) de manier waarop een element van dat type kan worden opgebouwd. Zo worden hieronder de elementtypen e-mail en proloog gedefinieerd. Er wordt vastgelegd dat een elektronisch bericht (e-mail) bestaat uit een proloog en een inhoud (met andere woorden: een e-mail-element wordt opgebouwd uit een proloog-element en een inhoud-element). De proloog van een elektronisch bericht bestaat uit een verplichte afzender, één of meer (aangegeven door de +) geadresseerden, een optioneel (aangegeven door het ?) onderwerp en nul of meer (aangegeven door het *) mensen die een kopie van het bericht moeten krijgen. Het uitroepteken geeft aan dat hier een nieuw type wordt gedefinieerd (en wel een elementtype, aangegeven door het sleutelwoord ELEMENT). Verder staat het geheel tussen <...> markers. Deze syntax is een vast bestanddeel in SGML van de manier waarop je DTD's definieert.

In een volledige DTD moeten uiteraard alle elementtypen gedefinieerd worden. Dit gebeurt in voorbeeld 1. Merk op dat het sleutelwoord hier ELEMENT is, terwijl het wellicht beter ELEMENTTYPE had kunnen zijn.

Met behulp van attributen kunnen extra eigenschappen van elementen gedefinieerd worden. Om dit mogelijk te maken, moet in de DTD een zogenaamde attribute definition list declaration opgenomen worden. Zo geeft onderstaande attribute definition list declaration aan dat een e-mail een verzenddatum en een status heeft. Attributen kunnen bijgevolg gebruikt worden om eigenschappen van elementen (en dus van documenten) vast te leggen die je op zich niet direct tot de inhoud van een document wilt rekenen. Denk bijvoorbeeld aan de verzenddatum van een brief of de autorisatie van een bepaalde tekening. Ook zullen we zien dat attributen gebruikt kunnen worden om verbanden tussen elementen te definiëren.

DATUM #REQUIRED

(vertrouwelijk | publiek) publiek>

Meteen achter het sleutelwoord ATTLIST staat de naam van het elementtype waar deze lijst attributen bijhoort; in dit geval is dat dus e-mail. Vervolgens wordt een aantal attributen gegeven. In de eerste kolom staat steeds de naam van het attribuut. De tweede kolom legt het type van het attribuut vast. De derde kolom bevat een default-waarde of

een aanwijzing omtrent hoe met het attribuut moet worden omgegaan. Zo'n aanwijzing wordt gegeven met een sleutelwoord, voorafgegaan door het #-teken. Zo heeft elke e-mail een verplicht attribuut dat aangeeft wanneer het betreffende bericht verzonden is. Het attribuut status kan twee waarden aannemen, waarbij de waarde publiek de default-waarde is. Wanneer de derde kolom een sleutelwoord bevat, kan een eventueel gewenste default-waarde in een vierde kolom worden opgenomen. Merk overigens op dat niet alle attributen een waarde hoeven te krijgen.

In SGML is het mogelijk verbanden tussen elementen aan te brengen door verwijzingen tussen elementen te creëren. Wanneer men naar een bepaald type elementen wil kunnen verwijzen, is een uniek identificerend attribuut van het type ID nodig. Zo zou de attribuu lijst van e-mail met een dergelijk attribuut uitgebreid kunnen worden tot:

ID #REQUIRED

DATUM #REQUIRED

(vertrouwelijk | publiek) publiek>

Verder is een apart elementtype nodig om de verwijzing te kunnen realiseren (dit wordt in voorbeeld 1 nader toegelicht):

IDREF #REQUIRED>

Veronderstel nu dat we een vertrouwelijk bericht willen sturen, waarbij we in de lopende tekst nog eens nadrukkelijk op deze vertrouwelijkheid willen wijzen, dan kan dat als volgt:

Zoals u kunt zien in de proloog van dit bericht (), dient u de inhoud van dit bericht met niemand te bespreken.

In dit voorbeeld zijn de attribuutwaarden eenvoudigweg ingetypt door de auteur. Men zou zich echter ook kunnen voorstellen dat een applicatie hiervoor ondersteuning biedt. Denk bijvoorbeeld aan het selecteren van de waarde vertrouwelijk of publiek uit een menu.

