Kennisgebaseerde uitwisseling van informatie in nieuwsproductie

Interoperability of Semantics in News Production

Erik Mannens

Promotor: prof. dr. ir. R. Van de Walle
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen: Computerwetenschappen

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. J. Van Campenhout
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2010 - 2011

UNIVERSITEIT
GENT

ISBN 978-90-8578-415-9
NUR 988
Wettelijk depot: D/2011/10.500/19

This One’s for You,
This One’s for Me,
This One’s for Anyone who ever Loved Someone.

(Ozark Henry)

And we never give up ...
It’s all about never giving in ...
And we never give up ...
It’s about the will who wants the most ...
And we never give up ...
It’s all about having the heart of a Champion!

(Paul Pierce - The Truth)

Dankwoord

Een ‘mid-life crisis’ op je veertigste? De avontuurlijke kerels onder ons kopen zich een zware motorfiets; degenen die zich wanhopig aan hun jeugd proberen vastklampen denken dat een maîtresse soelaas zal bieden; de saaie rakkers onder ons beginnen te tuinieren en dan blijven nog de ‘nerdy’ buitenbeentjes over die hun eigen auto willen bouwen of aan een doctoraat beginnen schrijven. Het ‘echte’ leven zit echter gecompliceerder in elkaar dan dit, waardoor mensen over het algemeen niet in één welbepaald hokje te duwen zijn. Ik, bijvoorbeeld, had een drang! Een drang naar kennis, een drang om uit te blinken, een drang om af te zien. Ik denk dat elke doctorandus deze symptomen, die nodig zijn om deze queeste tot een goed einde te brengen, wel (h)erkent. Wie me kent, zal glimlachend beamen dat dit typische kenmerken zijn van mijn karakter. ‘It’s beyond my control’ (Vicomte de Valmont, Dangerous Liaisons, 1988) (lees: ‘het is sterker dan mezelf’) om steeds opnieuw het nieuws te willen beluisteren/bekijken, om (onnozele) feiten te leren/debiteren, om om het even welk spelletje/quiz waaraan ik deelneem te willen winnen, en (helaas) om alledaagse woorden en afspraken te vergeten. Ik weet dat dit soms erg vermoeiend/frustrerend kan zijn voor de mensen waar ik echt van hou. Mijn oprechte excuses, mijn lieve schatjes. Laat het ons nu even over het afzien hebben. Ik hou ervan om mezelf fysiek en mentaal af te peigeren. Vooral de roes juist na het afwerken van een wedstrijd/job is verslavend. Fysiek had ik steeds die kick tijdens mijn basketbalcarrière, die helaas al lang achter de rug is. Mentaal heb ik dat nu wanneer ik artikels en projectvoorstellen kan afwerken.

Wat is Kunst? Het schrijven van een ‘perfect’ artikel, dat is Kunst! Je hebt een brilliant idee. Je werkt dit uit en schrijft er een artikel over. Je collega’s lezen het en vertellen je dat het een goed idee is, maar dat er toch wel een paar kleine foutjes in zitten die best opgelost worden. Met frisse moed werk je het idee verder uit en herschrijf je je artikel. Je bent uitermate tevreden, het is inderdaad veel beter nu. Tijd om het te laten reviseren door je promotor. Die vindt het wel een leuk idee, maar er zal nog veel water door de Schelde moeten vloeien vooraleer het matuur genoeg is om op te sturen naar een respectabel tijdschrift. Met nieuwe inzichten en hernieuwde moed herwerk je alles en je herschrijft het artikel volledig. Op dit ogenblik denk je dat dit zowat het beste artikel der tijden is en je stuurt het dan ook op naar een 'top' tijdschrift. Na een half jaar melden de revisoren je fijntjes dat het artikel toch niet zo uitmuntend is en dat het slechts in aanmerking kan komen voor publicatie als je er nog twee ideeën kan in verwerken die minstens dubbel zo vernieuwend zijn als je oorspronkelijke idee. Nu sta je op een tweesprong! Je glijdt weg in een depressie en begint alsnog te tuinieren, of je herstelt van deze mokerslag, vecht terug en volhardt in de studie met een nog grotere doortastendheid ... en dat is het moment waarop je twee nieuwe ingevingen krijgt! Als je artikel na nog een paar iteraties uiteindelijk gepubliceerd wordt, mag je het een echt Kunstwerk noemen! Je bent nu helaas twee jaar ouder, maar inderdaad wel heel wat wijzer. Sindsdien heb ik tijdens het schrijven van een resem andere artikelen dan ook nog ettelijke malen geluisterd naar 'You just haven't earned it yet, baby, you must suffer and cry for a longer time' (The Smiths, The World won't Listen, 1986) (lees: 'man, je hebt het nu nog niet verdiend, je zal nog wat langer moeten wenen en afzien') terwijl ik constant heen en weer geslingerd werd van het Walhalla naar Echternach en terug.

Hoe overleeft men 1200 eenzame werknachten zonder ravissante Sarah? 3 woorden: muziek, humor, en basketbal. Mijn neuronen maakten 's nachts de meest fantastische verbindingen als ik naar Klara Continuo luisterde. Mijn beste ideeën borrelden immers op bij 't horen van repetitieve renaissance-deuntjes. Als mijn brein even dreigde de pijp aan Maarten te geven, herbronde ik het door even te kijken naar Britse comedy. 't Zou geen leven zijn zonder Black Adder, The Fast Show, Little Britain en consorten! Dit is dan ook de enige verklaring voor de quotes die elk hoofdstuk vooraf gaan, immers 'There's no life without humour! It can make the wonderful moments of life truly glorious, and it can make tragic moments bearable' (Rufus Wainwright) (lees: 'Het is geen leven zonder humor! Het maakt de mooie momenten in het leven echt grandioos en het verzacht onze tragische levensmomenten'). En dan is er nog basketbal ... telkenmale ik in de Verenigde Staten vermoeiende standaardisatieworkshops bijwoonde, probeerde ik 's avonds (live of op TV) een NBA-wedstrijd mee te pikken. Zo heb ik in 2007 persoonlijk de heropstanding van de Boston celtics meegemaakt door toedoen van 'The Big Three', id est: 'The Big Ticket' (Kevin Garnett), 'The Truth' (Paul Pierce), en 'Jesus' (Ray Allen). Ook thuis bekeek ik 's nachts al 'spinnend' NBA-wedstrijden terwijl mijn lichaam het even overnam van mijn geest (al moet het gezegd zijn dat mijn lichaam de laatste 4 jaren veel minder oefening gekregen heeft dan mijn brein ... vanaf nu zal ik de weegschaal wat in de andere richting laten uitslaan, mijn lieveling :). Enkel deze mentale en fysieke oefeningen/relaxaties maakten het fysieke gemis aan Sarah nog net draagbaar.

Hoe bedank je iemand zonder al te sentimenteel te worden? Dit boekdeel zou je nu niet in je handen houden zonder de goede begeleiding van mijn promotor prof. dr. Rik Van de Walle. Vanaf mijn eerste werkdag bij MMLab kreeg ik van hem het volste vertrouwen en een ongelooflijke vrijheid om mezelf verder te ontplooien en geestelijk te 'verrijken', onder meer door MMLab op de kaart te zetten binnen het W3C-standaardisatieconsortium en als evangelist van het Semantische Web. Verder wil ik ook Raphaël Troncy bedanken om me te inroduceren in de wondere wereld van W3C en om samen met mij 'onze' Media Fragments groep voor te zitten. Ik hoop van harte nog lang met prof. dr. Rik Van de Walle en Raphaël Troncy te kunnen samenwerken. Dit werk zou vandaag nog niet af zijn zonder de ideeën/hulp van mijn collega's –zowel binnen als buiten MMLab– in de gezamenlijke projecten. In alfabetische volgorde zijn dit: Olivier Braet, Sam Coppens, Hendrik Dacquin, Andro Debevere, Pedro Debevere, Jan De Cock, Toon De Pessemier, Robbie De Sutter, Tom Evens, Laurence Hauttekeete, Peter Lambert, Ellen Lammens, Gaëtan Martens, Stijn Notebaert, Luk Overmeire, Chris Poppe, Davy Van Deursen, Wim Van Lancker, Dieter Van Rijsselbergen, Ruben Verborgh en Maarten Verwaest. Het feit dat ik de meeste kan aanspreken met 'mijn vriend' toont zonder enige twijfel dat ze intelligent, grappig en interessant zijn. Ook een oprechte dankbetuiging aan mijn dichte vrienden en familie voor de constante bevestiging en schouderklopjes, alsook voor de vele fijne avonden om mijn zinnen wat te verzetten. Blijf dit maar volhouden ;). Ik druk hierbij ook voor eens en voor altijd :) mijn appreciatie en liefdevolle warmte uit voor ma en pa, die mijn broer en ik zonder meer in poleposition geplaatst hebben na pakweg (hopelijk) $\frac{1}{4}$ van deze verrassende reis, 'Leven' genaamd. Tenslotte zijn we aanbeland bij 'Le moment Suprème' ... tromgeroffel ... boem, paukeslag (Paul Van Ostaijen, Bezette stad, 1921) ... ik kan mijn allesomvattende liefde voor mijn mooie Sarah amper in woorden uitdrukken. Deze '15 minutes of fame' zijn van ons beiden (!!!) maar verzengen in het niets bij het feit dat ik al ontelbare uren verliefd en gelukkig heb mogen doorbrengen met jou en onze drie (b)engeltjes Hanne, Robbe en Arne. Zonder al jullie hulp had ik dit huzarenstukje waarschijnlijk niet tot een goed einde gebracht (althans niet in deze korte tijdsspanne :). Hoedje af en diepe, nederige buiging !!!

Ik draag dit werk dan ook op aan:

- ‘Zie je wel!’ (ma & pa)
- ‘Als je ergens je zinnen op zet, en je geeft niet op ... dan lukt het wel –bis repetita placent– dan lukt het zeker!’ (mijn schatjes Hanne, Robbe en Arne)
- ‘Schatje, ’t beste moet nog komen ... enne ... ik ga nog wel een auto bouwen’ ;) (mijn diamant Sarah)

Erik Mannens
Maart 2011

Thanks

Now, how does one begin? Turning 40, a mid-life crisis? The adventurous ones buy a motorcycle, the feeling-sorry-for-themselves-ones have a young mistress, the dull ones just start gardening, and the nerdy ones build their own car or write a PhD. Real life, however, is much more complicated than this, so people cannot be just put into ready-made boxes like that. I, for one, just had the urge! An urge for knowledge, an urge to excel, an urge to suffer. All PhD students probably recognise these symptoms –one definitely needs to succeed– in this quest. For those who know me, the urge for knowledge and the urge to excel are a natural extension of my character. ‘It’s beyond my control’ (Vicomte de Valmont, Dangerous Liaisons, 1988) to always read/watch the news, learn/quote (silly) facts, and to win any kind of game/quiz I participate in and (alas) to forget everyday words and/or appointments. Sometimes that is very tiring/frustrating for the people I truly love. Sorry about that, my precious! Come to the suffering part. I tend to take great satisfaction in both physical and mental exhaustion, especially in the moments just after finishing a game/job, most likely in victory, sometimes in defeat. Physically I had that during my long lasting basketball ‘sporting career’ long-time ago, now I have that when biking with my friends. Mentally I nowadays have that when writing papers and research proposals.

Coming to the writing of perfect papers as an Art form. You have a brilliant idea, so you elaborate on it and write a paper. Your peers read it and tell you it’s a good idea, but there are some minor flaws that are better taken care of. So you rework your stuff and rewrite your paper. You’re content! Indeed, it is much better. Now it’s time for your supervisor to read it: It’s an idea all right, but there’s still much work to be done before it can be sent for review to a journal. So once again, you completely revise your work and you rewrite your paper once again. By now, you think it is the best paper ever written and you submit it to a journal. You wait at least half a year to discover that the reviewers don’t like it and that it can only be considered for publication if you add another two ideas to it that are twice better than your original one. At that moment in time you basically have two options: you get depressed, surrender, and start gardening after all, or you suffer, fight back, and try again with even more determination ... and that's when you get your two new ideas! If it gets approved after several more iterations, you may call it a true piece of Art! Now, unfortunately it just takes you about two years to come to that conclusion. Since then I've listened numerous times to the song 'You just haven't earned it yet, baby, you must suffer and cry for a longer time' (The Smiths, The World won't Listen, 1986) while being warped from Echternach to Walhalla, and back.

Now how did I survive the lonely working nights between 2007 & 2010 without my lovely Sarah: 3 words ... music, humour, and basketball. My brain was most vivid at night when listening to Klara Continuo. Especially the Renaissance repertoire made my neurons sparkle the brightest ideas! When utterly tired, I could revive my brain by watching BBC comedy. Thank God for Black Adder, The Fast Show, Little Britain, and the like, hence the quotes at the beginning of each chapter, because 'There's no life without humour! It can make the wonderful moments of life truly glorious, and it can make tragic moments bearable' (Rufus Wainwright). And then there's basketball ... When attending exhausting standardisation workshops in the States, going to watch live NBA games in the evening really made my day! I personally experienced the resurrection of the Celtics franchise together with 'The Big Ticket' (Kevin Garnett), 'The Truth' (Paul Pierce), and 'Jesus' (Ray Allen) –aka 'The Big Three'– back in 2007 and since. Back at home I watched numerous NBA games at night while working out on my spinning bike (although my body got much less exercise than my brains ... from now on I will work on that one too, my dear). All this mental and physical exercise/relaxation at night made the missing physical presence of Sarah only just bearable.

Now comes the most difficult part, being thankful without becoming too sentimental and most importantly without forgetting someone. First of all, this work would not have been possible without the support and guidance from my supervisor Prof. Dr. Rik Van de Walle. From the first day I was around, he believed in me and he gave me a lot of freedom degrees I could fill in myself, e.g., starting up Multimedia Lab's W3C standardisation activities and evangelising the Semantic Web to name just two. Furthermore, I profoundly thank Raphaël Troncy for introducing me into the world of W3C and co-chairing 'our' Media Fragments group; I hope we can keep working together within W3C and beyond. This work would also not have been finished today without the ideas/work of some of my co-workers both within Multimedia Lab and the projects I worked on. In alphabetical order these are: Olivier Braet, Sam Coppens, Hendrik Dacquin, Andro Debevere, Pedro Debevere, Jan De Cock, Toon De Pessemier, Robbie De Sutter, Tom Evens, Laurence Hauttekeete, Peter Lambert, Ellen Lammens, Gaëtan Martens, Stijn Notebaert, Luk Overmeire, Chris Poppe, Davy Van Deursen, Wim Van Lancker, Dieter Van Rijsselbergen, Ruben Verborgh, and Maarten Verwaest. The fact that most of them I call 'my friend' shows without any doubt their intelligence, wit and interesting personality. Thanks to all my close friends and family, for just being around when needed (or not ;). Furthermore, my deepest love and appreciation to both my mum and dad, who were able to still put me into poleposition after almost $\frac{1}{4}$ of this bumpy journey called Life. Finally, and most importantly boem, paukeslag (Paul Van Ostaijen, Bezette stad, 1921), I cannot put my love and appreciation into words for my lovely Sarah, who stood by me for the last decade. These '15 minutes of fame' are for both of us (!!!), but fade away in the numerous hours of joy and happiness I had until now together with you and our three precious ones you gave birth to. Without all of your support this was probably not possible, or at least not in this timely manner. My deepest appreciation!

I especially dedicate this work to:

- 'I told you so' (mum & dad)
- 'If you want something badly, and you don't give in ... you will, I repeat, you will get it! The sky isn't even your limit' (my precious ones Hanne, Robbe, and Arne)
- 'I know, the best is yet to come and I will build that car' ;) (my diamond Sarah)

Erik Mannens
March 2011

Samenvatting

Homo animal curiosum est! Een mens is inderdaad een nieuws-gierig dier: iedereen leest, luistert of bekijkt dagelijks nieuwsberichten. We staan er zelfs niet meer bij stil dat we dit eigenlijk altijd en overal doen: thuis, van/naar het werk en op het werk –velen zelfs als onderdeel van hun jobinhoud–. Als rechtgeaarde Europese burger is het inderdaad goed alle standpunten van de lokale, nationale en internationale politiek te kennen om naderhand de juiste stem te kunnen uitbrengen. Als loyale werknemer kennen we best de huidige en voorspelde trends op de lokale, nationale en internationale economische markten om naderhand met kennis van zaken de juiste beslissingen voor het bedrijf te kunnen nemen. Als fanatieke supporter willen we tijdens onze vrije tijd constant op de hoogte blijven van het reilen en zeilen bij onze favoriete ploeg, of willen we als fan op de hoogte blijven van de (privé) perikelen van onze idolen. Tegenwoordig is al deze informatie online beschikbaar en derhalve –mits het bezit van de juiste toestellen– dus overal bereikbaar.

In de huidige workflow van nieuwsprocessen worden nieuwsberichten typisch geproduceerd door nieuwsagentschappen, onafhankelijke journalisten of wakkere/geëngageerde burgers, waarna deze geconsumeerd en verrijkt worden door uitgevers (kranten en magazines) of omroepen (radio en televisie). Via kranten, magazines, radio en/of televisie worden deze nieuwsberichten dan uiteindelijk gelezen, beluisterd en/of bekeken door de eindgebruikers, die naderhand hun expressieve meningen over de aangeboden en verwerkte informatie terug als sporen achterlaten op het Internet door middel van blogberichten en/of tweets. Mochten nieuwsberichten aldus gedurende hun gehele levenscyclus ten gepasten tijde met de juiste beschrijvingen en metadata geannoteerd worden, dan zou het terugvinden van bepaalde nieuwsfeiten volgens de juiste chronologie veel gemakkelijker zijn. Helaas gaat er nu nog veel metadata verloren tijdens de nieuwsproductieworkflow wegens de incompatibiliteit van de verschillende productiesystemen die de (meta)data moeten uitwisselen, waardoor er kansen onbenut blijven om de eindgebruiker

de nodige extra informatie te verschaffen. Bijgevolg worden de eindgebruikers overspoeld door te veel individuele en losse stukjes informatie, waardoor men moeite heeft om het nieuws in de juiste context te plaatsen. Eén omvattende nieuwsgebeurtenis wordt gedefinieerd door een bundel statistisch gerelateerde nieuwsberichten, maar tot op heden wordt geen ontologisch begrip van zo een omvattende 'gebeurtenis' ondersteund. Bij de huidige nieuwsaanbieders draait daarentegen alles rondom primeurs en geplande nieuwsberichten, maar men mist de mogelijkheid om het nieuwe geproduceerde nieuws gemakkelijk te verbinden met de berichten uit het verleden die ze op dagdagelijkse basis beheren. Het semantisch verrijken en verbinden van nieuwsinformatie –uit heterogene bronnen, in verschillende media types en/of in verschillende talen– kan er evenwel voor zorgen dat individuele nieuwsberichten gebundeld worden tot betekenisvolle 'gebeurtenissen' die onderling verbonden zijn met extra achtergrondkennis.

Tijdens het lezen van dit proefschrift zal duidelijk worden dat *kennisgebaseerde uitwisseling van informatie* essentieel is tussen de verschillende workflow fases in nieuwsproductie. Eerst en vooral moet het mogelijk zijn om met alle échelons van een mediabedrijf te discussiëren over een probleem door middel van gezamenlijke concepten, waardoor de '*begripsverwarrende semantische kloof*' die historisch bestaat tussen het bedrijfsmanagement en de productieingenieurs gedicht wordt. Daardoor zal de workflow tussen de verschillende entiteiten van de mediaorganisatie geharmoniseerd worden, wat uiteindelijk zal leiden tot een technisch goed onderbouwde oplossing die iedereen snapt. Vervolgens moet de metadata zo snel mogelijk geïdentificeerd en semantisch beschreven worden door middel van standaarden, waardoor de '*technische semantische kloof*' verder gedicht kan worden. Daardoor zullen de verschillende interne systemen ergens in de workflow pijplijn, evenals externe Internet Webservices, voordeel kunnen halen uit deze extra kennis. Tenslotte moet de vergaarde kennis, i.e. de onderling verbonden nieuwsberichten, veilig voor de toekomst bewaard kunnen worden, waardoor de '*continuïteits semantische kloof*' gedicht wordt, aangezien metadata (vb. herkomstdata) van metadata essentieel zal blijken te zijn om te komen tot blijvende interoperabiliteit.

In een eerste hoofdstuk pakken we de '*begripsverwarrende semantische kloof*' aan. Daarin zijn we de eerste om van begin tot einde de nieuwsproductieprocessen te aligneren met de canonische mediaprocessen, waardoor alle échelons van een omroeporganisatie de procescycli van hun systemen kunnen begrijpen in de context van de meer generieke, standaard procescycli van bestaande mediasystemen. Aldus wordt deze voornoemde ‘*begripsverwarrende semantische kloof*’ gedicht. Deze canonische processen helpen de complexe interactie van de workflowprocessen ten opzichte van de verschillende verantwoordelijkeidsniveaus binnen een organisatie te verduidelijken. Bovendien kan men toekomstige scenario’s voorzien waar bepaalde processen in een productiesysteem kunnen uitgewisseld worden met deze van andere mediaproductiesystemen, vb. omroepen en productiehuizen, die de voormelde generieke principes reeds implementeren. Als extra resultaat hebben we een raamwerk opgezet dat duidelijke interfaces voorziet voor de informatiestromen tussen de mediaprocessen van verschillende nieuwsproductiefases, waardoor de systemen van verschillende media-aanbieders compatibel worden. Door het identificeren van recurrente canonische functionaliteit kunnen procesimplementaties vereenvoudigd worden en kunnen de input/output van de verschillende processen gecoördineerd worden zodanig dat het beter integreert met externe systemen.

In een volgende hoofdstuk pakken we de ‘*technische semantische kloof*’ aan. Daarin stellen we een semantische versie van de NAR/NewsML-G2 standaard voor als een unificerend (meta)datamodel voor dynamisch gedistribueerde informatie van nieuwsgebeurtenissen. Door deze ontologie van begin tot einde te gebruiken als een datacommunicatieinterface in een nieuwsdistributiearchitectuur, kunnen diverse diensten (aggregatie, categorisatie, verrijking, profilering, aanbeveling en distributie) in een workflowraamwerk ingebed worden waardoor de omroepen voor het eerst een manier gekregen hebben om automatisch (ontspinnende) nieuwsberichten 1-op-1 aan te bevelen aan een welbepaald doelpubliek. Op hetzelfde moment voorzien we de (inter)nationale (nieuws)gemeenschap van technieken om nieuwsgebeurtenissen en profielinformatie te beschrijven en uit te wisselen op een gestandaardiseerde manier, waardoor de ‘*technische semantische kloof*’ verder gedicht kan worden. Als extra resultaat demonstreren we het concept van generieke dataporteerbaarheid van gebruikersprofielen en hoe we aanbevelingen kunnen genereren op basis van zo ’n globaal profiel (waarin we stukjes informatie integreren van alle verscheidene sociale netwerksites die de gebruiker wil delen). Al onze ideeën werden geïmplementeerd door gebruik te maken van de open standaarden *OpenID*, *OAuth* en *OpenLike* waardoor onze architectuur vrij toegankelijk is voor andere nieuwsgebeurtenis- en/of profielaanbieders.

In een daaropvolgend deel pakken we de ‘*continuïteits semantische kloof* aan. Daarin identificeren we alle types metadata die nodig zijn voor de langetermijnpreservatie van digitale (nieuws)informatie. *Beschrijvende* *metadata* zijn nodig om de intellectuele entiteiten te verduidelijken. Verder zijn ook *binaire metadata*, *technische metadata* en *structurele metadata* essentieel om de data op alle lagere niveaus te beschrijven, i.e. op bitstroom-, op bestands- en op representatieniveau. Ook *preservatiemetadata* is onontbeerlijk om de herkomst van de data te beschrijven, om de authenticiteit van het digitale karakter van de data te garanderen, en om de algemene context van de data te borgen. Tenslotte moet ook de *rechtenmetadata* opgeslagen worden. We stellen een semantisch, tweelagig metadataschema voor die de vrijheid biedt om al deze types metadata te omvatten binnen de archiveringscontext. We zijn de eersten om zo'n generieke architectuur uit te denken voor zowel 'vrije toegang' als 'duurzame archivering', twee begrippen die tot voor kort in deze context steeds als orthogonaal en dus incompatibel beschouwd werden. De toplaag van ons vooropgestelde schema (de RDFS representatie van DC) omvat de beschrijvende metadata waardoor het mogelijk wordt om multimediale data –uit diverse domeinen met heterogene metadata– uit te wisselen. De grondlaag van ons metadataschema (de officiële OWL-representatie van PREMIS 2.0, goedgekeurd door het PREMIS-standaardisatiebestuur en Library of Congress) omvat daarentegen de nodige binaire, technische, structurele, preservatie- en rechtenmetadata om duurzame archivering te borgen. Door de data aan de hand van dit gelaagd metadatamodel te beschrijven, minimaliseren we de risico's van duurzame archivering, waardoor de voornoemde '*continuïteits semantische kloof*' gedicht kan worden. Aangezien het semantische metadataschema opgesplitst wordt in twee lagen, zorgt de toplaag met de beschrijvende metadata voor de publieke, vrije toegang en het verstrengelen van de data in het semantische Web van data op het Internet –indien de rechten dit toelaten natuurlijk–. De grondlaag kan dan weer (indien nodig) afgeschermd worden van publieke toegang en is daarbovenop verantwoordelijk voor de duurzame archivering van de data.