In voorbeeld 1 is een volledige DTD voor elektronische berichten gegeven. We gaan niet in op alle details, omdat dit buiten het bestek van deze beschouwing valt. De regels zijn genummerd om ernaar te kunnen verwijzen in de tekst. Dit is echter geen onderdeel van SGML zelf.

Voorbeeld 1

```
1 <
2
3
4
5
6 0 (#PCDATA) >
7
8
9
10
11
12
13 id ID #REQUIRED
14 verzenddatum DATUM #REQUIRED
15 status (vertrouwelijk | publiek) publiek>
16
17 idref IDREF #REQUIRED>
18
19 id ID #REQUIRED>
```

Op regel 1 ziet u dat commentaar tussen paren dubbele streepjes () wordt gezet. Overigens is het gebruik van commentaar enigszins verwarrend in SGML: soms wordt er namelijk informatie in het commentaar opgenomen waarvan een applicatie die het document moet presenteren, wel degelijk gebruik moet maken. Verder ziet u op regel 5 dat het blijkbaar toegestaan is meer elementtypen tegelijkertijd te definiëren. Opvallend is verder dat een ref-element blijkbaar leeg is (regel 9). Er is echter wel een attribuut voor ref gedefinieerd (regel 16-17). Hierdoor wordt het dus mogelijk verwijzingen in de lopende tekst van een elektronisch bericht op te nemen. In voorbeeld 2 is zo'n verwijzing (naar een opgenomen figuur) te zien. Tenslotte staat #PCDATA voor een string-achtig datatype, terwijl #NDATA voor niet-SGML-gegevens staat (in dit geval zullen dat bitmaps zijn voor het opnemen van plaatjes).

Een klein bericht dat past in de boven behandelde DTD, is in voorbeeld 2 weergegeven (het feit dat een bepaald document X past binnen een DTD Y wordt in SGML-jargon wel X conformeert zich aan Y genoemd).

Voorbeeld 2

Peter Verkoulen
Henk Blanken
Martijn Hoogeveen
ondergrondse aanleg
minister Jorritsma

Zoals beloofd in mijn bericht d.d. 27 september 1994 stuur ik u hierbij de nieuwe computerfoto's van de constructie welke gebruikt zou moeten worden bij ondergrondse aanleg van deeltrajecten. Met name foto geeft duidelijk nieuwe inzichten.

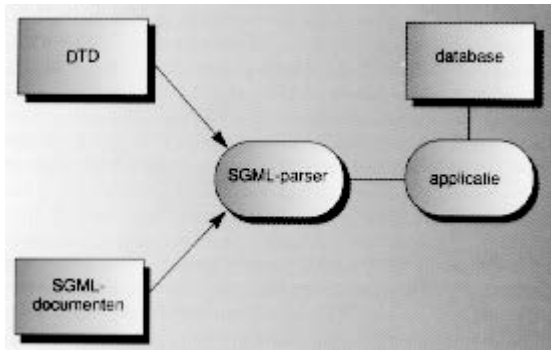
"projectondergronds oto1.bmp

"projectondergronds oto2.bmp

Ik verzoek u uiterlijk 5 november a.s. uw commentaar aan mij toe te sturen.

Een paar opmerkingen zijn op hun plaats. Ten eerste ziet u dat elementen beginnen met een starttag en beëindigd worden met een eindtag. De begintag wordt gevormd door de naam van het element tussen <...> haken; de eindtag bestaat uit de naam van het element tussen haken. Verder worden de eventuele attributen van een element vastgelegd in de betreffende starttag (bijvoorbeeld : dit bericht is blijkbaar verzonden op 19 oktober 1994). Verder ziet u dat elementen op hun beurt weer uit elementen kunnen bestaan: zo bestaat het inhoud-element in dit voorbeeld uit een paragraaf, gevolgd door twee figuur-elementen en tenslotte nog een paragraaf. Volgens de eerder geïntroduceerde DTD bestond een paragraaf-element uit referenties en PCDATA: denk bij dit laatste maar aan de letters en woorden die u hierboven ziet. Verder is in de DTD vastgelegd dat er één of meer geadresseerden moeten zijn; hier ziet u dat het er in dit geval twee zijn, hetgeen eenvoudigweg wordt aangegeven door de twee geadresseerde-elementen. In voorbeeld 2 zien we ook hoe referenties tussen onderdelen van een SGML-document (in dit geval een paragraaf en een figuur) gerealiseerd kunnen worden.

De afsluiting van dit bericht lijkt wellicht een beetje onvriendelijk. De auteur lijkt de geadresseerden niet te groeten en ook een ondertekening is niet terug te vinden. In dit geval zal men echter geredeneerd hebben dat dergelijke zaken niet met de inhoud, maar met het uiterlijk van doen hebben ofwel applicatiespecifiek zijn. Een applicatie voor het opstellen en presenteren van dergelijke e-mail-berichten zou zodanig gemaakt kunnen worden dat een groet en de naam en het fysieke adres van de afzender automatisch worden toegevoegd. Zo hoeft de tekst van de groet nog slechts eenmalig ingevoerd te worden. Per saldo zorgt SGML zo niet alleen voor standaardisatie en uitwisselbaarheid, maar ook voor tijdbesparing voor de gebruiker!



Figuur 1: De verwerking van SGML-documenten

Bovenstaande opmerking over een applicatie voor het opstellen en presenteren van elektronische berichten roept wellicht de vraag op hoe SGML-documenten nu precies verwerkt worden. Figuur 3.2 uit Brouwer (1994) geeft hiervoor het schema dat in figuur 1 van dit artikel is overgenomen.

Dit is in feite een aanpassing van figuur 9 uit Newcomb c.s. (1991). Het idee is dat een SGML-parser controleert of een bepaald SGML-document voldoet aan een gegeven DTD. Zo niet, dan kan de parser aangeven wat er mis is. En als alles in orde is, kan de parser het document in een bepaalde vorm doorgeven. In de database kunnen DTD en document (in een of andere vorm) opgenomen worden, zodat de applicatie hieraan de benodigde informatie kan ontleen. Ook zal deze database bijvoorbeeld de eerder genoemde gegevens voor de automatische toevoeging van een groet en van het adres van de afzender kunnen bevatten. Tenslotte zal de database een DSSSL-style sheet bevatten die de lay-out van het document bepaalt. Over de vrijgelaten aspecten neemt de applicatie zelf een beslissing.

Een ander voordeel van het gebruik van SGML is dat querying aan de hand van de structuur van de documenten mogelijk wordt. Bij documenten in platte tekst kunnen alleen vragen beantwoord worden als: Geef alle documenten die het woord ondergronds bevatten. Voor SGML-documenten kunnen ook vragen beantwoord worden als: Geef alle documenten die verstuurd zijn op 19 oktober 1994 en die foto's bevatten. HyTime breidt dit voordeel uit tot de hypermedia-aspecten van documenten. Een computerhulpmiddel dat het omgaan met SGML (of soortgelijke) documenten ondersteunt zou, naast de eerder genoemde essentiële parser, ook een query manager kunnen bevatten. Merk op dat hier dus een mengvorm van information retrieval en database retrieval mogelijk is. De query over het woord ondergronds is een eenvoudige vorm van information retrieval: de tekst moet doorzocht worden op het voorkomen van het woord ondergronds. De tweede query valt onder database retrieval, omdat hierbij de structuur en de eigenschappen van het document gebruikt worden.

3 HyTime

SGML is met name bedoeld om gestructureerde documenten te maken, met alleen primitieve hypertext-eigenschappen. Dat wil zeggen dat zaken als referenties binnen één

document goed ondersteund worden, maar dat referenties tussen verschillende documenten en (andere) aspecten die een rol spelen bij multimedia-applicaties, niet volledig worden ondersteund. Voor het specificeren van eigenschappen van informatie die niet tekstueel van aard zijn (plaatjes, video, geluid), biedt SGML geen standaardfaciliteiten: ze kunnen alleen atomair opgenomen worden, zoals gedemonstreerd is in het vorige hoofdstuk. Als je willekeurige multimedia-applicaties gaat ontwikkelen, zullen bijvoorbeeld synchronisatie-aspecten aan de orde komen: spreek een begeleidende tekst uit zodra de eerste video is opgestart en vul de spreektekst zonodig aan met achtergrondmuziek, totdat de video uiteindelijk afgespeeld is.