In het voorlaatste hoofdstuk wordt video opgewaardeerd tot een 'eersteklas burger' op het Internet door middel van een specificatie voor mediafragmenten. Het spreekt voor zich dat deze specificatie voor mediafragmenten een grote impact zal hebben op de complete nieuwsproductieworkflowketen. Van het moment dat multimediaal nieuwsmateriaal wordt opgenomen, geëditeerd en verrijkt met extra informatie tijdens elk specifiek proces gedurende de volledige productieketen, tot het moment dat een geïnteresseerde eindgebruiker een specifiek gekozen nieuwsfeit bekijkt, is het nu mogelijk om subclips van mediabronnen uniek te identificeren, te linken, te tonen, te browsen, te bookmarken, samen te stellen, te annoteren, en temporeel/spatiaal aan te passen. Een camera zou bijvoorbeeld automatisch juist opgenomen materiaal kunnen annoteren met de exacte geocoördinaten. Een nieuwsredacteur op zoek naar een specifiek nieuwsbericht zou bijvoorbeeld snel kunnen browsen doorheen het gezamenlijk materiaal van een paar maanden door gebruik te maken van de onderschriften van de hoogtepunten. Een eindgebruiker, daarentegen, zou een videocollage kunnen maken van zijn favoriete gebookmarkte videosegmenten die hij wil delen met zijn vrienden via een sociaal netwerk. Aldus bakenen we de gebruiksruimte van een mediafragmenten URI af en schetsen daarbovenop de syntax en de semantiek daarvan. Verder werkten we stapsgewijs uit hoe een mediafragment, gespecifieerd als een *URI Fragment* of een *URI Query*, kan worden beschreven en opgevraagd door middel van het HTTP-protocol. Tenslotte identificeren we ook wat de consequenties zijn bij het gebruik van de huidige mediaformaten op deze fragmentextractie. We zijn de eersten die een HTTP-implementatie hadden voor W3C's specificatie 1.0 voor mediafragmenten –zowel een clientimplementatie (i.e. Firefox-plugin) als een serverimplementatie (i.e. Ninsuna-module)– zodanig dat alle gedefinieerde testscenario's gevalideerd konden worden. Zodoende wordt uiteindelijk de nodige ondersteuning verschaft aan de reeds alomtegenwoordige derde dimensie 'tijd' op het Internet.

In een laatste hoofdstuk wordt al de opgebouwde kennis uit voorgaande hoofdstukken getoets aan de praktijk door middel van de use case voor de complete nieuwsproductie. Demonstratoren uit de projecten Archipel, BOM-Vl, CUPID en PISA (opgelijst in alfabetische volgorde aangezien ik elk project een even warm hart toedraag) werden (deels) aangepast om dit proefschrift te ondersteunen. Derhalve belichten we achtereenvolgens een use case voor het 'zoeken', een use case voor de 'distributie', een use case voor de 'archivering' en een use case voor de 'mediafragmenten' in een complete nieuwsproductieomgeving, gebruikmakend van alle besproken concepten uit voorgaande hoofdstukken.

Summary

News is something nearly every European citizen reads, watches, or listens to on a daily basis, at home, while commuting to and from work, at work, and even as part of their work. As voting citizens, we need to understand local, national, and international politics to allow us to cast our vote. As company employees, we need to understand the state and development of local, national, and international economies to enable us to understand our markets. As part of our leisure time, we want to know about our favourite sports teams, the lives of our soap idols, or the most recent books available. Nowadays, this information is on-line, and hence accessible from anywhere. In existing news workflow processes, news items are typically produced by news agencies, independent journalists or citizen media; afterwards consumed and enhanced by newspapers, magazines or broadcasters; then delivered to end-users; and finally perceived by these end-users that further leave a trace of what they have understood and felt facing these news events using blogs and tweets as means of expression. News items should therefore typically be accompanied by a set of metadata and descriptions that facilitate their storage, retrieval, and life cycle. However, much of the metadata is lost because of interoperability problems occurring along the production workflow. As such, opportunities for making use of the available metadata at the user interface level are often lost. Consequently, users are overwhelmed by too many individual and disconnected pieces of information, and cannot situate the news in a proper context. A news event is defined as a cluster of statistically related news items, but no ontological notion of event is supported. In contrast, the organisation of news providers is centred on the notion of scheduled and breaking news events, but they currently lack the tools necessary to easily relate the news they produce to the events they manage on a daily basis. Semantic processing of news information can improve the clustering and organisation of individual news items –from heterogeneous sources, in multiple media types and in multiple languages– into meaningful events linked to appropriate background knowledge.

In this thesis, we state that *Interoperability of Semantics* is key throughout the different workflow phases of news production. Firstly, within each media company all ranks of the organisation should be able to reason about a problem across levels given joint concepts, thereby closing the '*comprehensive semantic gap*' that historically exists between business management and technical engineering, thus smoothing the workflow between the organisation's entities and leading to a technical sound solution understood by everyone. Secondly, metadata should be identified, captured, and standard-based semantically described as soon as possible, thereby closing the '*technical semantic gap*' as different in-house systems down the workflow pipeline, as well as third party Internet Web services, will be able to harness this extra knowledge to their own favour. Finally, the gathered knowledge, i.e., the interlinked news items, should be archived in a future-safe way, thereby closing the '*continuity semantic gap*' as metadata, e.g., provenance data, about metadata will be key to long-lasting interoperability.

In a first chapter we tackle this '*comprehensive semantic gap*', as we are the first to align the end-to-end news production processes with the canonical media processes, making all ranks within a broadcast organisation understand the process cycles of their systems in the context of the more generalised, standard process cycles of existing media systems, thus closing the aforementioned '*comprehensive semantic gap*'. These canonical processes help us in clarifying the complex interleaving of workflow processes on the different levels of responsibilities within an organisation. Furthermore, one can envisage future scenarios where some of the processes within a system can be exchanged with those from other media production systems that adhered to aforementioned generic principles, e.g., production houses and local broadcasters. As such, we set up a framework to establish clear interfaces for the information flow across media processes among distinct news production phases so that compatibility across systems from different providers can be achieved. By identifying such recurring and canonical functionality, process implementations are simplified and input and output from different processes are coordinated for better integration with these external systems.

In a following chapter we tackle the '*technical semantic gap*', as we present a semantic version of the NAR/NewsML-G2 standard as a unifying (meta)data model dealing with dynamically distributed news event information. Using that ontology as a data communication interface within an end-to-end news distribution architecture, several services (aggregation, categorisation, enrichment, profiling, recommendation, and distribution) are hooked in the workflow engine giving broadcasters a tool to automatically recommend (developing) news stories 1-to-1 to the targeted customer for the first time. At the same first time, we provide the (inter)national (news) community with mechanisms to describe and exchange news events and profile information in a standardised way, thus contributing to the narrowing of the aforementioned '*technical semantic gap*'. We demonstrate the concepts of generic data portability of user profiles, and how to generate recommendations based on such a global profile –within which we integrate information fields from all the different social networks the user wants to share. All our ideas are implemented with open standards like *OpenID*, *OAuth*, and *OpenLike*, thus keeping the architecture open for other news event/profile providers.

Another chapter down my thesis workflow line tackles the '*continuity semantic gap*', as we identify the necessary types of metadata that need to be retained when preserving digital (news) information for the long-term. Descriptive metadata are needed to describe the intellectual entities, whereas binary metadata, technical metadata, and structural metadata are essential for the description of the data on all lower levels (bitstream, file, and representation). Preservation metadata is also necessary to describe the provenance of the data, to guarantee the authenticity of its digital nature, and to provide a context. Lastly, rights metadata also need to be stored. We propose a two-layered semantic metadata schema that offers the freedom to embrace all of these metadata types within an archiving context. We are the first to come up with this generic architecture for both open access and lasting archive, which were until recently considered orthogonal and not compatible. Our top layer (the RDFS representation of DC) takes care of the descriptive metadata and is also usable to initiate the exchange of multimedia data from different domains with non-homogeneous metadata. Our bottom layer, which is by now the official OWL representation of PREMIS 2.0 (acknowledged by both the PREMIS standardisation board and Library of Congress), on the other hand, encompasses the needed binary metadata, technical metadata, structural metadata, preservation metadata, and the rights metadata for future-safe archiving. By describing the data with this layered metadata schema, all the risks that come with long-term preservation are minimised, thus contributing to the narrowing of the aforementioned '*continuity semantic gap*'. By splitting up the semantic schema in two layers, the top layer with the descriptive metadata can be made public and weaved into the Web of data, if the rights permit it. The bottom layer can remain closed for the public and is responsible for the long-term preservation of that data.

In the one but last chapter we make video a 'first-class citizen' on the Web, as we present the rational for a Media Fragments specification. Needless to say that this Media Fragments specification will have a major impact on the complete end-to-end news production chain. From the moment news footage is shot, edited and enriched with extra information down the news production workflow chain, until a specifically chosen news item is viewed by an interested end-user, one is now able to uniquely identify, link to, display, browse, bookmark, re-composite, annotate, and/or adapt spatial and/or temporal sub-clips of media resources, e.g., a camera might automatically annotate footage with the exact geo-coordinates the moment it is shot, a news editor might quickly browse through a months' footage by means of highlight captions in search of one particular item, whereas an end-user might create a video mash-up from his bookmarked video segments to share with his friends on a social platform. As such, we outline the boundaries and semantics of a Media Fragments URI and show how the syntax should look like. We also elaborate on how a media fragment specified as an URI fragment & URI query can be resolved stepwise using the HTTP protocol, and finally, we identify the influence of current media formats on such fragment extraction. We are the first to have developed an HTTP implementation for the W3C Media Fragments 1.0 specification (both a Firefox client-side plugin and a Ninsuna server-side implementation) in order to verify all the test cases defined by our working group, thus providing at last the necessary support for this already omnipresent 'third dimension' *time* into the Internet.

In a final chapter, we put into practise what has been elaborated on in all the previous chapters, i.e., the overall use case being end-to-end news production. Proof-of-concepts from the Archipel-project, the BOM-Vl-project, the CUPID-project, and the PISA-project (in alphabetical order, as I worked on all of these projects with the same eagerness and enthusiasm) are (partly) altered to satisfy this use case. As such, we discuss a 'search use case', a 'distribution use case', an 'archiving use case', and last but not least a 'Media Fragments use case' within end-to-end news production using all concepts covered from all the previous chapters.

List of abbreviations

AAC	Advanced Audio Coding
AAF	Advanced Authoring Format
AIP	Archival Information Package
API	Application Programming Interface
Atom	XML-based Feeds Format
bNode	blank Node
BOM	Bill Of Material
BSD	Bitstream Syntax Description
BSDL	Bitstream Syntax Description Language
CB	Content-Based
CDWA	Categories for the Description of Works of Art
CF	Collaborative Filtering
CG	Character Generator
CMML	Continuous Media Markup Language
CMS	Content Management System
codec	coder - decoder
CURIE	Compact Uniform Resource Identifier
CWI	Centrum voor Wiskunde en Informatica
DAB	Digital Audio Broadcasting
DC	Dublin Core
DIG35	Digital Imaging Group – metadata for digital images
DIP	Dissemination Information Package
DMF	Digital Media Factory
DOI	Digital Object Identifier
DVB-H	Digital Video Broadcasting for Hand-helds
EAD	Encoded Archival Description
EBU	European Broadcasting Union
ENG	Electronic News Gathering
EPG	Electronic Programme Guide
ERP	Enterprise Resources Planning
EventsML-G2	Events Markup Language, Generation 2
EXIF	eXchangeable Image File format
FM	Frequency Modulation
FMO	Flexible Macroblock Ordering

FOAF	Friend Of A Friend
GOP	Group Of Pictures
GPFS	General Parallel File System
GPRS	General Packet Radio Service
GRDDL	Gleaning Resource Descriptions from Dialects of Languages
GSM	Global System for Mobile communications
HD	High Definition
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
H.264/AVC	MPEG-4 Part 10 – Advanced Video Coding
IBBT	interdisciplinary Institute for BroadBand Technology
IDR	Instantaneous Decoding Refresh
IEC	International Electro-technical Commission
IP	Internet Protocol
IPM	Intellectual Property Management
IPTC	International Press Telecommunications Council
ISAD(G)	(General) International Standard Archival Description
ISO	International Organisation for Standardisation
IT	Information Technology
JPEG	Joint Photographic Experts Group
JVT	Joint Video Team
LOD	Linked Open Data
MAM	Media Asset Management
MARC	MAchine Readable Cataloguing
MAWG	Media Annotations Working Group
MDE	Model Driven Engineering
MD5	Message Digest Algorithm 5
METS	Metadata Encoding and Transmission Standard
MFWG	Media Fragments Working Group
MIME	Multi-purpose Internet Mail Extensions
MODS	Metadata Object Description Schema
MPEG	Moving Picture Experts Group
MPEG-2	Generic Coding of Moving Pictures & associated Audio Information
MPEG-4	Toolbox of advanced Compression Algorithms for Audio and Visual Information
MPEG-7	Multimedia Content Description Interface
MP3	MPEG-1 Audio Layer 3
MP4	MPEG-4 Part 14 Multimedia File Format
MRPII	Manufacturing Resources Planning version II
MVP	Most Valuable Player
MXF	Material eXchange Format
M3U	Index File Format for describing Multimedia Playlists
NAR	News ARchitecture
NBA	National Basketball Association
NER	Named Entity Recognition
NewsML-G2	News Markup Language, Generation 2

NinSuna	NinSuna INtelligent Search framework for UNiversal multimedia Access
NITF	News Industry Text Format
NPT	Normal Playing Time
OAIS	Open Archival Information System
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
OAI-ORE	Open Archives Initiative Object Reuse and Exchange
OCR	Optical Character Recognition
OWL	Web Ontology Language
PDF	Portable Document Format
PDFa	Portable Document Format for Long-term Preservation
PE	Product Engineering
PREMIS	PREservation Metadata : Implementation Strategies
RAU	Random Access Unit
RDF	Resource Description Framework
RDFa	Resource Description Framework attributes
RDFS	Resource Description Framework Schema
RDF/XML	XML syntax for Resource Description Framework
ROE	Rich Open multi-track media Exposition
ROI	Region Of Interest
RSVP	Repondez, S'il Vous Plait
RTMP	Real Time Messaging Protocol
RTSP	Real Time Streaming Protocol
SIOC	Semantically-Interlinked On-line Communities
SIP	Submission Information Package
SKOS	Simple Knowledge Organisation System
SMEF	Standard Media Exchange Framework
SMIL	Synchronised Multimedia Integration Language
SMPTE	Society of Motion Picture and Television Engineers
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOP	Sales and Operations Planning
SPARQL	SPARQL Protocol And RDF Query Language
SPARUL	SPARQL Protocol And RDF Update Language
SQL	Structured Query Language
SVC	Scalable Video Coding
SVG	Scalable Vector Graphics
UMID	Unique Material IDentifier
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UUID	Universally Unique IDentifier
VRT	Vlaamse Radio en Televisie
WoC	Work Center

WSDL	Web Services Description Language
WWW	World Wide Web
W3C	World Wide Web Consortium
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XMP	eXtensible Metadada Platform
XSLT	eXtensible Stylesheet Language Transformations

Contents

Chapter 1

Introduction

You ride a horse rather less well than another horse would!

Lord Blackadder in Blackadder

1.1 Context

In recent years, the World Wide Web (WWW) has become the victim of its own success. There is so much fragmented information on-line, that users sometimes fail to see the real facts from some bended truth, let alone being able to interconnect different facts. One of the problems is that the Internet was initially designed as an instrument to share documents, and not intended as a platform for sharing the information that is enclosed within these documents. Tim Berners-Lee[1] himself was one of the first to suggest the idea of unleashing the real semantics of all the data out there. But if one wants to build machines that are able to interpret such information streams, one has to explain in detail the meaning, i.e.,'semantics', of the data in the information streams to these machines.

We can easily understand that the concept 'Pig' doesn't always relate to a tasty four-legged animal. For a butcher it means a slice of ham, whereas it is livestock and pure capital for the farmer that feeds them. In another context, e.g., the news production use case, it could relate to an influenza variant, or even an abuse from one member of parliament to another during a dispute over some amendments to a bill. Indeed, the exact meaning of a term depends on the context in which it is used within a certain domain. Only when

[1] Tim Berners-Lee WWW Profile, see also `http://www.w3.org/People/Berners-Lee/`

(electronic) information is unambiguously identifiable and interpretable, it can be exchanged over the Internet and reasoned on by (electronic) machines in a correct way. ‘Semantics’ is thus one of the key elements to interoperability. Even if we try to merge a number of information systems by using technological standards, information –be it text, audio, or video– cannot be exchanged in a meaningful way if the concepts cannot be matched.

News, to name a specific domain, is something nearly every European citizen reads, watches, or listens to on a daily basis, at home, while commuting to and from work, at work, and even as part of their work. As voting citizens, we need to understand local, national, and international politics to allow us to cast our vote. As company employees, we need to understand the state and development of local, national, and international economies to enable us to understand our markets. As part of our leisure time, we want to know about our favourite sports teams, the lives of our soap idols, or the most recent books available. Nowadays, this information is on-line, and hence accessible from anywhere.

The information is on-line and available, but through many different sources, including branded Web sites. Traditional news providers (e.g., journalists, news agencies, press, and broadcasters) make use of the Web for distributing their news. More recently, non-traditional news providers, often called citizen media or independent media, make use of Web technologies to publish alternative views and opinions of events. Furthermore, we observe an increasing tendency of social media sites playing a crucial role either in spreading news at a speed never seen (e.g., the death of Michael Jackson) or simply in informing citizens in countries or situations where other traditional communication means fail (e.g., the coverage of the protests following the controversial Iranian elections). Professional users have therefore access to continuous streams of incoming data from press agencies and archives of published news to tweets, images, and videos posted by anonymous people. Non-professional users have access to myriads of Web sites, offering push and pull sources of news information and the means of contributing to the expanding collection of news data. These billions of sources contain large amounts of isolated information that lack context, leaving users with the immense problem of manually finding related information. Typical tasks include: finding the same event covered from a different (political) angle, finding the role of the event in a wider historical perspective, finding and checking the original sources on which a story is based, etc. News aggregators only tend to amplify the problem by aggregating pointers to isolated articles, rather than analysing and contextualising events from the many continuous data streams. Faced with this upraise of citizen-based media and social networks, traditional media must re-think their fact-checking processes while citizens also need tools to enhance their user experience in terms of knowledge and confidence when reading and watching news on-line.

In existing news workflow processes, news items are typically produced by news agencies, independent journalists or citizen media; afterwards consumed and enhanced by newspapers, magazines or broadcasters; then delivered to end-users; and finally perceived by these end-users that further leave a trace of what they have understood and felt facing these news events using blogs and tweets as means of expression. News items should therefore typically be accompanied by a set of metadata and descriptions that facilitate their storage, retrieval, and life cycle. However, much of the metadata is lost because of interoperability problems occurring along the production workflow. As such, opportunities for making use of the available metadata at the user interface level are often lost.

News Web sites, such as De Standaard[2], The Times[3] or The Wall Street Journal[4] generally classify news in categories such as: World, National, Politics, Business, Science and Technology, Sport, Entertainment, and Health, while other services such as Google News[5] aggregate stories from multiple sources and offer personalised selections based on the user topics of interest. More advanced Web sites such as SiloBreaker[6] or Newstin[7] provide more flexible access to news stories by topic, person, organisation or region. Specific services attempt to extract trends from large amount of micro-blogging feeds. These services support the users' need for information more closely, but also add more sources of individual news items, which lead to overly complex interfaces. Current systems have a number of limitations that force the user to explore news information in an environment that contains large amounts of irrelevant, unreliable and repeated information, with insufficient access to background knowledge. In particular, current systems:

- Mainly deal with textual news articles in a single language (mostly English), and do not process audio-visual content at the same level of detail.

[2] `http://www.destandaard.be/`

[3] `http://www.timesonline.co.uk/`

[4] `http://europe.wsj.com/`

[5] `http://news.google.be/`

[6] `http://www.silobreaker.com/`

[7] `http://www.newstin.com/`

- Are unable to provide explicit relationships between different news on the same event to help the user form his/her own opinion on a particular topic, e.g., cannot automatically link a quote in a news article to the original statement in a video clip, or link a statement to the subsequent reactions.
- Are unable to handle the evolution of news events, e.g., do not link the first announcement of an explosion to its subsequent interpretation as a 'terrorist attack'.
- Are unable to provide a historic perspective of events, e.g., do not show the chain of events that led to the information the reader is focusing on, and do not highlight editorial news items summarising events that took place years ago.

1.2 Goals

Users are overwhelmed by too many individual and disconnected pieces of information, and cannot situate the news in a proper context. A news event is defined as a cluster of statistically related news items, but no ontological notion of event is supported. In contrast, the organisation of news providers is centred on the notion of scheduled and breaking news events, but they currently lack the tools necessary to relate easily the news they produce to the events they manage on a daily basis. Semantic processing of news information can improve the clustering and organisation of individual news items from heterogeneous sources, in multiple media types and in multiple languages into meaningful events linked to appropriate background knowledge.

Interoperability of semantics is therefore key throughout the different workflow phases of news production. Firstly, within each media company all ranks of the organisation should be able to reason about a problem across levels given joint concepts, thereby closing the '*comprehensive semantic gap*' that historically exists between business management and technical engineering, thus smoothing the workflow between the organisation's entities and leading to a technical sound solution understood by everyone. Secondly, metadata should be identified, captured, and standard-based semantically described as soon as possible, thereby closing the '*technical semantic gap*' as different in-house systems down the workflow pipeline, as well as third party Internet Web-services, will be able to harness this extra knowledge to their own favour. Finally, the gathered knowledge, i.e., the interlinked news items, should be archived in a future-safe way, thereby closing the '*continuity* *semantic gap*' as metadata, e.g., provenance data, on metadata will be key to long-lasting interoperability.

This dissertation is both a theoretical and hands-on 'Best Practices' document to start solving the above problems within the news context. We looked at the end-to-end news production workflow, taken our Flemish broadcaster VRT[8] as a use case, identified how and when the data and accompanying metadata should be described, interpreted, harvested, exchanged, archived, and distributed. It is my belief that lasting interoperability in any specific problem domain can only be achieved by constantly and consciously using standards, preferably open ones. As subsequent chapters will show, all my research heavily relies on the principles and standards of the (Semantic) Web.

1.3 Web Technologies

It is fair to say that the Internet paradigm of the early nineties redefined Information Technology (IT) as never before. At that time there was a collision of military need and voluntary academic research, sparked by the team of Tim Berners-Lee, that yielded a new kind of community –the Internet–, and with it a new type of information economy. With the necessary hiccups[9], this technology has proven to be a merit to IT, and I –for one– was a true believer from the start. The HyperText Transfer Protocol (HTTP) and the HyperText Markup Language (HTML) proved to be a 'deadly' team that took the IT world by storm within a decade.

Again, Tim Berners-Lee states that the essential property of the WWW is its universality [12] . The power of a hypertext link is that 'anything can link to anything'. Web technology, therefore, must not discriminate between the scribbled draft and the polished performance, between commercial and academic information, or among cultures, languages, media, and so on. As such, information varies along many axes. One of these is the difference between information produced primarily for human consumption and that produced mainly for machines. To date, however, that Web has developed most rapidly as a medium of documents for people rather than for data and information that can be processed automatically.

The *Semantic Web* is an evolving development of the WWW in which the meaning (semantics) of information and services on the Web is defined,

[8] Vlaamse Radio en Televisie, see also `http://www.vrt.be/`

[9] the Internet bubble in 1999 to name just one

making it possible for the Web to understand and satisfy the requests of people and machines to use that Web content. It's Tim Berners-Lee's renewed semantic vision of the Web as a universal medium for data, information, and knowledge exchange [153], to which I'm an advocate from the start too.

As the WWW Consortium's (W3C) Semantic Web Activity[10] states, the principal technologies of the Semantic Web fit into a set of layered specifications. The current components are the Resource Description Framework (RDF) Core Model, the RDF Schema language (RDFS), the Web Ontology language (OWL), and the Simple Knowledge Organisation System (SKOS). Building on these core components is a standardised query language, SPARQL Protocol And RDF Query Language (SPARQL) –pronounced 'sparkle'–, enabling querying decentralised collections of RDF data. The Gleaning Resource Descriptions from Dialects of Languages (GRDDL) and RDF in Attributes (RDFa) Recommendations furthermore aim at creating bridges between the RDF model and various eXtensible Markup Language (XML) formats, like eXtensible HTML (XHTML). In early December 2006 a blogpost[11] was made trying to explain the Semantic Web in '10 seconds' and it was concisely summarised as:

- Web 1.0 was about connecting-up documents.
- Web 2.0 is about connecting-up people.
- The Semantic Web is about connecting-up data.

One response stated that:

- The Web (be it 2.0 or the 'very old' one :) is all about curiosity. It requires people to explore (i.e., click on a link) the available space.
- The Semantic Web really is about laziness. You want some clever piece of software to take care of boring or otherwise non-exciting tasks for you.

It is very likely that the truth is somewhere in the 'complementary' middle. However, as a matter of fact the Semantic Web has started to lift off, meanwhile. This section explains the foundations of the Semantic Web, as can be seen on the (Semantic) Web layer-cake in Figure 1.1, ranging from its logical foundations to practical issues as most of the technologies, and implementations within this dissertation rely heavily on these Semantic Web foundations.