Om aan dit soort wensen tegemoet te komen is ISO in 1992 met de ontwerp-standaard HyTime gekomen. De afkorting HyTime staat voor Hypermedia/Time-based Structuring Language (ISO, 1992). HyTime is gedefinieerd bovenop SGML: het biedt een aantal standaardconcepten en -faciliteiten die met name bedoeld zijn voor de vastlegging van multimedia-applicaties en de dynamische aspecten ervan. Hierbij kan men denken aan faciliteiten om objecten te plaatsen in coördinaatruimten, om objecten te adresseren en om standaardbewerkingen op objecten te definiëren. SGML vormt echter de basis van HyTime. Dat is ook de reden waarom we in de vorige paragraaf al tamelijk uitgebreid op SGML zijn ingegaan.

Nu zult u zich wellicht afvragen wat HyTime dan toevoegt; we hebben immers net gezegd dat HyTime een laag bovenop SGML vormt. Het antwoord op deze vraag is tamelijk eenvoudig. Naast de SGML-manier van documenten structureren, bevat HyTime een aantal concepten die van belang zijn in een hypermedia-omgeving. Deze concepten zijn:

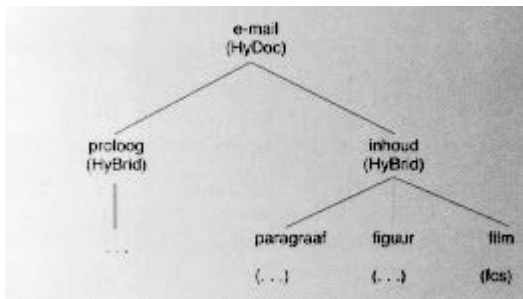
- ondersteuning van willekeurige referenties tussen multimedia-objecten;
- adresseren van willekeurige objecten op willekeurige lokaties;
- coördinaatsystemen met dimensies in de tijd, ruimte en/of met virtuele dimensies die door de gebruiker gedefinieerd zijn, zoals kilowattuur (KWh);
- synchronisatie van multimedia-objecten.

Omdat het structureren van hypermedia-documenten moeilijker is dan het beschrijven van gewone tekst met alleen plaatjes, zorgt HyTime voor een gestandaardiseerde representatie van voornoemde concepten. Deze extra restricties zijn nodig om de toegenomen complexiteit van multimedia-documenten onder controle te kunnen houden.

Wat bedoelen we nu precies met de frase HyTime is gedefinieerd op SGML? Wel, HyTime is gedefinieerd in de vorm van een grote SGML DTD. Deze DTD is echter niet bedoeld om documenten mee te maken, maar om HyTime DTD's mee te maken. Dit is de reden waarom deze HyTime DTD wel de (HyTime) metaDTD genoemd wordt. Wij zullen ons aan deze naamgeving houden, om onderscheid te kunnen maken tussen deze ene door ISO vastgelegde DTD die HyTime definieert en een door de gebruiker gedefinieerde HyTime DTD die een bepaalde klasse multimedia-documenten vastlegt. Bij het opstellen van HyTime DTD's kan gebruik worden gemaakt van standaard-HyTime-faciliteiten, waarvan de semantiek vastgelegd is in de standaard. Verder is in de HyTime-standaard een aantal spelregels vastgelegd, waaraan deze DTD's moeten

voldoen. In de HyTime-literatuur wordt ook wel gezegd dat zo'n DTD zich moet conformeren aan de HyTime-standaard. Met andere woorden, de standaard is een sjabloon waarin elke specifieke HyTime DTD en dus ook elk HyTime-document moet passen. Aan de andere kant zijn de specifieke HyTime DTD's ook gewone SGML DTD's die de SGML-syntax volgen.

In figuur 2 wordt een en ander geïllustreerd. Hiertoe wordt het voorbeeld uit het vorige hoofdstukje over elektronische berichten uitgebreid: het wordt nu ook mogelijk bewegende beelden op te nemen in een elektronisch bericht. Hier geven we de belangrijkste aspecten van deze HyTime DTD alleen in boomvorm weer; in voorbeeld 3 wordt de volledige DTD gegeven.