[10] http://www.w3.org/2001/sw/Activity.html

[11] http://www.oreillynet.com/xml/blog/2006/12/explaining_the_semantic_web_in.html

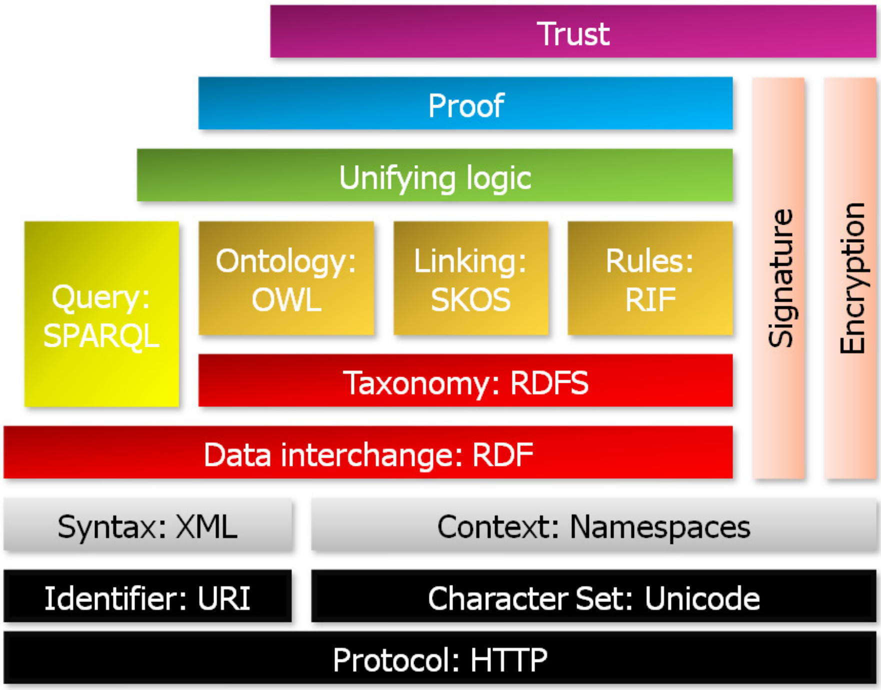


Figure 1.1: The Semantic Web Stack as introduced by Tim Berners-Lee.

1.3.1 HTTP

HTTP [78] is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes, and headers. A prominent feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

Practical information systems require more functionality than simple retrieval, including search, front-end update, and annotation. HTTP allows an open-ended set of methods and headers that indicate the purpose of a request. It builds on the discipline of reference provided by the Uniform Resource Identifier [79] (URI) , as a Uniform Resource Locator [74] (URL), or Uniform Resource Name [77] (URN), for indicating the resource to which a method is to be applied. Messages are passed in a format similar to that used by Internet mail as defined by the Multi-purpose Internet Mail Extensions [76] (MIME). HTTP is also used as a generic protocol for communication between user agents and proxies/gateways to other Internet systems as well. In this way, HTTP allows basic hypermedia access to resources available from diverse applications.

The HTTP protocol is a request/response protocol. A client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a connection with a server. The server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta-information, and possible entity-body content.

1.3.2 XML and XML Schema

It is important for Semantic Web developers to agree on the data's syntax and semantics before hard-coding them into their applications, since changes to syntax and semantics necessitate expensive modifications of applications. Current Semantic Web languages rely on an XML-based syntax [102]. Generally, XML [20] enables the specification and mark-up of computer-readable documents. It looks very much like HTML [144] in that special sequences of characters 'tags' are used to mark up the document content, and that XML data is stored as ordinary text. Unlike HTML (which is layout-oriented), however, XML (which is structure-oriented) can be used to annotate documents of arbitrary structure, and there is no fixed tag vocabulary. Typically, XML tags contain information indicating the human interpretation of pieces of the document's content, such as <employees>, <person>, and <phone>. Thus XML lets people meaningfully annotate documents by adding context to and indicating the meaning of the data. Someone can define their own custom tags to represent data logically, making XML documents self-describing for one another (because the tags describe the information the documents contain).

However, XML does not itself imply a specific machine interpretation of the data, i.e., the meaning is not formally specified. The information is only encoded in an unambiguous syntax, but its use and the semantics are not specified. In other words, XML is aimed only at the structure of a document, not at a common machine interpretation of it. It provides only a data format for structured documents, without specifying a vocabulary. On the other hand, owing to the standardised data format and structure of XML documents, software programs and scripts can dynamically access and update the content, structure, and style of such documents [110]. Appropriate parsers and other processing tools for XML documents are readily available. Moreover, XML is extensible in a standardised way, and hence enables customised mark-up languages to be defined for unlimited types of documents. Using XML for document and data exchange among applications requires prior agreement on the vocabulary, its use, and the meaning of its terms though. Such agreement can be partly achieved by using XML Schema [15, 162], which provides mechanisms to specify the structure and grammar of XML documents. Every XML schema provides the necessary framework for creating a category of XML documents. The schema describes the various tags, elements, and attributes of an XML document of that specific category, the valid document structure, the constraints, and the custom data types (these are based on built-in types, such as *integer* and *string*). The XML Schema language also provides some limited support for specifying the number of occurrences of child elements, default values, choice groups, etc. The encoding syntax of the XML Schema language is XML, and just like XML itself XML Schema documents use namespaces [19]. Namespaces define contexts within which the corresponding tags and names apply.

1.3.3 RDF and RDF Schema

XML provides an easy-to-use syntax for encoding all of the kinds of data that are exchanged between computers, by using XML schemas to prescribe the data structure. However, since it does not provide any interpretation of the data beforehand, it does not contribute much to the semantic aspect of the Semantic Web. To provide machine interpretation of Web data, a standard model is needed to describe facts about Web resources. Such a standard model can be specified by use of RDF [103] and RDFS [22]. RDF's model for representing data about 'things on the Web' (resources) is that of (*Subject, Predicate, Object*) triples and semantic networks. It is based on the idea that the 'things' being described have properties which have values, and that resources can be described by making statements, that specify those properties and values. RDF uses a particular terminology for talking about the various parts of statements. Specifically, the part that identifies the 'thing' the statement is about (a Web resource) is called the *subject*. The part that identifies the property or characteristic of the subject that the statement specifies is called the *predicate*, and the part that identifies the value of that property is called the *object* [10]. The value can be a literal (text), a typed literal (e.g., int, an XML Schema built-in datatype), another resource (via an URI [79]), or 'blank node' (bNode), which refers to the fact that the corresponding nodes in the RDF graph are 'blank', i.e., have no URI. Every RDF description also can be represented as a directed labelled graph (a semantic network). An RDF model itself provides only a domain-neutral mechanism to describe individual resources. It neither defines (a priori) the semantics of any application domain, nor makes assumptions about a particular domain. Defining domain-specific features and their semantics, i.e., ontologies, requires additional facilities.

RDFS, built on top of RDF, furthermore provides a vocabulary to specify classes and their relationships, to define properties and associate them with classes, and to enable the creation of taxonomies. To do all of this, RDFS uses frame-based modelling primitives such as *Class*, *subClassOf*, *Property*, and *subPropertyOf*. The *Resource* concept occurs in the root of all hierarchies and taxonomies. There is an important departure in RDFS from the classical frame-based paradigm: properties are defined separately from classes. An implication is that anyone, anywhere, anytime can create a property and state that it is usable with a class, or with multiple classes. Each property can be described by *rdfs:domain* and *rdfs:range*, which restrict the possible combinations of properties and classes. On the other hand, a property may be defined so as to feature multiple classes. RDF(S) provides a standard model to describe facts about Web resources, but modellers often need even richer and more expressive primitives to specify the formal semantics of Web resources. RDFS is quite simple compared with full-fledged knowledge representation languages. For example, one cannot state in RDFS that 'this class is equivalent to this other class', and cannot specify cardinality constraints.

1.3.4 OWL and OWL2

OWL [121] is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDFS by providing an additional vocabulary along with a formal semantics. These are used to describe domain terms and their relationships in an ontology. In fact, the OWL vocabulary is built on top of the RDF(S) vocabulary. Things such as *Class* and *subClassOf* exist in OWL as well, and so do *ObjectProperty*, *DatatypeProperty*, and many more. An important feature of the OWL vocabulary is its extreme richness for describing relations among classes, properties, and individuals. For example, we can specify in OWL that a property is *Symmetric*, the *InverseOf* another one, an *equivalentProperty* of another one, and *Transitive*; that a certain property has some specific *cardinality*, or *minCardinality*, or *maxCardinality*; and that a class is defined to be an *intersectionOf* or a *unionOf* some other classes, and that it is a *complementOf* another class. Similarly, a class instance can be the *sameAs* another instance, or it can be required to be *differentFrom* some other instance. A nice consequence is that reasoning can be performed in spite of such terminological differences.

Another important asset is OWL's layered structure. In fact, OWL is not a closed language; it is, rather, a combination of three increasingly expressive sublanguages [154] building on top of each other, designed to suit different communities of implementers and users. OWL Lite is intended to support the building of simple classification hierarchies and simple constraints. To this end, the ability to specify constraints in OWL Lite is rather restricted; for example, the only cardinality values permitted in OWL Lite are 0 and 1. OWL DL reflects the description-logic foundation of its predecessor, DAML+OIL [100]. OWL DL provides more expressiveness, and also guarantees that all conclusions are computable and will finish in a finite time. It includes all OWL language constructs, although it imposes certain restrictions on using them. OWL Full supports users who want maximum expressiveness and the syntactic freedom of RDF, but does not guarantee computational completeness and decidability. OWL Full can be viewed as an extension of RDF, whereas OWL Lite and OWL DL can be viewed as extensions of a restricted view of RDF.

OWL 2 [122] has a very similar overall structure to OWL 1. Looking at Figure 1.2, almost all the building blocks of OWL 2 were present in OWL 1, albeit possibly under different names. The central role of XML Syntax for RDF [10] (RDF/XML), the role of other syntaxes, and the relationships between the *direct* and *RDF-based* semantics have not changed. More importantly, backwards compatibility with OWL 1 is, to all intents and purposes, complete: all OWL 1 ontologies remain valid OWL 2 ontologies, with identical inferences in all practical cases. OWL 2 adds new functionality with respect to OWL 1 though. Some of the new features are syntactic sugar (e.g., disjoint union of classes) while others offer new expressiveness, including:

- Keys.
- Property chains.
- Richer datatypes, and data ranges.
- Qualified cardinality restrictions.

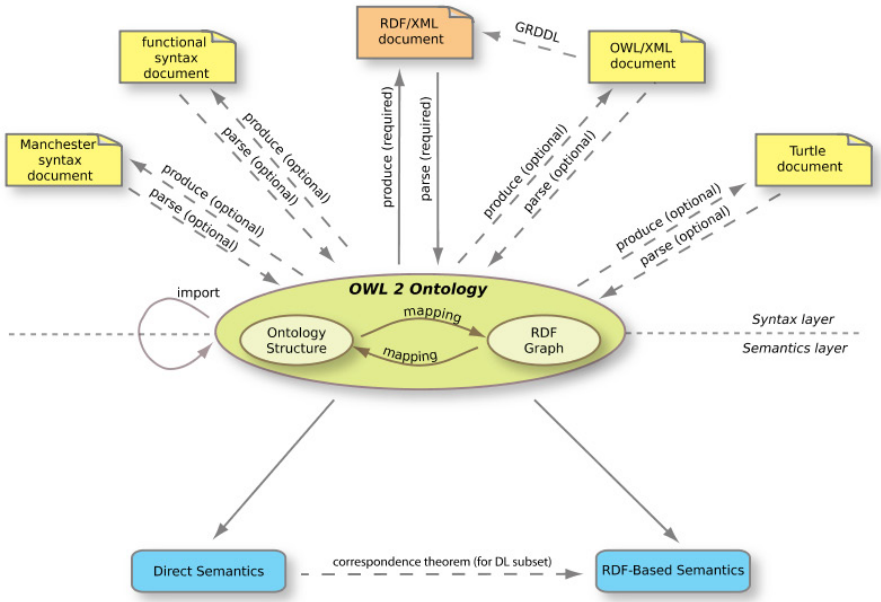


Figure 1.2: The Structure of OWL 2.

- Asymmetric, reflexive, and disjoint properties.
- Enhanced annotation capabilities.

1.3.5 SPARQL

Unlike OWL and RDF(S), SPARQL is not intended for ontology and resource representation, but for querying Web data; precisely, it is a protocol [32] and query language [143] for RDF. To understand SPARQL, the view of RDF resources as semantic networks (set of triples) helps. SPARQL can be used to:

- Extract information from RDF graphs in the form of URIs, bNodes, and plain and typed literals.
- Extract RDF subgraphs.
- Construct new RDF graphs based on information in queried graphs.

SPARQL queries match graph patterns against the target graph of the query. The patterns are like RDF graphs, but may contain named variables in place of some of the nodes (resources) or links/predicates (i.e., properties). The simplest graph pattern is like a single RDF triple. A binding is a mapping from a variable in a query to terms. Each triple is a pattern solution (a set of correct bindings) for the pattern. Query results in SPARQL are thus sets of pattern solutions. Simple graph patterns can be combined using various operators into more complicated graph patterns. Syntactically, SPARQL queries closely resemble the syntax of database query languages such as Structured Query Language [83] (SQL) (cf. *Select*, *Construct*, *Ask*, and *Update* –SPARUL submission [151]– statements). The SELECT clause contains variables, beginning with '?' or '$'. The WHERE clause contains a pattern. Prefixes are used as an abbreviation mechanism for URIs and apply to the whole query.

The SPARQL Protocol on the other hand uses Web Services Description Language (WSDL) version 2.0 [29] to describe a means for conveying SPARQL queries to a SPARQL query processing service and returning the query results to the entity that requested them. SPARQL Protocol has been designed for compatibility with the SPARQL Query Language for RDF, and can be described in two ways: first, as an abstract interface independent of any concrete realisation, implementation, or binding to another protocol; second, as HTTP and Simple Object Access Protocol [59] (SOAP) bindings of this interface.

1.3.6 SKOS

Like OWL, SKOS [81, 124] can be used to define vocabularies. But the two technologies were designed to meet different needs. SKOS is a simple language with just a few features, tuned for sharing and linking knowledge organisation systems such as thesauri and classification schemes, whereas OWL offers a general and powerful framework for knowledge representation, where additional 'rigor' can afford additional benefits (for instance, business rule processing). SKOS is a model for expressing the basic structure and content of concept schemes, such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies, and other types of controlled vocabularies. The SKOS Core Vocabulary is an RDF application. Using RDF allows data to be linked to and/or merged with other RDF data by Semantic Web applications. In practice, this means that data sources can be distributed across the Web in a decentralised way, but still be meaningfully composed and integrated by applications, often in novel and unanticipated ways. The SKOS Core Vocabulary is a set of RDF properties and RDFS classes, that can be used to express the content and structure of a concept scheme as an RDF graph.

1.3.7 RDFa and GRDDL

Current Web pages contain inherent structured data: calendar events, contact information, photo captions, song titles, copyright licensing information, etc. When authors and publishers can express this data precisely, and when tools can read it robustly, a new world of user functionality becomes available, letting users transfer structured data between applications and Web sites. W3C has –motivated by the success of the microformats[12]– a standard to embed RDF in (X)HTML: RDFa [1, 2] is a syntax that allows for embedding an RDF graph into an (X)HTML document via attributes. RDFa lets (X)HTML authors express this structured data using extra (X)HTML attributes (*@rel*, *@ref*). The RDFa specification introduces new, RDFa-specific attributes allowing to represent an arbitrary RDF graph (*@about*, *@property*, *@resource*, *@datatype*, *@instanceof*). The RDFa Syntax document defines not only how RDFa must be processed, but also introduces the notion of Compact URIs [14] (CURIE). A CURIE is comprised of two components, a prefix which maps to an URI, and a reference. The prefix is separated from the reference by a colon.

While adding metadata explicitly to an HTML page is an option especially for new or evolving systems; current Web sites (including microformats) can be easily turned into Semantic Web sites when using GRDDL [35]. The GRDDL specification proposes mechanisms for declaring that an XML document includes data that is compatible with RDF. Further, GRDDL defines the linking to algorithms –which are typically represented in eXtensible Stylesheet Language Transformations [31] (XSLT)– for extracting this data from the document by enabling this transformation on the authors' side.

1.3.8 Ontologies' Key Application Areas

There are many potential application areas of ontologies, but [57] has offered a high-level list of key application areas, which we all used somehow throughout this dissertation as it was an influencing factor in numerous projects of the Interdisciplinary Institute of BroadBand Technology (IBBT), as can be seen in Appendix E, that helped to shape this dissertation: *collaboration*, *interoperation*, *learning*, and *modelling*.

- *Collaboration*. Different people may have different views of the same problem area when working on a team project. This is particularly true for interdisciplinary teams, with specialists from different branches of science, technology, and development having different foci of interests

[12] see also `http://microformats.org/`

and expertise. For such specialists, ontologies provide a unifying knowledge skeleton that can be used as a common, shared reference for further development and participation. These people can simply talk more easily to each other when they have such a stable, consensual knowledge armature to rely on. Perhaps even more importantly, ontologies play the same role in collaboration between intelligent agents in terms of agent-to-agent communication. When an agent sends a message to another agent that it is communicating with, the other agent must have the same world model (i.e., the same ontology) in order to interpret the message correctly. Knowledge exchange between agents is much more feasible when the agents are aware of the ontologies that the other agents are using as world models.

- *Interoperation*. Ontologies enable the integration of information from different and disparate sources. End-users typically do not show much interest in how they get their information; they are much more interested in getting the information they need, and getting all of it. Distributed applications may need to access several different knowledge sources in order to obtain all the information available, and those different sources may supply information in different formats and in different levels of detail. However, if all the sources recognise, and understand the different views, and the mapping between those ontological views, data conversion and information integration are easier to do automatically and in a more natural way.
- *Learning*. Ontologies are also a good publication medium and source of reference. Since they presumably always result from a wide consensus about the underlying structure of the problem domain they represent, they can provide reliable and objective information to those who want to learn more about the domain. Simultaneously, domain experts can use ontologies to share their understanding of the conceptualisation and structure of the domain.
- *Modelling*. In modelling intelligent, knowledge-based applications, ontologies represent important reusable building blocks, which many specific applications should include as pre-developed knowledge modules.

1.4 Outline

In fact, one could consider end-to-end news production to be such an application domain for ontologies where all of the four categories (roles) can be applied. Chapter 2 ('Identifying Generic Media Processes in News Production') lays the theoretical, *collaborative* foundations of the processes of the news workflow providing agreed upon and rigorous descriptions for exchanging semantically annotated media assets among different companies and/or applications. Chapter 3 ('Personalising Enriched News Events') further elaborates on enabling *interoperable* machine-based communication between news providers and consumers using ontologies, whereas Chapter 4 ('Making Disclosure of News Future Proof') gives *educational* insights in the solving of the continuous audio/video archiving issues. Chapter 5 ('Making Time Uniformly Identifiable in News Items') then *models* and formalises media fragments' reuse between different news/events marketplaces. Finally Chapter 6 ('Proof of Concepts') demonstrates the use of our proposed solutions mentioned in prior chapters and Chapter 7 draws some final conclusions.

1.5 Overview Publications

The research activities that led to this dissertation resulted in 22 Journal publications (19 accepted, 3 submitted). Furthermore, the work described in this dissertation contributed to 35 papers that were presented at international conferences. Next to this, the author contributed to two W3C Specifications.

1.5.1 Journal Publications

1. E. Mannens, S. Coppens, T. De Pessemier, H. Dacquin, D. Van Deursen, R. De Sutter, and R. Van de Walle. Automatic News Recommendations via Profiling. Submitted to *Multimedia Tools and Applications – Special Issue on Automated Information Extraction in Media Production*, Springer-Verlag

2. E. Mannens, D. Van Rijsselbergen, M. Verwaest, R. De Sutter, L. Overmeire, and R. Van de Walle. Generic Architecture Guidelines for a Digital Media Factory. *SMPTE Motion Imaging Journal*, Accepted for publication, SMPTE

3. E. Mannens, D. Van Deursen, S. Pfeiffer, C. Parker, R. Troncy, Y. Lafon, J. Janssen, M. Hausenblas, and R. Van de Walle. Universally Addressing Media Fragments. *Multimedia Tools and Applications – Special Issue on Multimedia Data Semantics*, Online First (DOI: 10.1007/s11042-010-0683-z), Springer-Verlag, December, 2010

4. E. Mannens, S. Park, J. Soderberg, G. Adams, P. Le Hegaret, R. Van de Walle, and C. Seon Hong. Video in the Web: Technical Challenges and Accompanying Standardisation Activities. *IEEE Multimedia*, volume 17, issue 4, pages 90–93, IEEE Computer Society, October, 2010

5. E. Mannens, S. Coppens, L. Hauttekeete, T. Evens, and R. Van de Walle. Semantic Bricks for Performing Arts Archives and Dissemination. *IASA Journal*, issue 35, pages 40–49, IASA, June, 2010

6. E. Mannens, M. Verwaest, and R. Van de Walle. Production and Multichannel Distribution of News. *Multimedia Systems – Special Issue on Canonical Processes of Media Production*, volume 14, issue 6, pages 359–368, Springer-Verlag, December, 2008

7. W. Van Lancker, D. Van Deursen, E. Mannens, and R. Van de Walle. Implementation Strategies for Efficient Media Fragment Retrieval. *Multimedia Tools and Applications – Special Issue on Recent Advances and Future Directions in Multimedia and Mobile Computing*, Accepted for publication, Springer-Verlag, March, 2011

8. L. Hauttekeete, K. De Moor, D. Schuurman, T. Evens, E. Mannens, and R. Van de Walle. Archives in Motion: Concrete Steps towards the Digital Disclosure of Audio-visual Content. *International Journal of Cultural Heritage*, Accepted for publication, Elsevier, March, 2011

9. S. Coppens, E. Mannens, T. De Pessemier, K. Geebelen, H. Dacquin, D. Van Deursen, and R. Van de Walle. Unifying and Targeting Cultural Activities via Events Modelling and Profiling. *Multimedia Tools and Applications – Special Issue on Events in Multimedia*, Online First (DOI: 10.1007/s11042-011-0757-6), Springer-Verlag, February, 2011

10. R. Verborgh, D. Van Deursen, E. Mannens, and R. Van de Walle. Enabling Context-aware Multimedia Annotation by a Novel Generic Semantic Problem-solving Platform. *Multimedia Tools and Applications – Special Issue on Multimedia and Semantic Technologies for Future Computing Environments*, Online First (DOI: 10.1007/s11042-011-0709-6), Springer-Verlag, January, 2011

11. T. De Pessemier, S. Coppens, K. Geebelen, C. Vleugels, S. Bannier, E. Mannens, K. Vanhecke, and L. Martens. Collaborative Recommendations with Content-based Filters for Cultural Activities via a Scalable

Event Distribution Platform. *Multimedia Tools and Applications*, Online First (DOI: 10.1007/s11042-011-0715-8), Springer-Verlag, January, 2011

12. D. Van Rijsselbergen, M. Verwaest, C. Poppe, E. Mannens, and R. Van de Walle. Semantic Mastering: Content Adaptation in the Creative Drama Production Workflow. *Multimedia Tools and Applications – Special Issue on Intelligent Interactive Multimedia Systems and Services*, Online First (DOI: 10.1007/s11042-011-0710-0), Springer-Verlag, January, 2011

13. C. Hollemeersch, B. Pieters, A. Demeulemeester, B. Van Semmertier, E. Mannens, P. Lambert, and R. Van de Walle. Infinitex: An Interactive Editing System for the Production of Large Texture Data Sets. *Computer Graphics – Special Issue on Serious Gaming*, volume 34, issue 6, pages 643–654, Elsevier, December, 2010

14. T. Evens, L. De Marez, L. Hauttekeete, D. Biltereyst, E. Mannens, and R. Van de Walle. Attracting the Un-served Audience: The Sustainability of Long Tail-based Business Models for Cultural Television Content. *New Media & Society*, volume 12, issue 6, pages 1005–1023, SAGE Publications, September, 2010

15. D. Van Rijsselbergen, M. Verwaest, E. Mannens, and R. Van de Walle. On how Metadata Enables Enriched File-Based Production Workflow. *SMPTE Motion Imaging Journal*, pages 27–38, SMPTE, June, 2010

16. L. Hauttekeete, K. Berte, P. Mechant, S. Paulussen, E. De Waele-De Guchtenaere, E. Mannens, and R. Van de Walle. Using WEB 2.0 to Support the Independent Film Production Process. *Journal of Film and Film Culture*, volume 5, pages 133–156, SAGE Publications, March, 2010

17. D. Van Deursen, W. Van Lancker, S. De Bruyne, W. De Neve, E. Mannens, and R. Van de Walle. Format-independent and Metadata-driven Media Resource Adaptation using Semantic Web Technologies. *Multimedia Systems Journal*, on-line first, pages 85–104, Springer-Verlag, January, 2010

18. D. Van Deursen, W. Van Lancker, W. De Neve, T. Paridaens, E. Mannens, and R. Van de Walle. NinSuna: a Fully Integrated Platform for Format-independent Multimedia Content Adaptation and Delivery

based on Semantic Web Technologies. *Multimedia Tools And applications – Special Issue on Data Semantics for Multimedia Systems*, volume 46, issue 2, pages 371–398, Springer-Verlag, January, 2010

19. S. Coppens, E. Mannens, and R. Van de Walle. Disseminating Heritage Records as Linked Open Data. *International Journal of Virtual Reality*, volume 8, issue 3, pages 39–44, IPI Press, September, 2009

20. C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. Personal Content Management System: A Semantic Approach. *Journal of Visual Communication and Image Representation – Special Issue on Emerging Techniques for Multi-media Content Sharing, Search and Understanding*, volume 20, issue 2, pages 131–144, Elsevier, February, 2009

21. S. Coppens, C. Poppe, E. Mannens, D. Van Deursen, P. Hochstenbach, B. Janssens, and R. Van de Walle. Digital Long-term Preservation - A Provenance Story. Submitted to *International Journal of Web Semantics – Special Issue on Using Provenance in the Semantic Web*, Elsevier

22. P. De Potter, H. Cools, K. Depraetere, G. Mels, P. Debevere, J. De Roo, C. Huszka, D. Colaert, E. Mannens, and R. Van de Walle. Semantic Patient Information Aggregation and Medicinal Decision Support. Submitted to *IEEE Transactions on Information Technology in Biomedicine*, IEEE

1.5.2 Conference Publications

1. E. Mannens, D. Van Deursen, and R. Van de Walle. Making Space and Time Uniformly Identifiable in the Web via Media Fragments. In *the Proceedings of the 8th IEEE Consumer Communications and Networking Conference – 1st International Workshop on Semantics to Enable Convergence for Consumer Communications and Applications*, pages 1023–1028, January 2011, Las Vegas, USA

2. E. Mannens, S. Coppens, and R. Van de Walle. A Network-centric Approach to Sustainable Digital Archives. In *the Proceedings of the 41th International Conference of the International Association of Sound and Audio-visual Archives*, pages 1–1, November 2010, Philadelphia, USA

3. E. Mannens, S. Coppens, T. De Pessemier, H. Dacquin, D. Van Deursen, and R. Van de Walle. Automatic News Recommendations via Profiling. In *the Proceedings of the ACM International Conference on Multimedia*

- the 3rd International Workshop on Automated Information Extraction in Media Production, pages 45–50, October 2010, Florence, Italy

4. E. Mannens, S. Coppens, L. Hauttekeete, R. De Sutter, and R. Van de Walle. Cloudcomputing Approach to Sustainable Media Archives. In *the Proceedings of the International Federation of Television Archives World Conference*, pages 1–1, October 2010, Dublin, Ireland

5. E. Mannens, R. Troncy, J. Sendor, and D. Van Deursen. Implementing W3Cs Media Fragments URI Specification. In *the Proceedings of the Open Video Conference*, pages 1–1, October 2010, New York, USA

6. E. Mannens, D. Van Rijsselbergen, L. Hauttekeete, R. De Sutter, and R. Van de Walle. The Repurposing of Archive Content: The MEMENTO Project. In *the Proceedings of the International Federation of Television Archives World Conference*, pages 1–6, October 2009, Beijing, China

7. E. Mannens, S. Coppens, T. De Pessemier, K. Geebelen, H. Dacquin, D. Van Deursen, and R. Van de Walle. Unifying and Targeting Cultural Activities via Events Modelling and Profiling. In *the Proceedings of the ACM International Conference on Multimedia - the 1st International Workshop on Events in Multimedia*, pages 33–40, October 2009, Beijing, China

8. E. Mannens, S. Coppens, and R. Van de Walle. Semantic Bricks for Performing Arts Archiving & Dissemination. In *the Proceedings of the 40th International Conference of the International Association of Sound and Audio-visual Archives*, pages 85–86, September 2009, Athens, Greece

9. E. Mannens, O. Braet, and R. Van de Walle. Technical Criteria & Business Motives for Evaluating the Mobile DRM Landscape. In *the Proceedings of the the 13th IEEE International Symposium on Consumer Electronics*, pages 503–506, May 2009, Kyoto, Japan

10. E. Mannens, O. Braet, and R. Van de Walle. Mobile DRM Business Motives. In *Abstract book of the 13th IEEE International Symposium on Consumer Electronics*, pages 47–47, May 2009, Kyoto, Japan

11. E. Mannens, R. Troncy, K. Braeckman, D. Van Deursen, W. Van Lancker, R. De Sutter, and R. Van de Walle. Automatic Information Enrichment in News Production. In *the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 61–64, May 2009, London, United Kingdom

12. R. Verborgh, D. Van Deursen, E. Mannens, and R. Van de Walle. Application of Semantic Web Technologies for Automatic Multimedia Annotation. In *the Proceedings of the 8th IEEE Consumer Communications and Networking Conference – 1st International Workshop on Advanced Future Multimedia Services*, pages 1–10, December 2010, Gwangju, South Korea

13. D. Van Deursen, W. Van Lancker, E. Mannens, and R. Van de Walle. NinSuna: Metadata-driven Media Delivery. In *the Proceedings of the 3th International Service Wave Conference*, pages 1–2, December 2010, Ghent, Belgium

14. R. Verborgh, D. Van Deursen, E. Mannens, and R. Van de Walle. Enabling Advanced Context-Based Multimedia Interpretation Using Linked Data. In *the Proceedings of Linked Data in the Future Internet workshop at the 4th Future Internet Assembly*, pages 1–8, December 2010, Ghent, Belgium

15. P. Debevere, D. Van Deursen, D. Van Rijsselbergen, E. Mannens, M. Matton, R. De Sutter, and R. Van de Walle. Enabling Semantic Search in a News Production Environment. In *the Proceedings of the 5th International Conference on Semantic and Digital Media Technologies*, pages 1–16, December 2010, Saarbrucken, Germany

16. R. Verborgh, D. Van Deursen, J. De Roo, E. Mannens, and R. Van de Walle. SPARQL Endpoints as Front-end for Multimedia Processing Algorithms. In *the Proceedings of the 9th International Semantic Web Conference – 4th International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web*, pages 1–16, November 2010, Shanghai, China

17. S. Coppens, and E. Mannens. Workflow Engines PREMIS OWL Binding for Long-Term Preservation. In *the Proceedings of the 41th International Conference of the International Association of Sound and Audiovisual Archives*, pages 1–1, November 2010, Philadelphia, USA

18. D. Deursen, W. Van Lancker, E. Mannens, and R. Van de Walle. NinSuna: a Server-side W3C Media Fragments Implementation. In *the Proceedings of the 11th IEEE International Conference on Multimedia & Expo*, pages 270–271, July 2010, Singapore, Singapore

19. D. Van Deursen, R. Troncy, E. Mannens, S. Pfeiffer, Y. Lafon, and R. Van de Walle. Implementing the Media Fragments URI Specification. In

the Proceedings of the 19th International World Wide Web Conference, pages 1361–1364, April 2010, Raleigh, USA

20. S. Coppens, E. Mannens, and R. Van de Walle. Digital Long-term Preservation using a Layered Semantic Metadata Schema of PREMIS 2.0. In *the Proceedings of the 2nd CULTURAL HERITAGE on line Conferenc*, pages 1–6, December 2009, Florence, Italy

21. D. Van Rijsselbergen, M. Verwaest, E. Mannens, and R. Van de Walle. On how Metadata Enables EnrichedFfile-based Production Workflows. In *the Proceedings of the SMPTE Annual Tech Conference and Expo*, pages 1–19, October 2009, Hollywood, USA

22. L. Hauttekeete, T. Evens, E. Mannens, and R. Van de Walle. The Repurposing of Archive Content: The PokuMOn Project. In *the Proceedings of the International Federation of Television Archives World Conference*, pages 1–20, October 2009, Beijing, China

23. D. Van Rijsselbergen, B. Van De Keer, M. Verwaest, E. Mannens, and R. Van de Walle. Movie Script Markup Language. In *the Proceedings of the 9th ACM Symposium on Document Engineering*, pages 161–170, September 2009, Munich, Germany

24. P. De Potter, P. Debevere, E. Mannens, and R. Van de Walle. Next Generation Assisting Clinical Applications by using Semantic-aware Electronic Health Records. In *the Proceedings of the 22nd IEEE International Symposium on Computer-Based Medical Systems*, pages 1–5, August 2009, Albuquerque, USA

25. R. De Sutter, K. Braeckman, E. Mannens, and R. Van de Walle. Integrating Audio-visual Feature Extraction Tools in Media Annotation Production Systems. In *the Proceedings of the 13th IASTED International Conference on Internet and Multimedia Systems and Applications*, pages 76–81, August 2009, Honolulu, USA

26. B. Van De Keer, D. Van Rijsselbergen, M. Verwaest, E. Mannens, and R. Van de Walle. Extending a Data Model for a Drama Product Manufacturing System with Re-purposing Support. In *the Proceedings of the 13th IASTED International Conference on Internet and Multimedia Systems and Applications*, pages 105–110, August 2009, Honolulu, USA

27. D. Van Rijsselbergen, B. Van De Keer, M. Verwaest, E. Mannens, and R. Van de Walle. On the Implementation of Semantic Content Adaptation

in the Drama Manufacturing Process. In *the Proceedings of the 10th IEEE International Conference on Multimedia & Expo*, pages 822–825, June 2009, New York, USA

28. D. Van Rijsselbergen, B. Van De Keer, M. Verwaest, E. Mannens, and R. Van de Walle. Enabling Universal Media Experiences through Semantic Adaptation in the Creative Drama Production Workflow. In *the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 296–299, May 2009, London, United Kingdom

29. L. Hauttekeete, T. Evens, E. Mannens, and R. Van de Walle. Browsing through Memories: the Online Disclosure of Oral History in Flanders. In *Abstract Book of the 1st Global Conference on Digital Memories*, pages 22–22, March 2009, Salzburg, Austria

30. D. Van Deursen, W. Van Lancker, T. Paridaens, W. De Neve, E. Mannens, and R. Van de Walle. NinSuna: a Format-independent Multimedia Content Adaptation Platform based on Semantic Web Technologies. In *Proceedings of the 10th International Symposium on Multimedia*, pages 491–492, December 2008, Berkeley, United States

31. D. Van Deursen, C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. XML to RDF Conversion: a Generic Approach. In *Proceedings of the 4th International Conference on Automated Production of Cross Media Content for Multi-channel Distribution*, pages 138–143, November 2008, Florence, Italy

32. B. Van De Keer, D. Van Rijsselbergen, E. Mannens, and R. Van de Walle. A Framework for RESTful Object Exchange through Schematized XML (unRESTRricted). In *Proceedings of the 3rd IEEE International Conference on Digital Information Management*, pages 593–598, November 2008, London, United Kingdom

33. R. De Sutter, E. Mannens, D. Van Rijsselbergen, and R. Van de Walle. Automatic News Production. In *Proceedings of the International Broadcasting Conference*, pages 158–165, September 2008, Amsterdam, The Netherlands

34. O. Braet, L. Van Audenhove, E. Mannens, and R. Van de Walle. The Business Ecosystem Surrounding Digital Video DRM. In *Proceedings of the 19th International Telecommunications Society - European Regional Conference*, pages 1–16, September 2008, Rome, Italy

35. O. Braet, L. Van Audenhove, E. Mannens, and R. Van de Walle. The Business Ecosystem Surrounding Digital Video DRM. In *Abstract book of the 19th International Telecommunications Society - European Regional Conference*, pages 21–21, September 2008, Rome, Italy

1.5.3 W3C Standardisation

1. Editor of Working Draft on 'Media Fragments URI 1.0' (Last Call), June 2010, see `http://www.w3.org/TR/media-frags/`

2. Editor of Working Draft on 'Use Cases and Requirements for Media Fragments', December 2009, see `http://www.w3.org/TR/media-frags-reqs/`

3. Contributor to Working Draft on 'API for Media Resource 1.0' (Last Call), June 2010, see `http://www.w3.org/TR/mediaont-api-1.0/`

4. Contributor to Working Draft on 'Ontology for Media Resource 1.0' (Last Call), June 2010, see `http://www.w3.org/TR/mediaont-10/`

5. Contributor to Working Draft on 'Use Cases and Requirements for Ontology and API for Media Object 1.0', January 2010, see `http://www.w3.org/TR/media-annot-reqs/`

6. Contributor to XG Report on 'Multimedia Vocabularies on the Semantic Web', July 2007, see `http://www.w3.org/2005/Incubator/mmsem/XGR-vocabularies/`

7. Advisory Committee Representative for IBBT, and co-Chair of the Media Fragments Working Group

8. Active Member of Multimedia Semantics Incubator Group, Media Annotations Working Group, and Media Fragments Working Group

1.5.4 Other Publications

1. E. Mannens, S. Coppens, and R. Van de Walle. Een Gelaagd Semantisch Metadatamodel voor Langetermijnarchivering. *Bibliotheek- & Archiefgids*, number 5, pages 17–22, Vlaamse vereniging voor Bibliotheek-, Archief- en Documentatiewezen, September, 2009

2. E. Mannens, S. Coppens, P. Bastijns, S. Corneillie, P. Hochstenbach, L. Van Melle, S. Van Peteghem, and R. Van de Walle. In *Book '(Meta)datastandaarden voor Digitale Archieven'*, pages 1–204, Universiteitsbibliotheek Gent, June, 2009

3. E. Mannens, T. Paridaens, L.Hauttekeete, T. Evens, and J. Gysels. In *Book 'Van Horen Zeggen III - Haalbaarheidsstudie naar een Innovatieve Applicatie voor de Ontsluiting van Mondelinge Bronnen'*, pages 1–172, Universiteitsbibliotheek Gent, September, 2007

4. C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. Creating Personal Content Management Systems using Semantic Web Technologies. In *Book Chapter in 'Data Management in Semantic Web'*, Accepted for publication, Nova Science Publishers

5. S. Coppens, E. Mannens, J. Haspeslagh, P. Hochstenbach, I. Van Nieuwerburgh, and R. Van de Walle. Metadatastandaarden, Dublin Core en het Gelaagd Metadatamodel. In *Book Chapter in 'Bewaring en Ontsluiting van Multimediale Data in Vlaanderen'*, pages 46–62, Lannoo Campus, June, 2010

6. T. Evens, D. Moreels, E. Mannens, and R. Van de Walle et al. In *Book 'Access to Archives of Performing Arts Multimedia'*, pages 1–150, VTi–IBBT, August, 2009

7. S. Notebaert, J. De Cock, S. Coppens, E. Mannens, M. Jacobs, J. Barbarien, P. Schelkens, and R. Van de Walle. Digital Recording of Performing Arts: Formats and Conversion. In *Book Chapter in 'Access to Archives of Performing Arts Multimedia'*, pages 95–119, VTi–IBBT, August, 2009

8. S. Coppens, E. Mannens, and R. Van de Walle. Semantic BRICKS for Performing Arts Archives and Dissemination. In *Book Chapter in 'Access to Archives of Performing Arts Multimedia'*, pages 121–141, VTi–IBBT, August, 2009

9. L. Hauttekeete, T. Evens, E. Mannens, and R. Van de Walle. Browsing through Memories: the Online Disclosure of Oral History in Flanders. In *Book Chapter in 'Digital Memories: Exploring Critical Issues'*, pages 139–147, Inter-Disciplinary Press, March, 2009

10. T. Evens, L. Hauttekeete, E. Mannens, and R. Van de Walle. Surfen naar het Verleden. De Ontsluiting van Mondelinge Historische Bronnen in

Vlaanderen. *FARO: tijdschrift over cultureel erfgoed*, volume 1, number 1, pages 26–32, FARO, March, 2008

Chapter 2

Identifying Generic Media Processes in News Production

It's amazing that the amount of news that happens in the world every day always just exactly fits the newspaper!

Jerry Seinfeld in Seinfeld

2.1 Introduction

The Internet –and IT in general– is having a major impact on the broadcast industry. Physical carriers of audio-visual material are being replaced by files, and analogue networks by Internet Protocol (IP) based networks. Therefore, international news agencies can now distribute multimedia news items enriched with metadata to their customers as soon as they are produced. Consumer expectations have also raised far beyond the traditional radio and television medium. Cross-functional and configurable content [134], cleverly engineered to scale from a three inch screen up to High Definition (HD), will gracefully complement current uni-size news television and radio programmes. Eventually, mechanised and sequential news manufacturing methods will be often complemented by scalable, configurable, and agile methods [120] capable of supporting exactly the window of interactivity desired by the end-user [135]. Broadcasters daily distribute a number of news bulletins. These are created according to a rigid format, characterised by a limited number of items and being restricted in duration. While this is suitable for conventional broadcast, it does not meet the expectations of a news consumer using an alternative channel (e.g., Internet) or device (e.g., hand-held devices). Furthermore, the broadcaster usually discards a lot of material that does not fit the conventional format leaving potentially interested stories for groups of users unharnessed.

Moreover, the metadata life cycle in the massive audio-visual content creation news environments has undergone a significant development during the last decade. The digital evolution has changed the way how metadata is generated and how agents are involved in the metadata workflow. This, of course, opens new opportunities to exploit and enrich the content. Nowadays the annotations of the broadcasters' archives are not generated and not automatically enriched, but managed only by the archivist. Consequently, the metadata related to the production/distribution and the content itself are not treated in a uniform way. Besides, despite the existence of different standards for the creation, manipulation and querying of the metadata, most of the solutions in the broadcast industry are still proprietary or customised solutions.

However, the transformation of only the audio-visual essence chain is not sufficient for the implementation of efficient IT-based workflows. The workflow information generated and gathered during the production, i.e., the metadata, should become an integral part of the production process [145] and should be modelled and employed as such by production systems. A lot of this information is still used inefficiently and in an unstructured way, so that no automated system can take advantage of its semantics, leaving manual processing as the only option. Additionally, most tools available are concerned only with a limited subset of the production process and fail at appropriately integrating artefacts from external systems. The next level of news production is therefore characterised by complex and dynamic workflows in which it is important to produce and distribute news items as fast as possible in a quality as good as possible. Using a Digital Media Factory (DMF) as a backbone, personalised distribution and consumption of news items is enabled transparently through all levels of production by all ranks of the organisation, thus providing a more pleasant experience to fastidious consumers.

The problem within the news domain to be solved is therefore twofold. Firstly, within each media company all ranks of the organisation should be able to reason about a problem across levels given joint concepts, thereby closing the '*comprehensive semantic gap*' that historically exists between business management and technical engineering, thus smoothing the workflow between the organisation's entities and leading to a technical sound solution understood by everyone. Secondly, gaining community agreement on a small number of very basic media production processes would be a first step towards the longer term goal of providing agreed upon and rigorous descriptions for exchanging semantically annotated media assets among different companies and/or applications. This chapter therefore identifies generic media processes in news production. Section 2.2 first introduces the notion of the canonical processes of media production as related work. In Section 2.3 this knowledge is applied to the specific news production use case. Using this theoretical knowledge the requirements and architecture of the DMF are laid out in Section 2.4. Afterwards, Section 2.5 defines the global generic business processes that can be used throughout the complete organisation using the Manufacturing Resource Planning II [24] (MRPII) method, whereas Section 2.6 introduces a reliable and structured data model that models and centralises all production workflow information and metadata by employing a formal definition of the processes involved in news production. Finally, conclusions are drawn in Section 2.7.

2.2 Related Work

For a decade there is already substantial support within the multimedia research community for the collection of machine-processable semantics during established media workflow practices [41, 49, 106, 126]. An essential aspect of these approaches is that a media asset gains value by the inclusion of information about how or when it is captured or used, and how it is manipulated and organised. For example, metadata captured from a camera on pan and zoom information can later be used for supporting an editing process [18]. Though the combination of description structures for data, metadata, and work processes is promising, current approaches share an essential limitation, namely that the descriptions are not shareable. The problem is that each approach provides an implicit model for exchanging information that serves the particular functionality and process flow addressed by a particular environment.

To address the issues of metadata capture, preservation, and exchange, [63] proposed an approach to improve interoperability of (semantically-rich) multimedia systems based on a model of canonical processes of media production. The hypothesis is that the existence of such a model can facilitate interoperability among software built by the multimedia community. The term canonical[1] has been chosen to *indicate that a canonical process represents the highest level of abstraction for the description of a process that can be shared among systems.* Non-canonical processes can be constructed by combining canonical ones. While there exist many standards that try to facilitate the exchange between the different media process stages [136] –such as the Material Exchange Format [157] (MXF) and the Advanced Authoring Format [4] (AAF)–, the goal was to capture and model the whole media production process, starting from the early ideas sprouting of authors' genius. This model is therefore *not* a replacement for, but complementary to existing models, where the aim is defining a framework where existing standards can be integrated as detailed specialisations or instances of particular concepts of the model of these canonical processes of media production.

Based on an examination of existing multimedia systems, nine canonical processes of media production have been identified [63]. Every process introduced into the core model has at least several instances in existing systems. The core model, therefore, does not contain processes that are specific for particular systems. In short, these nine processes can be defined as follows:

- *Premeditate*, where initial ideas about media production are established.
- *Create media asset*, where media assets are captured, generated or transformed.
- *Annotate*, where annotations are associated with media assets.
- *Package*, where process artefacts are logically and physically packaged.
- *Query*, where one (both human and computer) retrieves a set of process artefacts.
- *Construct message*, where one specifies the message they wish to convey.
- *Organise*, where process artefacts are organised according to the message.
- *Publish*, where final content and user interface presentation is created.
- *Distribute*, where final interaction between end-users and produced media occurs.

[1] Canonical: 'reduced to the simplest and most significant form possible without loss of generality', see http://wordnet.princeton.edu/perl/webwn?s=canonical

The canonical processes help the multimedia community to communicate about their systems at both a descriptive and computational level. The next two subsections give an overview of advantages of specifying these canonical processes explicitly and their relationship to workflow patterns.

2.2.1 Benefits of descriptions in terms of the canonical processes

2.2.1.1 Identifying omitted functionality

Identifying and aligning the processes implemented by a particular system with the canonical processes enables a comparison of system functionality. This allows implementers to identify functionality not currently implemented and to make informed decisions as to whether to implement the missing functionality in their own system or to search for an existing, compatible component.

2.2.1.2 Improving interoperability

Comparing current system functionality with the canonical processes also improves interoperability. Defining both application and framework functionalities in terms of canonical processes allows the decoupling of the actual abstraction steps and visualisation and avoids hard-wiring design decisions and context information, thus making the Application Programming Interfaces (API) between different systems within the same workflow more robust. The canonical processes also help to envisage future scenarios where some of the processes within a system could be exchanged with those from another using those API's.

2.2.1.3 Using annotations in different processes

Much work in the multimedia community deals with analysing visual or audio media and extracting higher-level representations of the features found. By identifying a number of other processes, the role of annotations in each of these can now be discussed. For example, the *organise* process can use semantics from both the *message* and the results retrieved by a *query* for more user-friendly ways of grouping and ordering material. The *publish* process can also benefit from information about the media characteristics (such as bandwidth required, or aspect ratio) for selecting items for a specific platform.

2.2.2 Canonical Processes and Workflows

There are several established workflows in multimedia production [7,50]. The relationship of the canonical processes of media production to workflow patterns is that we propose basic building blocks that can be coordinated using different workflow patterns. In other words, we are *not* interested in defining a workflow of media production, but only the building blocks where workflow patterns can play a role of a higher-level coordination language for our canonical processes.

2.3 Canonical Processes of News Production

Multimedia news production differs from other genres or formats, and an important differentiator is the perception of quality. Given an intrinsic editorial quality, it is considered important to produce and distribute news as soon as possible to as many channels as possible, in an audio-visual quality as good as possible [64]. Therefore the news format is an excellent candidate for multi-channel distribution [9], including Internet terminals that need to cope with varying distribution capacities and mobile devices with limited rendering capabilities [111].

Items are brought on-air or on-line as soon as possible and they will be continuously enhanced by additional material during their life cycle [127]. Given an arbitrary life cycle of an item being 24 hours and the requirement to issue a news bulletin at least once per hour, we conclude that news production is in fact a moving target. This poses complex architectural constraints on the production facility. Driven by the highly dynamic nature of the format, the phases of news bulletin composition (*organise*), story editing (*construct message*), production, publishing, and distribution overlap and interfere. The process artefacts exchanged between them are continuously in a ‘draft’ state. The audio-visual product is never finished as such; it is in fact the mastering (*publish*) process that ‘de facto’ executes a final assembly. Modelling by Unified Modelling Language[2] (UML) has allowed us to make abstraction of this concurrency problem, to formally define each important input and output, and eventually to define a scalable architecture.

In the remainder of this section, we examine per business process how news production is represented by the canonical processes as defined in Section 2.2. The logistic processes of Sales and Operations Planning (SOP) (depicted in Figure 2.1) is an implementation of *premeditate* and Product Engineering (PE) (depicted in Figure 2.2) includes *organise* and *construct message* representing respectively news bulletin composition and story editing. News

[2] Object Management Group’s UML, see `http://www.uml.org/`

Material Procurement (depicted in Figure 2.3) is a particular type of *media asset creation* and includes goods receipt, Electronic News Gathering (ENG), and material editing. Based on the assumption of scalability of the publishing process, we then deduct the requirements of the material warehouse (depicted in Figure 2.3), multi-channel publishing, and distribution systems (depicted in Figure 2.4).

2.3.1 Planning and Sales – Premeditate

The SOP is the process that formally integrates the sales, production, and distribution logic by an overall planning implemented by an Enterprise Resources Planning (ERP) system. It is a reflection of the business strategy and therefore it involves the management, represented by the producer. Input is based on management decisions in general, in particular the product portfolio and the forecast, which is in fact a long-term schedule of production and distribution. The SOP issues production orders and distribution orders, based on which the DMF operates. The SOP is a typical specialisation of *premeditate*. Before the actual production of news is started by the SOP through the release of a production order, the planning system expects a sales order from the sales process (see Figure 2.1). While the sales process is usually implicit in the context of news broadcasting operations –there is no sell operation or financial settlement– the producer's commissioning decision is based on indirect user input and it is formally implemented by a sales order. In the context of video-on-demand, broadband Internet, or mobile access, sales processes and orders are explicit, and different business models co-exist, including pay-per-view, subscription-based (on-line) channels, or services paid by advertisement. Therefore, it is plausible to generalise the concept of a sales process, which is a specialisation of the *premeditate* process as well. In this context, interactivity can be easily understood as an extension of the sales process, which, as indicated in Figure 2.1, receives explicit or implicit input from the user. As opposed to the current situation where pay-per-view is considered the maximum level of interactivity, it is expected that the intelligence of future sales processes will drastically increase, allowing the user to express his interest and a desired level of quality.