Figuur 2: De belangrijkste aspecten van de HyTime DTD in de vorm van een boom

We zien in dit voorbeeld dat het e-mail-elementtype gebaseerd is op HyDoc, hetgeen binnen HyTime staat voor HyTime Document.

HyDoc representeert altijd het elementtype op het hoogste niveau in een (HyTime-)documentstructuur. De standaard beschrijft welke knopen onder HyDoc mogen voorkomen. De elementtypen die in figuur 2 onder de knoop e-mail hangen, moeten gebaseerd zijn op zo'n knoop die onder HyDoc mag voorkomen. Wie de standaard bestudeert, kan zien dat we ons in dit voorbeeld daaraan gehouden hebben.

De knopen (elementtypen) in de HyTime-standaard (zoals HyDoc) worden in HyTime-terminologie architectural forms genoemd. Zo'n architectural form heeft een betekenis in termen van multimedia-concepten. In voorbeeld 1 en 2 hebben we een SGML-elementtype ref gedefinieerd, waarmee het mogelijk is verwijzingen tussen onderdelen van één document te realiseren. Door mutatis mutandis het woord ref te vervangen door bijvoorbeeld verwijzing wordt echter precies hetzelfde gerealiseerd. Met andere woorden, de gekozen namen zijn alleen handig voor de gebruiker, maar ze hebben geen onderliggende semantiek. Dit betekent ook dat de gewenste semantiek volledig door de gebruiker gerealiseerd moet worden: er zijn geen standaardfaciliteiten beschikbaar. Voor HyTime ligt dit anders. Zo heeft HyTime bijvoorbeeld een architectural form met de vaste naam ilink. Dit is een afkorting voor independent link. Een ilink wordt gebruikt als hyperlink tussen twee of meer informatie-objecten (die overigens deel mogen uitmaken van verschillende documenten). De HyTime-standaard definieert de semantiek van de architectural form ilink en biedt zo standaardfaciliteiten voor het realiseren van

verwijzingen. Door nu het ref-elementtype uit voorbeeld 1 te baseren op de ilink architectural form bereikt men dat ref de expliciete semantiek van ilink krijgt, in plaats van de impliciete semantiek die alleen in de naam van ref zat. Dit is gebeurd in de DTD die in voorbeeld 3 is weergegeven.

In voorbeeld 3 geven we een deel van de HyTime DTD van figuur 2. In dit voorbeeld is goed te zien hoe de koppeling met de (architectural forms uit de) HyTime metaDTD plaatsvindt. Dit gebeurt namelijk met behulp van attributen. Zo heeft e-mail (net zoals alle andere elementtypen) een extra (verplicht) attribuut HyTime gekregen, met als waarde HyDoc. Hiermee wordt het elementtype e-mail gekoppeld aan de architectural form HyDoc uit de metaDTD. Hierdoor wordt dus onder andere expliciet vastgelegd dat e-mail het hoogste elementtype is in deze DTD. In de SGML DTD uit voorbeeld 1 was dit alleen impliciet vastgelegd, doordat er geen elementtypen gedefinieerd waren waarvan het type e-mail een onderdeel is.

Evenzo wordt de semantiek van ref als verwijzing in onderstaande DTD expliciet vastgelegd door in de lijst met attributen voor ref aan het HyTime-attribuut de naam ilink toe te kennen en gebruik te maken van het attribuut linkends uit de metaDTD, waarmee de verwijzingen uiteindelijk gerealiseerd worden.

Overigens hebben alle elementen van één en hetzelfde type steeds een vaste waarde voor hun HyTime-attribuut, hetgeen aangegeven wordt met het sleutelwoord #FIXED in de derde kolom van de lijst met attributen. In de extra vierde kolom staat de specifieke waarde vastgelegd.