The product portfolio implements a generic Bill Of Material (BOM) per type of product that can be produced and distributed by the DMF, including non-news programmes [178]. The BOM is a formal definition of the product in terms of raw material and semi-finished components, based on which product engineering processes will specify the content of each audio-visual product.

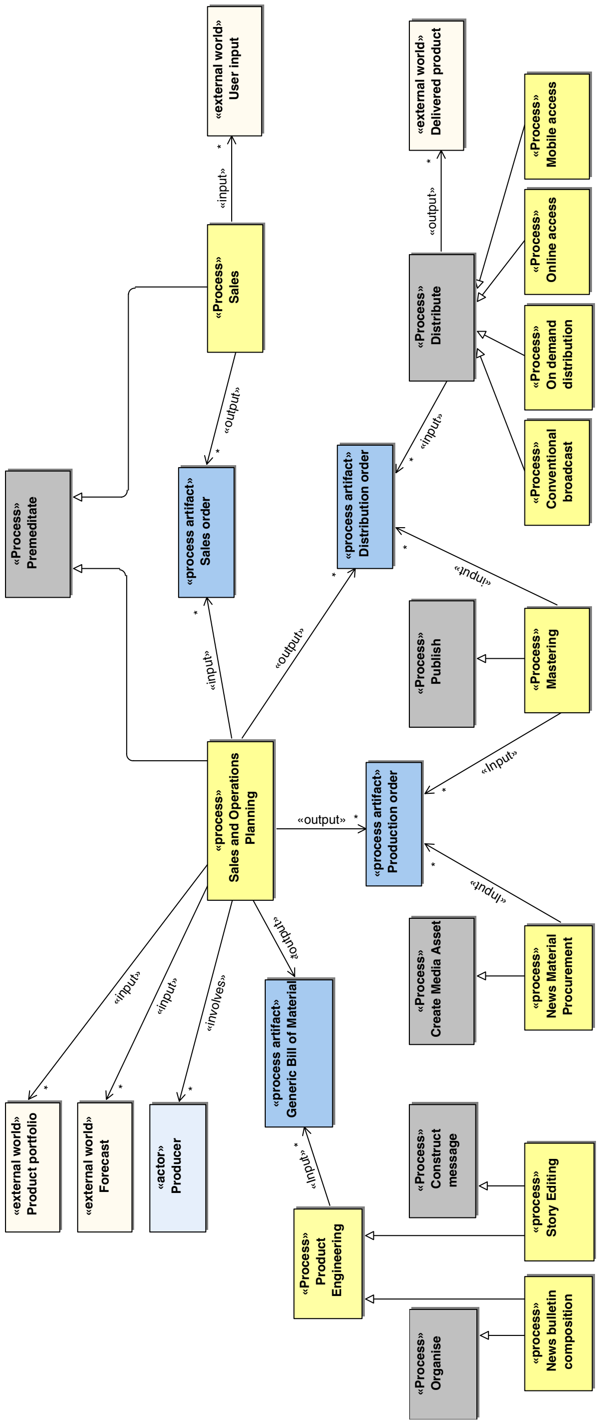


Figure 2.1: Sales and Operations Planning.

In the context of news production, audio-visual products do usually not match the individual news bulletins. Given an arbitrary assumption that an item is usually newsworthy during 24 hours, the news product is arbitrarily defined as the entire news production during a period of 24 hours and referred to as a news day. During this period, the amount of original footage that can be produced for the news day is primarily constrained by the production capacity, while in principle an unlimited number of news bulletins can be published and distributed. In fact, news is considered mass production. As an example, the VRT news department issues 4 bulletins per day for television, 26 for radio, and mobile and broadband Internet support 'ad hoc' mastering of a bulletin that contains the most up-to-date collection of news items.

2.3.2 Product Engineering – Construct Message and Organise

Media production is usually preceded by one or more PE processes and in ideal circumstances the product specifications are approved before the production starts. In the case of news production, the product specification is in a continuous draft state and the PE processes will be executed in parallel with the material production, publishing and distribution processes.

Whereas the SOP issues a formal definition of the news day as a BOM, PE processes are rather concerned with the editorial specification of the news day and in this news case by extension of each individual news bulletin. In the context of news operations, we have identified two PE processes, as depicted in Figure 2.2, based on different outputs and different actors that take editorial decisions.

- *Story Editing* is a specialisation of *construct message*. It is an interpretation of a news event by a news reporter. Eventually the news item may also refer to the specifications of the material that will be captured, retrieved from the archive or edited in order to illustrate the item. These annotations are referred to as component descriptions in Figure 2.2. In particular cases, story editing can be partially outsourced to an external news agency. Then again, the news agency would issue a formal description of the news event that is then used as input of an internal re-interpretation.

- *News bulletin composition* is a specialisation of *organise*. While the structure of a news bulletin is usually based on a fixed template, it is populated by news items delivered by the story editing (construct message) process. The output is a *rundown*; a document structure that includes

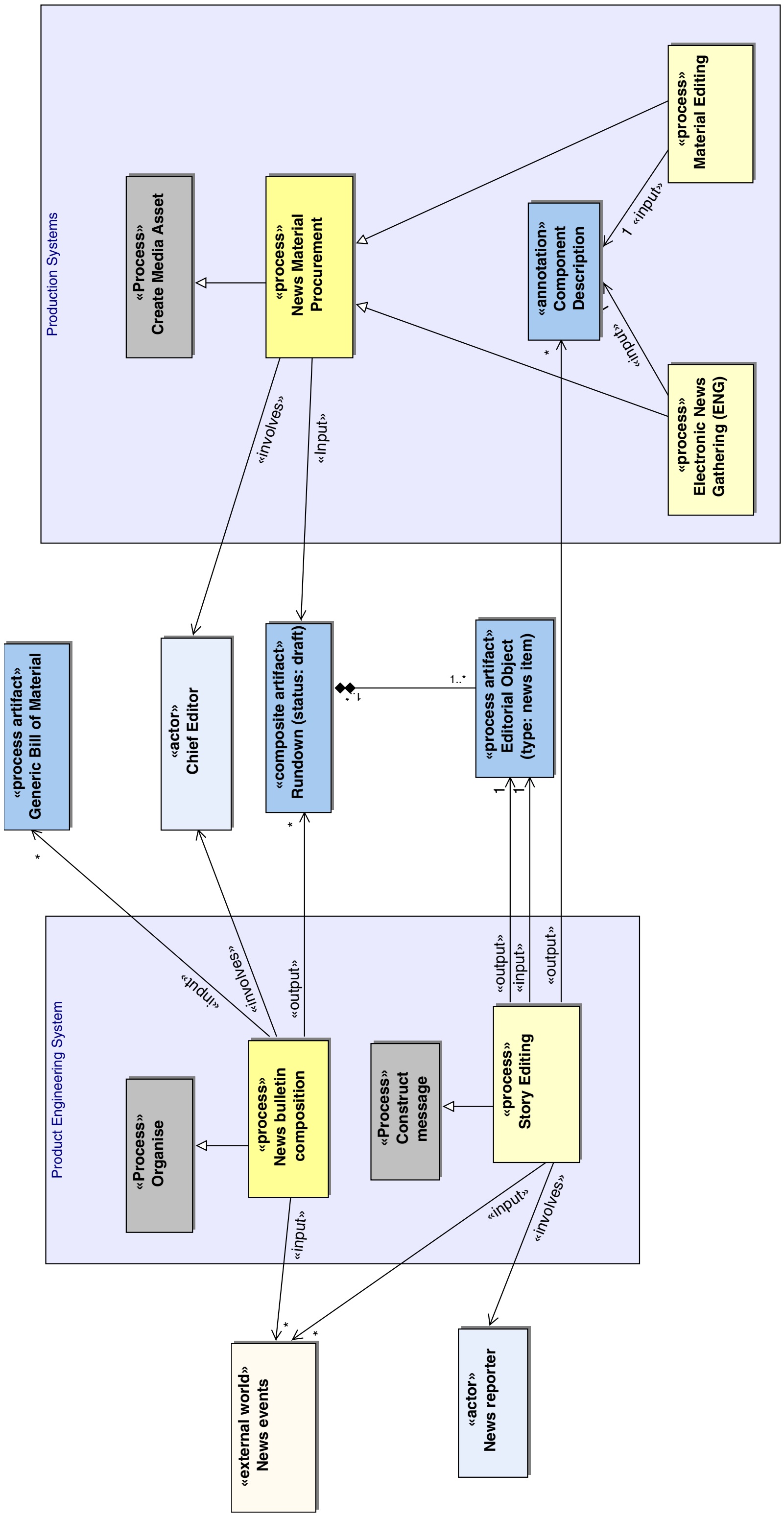

Production Systems
«Process» Create Media Asset
«process» News Material Procurement
«annotation» Component Description
«process» Material Editing
«process» Electronic News Gathering (ENG)
1 «input»
1 «input»
*
«involves»
«input»
«process artifact» Generic Bill of Material
«actor» Chief Editor
«composite artifact» Rundown (status: draft)
«process artifact» Editorial Object (type: news item)
1..*
1
*
*
*
1
1
Product Engineering System
«Process» Organise
«process» News bulletin composition
«Process» Construct message
«process» Story Editing
«input»
«involves»
«output»
«output»
«input»
«output»
«input»
«input»
«involves»
*
*
«external world» News events
«actor» News reporter

Figure 2.2: Product Engineering.

individual news items that contain pointers to the media assets that will be or have been created in the context of this news item. Each decision to include/exclude an item in/from the rundown is formally taken by the chief editor. As such, the rundown ideally contains any decision that needs to be taken during production and mastering.

As of today, an important fraction of the necessary creative decisions are not documented and a fair amount of human interaction is required during production and mastering. Automatic mastering of live programmes, such as news, would require a virtual modelling environment capable of pre-visualising the news and delivering a precise model based on which the master control can be automated. Research was also done within the PISA-project, as can be seen in Appendix E.12, in order to stretch the idea of virtual modelling and to estimate to what extent production and mastering can be automated.

2.3.3 Material Procurement – Create Media Asset

Production refers to any process that converts raw or semi-finished material to a state of further completion. Procurement is a generalisation of production that includes any other type of material acquisition, e.g., the intake of news agency material. In any case, news material procurement is a typical specialisation of *create media asset*. In the context of news production we have identified three types of material procurement, as indicated by Figure 2.3. Any of these require in principle a production order issued by the SOP, and an approval decision of the director is reflected as an approved update of the rundown. A process formally classified as a *media asset creation* process, not news related, is the goods receipt or ingest process. After news footage is captured, aggregated, and issued by a third party such as a news agency or a correspondent, material is recorded and transferred to the material warehouse as news feeds, i.e, raw material of type ‘news feed’. In case metadata is available, it is usually presented as a separate unstructured document or embedded as an XML-wrapped structured document [20, 80]. This document, referred to as a ‘dope sheet’, is extracted or retrieved and issued to the material annotation process as a basis for formal annotation.

- The most common process of news material production is ENG. Given the description of a news item and an indication of the material that needs to be captured to illustrate the material, a news reporter (if necessary assisted by technical staff) captures material on location. Material is checked in as a collection of shots (raw material of type ‘shot’) and the creation metadata is issued to the material annotation process as a basis for formal annotation.

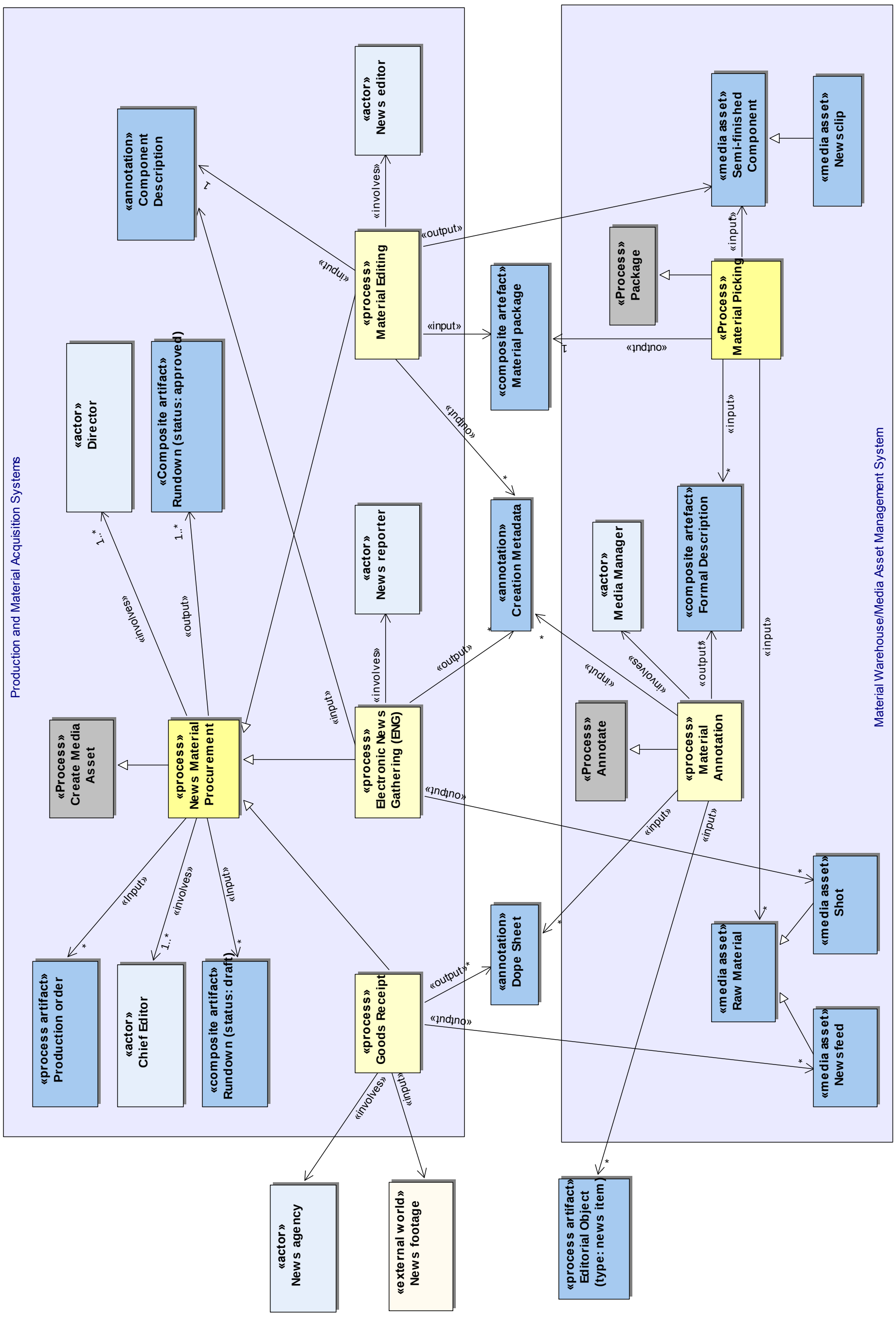

Production and Material Acquisition Systems
Material Warehouse/Media Asset Management System
«annotation» Component Description
«actor» News editor
«process» Material Editing
«actor» Director
«Composite artifact» Rundown (status: approved)
«actor» News reporter
«composite artefact» Material package
«annotation» Creation Metadata
«Process» Create Media Asset
«process» News Material Procurement
«process» Electronic News Gathering (ENG)
«process artifact» Production order
«actor» Chief Editor
«composite artifact» Rundown (status: draft)
«process» Goods Receipt
«annotation» Dope Sheet
«actor» News agency
«external world» News footage
«process artifact» Editorial Object (type: news item)
«media asset» Semi-finished Component
«media asset» News clip
«Process» Package
«Process» Material Picking
«composite artefact» Formal Description
«actor» Media Manager
«Process» Annotate
«process» Material Annotation
«media asset» Shot
«media asset» Raw Material
«media asset» News feed
«involves»
«input»
«output»
1
1..*
*

Figure 2.3: Material Procurement and the Material Warehouse.

- In normal circumstances, raw material received by the ‘goods receipt’ or ENG processes will not be distributed as such. It needs to be cut and pasted, mixed with archive material, and sound engineered. In fact, in the context of news, material editing refers to any activity associated with the creation of a news report (semi-finished material of type ‘news clip’) based on existing material, with the intention of creating a media asset that can be distributed as such.

2.3.4 The Material Warehouse - Annotate and Package

Before the DMF had to support scalable multi-channel publishing, the processes of television and radio production were tightly integrated and optimised to deliver exactly one version of a product. The BOM was trivial: a single production process delivered a finished product and a notion of intermediary components was considered overhead. Mainly intended for preservation of cultural heritage, the ‘deep archiving’ process was the only instance of a formal annotation process where both annotation and archiving were optimised for long-term conservation and retrieval of end-of-life media assets. As suggested, multi-channel publishing has introduced a more complex BOM, supporting a configurable product and a modular production process. For example, in the context of news we have defined production processes that deliver semi-finished components and a dedicated mastering process that will be discussed in the next paragraph. Within the PISA-project (see also Appendix E.12) we have built an intermediary system that implements material warehouse operations that store intermediary components and make them available again. The system validates or provides the required identifiers, analyses the material and eventually normalises all available metadata in order to enable efficient query and packaging processes. The process of collecting all available information and normalisation is indicated in Figure 2.3 as material annotation, which is a typical specialisation of the canonical process *annotate*. As shown in Figure 2.3, the material picking process, which is a typical specialisation of *package*, issues a material package to the material editing or mastering processes. Although supervised by a media manager, material annotation is a largely automated process that delivers a formal and machine-readable description of the asset, which may include any type of identification and descriptive properties [48].

- Identification properties range from low-level object identification such as a Unique Material IDentifier (UMID) [135, 137] or a Universally Unique IDentifier (UUID) [97] up to logistic identifiers such as the product identifiers and sales order numbers.

- Descriptive information strictly accounts for the content of the material. In case no descriptive information would be available during check-in of the material, we expect that the process of material description has the potential to be automated completely. Based on available research results, we currently implemented automatic shot and scene detection [42], speech analysis, automatic summarisation, and object detection based on image processing [40, 61].

- Rights related information identifies the copyright holder and the rights that have been granted. Thus, the material warehouse system in fact executes the Intellectual Property Management (IPM) processes and inhibits improper reuse of material to the extent it has been provided correct information.

The material warehouse is implemented by configuration of a Media Asset Management (MAM) system. Being the central node of the hub-and-spoke architecture, as opposed to a conventional linear approach where it strictly implements archiving processes, it is optimised for scalable throughput and fast response times. Archive related activities are strongly entwined with the material warehouse operations. In fact, production-archiving has become a trivial operation. Since the introduction of a MAM system that supports the complete production process, the archived state is rather a logical setting in the MAM system and no additional archive records are created. The media manager (formerly known as the archivist) still removes duplicates, improves the available metadata, and creates clusters of material in order to improve *query* and *packaging* processes. The canonical process *query* represents activities associated with the retrieval of objects from the archive. Since (active) archiving as such is not part of the production logic, it is not elaborated on prior to Chapter 4.

2.3.5 The Distribution System - Publish and Distribute

As can be seen in Figure 2.4, a distribution system usually implements the adaptation logic or the final assembly as function of a particular distribution channel, as well as the actual distribution. As suggested in the introduction, it is primordial to automate these components to a large extent in order to create a scalable multi-channel publishing system.

The canonical process *publish*, commonly known as *mastering* in the environment of audio engineering and as recently being re-introduced in the context of television production by AMWA[3], represents the various packaging and labelling operations of a finished product as function of the capabilities of a physical distribution channel, the type of client used by the consumer, and/or particular user preferences. Mastering may include the following operations:

- Labelling is easily automated by simple in-line systems. It includes the insertion of logos, embedded programme information, and navigation anchors. It also applies watermarks and copy protection artefacts, such as scrambling.
- Insertion of interstitials and commercials is automated by integration of scheduling and distribution events. It requires a tight and semantically rich interface between the distribution scheduling and execution systems.
- The final down-mix of the available audio into one, two, or six audio-channels is considered a mastering process as well. This is not a trivial operation because the amplitude must be manipulated as function of the available dynamic range and per type of audio component, being the dialogue speech, the background, the soundtrack, and the bruitage or effects. This is a typical example where it is important to define a meaningful BOM and to formally identify the different audio components as different semi-finished components.
- A complicated task and a problem which is subject of current research, is intelligent scaling and cropping of the video aspect [105,166]. Given the display capabilities of the client device, the image adaptation requires a notion of the Region Of Interest (ROI), which, if not available as a result of PE, requires image analysis in order to intelligently re-frame the picture.

As opposed to the definition of a single mastering process, we hereafter define a specific distribution process per logical distribution channel, which can be any one or combination of physical distribution channels. For example, simultaneous broadcasting over analogue FM[4] and digital DAB[5] would be considered one logical distribution channel. In the context of news production, the canonical process *distribution* represents four typical processes according to four logical distribution channels as depicted in Figure 2.4:

[3] Advanced Media Workflow Association, see `http://www.aafassociation.org/`

[4] FM: Frequency Modulation

[5] Digital Audio Broadcasting, see `http://www.worlddab.org/`

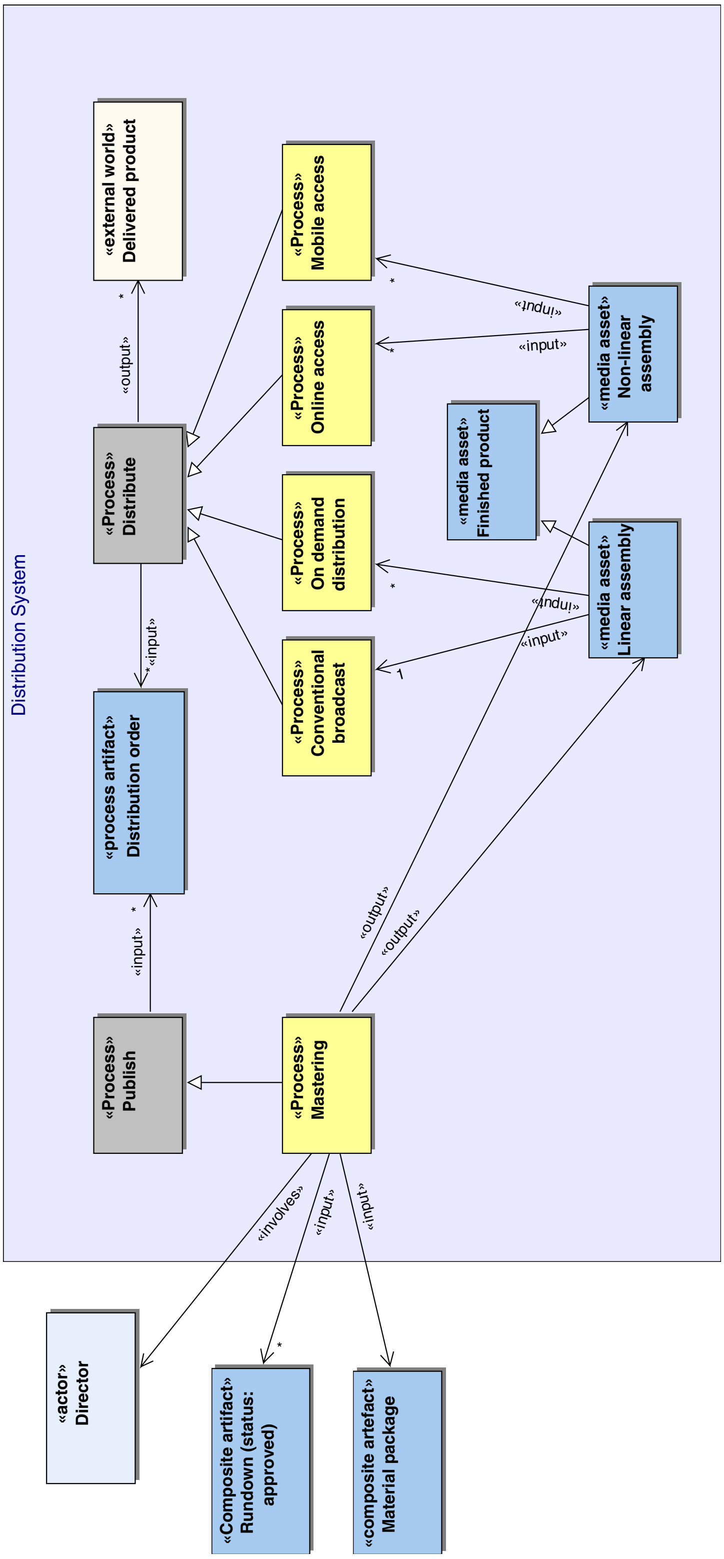


Figure 2.4: Mastering and Distribution.

- Conventional broadcast refers to linear radio and television.

- On-demand distribution refers to 'ad hoc' distribution of linearly assembled audio-visual content. As of today, this distribution process often reuses the product that had been mastered for conventional broadcast services. This compromises the quality of the information, because the original assembly usually includes irrelevant promotional references or interrupts (e.g., traffic information) and therefore it is considered better mastering a specific version of the news bulletin intended for on-demand distribution.

- The news bulletin assembled for on-line access is presented as a structured overview and the end-user explicitly selects item per item. In fact, the on-line news bulletin behaves like a richly featured Web site and the end-user uses a browser that 'ad hoc' fetches the required files.

- Mobile access is promising as the possibilities of mobile access (see also the MADUF-project in Appendix E.17) are explored. Mobile and handheld devices are featured with multiple radios and it is expected that they will include a broadcast receiver (DVB-H[6]), a communication channel (GSM[7], GPRS[8], or UMTS[9]), and sufficient local storage. They have the potential to overcome their limited rendering capabilities as it is probable they will be equipped with wireless interfaces supporting communication with high-end play-out devices. As such, despite their limitations in terms of quality and capacity, mobile terminals are becoming the most feature-rich terminals.

2.4 DMF Principals and Architecture

Now that we identified the generic media processes in news production, we can start reasoning about the requirements and architectural constraints to build a future-proof DMF on top of a (meta)data model that adheres to different types of media production and that is well-understood by the complete hierarchy of the broadcasting organisation.

[6] DVB-H: Digital Video Broadcasting for Hand-helds

[7] GSM: Global System for Mobile communications

[8] GPRS: General Packet Radio Service

[9] UMTS: Universal Mobile Telecommunications System

2.4.1 General Prerequisites and Observations

2.4.1.1 File-based Workflow Paradigm

The underlying philosophy of the DMF is to realise an end-to-end, fully integrated file-based workflow, from acquisition (i.e., file-based cameras and production servers) to play-out and (short-term) archive. Traditional tape-based workflows are replaced by content-centric non-linear workflows, where different operators can perform different production steps at the same time. The file-based paradigm enables cross-media reuse and repurposing. Media files are preferably stored in their contribution or production format, while keeping transcoding and file transfers to an absolute minimum. Each new investment is oriented towards this file-based philosophy. Furthermore, generic IT tools are used as much as possible. Flexibility and programmability are some of the obvious advantages. An optimal selection and fine-tuning of mostly IP-based components to the specific needs of media production and processing, ideally result in both higher performance and lower cost. For the central storage architecture, IBM's General Parallel File System[10] (GPFS) clustered file system has been selected to build up a high capacity file server and storage environment. Specialised equipment is mainly utilised in Work Centres (WoC) such as editing and play-out automation. Tape-robots are used for deep archiving, but a link to the short-term MAM archive is always maintained, as the proxy copy remains on-line at all times.

2.4.1.2 Integrated and Layered Architecture

The DMF is based on an open and layered architecture. The different layers (storage/archive and network infrastructure, production, information and business service layer) are loosely coupled. In order to facilitate the integration and coupling of the different system components and domains, the single sourcing principle –i.e., one vendor solution for each functional area– is applied as much as possible. Open standards such as MXF [157], AAF [4], P/Meta [51], SOAP [59], and XML [20] have been adopted to ensure interoperability. More specifically, MXF OP1a [158] has been selected as the standard file format, wrapping essence in either DV-25 [33] or D-10 (IMX 50) [156] using UMID's [155] along the workflow (re-)travelling.

[10] GPFS Resources, see http://www-03.ibm.com/systems/clusters/software/gpfs/resources.html

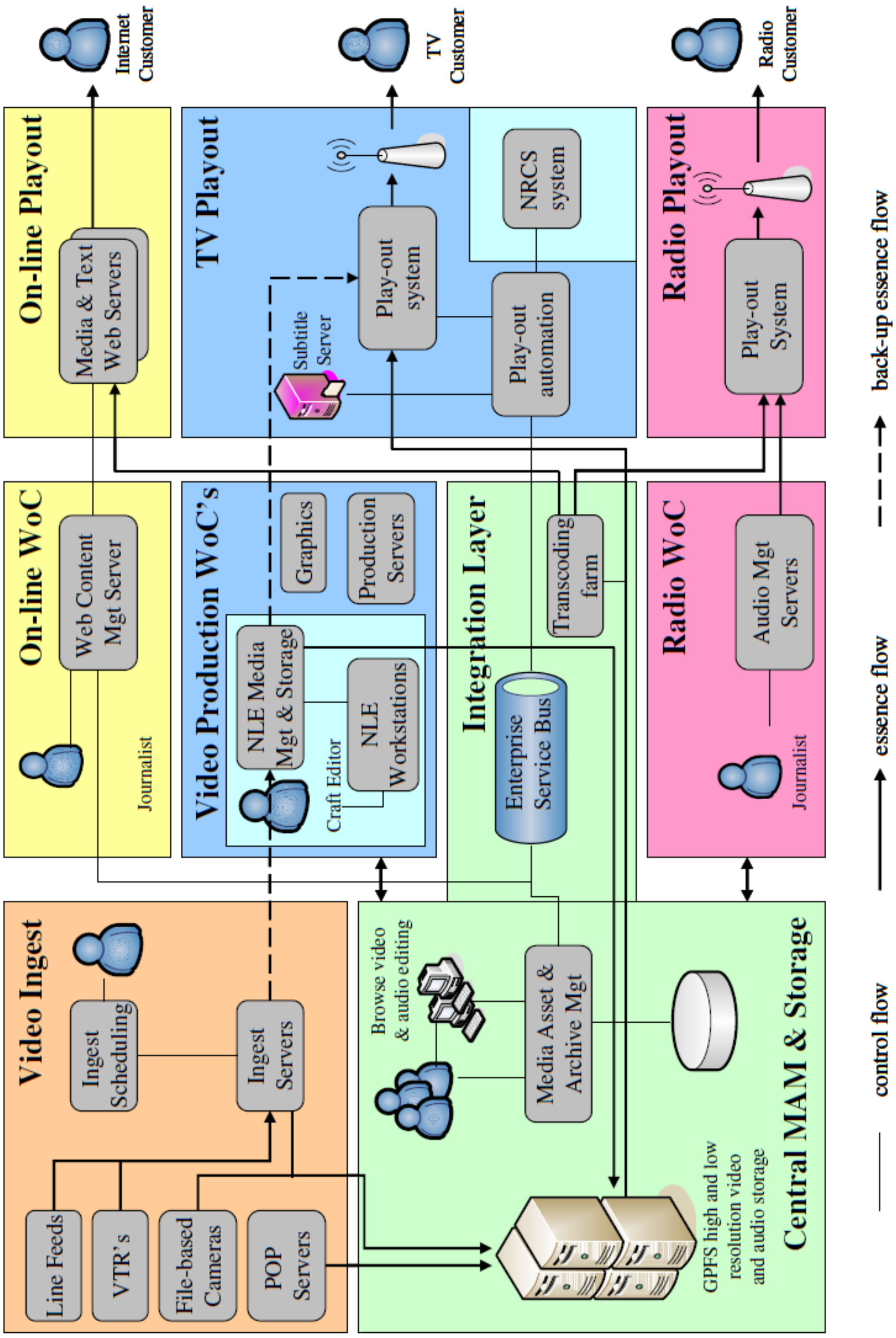


Figure 2.5: Overview of the DMF Architecture.

2.4.1.3 WoC and Central MAM

The boundaries of the DMF are demarcated at the transition between the real-time infrastructure and the file-based production environment. It consists of different, specialised WoCs such as audio and video editing, subtitling, and graphics which are connected to a central MAM system (Ardome[11] by Vizrt), as can be seen in Figure 2.5. The MAM system serves as a media-aware hub and repository for production and archived material. The Ardome system is the main access point to media essence for a large media production user community (e.g., journalists, programme assistants), by offering search, retrieval, browse, and simple editing functionality on low resolution material. By contrast, the WoCs are typically specialised, optimised for a particular task, and targeted to a smaller group of craft users (e.g., video editors).

2.4.1.4 Separation of Essence and Metadata

Metadata is gradually enriched throughout the different steps of the production lifecycle of the related media content. Notwithstanding the ubiquitous application of MXF as the storage and interchange format in the DMF and its inherent capability to wrap media and metadata, the latter is strictly confined to a few technical parameters, such as time code, aspect ratio, video and audio format. As a general rule, media and metadata are handled separately in order to ensure the integrity in case of parallel usage of metadata in different parts of the process (e.g., in the craft editors). Synchronisation of media and metadata between MAM and WoCs is handled by the SOAP-based [59] integration layer. Within the MAM's proxy all ingested high-res material is converted to low-res I-frame only MPEG-1 [84], which is able to be used by a preview tool (e.g., PreCut), thus saving valuable bandwidth when previewing the essence. Within the craft editing Stations (e.g., AVID[12] & Final Cut Pro[13]) the original ingested high-res material within MXF is constantly used.

2.4.1.5 Production Models and Application Areas

In principal, two types of media production can be distinguished: item-based cross-media production on the one hand (news use case) and project-based cross-media production on the other hand (drama use case). In item-based production, the emphasis lies on simultaneous, short-term, high-volume media production, and reuse by a large number of users in a heterogeneous

[11] Viz Ardome Media Asset Management solution, see http://www.vizrt.com/products/article138.ece

[12] http://www.avid.com

[13] http://www.apple.com/finalcutstudio

environment (television, radio, and on-line). Typical examples are news, sports, and cultural programmes. In this type of production, automation is an indispensable requisite and integrations between the central MAM system and WoCs are limited to the level of a media item or a container of media items, and simple timeline information. In this context, the DMF model, as depicted in Figure 2.5, with a prominent role for the central MAM system, prevails. As the broadband Internet medium is ideally suited for item-based distribution, it becomes an equally important outlet instead of just a derivative of television or radio production.

Project-based production typically relies on smaller teams of people and is generally more spread in time and thereupon less time-critical. Typical examples include drama and documentary production [175]. In this case, complex timeline-based integrations between the different components in the production workflow are imposed. As a consequence, the role of the WoCs becomes prevalent by combining multiple workflow steps (e.g., ingest, editing, sonorisation, and graphics) before handing it over to the central system. Although cross-media reuse remains important, the focus rather lays on a specific medium (e.g., television) in this case. From a qualitative perspective, one can identify the following application areas of television production:

- High-end production: landmarks and drama.
- Mainstream production: soap series and documentaries.
- News and sports.
- Video journalism (consumer cameras).
- User generated content.

Each of these areas corresponds to different requirements regarding quality, compression, storage, etc. From this discussion, it is clear that continuously increasing diversification in media production is inevitable and that for each particular area efficiency can only be guaranteed by clear rules and technical guidelines, optimised infrastructure and integrations, and distinct application specifications. Given the above principles and the opportunities provided by the virtualisation of the media production supply chain, we have sufficient indications about the expected behaviour of the DMF, as one will read in the following subsection.

2.4.2 Requirements and Impact on the Architecture

2.4.2.1 The DMF needs to increase its throughput

We assume that the overall consumption of media will increase as the amount of output channels increases and, e.g., mobile devices will be the bulk of the media consumers, so the amount of available material procured by the DMF needs to be increased as well. Assuming that it is not desirable and not feasible to multiply the editorial staff equally, we face a drastic rationalisation of the production processes. The more accurate and complete the specifications of the product can be delivered by the development processes, the less 'brainless' human intervention is required during production. Eventually, the actual production will be augmented leaving more time for 'creative thinking' as cumbersome, and repetitive tasks will have been automated. Model Driven Engineering (MDE) [152] and production automation are two key elements in this rationalisation. MDE moves the creative processes from being script- and document-centric to model-centric, crafting the product virtually instead of simply describing it [176, 177]. MDE makes it easier to reuse existing information and allows some automation of the production processes. Using MDE, the editorial staff will eventually take more and better creative decisions, throughout the life cycle of the product, as it is easier and cheaper to retry new and altered creative decisions.

2.4.2.2 The DMF needs to produce adaptable content

We assume that the consumer is able to access any media, whatever terminal he is using, and whatever network he is connected to [134], and this by a coherent and consistent user interface [135]. This means the DMF should produce adaptable content, unless the media production company supports manual reproduction per distribution channel, which is in conflict with the required rationalisation discussed earlier. The production process and the product can be considered adaptable if the production cost per (additional) distribution channel is marginal. To be so, MDE and production automation are essential elements. For example, in the context of drama production [175], when the storyboard would be precise enough, the editing and mastering processes can be automated and it is exactly during these processes that better channel- and terminal-specific decisions are taken. In conclusion, assuming that the 'ad hoc' creation of a different product per distribution channel is considered not adaptable, then MDE is essential in order to deliver a nominal and generic specification of the content. These (configurable) specifications can then be used as a basis for final assembly by partly unsupervised production processes.

2.4.2.3 The DMF needs to support interactivity

In the overall environment of digital media production in general, multiple DMFs will be rationalising their operations. It can be expected that the overall amount of available original audio-visual material is growing faster than the consumer's ability to process content. The DMF will need to differentiate its available material under supervision of the director, in other words setting up categories of material and being able to deliver as a function of the consumer's profile and terminal capabilities. The correlation between different types of content, defined as audio-visual formats in a particular genre, where '*format*' refers to the formal aspects and '*genre*' strictly accounts for the content (cf. P/Meta), and the profile of the consumer, requires sophisticated and conscientious planning, and it is proportional with the level of interactivity. In that sense, interactivity is more than just adaptivity, i.e., it is a measure for the understanding of the user requirements and the ability to react accordingly.

2.5 Generic Business Processes

While the media industry, and in particular radio, television, and feature film, is characterised by media-specific processes, we can easily see that it acts and interacts like any other industry on a higher level of abstraction. The generic business model which we are referring to is known as the MRPII-model [24]. The main concepts of the MRPII-model are defined by APICS[14]: The SOP is the executive planning process that correlates product engineering, production, and distribution processes.

PE is the creative process and it refers to any activity related to the design and the specification of the product. Manufacturing or production is the series of operations performed upon material to convert it from the raw material or a semi-finished state to a state of further completion. Sales processes deliver a forecast and a product catalogue. Sales orders eventually trigger the production and distribution logic. Distribution refers to any kind of dissemination of the labelled and packaged product to the consumer, by physical transport, transmission, or any other means of making the end-product available. In a digital and networked media production environment, the core business of the media industry will still be the production and distribution of feature film, radio, and television programmes. As such, we can assume that the MRPII-model still holds. However, since more distribution channels emerge, the mixture of

[14] the American Production and Inventory Control Society, is an industry forum that represents the body of knowledge in operations management, including production, inventory, supply chain, materials management, purchasing, and logistics, see http://www.apics.org/

those distribution channels is becoming more complex. The production, and by extension the engineering of audio-visual material, will therefore need a structural re-engineering. Figure 2.6 illustrates how the generic business processes of the MRPII-model map onto the media industry, which will act as an anchor point for the remainder of this section.

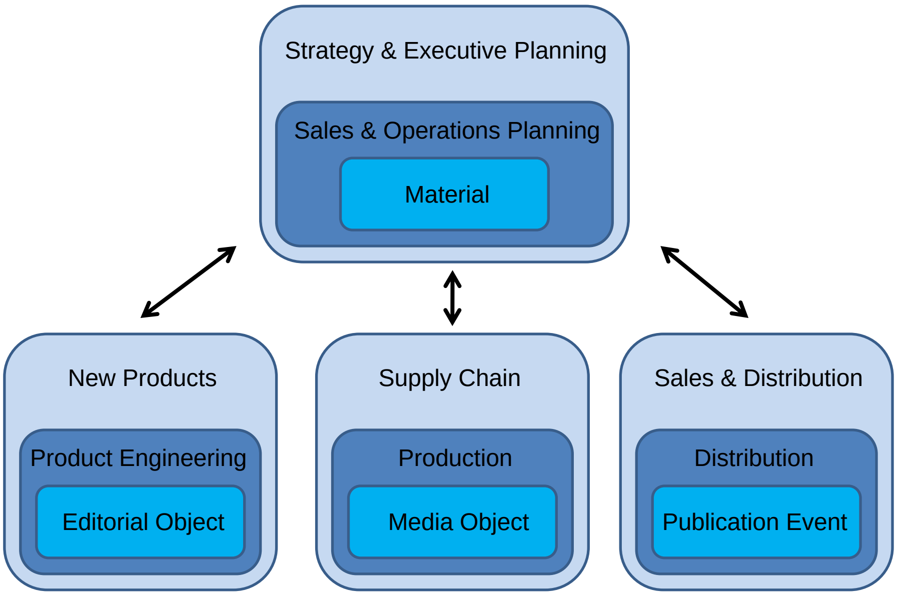


Figure 2.6: Generic Business Processes of the MRPII model.

2.5.1 Sales and Operations Planning

The SOP is a high-level planning process basically defining the context for product engineering, production, and distribution. The SOP defines product categories and types, the BOM per type of product (the raw material and semi-finished components that make up a product), and the routings (the sequence of production steps required to realise a semi-finished or finished product). Within the news scope the SOP is run by management and the editor-in-chief, whereas in the drama scope it is run by management, the producer, and the director. Furthermore, the SOP uses a representation of consumers or types of consumers, in order to make up a layered and incremental calculation of the impact of new products or changing production, or sales rates, given the actual production and distribution capacities. As such, the executive users running the SOP can schedule new products or new product types, include new distribution channels, and adapt the required production capacities accordingly. At lower levels, the SOP eventually takes care that a sales order is converted into one or more production orders or purchase orders, and that the distribution is executed in time.

2.5.2 Product Engineering

Production processes are usually preceded by a phase of design and development, which has been defined as PE. Within the news scope the PE is run by journalists and editors, whereas in the drama scope it is run by the director, and the script editor. PE is a creative process, iterative and ongoing, during which in principle all decisions are taken about the structure and the content of the product. Industries in a competitive environment usually structure and improve the PE processes, for example by MDE, in order to lower the start-up time of a production cycle as all is documented in an earlier iteration, thus indirectly improving the production process as well. In logistic terminology, delivering a range of products that slightly differ as a function of the context they will be applied in, is usually solved by designing a generic or full-featured version of the product. The notion of a configurable BOM allows options to be switched on or off during production, delivery, or while installing the product at the consumer's premises. It is clear that excellence in PE using a model-driven approach is an essential requirement in order to support the notion of a configurable BOM.

2.5.3 Production

Although a product cannot be designed without firm implicit knowledge about the production processes, the product engineer makes abstraction of the production process. For example, it does not matter for the product engineer whether the product is realised by purchasing or by production, or which production methods will be applied. Production is the process during which any decision is taken about the realisation of the product, including the actual manufacturing, by either producing or purchasing material, outsourcing, or by any combination of these. Within the news scope the production is run by journalists, editors, cameramen, and operators, whereas in the drama scope it is run by the director, script editors, cameramen, and operators.

2.5.4 Distribution

In general terms, products will only be produced and distributed when they can be sold. The sales process can be explicit, by sales orders and invoices, or implicit as for a commercial broadcast operator that distributes television programmes as if they were sold, since his advertisement income is a function of the number of viewers. Whether the sales order is a requirement before the production can be initiated (make-to-order) or not (make-to-stock), is a management decision. Both within the news scope as in the drama scope the decision making is done through the same roles as in the SOP. In any case, based on the predicted sales figures, a company can make up a forecast and a product catalogue referred to as a broadcast schedule and an Electronic Programme Guide (EPG) in media terminology. As soon as the products are sold explicitly or implicitly, have been manufactured, and are made available by packaging and labelling, distribution is executed and, if applicable, the billing processes are started.

2.6 Media Production Metadata Model

Depending on the type of media production industry, the product would be a feature film or a television programme and then again, based on the type of programme format, a programme would be referred to as a news bulletin made up of items or a drama episode which is made up of scenes. Therefore, we should identify generic and universally applicable business objects and provide self-contained definitions for each of them. Before we translate the generic business processes to media specific processes, we need some intermediary definitions: the business objects or the classes of information that are manipulated by these individual processes. The core DMF business objects are the conceptual level of the information aspect of the business architecture, i.e., they are the highest abstraction of the data model, as illustrated in Figure 2.7, which can be reasoned on by all ranks within the organisation.

2.6.1 Material

Material is defined as the logistic logical unit of work and it is the key concept which is communicated between engineering, procurement, sales and distribution processes. It is in fact the logical unit of the SOP, and by extension, of any logistic operation in any type of factory, including one for media production. The *material* type may discriminate raw material, semi-finished components, and finished products. For example, in the context of television production, the programme as it is developed, manufactured, and broadcast, would be a finished product. The individual items, edited to be assembled in a news bulletin, would be represented as semi-finished components; to such extent the notion of a configurable news bulletin is supported. The uncut news feeds, originating

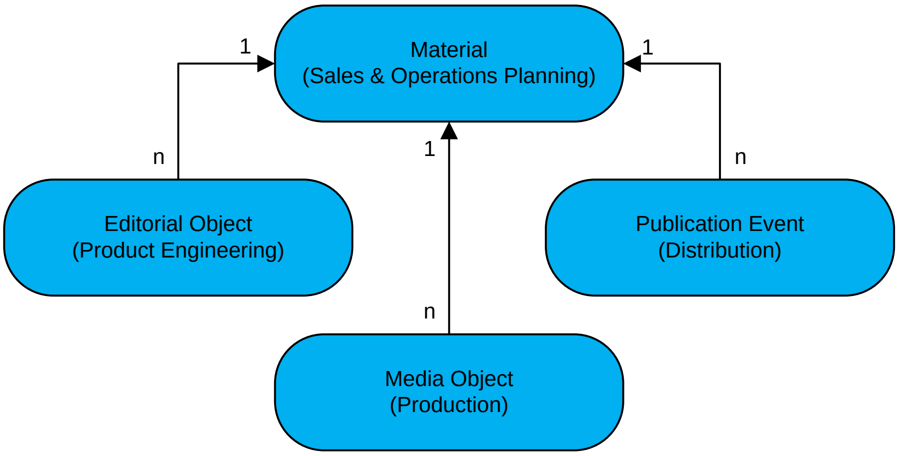


Figure 2.7: Conceptual Data Model.

from a news agency, would be represented as raw material [120]. The BOM implements the logistic structure of a product: it may express how many raw material or components are required to assemble a television programme as far as this is logistically relevant. For example, the items of a news bulletin are not represented individually by the BOM if the details of the item do not influence the planning, the delivery time, or the cost of the overall programme.

2.6.2 Editorial Object

The *Editorial Object* is the logical unit of creative work, and hence of PE; and it represents any creative decision taking during the iterative design phase of the audio-visual product. For example, in drama production, an episode is a logistic unit of work and thus represented as material, but it is actually engineered as a collection of scenes whereby each scene is an action on a particular place and point in time. During the iterative design process, scenes can be removed but the director's cut may eventually contain additional scenes. Therefore, logic suggests that each scene should be represented as an *Editorial Object* [178].

2.6.3 Media Object

Any sequence of audio-visual material is represented by a *Media Object* instance, and in case of multiple copies or files that contain the same media object, each individual instance would be represented by a *Media Ob-*

ject instance. Each *Media Object* represents the production decisions, any technology-specific details, and the technical characteristics of the audio-visual material. During the capturing of audio-visual material by a camera, each shot would be identified as a *Media Object*. During editing, each track and/or clip can be represented as a *Media Object*, although it is up to the discretion of the editing operator whether each individual component is formally identified. Usually, any *Editorial Object* instance is realised by editing one or more clips of audio-visual material. This would suggest that each *Media Object* would relate to an *Editorial Object* while *Editorial Objects* would be associated with one or more *Media Objects*. However, because *Media Objects* can exist without being formally engineered, or since multiple *Editorial Object* instances can be assembled as a single sequence or *Media Object* instance, it is more appropriate to define a relationship between the material and the *Media Object*, where the material is related with one or more *Media Objects*. As such, the DMF provides lots of freedom once again to take creative decisions at any time in the product's workflow. As such, a single programme or product, represented by a material, can be associated with multiple *Editorial Objects* while being delivered as a single *Media Object*. Otherwise, the programme can be described by a single script of an episode, represented by an *Editorial Object* while being produced and delivered as a collection of *Media Objects*, such as the video track, multiple audio tracks, and the subtitles.

2.6.4 Publication Event

The distribution aspect is implemented by another object, referred to as the *Publication Event*, which is the logical unit of distribution. The *Publication Event* represents any decision taken during distribution scheduling, which often introduces additional layers of information such as logo's, watermarks, and captions. Using the same design pattern, a programme or product is distributed as one or more individual elements.

2.7 Conclusions and Original Contributions

As we have suggested, there is a remarkable correspondence between media production and a generally applicable business logic. Media production involves strategic planning in the form of product categories in terms of formats and genres. Long-term planning in order to make predictions about production and distribution capacities can be modelled as a SOP. We have discussed that the iterative creative decision making process is a type of PE and that a model-driven approach is essential to support the production of scalable and cross-channel content. At the other end of the chain, programmes or films are sold, scheduled, and distributed by means of transmission, on-line distribution, or by packaged media.

By implementing a DMF (see also the PISA-project in Appendix E.12), conformant to the identified canonical processes of news production and conformant to the formulated principals and requirements, the file-based production paradigm has finally materialised, hereby replacing the old concept of a linear, tape-driven workflow model by a content-centric pull model. This new principle brings about a lot of salient and unprecedented advantages: concurrent engineering, edit-while-ingest, faster than real-time processing, enriched integrations, etc. To a large extent, these advantages can be attributed to the adoption of the standard file formats MXF and AAF in the integrated DMF architecture. Furthermore, additional user requirements, such as UMID, media tracking (as will be further elaborated on in Chapter 5 and Chapter 6), compression agnosticism, random access to material, and transparent metadata exchange (as will be further elaborated on in Chapter 4) are also intrinsically satisfied.