Voorbeeld 3

```
1 <
2
3
4
5
6 0 (#PCDATA) >
7
8
9
10
11 ...
12
13 HyTime NAME #FIXED HyDoc
14 id ID #IMPLIED
15 verzenddatum DATUM #REQUIRED
16 status (vertrouwelijk | publiek) publiek>
17
18 HyTime NAME #FIXED HyBrid
19 id ID #REQUIRED>
20
```

21 HyTime NAME #FIXED ilink
22 linkends IDREF #REQUIRED
23 id ID #IMPLIED>
24
25
26 HyTime NAME #FIXED fcs
27 file #CDATA #REQUIRED
28 id ID #IMPLIED>
29 ...

De HyTime-standaard kent zo'n 90 architectural forms, die zijn onderverdeeld in 6 modules. Elke module staat voor een bepaald cluster multimedia-concepten. Zo ziet u in voorbeeld 3 ook een elementtype film, waarmee op een bepaalde manier moet worden omgegaan. Het elementtype film is hier gebaseerd op het HyTime-elementtype (architectural form) fcs, hetgeen staat voor finite coordinate space. Hiermee kan de semantiek van film vastgelegd worden.

Voor de verwerking van HyTime-documenten stellen Newcomb c.s. (1991, figuur 9), zoals gezegd, een model voor. Hierbij wordt tussen de applicatie en de SGML-parser nog een zogenaamde HyTime engine geplaatst. Nadat de parser zijn werk gedaan heeft, kan de applicatie extra informatie verkrijgen van de HyTime engine. Zo zorgt de engine er bijvoorbeeld voor dat fcs-schedules nageleefd worden en zorgt hij voor het volgen van referenties tussen verschillende informatie-objecten.

4 Toepasbaarheid

Een van de grote voordelen van SGML/HyTime is de ondersteuning van het genereren van documenten door veel partijen tegelijk. Bij het eerdergenoemde Betuwelijn-project is er sprake van een dergelijke situatie. Er zullen duizenden documenten geschreven worden, waartussen enorm veel verbanden zullen bestaan. De betrokken partijen bij dit project zijn onder andere de gebruikers van de spoorweg (de NS), de overheid en een aantal andere investeerders, een onderzoeksbureau voor het schatten van de haalbaarheid en de kosten, een ontwerp bureau, een consortium van aannemers, leveranciers van grondstoffen, vergunningverlenende instanties, belangengroepen.

Deze partijen zullen op alle mogelijke wijzen met elkaar willen/moeten communiceren. HyTime kan helpen deze communicatie te standaardiseren en zo bijdragen aan de totale produktiviteit. Voorbeelden van de ondersteuning die HyTime bij een dergelijk project kan bieden, zijn te vinden bij Brouwer (1994).

Wil HyTime inderdaad de brede mate van toepassing vinden zoals boven geschetst, dan zal aan een aantal voorwaarden moeten worden voldaan. Hierbij moet in ieder geval worden gedacht aan faciliteiten voor de verwerking van HyTime-documenten. Spoedige beschikbaarheid van dergelijke faciliteiten is een levensnoodzakelijke voorwaarde voor HyTime.

Als de software-industrie het eens zou kunnen worden over een standaard, dan zou dit de uitwisseling van documenten in verschillende formaten sterk verbeteren. HyTime zou een geschikte kandidaat kunnen zijn om mee te beginnen. Echter, HyTime is dermate complex dat de ondersteunende computerhulpmiddelen zo goed zullen moeten zijn dat de gebruikers van deze complexiteit niets merken. Als dit niet lukt, zal de standaardisatie-inspanning een doodgeboren kindje blijken te zijn. Overigens is al gebleken dat standaardisatie wel degelijk problemen kan oplossen: bij de opzet van het eerdergenoemde World Wide Web (WWW) wordt gebruik gemaakt van de taal HTML. Onder andere hiermee heeft men de problemen van verschillende soorten machines en dergelijke in het web weten op te lossen (ook het transportprotocol (HTTP) en al bestaande standaarden (zoals TCP/IP) helpen daaraan mee).

Van het Web kunnen we (nog eens) leren dat beschikbaarheid van software en gemakkelijk maken van gebruik eerste voorwaarden zijn voor de acceptatie van bepaalde software of standaarden. Voor HyTime betekent dit dat je moet kijken in hoeverre kleine stukjes ervan binnen het Web kunnen worden ondersteund, en vervolgens daarvoor benodigde software beschikbaar moet stellen. Verder moet een inventarisatie worden gemaakt van de concrete dingen die in het Web niet goed worden ondersteund en waarvoor HyTime een oplossing kan bieden. Dit is zeker een punt voor verder onderzoek, waaraan de WWW-wereld en de SGML/HyTime-onderzoekers samen zouden kunnen gaan werken. Trends voor samenwerking zijn al meer en meer waarneembaar.