MXF replaces the classical video tape and accompanying note in file-based workflows. It is a comprehensive toolkit, by which essential system integration and media exchange can be accomplished. As such, interoperability between the different production systems (acquisition, central MAM, post production, and play-out) is primarily based on MXF in its different flavours –the so-called ‘operational patterns’. For media exchange and central archiving on the one hand, MXF OP1a is typically applied, such that the different audio-visual components can be handled efficiently and coherently, thus also enhancing network transport efficiency. In post-production environments on the other hand, MXF OP-Atoms and AAF compositions meet the particular, task-related requirements of craft editing and sonorisation. In this case, video and audio are stored separately to be edited apart.

We have identified interactivity as an extension of a sales process. Before being able to develop and support applications that will be perceived as inter-active [135], we will need to be able to formally describe the process that issues the ‘buy’ transaction and that receives a product from the distribution processes and therefore we would suggest considering an additional canonical process that represents ‘consumption’. Including the generic processes of SOP, PE, Production, and Distribution, we formally described generic news production and indicated how these processes implement the generic canonical processes of media production. Based on the assumption that the news format is the primary candidate for multi-channel publication and distribution, we made a distinction between production and mastering processes and we explained the role of the material warehouse process implemented by a MAM system. We also identified interactivity as an extension of the sales and distribution processes.

As such, we addressed the twofold problem of Section 2.1. We are the first to identify the canonical processes within end-to-end news production. By aligning the news production processes with these canonical processes, all ranks within an organisation can now fully understand the process cycles of their systems in the context of the more generalised, standard process cycles of existing media systems. The identified canonical processes help in clarifying the complex interleaving of workflow processes on the different levels of responsibilities within an organisation. Furthermore, one can envisage future scenarios where some of the processes within a system could be exchanged with those from other media production systems that adhered to aforementioned generic principles, e.g., production houses and local broadcasters. We set up the framework to establish clear interfaces for the information flow across media processes among distinct news production phases so that compatibility across systems from different providers can be achieved. By identifying recurring and canonical functionality, process implementations are simplified and input and output from different processes are coordinated for better integration with these external systems. However, the workflow information generated and gathered during the production, the so-called metadata, must become an integral part of the production process and must be modelled and employed as soon as possible by production systems. This successful mapping then allows implementers of new manufacturing methods and of foreign integrating systems to better coordinate processes in terms of inputs, outputs, and functionality. Further work is also envisioned by us within W3C's Media Annotations Working Group [179] (MAWG) for standardisation of media applications' portability. Interdisciplinary research could also be done on identifying the different actors and their respective 'profiled' canonical processes within the whole end-to-end news production workflow chain, as production houses and local broadcasters are only covering certain parts of this end-to-end workflow chain. Point by point my own research contributions can thus be summarised as follows:

- First to identify the canonical processes within end-to-end news production.

- Set up a theoretical framework to establish clear interfaces for the information flow across media processes in news production to enhance interoperability across different system providers.

The research that has led to this chapter is also described in the following publications:

1. E. Mannens, D. Van Rijsselbergen, M. Verwaest, R. De Sutter, L. Overmeire, and R. Van de Walle. Generic Architecture Guidelines for a Digital Media Factory. *SMPTE Motion Imaging Journal*, Accepted for publication, SMPTE

2. E. Mannens, M. Verwaest, and R. Van de Walle. Production and Multichannel Distribution of News. *Multimedia Systems – Special Issue on Canonical Processes of Media Production*, volume 14, number 6, pages 359–368, Springer-Verlag, December, 2008

3. D. Van Rijsselbergen, M. Verwaest, C. Poppe, E. Mannens, and R. Van de Walle. Semantic Mastering: Content Adaptation in the Creative Drama Production Workflow. *Multimedia Tools and Applications – Special Issue on Intelligent Interactive Multimedia Systems and Services*, Online First (DOI: 10.1007/s11042-011-0710-0), Springer-Verlag, January, 2011

4. D. Van Rijsselbergen, M. Verwaest, E. Mannens, and R. Van de Walle. On how Metadata Enables Enriched File-Based Production Workflow. *SMPTE Motion Imaging Journal*, pages 27–38, SMPTE, June, 2010

5. P. Debevere, D. Van Deursen, D. Van Rijsselbergen, E. Mannens, M. Matton, R. De Sutter, and R. Van de Walle. Enabling Semantic Search in a News Production Environment. In *the Proceedings of the 5th International Conference on Semantic and Digital Media Technologies*, pages 1–16, December 2010, Saarbrucken, Germany

6. D. Van Rijsselbergen, M. Verwaest, E. Mannens, and R. Van de Walle. On how Metadata Enables Enriched File-based Production Workflows. In *the Proceedings of the SMPTE Annual Tech Conference and Expo*, pages 1–19, October 2009, Hollywood, USA

7. B. Van De Keer, D. Van Rijsselbergen, M. Verwaest, E. Mannens, and R. Van de Walle. Extending a Data Model for a Drama Product Manufacturing System with Re-purposing Support. In *the Proceedings of the 13th IASTED International Conference on Internet and Multimedia Systems and Applications*, pages 105–110, August 2009, Honolulu, USA

8. D. Van Rijsselbergen, B. Van De Keer, M. Verwaest, E. Mannens, and R. Van de Walle. On the Implementation of Semantic Content Adaptation in the Drama Manufacturing Process. In *the Proceedings of the 10th IEEE International Conference on Multimedia & Expo*, pages 822–825, June 2009, New York, USA

Chapter 3

Personalising Enriched News Events

Me? The 13th Duke of Wybourne? Here? In a sixth form girl's dormitory? At three o'clock in the morning? With my reputation? What were they thinking of?

The 13th Duke of Wybourne in The Fast Show

3.1 Introduction

The nature of information has changed significantly in the last two decades. Now information is multimedia on the Internet, sensitive to its spatio-temporal roots, live, and dynamic. This also holds true for the news scene, where supply and demand has shifted from only broadcasting and hard-copy press to the Internet as well. As such, an important role is granted to both news providers and broadcasters to grasp the vast and complex supply of these news event streams. Broadcasters and news agencies are no longer only traditional gate keepers who determine what the public needs to see and read, but their primary function is shifting to being a guide that puts emphasis and structure on the supply and demand of these news events and their accompanying assets.

Furthermore, with the emergence of both citizen-based media and social media, broadcasters and news providers have to fundamentally re-think their production and distribution workflow processes as discussed in Chapter 2. In addition, traditional text-based products are losing market share for other media products, in particular video. At the same time, users have access to multiple news portals, which provide on-line access to different sources and services for commenting and debating on the news, and use social media to instantaneously (a.o., twitter[1]) spread news information. This results in large amounts of (possibly) unreliable and repeated information, leaving the user exploring on their own to try to build their own version of a news event from large amounts of potentially related information, or simply to find the truth in the middle of an ocean of rumours and hoaxes.

The ultimate goal is to create an environment that facilitates end-users in seeing meaningful connections among individual news events (stories, photos, graphics, and videos) through underlying knowledge of the descriptions of the items, their relationships, and related background knowledge. This can be solved by semantic metadata models improving metadata interoperability along the entire news production chain. The underlying research problem tackled in this chapter covers two ends of the news workflow spectrum: how to model, exchange, and represent semantic multimedia metadata along the news workflow and how an end-user can benefit from that via personalised news recommendations. For this open-knowledge approach of news workflows to work on an industrial scale in practice, all information components need to conform to open standards. This will ensure a broad uptake for establishing metadata workflows in the production and consumption chains throughout the news industry.

The contributions of this chapter are therefore twofold. On the one hand, in Section 3.2 we report on the modelling of the ontologies for the IPTC[2] family of languages using the IPTC NewsCodes SKOS [124] taxonomy conversion and we demonstrate in Section 3.3 how the news metadata can be automatically enriched and further integrated with the knowledge already formalised on the Web. On the other hand, in Section 3.4 we report on the modelling of a generic global profile and discuss these modelling decisions with respect to their consequences on the profiling and personalised recommendation of news events to an end-user. Furthermore Section 3.5 elaborates on a possible distribution scheme of these personalised news events, whereas related work, future directions and final conclusions are drawn in Section 3.6, Section 3.7 and Section 3.8.

[1] `http://twitter.com/`

[2] IPTC: International Press Telecommunications Council, see `http://www.iptc.com/`

3.2 Semantic-Web-based Infrastructure for News Metadata

Broadcasters receive news information from various sources. For example, VRT gathers its material from both its own news crews and from several international news agencies, such as Reuters[3] and EBU[4] Eurovision. The rough-cut and mastered essence created by the news crews is then stored into the MAM system. Reporters use AVID's iNews[5] application to enrich the essence by adding descriptive information, such as captions, anchor texts, and other metadata, and to organise the rundown of a classical television news broadcast. Then the essence needs to be packed into an MXF instance [157] before the MAM can process it. Afterwards, the essence is transcoded into a consumer format, such as H.264/AVC [98]. As the use of NewsML is key within our framework, it is being elaborated on over the next subsections.

3.2.1 NewsML

The IPTC has developed NITF[6] and NewsML, two XML-based languages for describing the structure and the content of news articles. These languages proved, however, to be inadequate to describe all kinds of multimedia news and were often judged too verbose. Recently, IPTC has released the NAR[7] framework which provides the framework for the second generation (G2) of IPTC standards. NAR is a generic model that defines four main objects (*newsItem*, *packageItem*, *conceptItem*, and *knowledgeItem*) and the processing model associated with these structures. Specific languages, such as NewsML-G2 [73] or EventsML-G2 [72], are then built on top of this architecture. For example, the generic *newsItem* is specialised into media objects (textual stories, images, or audio clips) in NewsML-G2. Finally, IPTC maintains a number of controlled vocabularies called the IPTC *NewsCodes*, that are used as values while annotating news items. Among others, the *Subject Codes* is a thesaurus of 1300 terms used for categorising the main topics (*subjects*) of all news items. As stated in [163], various use cases in need of a news ontology have been reported. The NEWS[8] project –as will be further elaborated on in the Related Work Section 3.6– aims to combine Semantic Web technologies and Web services for improving the news agencies' workflow. The project

[3] Reuters, see `http://www.reuters.com/`

[4] European Broadcasting Union, see `http://www.ebu.ch/`

[5] `http://www.avid.com/products/iNews/index.asp`

[6] News Industry Text Format, see `http://www.nitf.org/`

[7] News ARchitecture, see `http://www.iptc.org/NAR/`

[8] `http://www.news-project.com/`

has developed a lightweight RDFS news ontology (in English, Spanish and Italian) based on the IPTC Subject Codes for categorising the news items and on IPTC's NITF and NewsML for the metadata management [53, 54]. The Neptuno[9] research project has also modelled a lightweight RDFS news ontology representing a newspaper archive. The ontology is again a mix between news management metadata based on the NewsML standard and on the IPTC Subject Codes aligned with a news agency thesaurus for categorising the news items [28].

In contrast to these projects, our approach is to decouple the taxonomies used in the metadata values from the ontology that describes the management of the news items according to the journalist's point of view. This separation of concerns provides a more flexible infrastructure where the Subject Codes can be aligned to other controlled vocabularies. We expose these aligned taxonomies on the Semantic Web, providing dereferencable URIs for every term. Furthermore, we conform to the latest standard for the news metadata (NAR) and we design the ontology to be linked with other media ontologies. Already a few methods to automatically convert XML schemas into OWL ontologies exist [169]. To name just one, the ReDeFer compendium of RDF-aware utilities[10] is used in the journalism domain for converting the NewsML and NITF document formats, the IPTC Subject Codes thesaurus, and the MPEG-7 [87] multimedia format into OWL/RDF [56]. The resulting ontology, however, fails to capture the intended semantics of these standards that cannot be represented in XML schemas [164]. It recreates the complex nested structures used in the original schema (e.g., the definition of intermediate containers defining the XML Schema types and elements) that should generally not be modelled in the ontology. We propose, on the contrary, to re-engineer the ontology following some good practices detailed hereafter.

As described above, NAR is a generic model for describing news items as well as their management, packaging, and exchange. Interestingly, this model shares the principles underlying the Semantic Web:

- News items are distributed resources that need to be uniquely identified like the Semantic Web resources.
- News items are described with shared and controlled vocabularies.

NAR, however, is defined by means of an XML schema and has thus no formal representation of its intended semantics (e.g., a *NewsItem* can be a

[9] http://nets.ii.uam.es/neptuno/

[10] ReDeFer project, see http://rhizomik.net/redefer/

TextNewsItem, a *PhotoNewsItem*, or a *VideoNewsItem*). Extensions to other standards are cumbersome since it is hard to state the equivalence between two XML elements in a machine-understandable way. By modelling NAR in RDFS, we expect the following benefits:

- Better control of NewsML-G2 descriptions enabled by logical consistency checks.
- Enhanced search of news items enabled by logical inferences from the taxonomy and the knowledge formalised on the Web.
- Unified semantic interfaces for searching and browsing seamlessly news content and background knowledge.

The following subsections therefore describe the necessary steps for modelling such an ontology infrastructure. The various interconnected ontologies (NAR, NewsML-G2, and EventsML-G2) are available at `http://multimedialab.elis.ugent.be/ontologies/` and are extensions/evolutions of ontologies from *CWI*[11], and *Eurecom*[12]. At the time of writing a more elaborated version of the complete NewsML-G2 specification is available through *EBU's* technical Website, as our news use case was more availed with a simple, generic approach.

3.2.2 Designing the NAR Ontology

The first step aims to formally capture the intended semantics of NAR and the family of IPTC G2 standards. Even though these models exist in UML diagrams, their translation into Semantic Web models/languages is not trivial. We discuss below the rationale of our modelling decisions [118, 163].

Flattening the XML structure. XML Schema provides the means to have a very rich structure but is rather limited when expressing the meaning of this structure as the language is (only) concerned with providing typing and structuring information for isolated chunks of data. Likewise, the NAR model defines intermediate structures and containers whose only goal are to group a number of properties without particular semantics. These structures should not be represented in the ontology, thus making a much slicker ontology structure –no extra overhead of zero-information triples–, which in turn will correspond to much better performing SPARQL queries, as hierarchical, complicated structures hamper the SPARQL engines. As a side effect, blank

[11] Centrum voor Wiskunde en Informatica, see also `http://www.cwi.nl/en/`

[12] `http://www.eurecom.fr/index.en.htm`

nodes generation at the instance level will be minimised, so visualisation in any Semantic Web browser will not be complexified. Therefore, we propose to flatten the XML structure keeping only the properties that will be instantiated.

Provenance. Statements about news items need often to be reified. For example, an editor registered as *team:md* can classify a news item as *diplomacy* at *2005-11-11T08:00:00Z*. Using the RDF reification –using the known indirection of *diplomacy* to its language agnostic QCODES scheme, as explained hereafter– and the N3 syntax, this yields the following statements from Listing 3.1. Mind you, that future solutions tend to use named graphs as already stipulated in the W3C's RDF Next Steps Workshop[13] and W3C's Provenance Incubator Group[14].

Listing 3.1: NAR Reification Example.

```
{<> nar:subject cat:11002000}
  dc:creator team:md ;
  dc:modified "2005-11-11T08:00:00Z".
```

Modelling unique identifiers. News items metadata rely on numerous taxonomies that implement a coding scheme for identifying the terms in order to be language agnostic. For example:

Listing 3.2: Language Agnostic Coding Scheme.

```
<pubStatus code="stat:usable"/>
<locCreated code="city:Paris"/>
<creator code="team:DOM"/>
<subject code="cat:04000000"/>
<subject code="isin:NL0000361939"/>
<subject code="pers:021147"/>
```

IPTC has therefore defined the notion of QCODES (by analogy to the XML QNAMES) with the following properties:

- Each coding scheme is associated with an URI, which must resolve to a resource (or resources) containing information about the scheme.
- The prefix represents the URI of the scheme within which the local part is allocated.

[13] http://www.w3.org/2009/12/rdf-ws/

[14] http://www.w3.org/2005/Incubator/prov/

- There are almost no constraints on the values of the local part. For example, the local part (the code) is allowed to start with a digit.
- The two taken together must form a legal URI.
- This URI should provide access to a definition of the concept represented by that code within that scheme, i.e., it is dereferencable.

The tuple *prefix:localname* is however not identical to a CURIE [14] since the two parts (scheme and code) each have a meaning. For solving this issue, [163] proposes the ‘slash’ rule, i.e., the concatenation of the scheme URI, a slash, and the code, for the construction of a valid and dereferencable code URI.

3.2.3 Linking with Media Ontologies

Within the media industry, other multimedia standards, such as EXchangeable Image File Format [5] (EXIF), Dublin Core [71] (DC), eXtensible Metadata Platform [3] (XMP), Metadata for Digital Images [58] (DIG35), or MPEG-7 [87] also exist. These standards have generally been converted into OWL ontologies and can thus be integrated within our ontology infrastructure [181]. Therefore, this step consists in adding RDFS statements stating the relationship between resources defined in different but strongly overlapping ontologies. For example, the NAR ontology (see Figure 3.1 and Appendix B.1) contains the following axioms:

Listing 3.3: NAR Axioms.

```
nar:subject rdfs:subProperty dc:subject.
nar:Person rdfs:subClass foaf:Person.
```

Semantic Web search engines, such as Sindice[15], Watson[16], or Falcon[17] are useful tools for discovering concepts and properties defined in other ontologies that share the same semantics as the ones defined in our news infrastructure and could therefore be linked to them.

3.2.4 Converting IPTC NewsCodes into a SKOS Taxonomy

The IPTC *NewsCodes* define 36 thesauri used as metadata values in the NAR architecture. Although the terms are sometimes organised in a taxon-

[15] http://sindice.com/

[16] http://watson.kmi.open.ac.uk/WatsonWUI/

[17] http://www.falcons.com.cn/

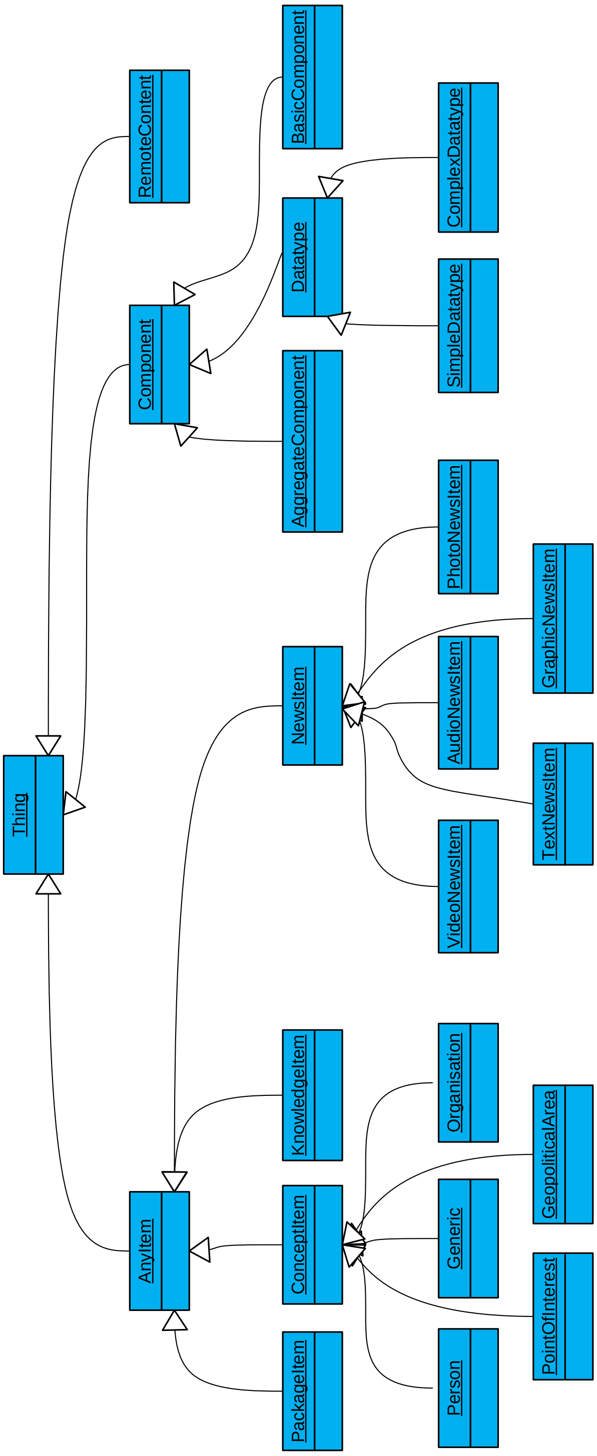


Figure 3.1: NAR-taxonomy.

omy, the subsumption relationship is not explicit but instead encoded into the coding scheme identifying the terms. For example, 'cancer' (cat:07001004) is narrower than 'disease' (cat:07001000), which is narrower than 'health' (cat:07000000) because they share a number of digits. *CWI* has converted these thesauri into SKOS, an application of RDF, making the subsumption relationships explicit (*skos:narrower*, *skos:broader*). This RDF compatibility allows us to define some concepts in the NAR ontology in terms of *owl:Restriction*: the value of a property can be a *skos:Concept* or must come from a given *skos:ConceptScheme*. These taxonomies can be found at `http://newsml.cwi.nl/NewsCodes` following the best practice recipes for publishing RDF vocabularies [13] and Cool URIs for the Semantic Web notes [8]. Each term is thus identified by a dereferencable URI. Consequently, sending an HTTP-request with the requested type *Accept: application/xml* will deliver the original XML version from IPTC of the taxonomies, while the requested type *Accept: application/rdf+xml* will return the SKOS/RDF machine-processable version of the taxonomy.

3.3 Enriching News Metadata with the Linked Data Cloud

3.3.1 Linked Open Data

Linked Open Data [16] (LOD) is all about using the Web to create typed links between data from different sources. These may be as diverse as databases maintained by two organisations in different geographical locations, or simply heterogeneous systems within one organisation that, historically, have not easily interoperated at the data level. Technically, LOD refers to data published on the Web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets, and can in turn be linked to from external data sets. While the primary units of the hypertext Web are HTML documents connected by untyped hyperlinks, LOD relies on documents containing data in RDF format. However, rather than simply connecting these documents, LOD uses RDF to make typed statements that link arbitrary things in the world. The result, which we will refer to as the Web of Data, may more accurately be described as a Web of Things in the world, described by Data on the Web.

Based on Tim Berners-Lee's LOD principles [11], we identify a lowest common denominator for Semantic Web interoperability:

- All items of interest, such as information resources, real-world objects, and vocabulary terms, are identified by URI references.
- Use HTTP URIs, so that people can look up all those items of interest.
- When someone looks up an URI, provide useful information, using the standards (RDF, SPARQL) –so descriptions can be provided using the RDF/XML syntax.
- Every RDF triple is conceived as an URI hyperlink that links to related information from the same or a different source that can be followed by Semantic Web agents.

These have become known as the ‘Linked Open Data principles’, and provide a basic recipe for publishing and connecting data using the infrastructure of the Web while adhering to its architecture and standards. LOD relies on two technologies that are fundamental to the Web: URIs [79] and HTTP [78]. While URLs [74] have become familiar as addresses for documents and other entities that can be located on the Web, URIs [79] provide a more generic means to identify any entity that exists in the world. Where entities are identified by URIs that use the ‘http://’ scheme, these entities can be looked up simply by dereferencing the URI over the HTTP protocol. In this way, the HTTP protocol provides a simple yet universal mechanism for retrieving resources that can be serialised as a stream of bytes (such as a photograph of an NBA[18] player), or for retrieving descriptions of entities that cannot be sent across the network in this way themselves (such as the Celtic’s Paul Pierce –aka ‘The Truth’– himself).