Verder lijkt het nog de vraag of de industrie wel bereid zal zijn over te gaan tot de ondersteuning van standaarden. Rada & Carson (1994) geven een aantal redenen aan waarom deze bereidheid wellicht laag zou kunnen zijn. Sommigen zeggen volgens hen dat de hypermedia-markt nog te jong is, waardoor er geen totaaloplossing is, maar verschillende deeloplossingen voor verschillende toepassingen. Anderen geven aan dat de software-leveranciers simpelweg denken dat standaardisatie voor hen economisch niet verstandig zal zijn. Hoe het ook zij, de mogelijkheid van uitwisselen van multimedia-documenten is cruciaal voor het succes van multimedia in de komende jaren.

In het snel evoluerende gebied van multimedia lijkt het verstandig de ontwikkelingen rondom SGML en HyTime goed te blijven volgen. Organisaties die daadwerkelijk problemen hebben met de uitwisseling en gezamenlijke ontwikkeling van (multimedia-)documenten, zouden in een proefproject eens kunnen bekijken of SGML/HyTime voor hen een oplossing zou kunnen bieden. Wanneer WordPerfect en MS-Word SGML (en wellicht later ook HyTime) daadwerkelijk (gaan) ondersteunen, kan het snel gaan met de acceptatie door de potentiële gebruikers, mits er op korte termijn goede hulpmiddelen beschikbaar komen.

Een woord van dank

De auteurs zijn veel dank verschuldigd aan drs. R.D.J. Post van de Technische Universiteit Eindhoven. Zijn commentaar op een eerdere versie van dit artikel heeft de kwaliteit ervan zeker verbeterd.

Verder zijn wij erkentelijk voor de constructieve opmerkingen van de redactie van Informatie en van de gastredacteur van dit nummer, dr. M.J. Hoogeveen.

Literatuur

- Brouwer, Jan J. (1994), The application of HyTime on multi-author documents,afstudeerverslag Universiteit Twente, oktober 1994.
- Chamberlin, D.D. en C.F. Goldfarb (1987), Graphic applications of the Standard Generalized Markup Language (SGML),Computer Graphics, jaargang 11, nr. 4.
- Goldfarb, C.F. (1990), The SGML Handbook,Oxford University Press.
- Herwijnen, E. van (1990), Practical SGML, Kluwer Academic Publishers, Dordrecht.
- ISO (1986), Information Processing Text and Office Systems Standardized Generalized Markup Language (SGML), ISO 8879-1986 (E), International Organization for Standardization.
- ISO (1991), Document Style Semantics and Specification Language (DSSSL), ISO/IEC 10179-1991 (E), International Organization for Standardization.
- ISO (1992), Information Technology Hypermedia/Time-based Structuring Language (HyTime), ISO/IEC 10744-1992 (E), International Organization for Standardization.
- Newcomb, Steven R., Neill A. Kipp en Victoria T. Newcomb (1991), The HyTime Hypermedia/Time-based Document Structuring Language,Communications of the ACM, Vol. 34, Nr. 11 (november 1991), blz. 67-83.
- Rada, Roy en George S. Carson (1994), The New Media,Communications of the ACM, Vol. 37, Nr. 9 (september 1994), blz. 23-25.

Dr. Henk M. Blanken is als UHD verbonden aan de Faculteit der Informatica aan de Universiteit Twente.

Dr. ir. Peter A.C. Verkoulen is als senior IT-Specialist verbonden aan de Adviesdienst Verkeer en Vervoer van Rijkswaterstaat.

Daarvoor was hij UD bij de Faculteit der Informatica aan de UT.

Correspondentie-adres: AVV, t.a.v. Peter Verkoulen, Postbus 2510, 6401 DA Heerlen, tel. 045-(5) 735 279, e-mail P. Verkoulen @ cs.utwente.nl.