Consequently, it is possible to think of RDF triples that link items in different data sets as analogous to the hypertext links that tie together the Web of documents. RDF links[19] take the form of RDF triples, where the subject of the triple is an URI reference in the namespace of one data set, while the object of the triple is a URI reference in the other. RDFS and OWL provide a basis for creating vocabularies that can be used to describe entities in the world and how they are related. Such vocabularies are collections of classes and properties which are themselves expressed in RDF, using terms from RDFS and/or OWL, which provide varying degrees of expressiveness in modelling domains of interest. Anyone is free to publish vocabularies to the Web of Data [13], which in turn can be connected by RDF triples that link classes and properties

[18] National Basketball Association, see `http://www.nba.com/`

[19] Tutorial: How to publish Linked Data on the Web, see `http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/`

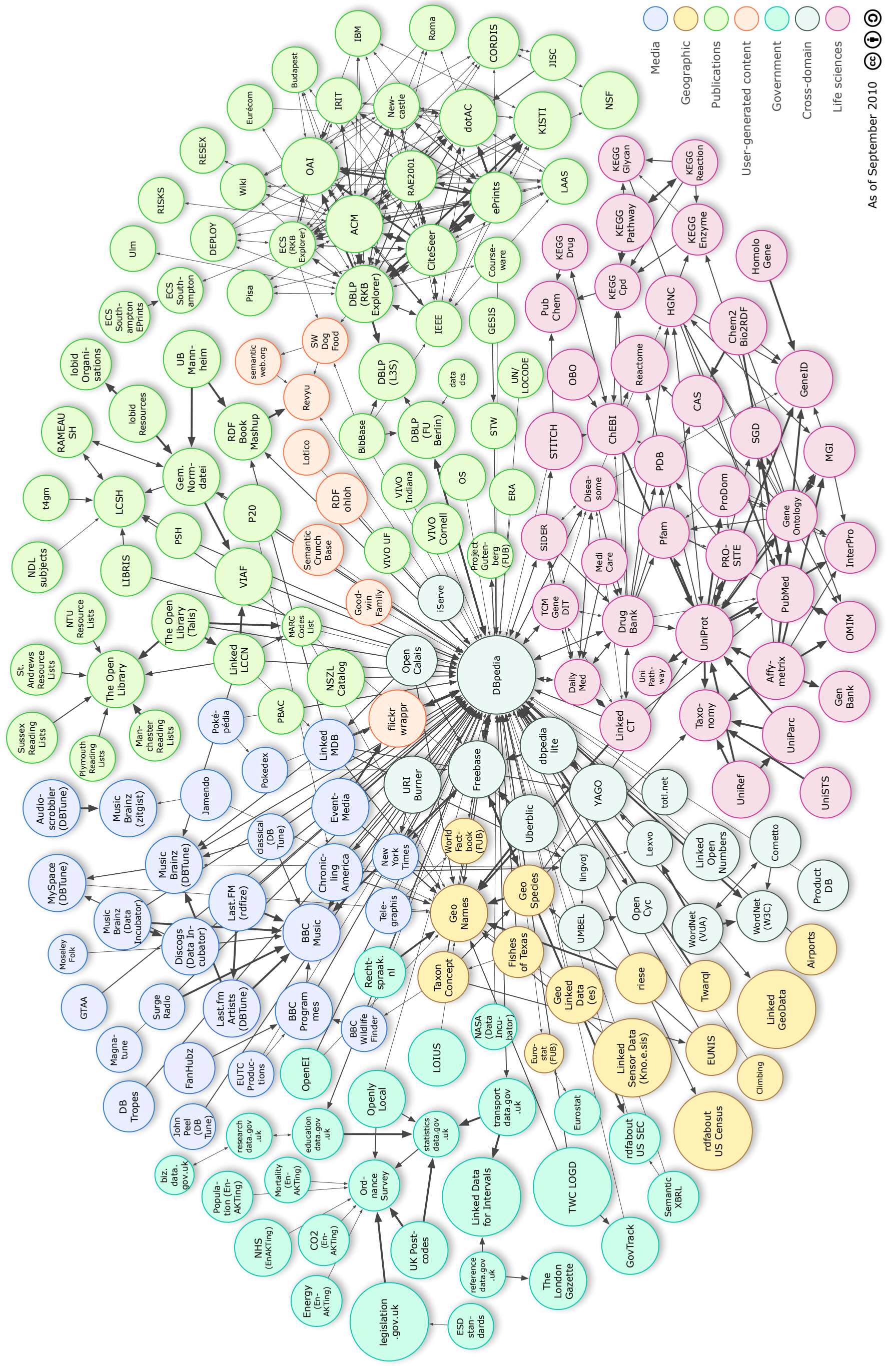


Figure 3.2: Linking Open Data Dataset Cloud, as of September 2010.

in one vocabulary to those in another, thereby defining mappings between related vocabularies. By employing HTTP URIs to identify resources, the HTTP protocol as retrieval mechanism, and the RDF data model to represent resource descriptions, LOD directly builds on the general architecture of the Web [99]. The Web of Data can therefore be seen as an additional layer that is tightly interwoven with the classic document Web and has many of the same properties:

- The Web of Data is generic and can contain any type of data.
- Anyone can publish data to the Web of Data.
- Data publishers are not constrained in choice of vocabularies to represent data.
- Entities are connected by RDF links, creating a global data graph that spans data sources and enables the discovery of new data sources.

From an application development perspective, the Web of Data has the following characteristics:

- Data are strictly separated from formatting and presentation aspects.
- Data are self-describing: if an application consuming LOD encounters data described with an unfamiliar vocabulary, the application can dereference the URIs that identify vocabulary terms in order to find their definition.
- The use of HTTP as a standardised data access mechanism and RDF as a standardised data model simplifies data access compared to Web APIs, which rely on heterogeneous data models and access interfaces.
- The Web of Data is open, meaning that applications do not have to be implemented to a fixed set of data sources, but can discover new data sources at run-time by following RDF links.

3.3.2 News Metadata Enrichment

Once the NAR and NewsML-G2 ontologies have been modelled and linked to other media ontologies, the conversion of the metadata of individual news items into RDF according to this ontology can be automated. However, we propose a further step aiming at semantically enriching the news metadata following the LOD principles [16] as described above. In our case, we apply linguistic processing on the plain text contained into some elements of the metadata, such as title, caption, description, and the body of the news stories. The linguistic processing consists of extracting named entities, such as persons, organisations, companies, brands, locations, and events.

We use the OpenCalais[20] infrastructure for extracting these named entities. For example, the processing of the headline "Paul Pierce to be named MVP in the NBA finals." will result in three named entities: 'Paul Pierce', 'MVP', and 'NBA' together with their type (i.e., person, event, organisation, etc.). Once the named entities have been extracted, we map them to formalised knowledge on the Web available in GeoNames[21] for the locations or in DBPedia[22]/FreeBase[23] for the persons, organisations, and events. The string 'Paul Pierce' is therefore mapped to its URI in DBPedia[24] that provides i) a unique identifier for the resource and ii) formalised knowledge about this person, such as his biography, career, and genealogy in multiple languages. Therefore, the use of, e.g., the OpenCalais Web service allows us to populate the knowledge base by providing a list of possible instances for all named entities discovered.

The main challenge in this semantic enrichment step is to deal with the ambiguity. For example, the GeoNames Web service tends to return primarily a US city when a single string is passed as an argument. Fortunately, news items always contain information about the city and the country yielding accurate recognition of the location mentioned in the story. When the accuracy is the primary concern, one could envision a semi-automatic approach where suggestions will be proposed to the journalist during the annotation process, e.g., using extra services like Zemanta[25], Apture[26], Wibiya[27], or multi-lingual tools using Word-Net[28] & Cornetto[29]. Furthermore, we integrated the extracted metadata from shot segmentation tools, scene detection tools, and face recognition tools of the PISA-project (see also Appendix E.12) to be able to also further enrich these extracted named entities [47].

[20] http://www.opencalais.com/

[21] http://www.geonames.org/

[22] http://dbpedia.org/

[23] http://www.freebase.com/

[24] http://dbpedia.org/page/Paul_Pierce/

[25] http://www.zemanta.com/

[26] http://www.apture.com/

[27] http://www.wibiya.com/

[28] http://semanticweb.cs.vu.nl/lod/wn30/

[29] http://www2.let.vu.nl/oz/cltl/cornetto/

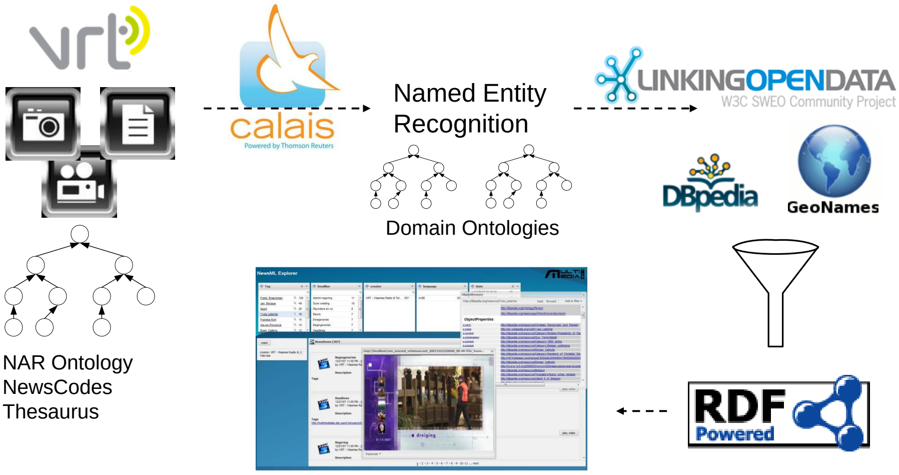


Figure 3.3: Enrichment Stages.

3.4 Profiling & Recommending the End-user

3.4.1 User Profiling

In recent years technologies have appeared that also empower the user to have more control over his experience, e.g., RSS[30] was designed to empower the user to view the content he or she wants, when it's wanted, not at the behest of the information provider [130]. Beyond this control over the view, the user needs adequate filtering mechanisms in order to work through this real-time stream of (news) data. Recommendation systems offer content tailored to the user's needs. A common way of serving selected content is to relate it to the user's profile information. *Amazon*[31] is considered a leader in on-line shopping and particularly recommendations. They have built a smart set of recommendations that tap into a user's browsing history, past purchases, and purchases of other shoppers [82]. *Pandora*[32] is a music recommendation system that leverages similarities between pieces and music, and is thus a recommendation system based on genetics. While both *Amazon* and *Pandora* offer an excellent service, they do not have access to the massive amount of information about a user that is stored in his preferred social network. Together with the evolution of recommendation engines, social networks are growing according to a recent

[30] RSS: Really Simple Syndication, also see http://www.rss-specifications.com/

[31] http://www.amazon.com/

[32] http://www.pandora.com/

Forrester Research study [36]. The giant in the space remains *Facebook*[33], which gets 87.7 million unique viewers per month according to *ComScore*[34]. Although *Facebook* is the most popular social network at the moment, users don't limit themselves to one dedicated network. There are a plethora of popular social networks with more than 1 million monthly visitors: *Myspace*[35], *Twitter*[36], *LinkedIn*[37], and *Netlog*[38] are among the more popular ones. There are in fact already also a number of very popular social news Web sites, a.o., *Reddit*[39], *Digg*[40], and *Propeller*[41]. Generally, a user's profile consists of three types of information:

- *Static information*, e.g., the user's birth date, address, favourite books, etc.
- *Dynamic information*: this is information coming from the user's activity stream, e.g., what is the user listening to, what is the user's current location, feedback of the user on offered content using the *OpenLike* paradigm[42], etc.
- *The social graph*: this contains all the user's connections to other users, e.g., a friend list.

Current recommendations are mostly offered within the closed context of a single community: e.g., *Facebook* recommends events based on RSVP event invitations[43] from other users connected to the *Facebook* user's social graph. *Facebook* does not automatically recommend events based on the static and dynamic profile information of a user, nor does it take into account possibly interesting data coming from other social networks. That is why we created a *global profile*, which is aggregated from other profiles the user has in different communities. This global profile is consumed together with the news events information by the recommendation system to yield recommendations of news items.

[33] `http://www.facebook.com/`

[34] `http://www.comscore.com/`

[35] `http://www.myspace.com/`

[36] `http://www.twitter.com/`

[37] `http://www.linkedin.com/`

[38] `http://www.netlog.com/`

[39] `http://www.reddit.com/`

[40] `http://digg.com/`

[41] `http://www.propeller.com/`

[42] `http://openlike.org/`

[43] 'Respondez, S'il Vous Plait', see `http://wiki.developers.facebook.com/index.php/Events.rsvp`

This global profile allows storing the necessary elements to yield a better profile, more useful to a recommendation system, because it combines information from different user communities. This information needs to be aggregated from the profiles the user has in several user communities. For this, we used an *OpenID*[44] identity provider service. It provides an identity, e.g., `http://myname.newsfeed.be/`, together with a profile, i.e., the aggregated global profile. By letting users link this identity to the identities they already possess in different user communities, this identity service identifies uniquely the user with all his other identities. This *OpenID* identity service is also used as an authentication mechanism, for proving who you are and what your other identities are.

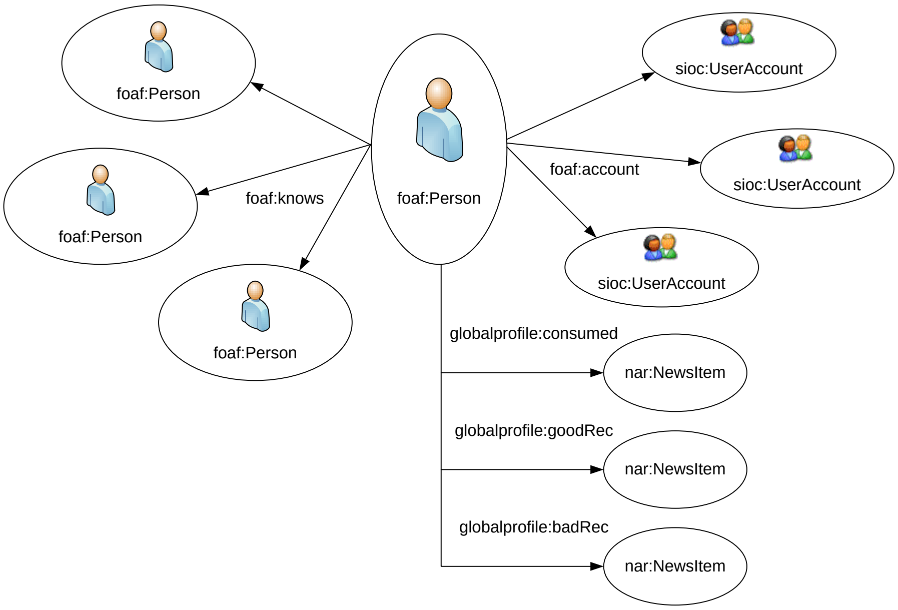


Figure 3.4: The Global Profile.

This global profile (as can be seen in Figure 3.4 and in Appendix B.2) is modelled as a *foaf:Person*. This person has some static information, e.g., first name, last name, birth date, etc. For this, Friend-Of-A-Friend[45] (FOAF) properties are used, resp., *foaf:firstName*, *foaf:lastName*, *foaf:birthday*, etc. The different on-line accounts the person has, are modelled using the

[44] `http://openid.net/`

[45] `http://www.foaf-project.org/`

sioc:UserAccount (Semantically-Interlinked On-line Communities[46], or SIOC) and linked to the *foaf:Person* using the *foaf:account*. This allows us to link the *OpenID* account to other on-line accounts, e.g., *Facebook* or *Digg* the user has. The social graph of the person is formalised using the *foaf:knows* property. The dynamic information consists of the consumptions of news events and RSVP feedback on events. For this, the *foaf:Person* is extended with *globalprofile:consumed* for denoting the person has consumed a certain news event, and with *globalprofile:goodRec* and *globalprofile:badRec* for describing the feedback on news events recommendations.

For populating the global profile, the identity service has connectors to *OpenID* identity providers, e.g., *Digg* or *Facebook*. This way, the user can keep control over the global profile by selecting the identity providers he/she trusts. The global profile then gets aggregated from the identity services he has trusted. *OpenID* is a good authentication mechanism, but not a good authorisation mechanism. Indeed, we need a mechanism for the authenticated user to explicitly permit the data/profile provider to use his *OpenID* credentials to connect to a profile provider and retrieve a particular piece of the user's private information. A combination of *OpenID* and *OAuth*[47] lets the user control his permissions for Web services in a fine-grained manner. Our connectors use a combination of the *OpenID* protocol and the *OAuth* protocol for retrieving the profile information.

By providing these connectors, the user can also use this authentication information from the identity service to log on to the platforms that support *OpenID* identification. At the same time any application that consumes profile information, can use this identity service as long as they have an *OpenID* connector as profile provider and authentication mechanism and an *OAuth* mechanism for authorisation.

3.4.2 News Recommendation System

The over-abundance of (news) content and the related difficulty to discover interesting (news) content items have already been addressed in several contexts. On-line shops, like *Amazon*, apply *Collaborative Filtering* (CF) to personalise the on-line store according to the needs of each customer [115]. Purchasing and rating behaviour are valuable information channels for on-line retailers to investigate consumer's interests and generate personalised recom-

[46] http://sioc-project.org/

[47] http://oauth.net/

mendations [101]. Because of the success of recommendation techniques for a big variety of items (books, DVDs, and TV programmes), it is logical to use the same recommendation techniques for suggesting news event items. However, some problems arise due to the inherent nature of news events [38].

CF techniques are the most commonly used recommendation algorithms because they generally provide better results than *content-based* (CB) techniques [68]. Most user-based CF algorithms start by finding a set of neighbours whose consumed or rated items overlap the user's consumed or rated items. Users can be represented as an *N*-dimensional vector of items, where *N* is the number of distinct catalogue items. Consumed or rated items are recorded in the corresponding components of this vector. For the big majority of users who consumed or rated only a very small fraction of the catalogue items, this vector is extremely sparse. The similarity *–Sim()–* between two users, j and k, symbolised by their respective consumption vectors, $\vec{U_j}$ and $\vec{U_k}$, can be measured in various ways. The most common method is to measure the cosine of the angle between the two vectors [147]:

$$Sim(\vec{U_j}, \vec{U_k}) = \frac{\vec{U_j} \cdot \vec{U_k}}{||\vec{U_j}||\ ||\vec{U_k}||}, with ||\vec{U_j}|| = \sum_{i=1}^{N} \vec{U_j}[i]. \tag{3.1}$$

Next, the algorithm aggregates the consumed items from these similar neighbours –the threshold being the 20 friends that have the most resembling consumption vectors, i.e., in which the cosine is close to 1–, eliminates items the user has already consumed or rated, and recommends the remaining items to the user [115]. An alternative to this user-based CF technique is item-based CF, a technique that matches each of the user's consumed or rated items to similar items and then combines those similar items into a recommendation list. For measuring the similarity of items, the same metrics can be employed as with the user-based CF. Because of scalability reasons, this technique is often used to calculate recommendations for big on-line shops, like *Amazon*, where the number of users is orders of magnitude bigger than the number of items. Despite the popularity of CF, its applicability is limited due to the sparsity problem, which refers to the situation that the consumption data in the profile vectors are insufficient to calculate reliable recommendations. Especially news systems suffer from sparse datasets, since most users only consume/read a small fraction of all the available news events. A direct consequence of a sparse data matrix is that the number of neighbours for a user/item might be very limited in a user-based/item-based CF system. Indeed, the majority of the similarity metrics that are used in CF systems rely on the vector overlap to determine the similarity of two users/items. Sparse profile vectors lead to a limited overlap, which obstructs the creation of accurate and extensive neighbourhoods of like-minded people or similar items. Furthermore, because of this sparsity, the majority of these neighbours will also have a small number of consumptions in their profile vectors. Because the prospective personal recommendations are limited to this set of consumptions of neighbours, the variety, quality, and quantity of the final recommendation list might be inadequate.

In an attempt to provide high-quality recommendations even when data profiles are sparse, some solutions are proposed in the literature [133]. Most of these techniques use trust inferences, transitive associations between users that are based on an underlying social network, to deal with these sparsity and the cold-start problems [182]. Nevertheless, the underlying social networks are in many cases insufficiently developed or even non-existing for (new) Web-based applications that desire to offer personalised content recommendations. Default voting is an extension to the traditional CF algorithms which tries to solve this sparsity problem without exploiting a social network. A default value is assumed as 'vote' for items without an explicit rating or purchase [21]. Although this technique enlarges the profile overlap, it cannot identify more significant neighbours than the traditional CF approach. Grouping people or items/events into clusters based on their similarity can be another solution, but finding the optimal clusters is not trivial [37].

We, on the other hand, used a 5 phased approach [44, 45]. In a first phase, we calculate the probability that an item will be consumed by the user based on the user's profile as 'a priori' knowledge. This probability is inverse proportional to the index of the item in a personal top-*N* recommendation list, and is estimated by the confidence value which is calculated by a traditional CF system. After all, this top-*N* recommendation list is a prediction of the items which the user will like/consume in the near future. Based on this calculated general and profile-based probability, the user or item profile is completed until the minimum profile threshold is reached. However, these predicted consumptions will be marked as uncertain in contrast to the initial assured consumptions. For a news event service, e.g., the real news item consumption corresponds to a value of 1, which refers to a 100% guaranteed consumption, while the potential future consumptions are represented by a decimal value between 0 and 1 according to the probability value in the profile vector. This second phase can consist of several successive iterations to complete the profiles.

Based on these extended profile vectors, the similarities are recalculated with the chosen similarity metric, e.g., the cosine similarity metric (equation 3.1), in a third phase. Because of the added future consumptions, the profile overlap and accordingly the number of neighbours will be increased compared to phase 1. To produce personal suggestions, a recommendation vector will be generated based on these extended profile vectors, in a fourth phase. The recommendation vector, R_j, for user *j* can be calculated as:

$$\vec{R_j} = \frac{\sum_{k=1,k\neq j}^{M} \vec{U_k} \cdot Sim(\vec{U_j}, \vec{U_k})}{\sum_{k=1,k\neq j}^{M} Sim(\vec{U_j}, \vec{U_k})}, \tag{3.2}$$

where $\vec{U_j}$ and $\vec{U_k}$ represent the consumption vectors of users j and k, which might contain real values, and M is the total number of users within user's j neighbourhood. Subsequently, the top-*N* recommendations are obtained by taking the indices of the highest components of the recommendation vector, R_j, and eliminating the items which are already consumed by user j in the past.

Finally, in order to take into account personal news event selection criteria that are not related to the subject of the news event, we added contextual post-filters to the recommendation system in the fifth phase. These filters operate after the main recommendation algorithm and remove or penalise the candidate recommendations which do not satisfy the personal selection criteria. These personal selection criteria, which can be specified by the end-user, are for example the location, the language, the preferred categories, and the date of the news event.

3.5 Distribution of News Events

As we have enriched our semantic news items (see Section 3.3.2), we can now start publishing them as LOD [16]. These news items are distributed to the end-user via a portal site (ongoing implementation work). This portal relies on the *OpenID* identity service for authentication (see Section 3.4.1), on the recommendation engine (see Section 3.4.2) for getting the rightly targeted news items, and on the LOD server for the effective, enriched information of these news items. There, people will find the latest news items, search for particular new items, and view their personal recommended news items based on their global profile by exploring it using our faceted browser. Because news is very volatile and we want the user to be constantly updated on new/developing news items, we offer them a *personal* RSS feed –containing a unique URI for each individual registered *OpenID*. This personal RSS feed contains updates on the top 20 recommended news items for that user. The recommendation engine only takes news items of no more than five days old into account and for performance reasons all newly recommended (developing) news items are aggregated and pushed to the end-users only twice a day. These feed items, a.o. things, contain a link to the LOD published news items, a description of these news items, their date, and their location. By providing such a *dynamic* personal RSS feed, which is updated twice a day, the users stay on top of the latest news items they prefer most.

3.6 Related Work

A number of EU projects are related to this research. *NEWS*[48] developed an automated multi-lingual textual news classification and annotation engine that is able to categorise information (also using the IPTC subject taxonomy) and extract named entities in English, Italian and Spanish. However, the project never achieved to build a feature extraction engine while we used such tools in the PISA-project for inferring news events from content annotations. The NEWS project also provided a prototype ontology model applicable to the news domain [54]. We built on the results of this work for creating a reference OWL description of the news domain, based on the NewsML-G2 IPTC standard. We also tackled multiple types of media, in addition to only text within the NEWS project, as it is our belief that video will gradually become the dominant medium for news content anyway.

Likewise, *Citizen Media*[49] enabled lay users to consume, author and publish their content as part of a networked audio-visual system. The project focused on automatic analysis of visual information and on scalability issues since the system has to be able to handle a massive amount of user-generated content in different formats in real-time, and annotate and store this content in huge databases in order to better reuse all these pieces of user-generated content. We, on the other hand, emphasised the role of semantic metadata for solving interoperability problems and empowering end-user interfaces.

The *MESH*[50] project aimed to extract, compare and combine content from multiple multimedia news sources, to automatically create advanced personalised multimedia summaries, syndicate summaries and content based on the extracted semantics, and provide end-users with a multimedia mesh

[48] `http://www.news-project.com/`

[49] `http://www.ist-citizenmedia.org/`

[50] `http://www.mesh-ip.eu/`

news navigation system. While the project has made progress across this broad set of goals, it focused mainly on news distribution on mobiles. We concentrate, however, on the use of the underlying technological semantic infrastructure to reduce the amount of information exposed to the user in a simplified interface by using our recommendation engine.

SEMEDIA's[51] overall objective was to develop a collection of audio-visual search tools that are user driven, preserve metadata along the (workflow) chain, and are generic enough to be applicable to different fields (e.g., broadcasting production, cinema post-production or social Web). We also stressed on the preservation of the metadata along the workflow and even solve the current interoperability problems using a knowledge infrastructure based on current standards and practices in the media industry. Given the specificity of the news domain and the existence of the IPTC subject codes, we can also concentrate on the macro events detection problem together with the implementation of clustering and ranking algorithms for news items in the future.

Furthermore, the *PENG*[52] (Personalised News Content Programming) project aimed at providing news professionals with an interactive and personalised tool for multimedia news gathering and delivery. This was achieved by developing a flexible prototype for a personalised filtering, retrieval and composition of multimedia news. The personalisation was obtained by 'tuning' the contribution of the distinct content types and sources of information by associating a trust score to each information source. However, the hierarchical organisation and categorisation of the news remains fuzzy while we are able to automatically find relationships between news items coming from different sources, in different languages and on different media by using our knowledge infrastructure and multimedia analysis tool-kits. Together with our recommendation engine this results in an intelligent and intuitive clustering and ranking of personalised news items.

On the other hand, *PAPYRUS*[53] (Cultural and Historical Digital Libraries dynamically mined from News Archives) aims at creating a cross-discipline digital library engine that allows for drawing content from one domain and making it available and understandable to the users of another. PAPYRUS starts from the historical perspective (within existing digital libraries) and tries to relate that back with legacy content from News Providers. In that sense, the

[51] http://www.semedia.org/

[52] http://www.peng-project.org/

[53] http://www.ict-papyrus.eu/

definition of an event pre-exists and even if it can be controversial, some facts are unambiguous (why it happened; what were the consequences; and how it may have been avoided). We take the point of view of news providers for formally defining the notion of an event and thus our recommendation system has to deal with the dynamics of the news where events are not necessarily expected.

3.7 Near Future Directions

In parallel, the W3C MAWG aims to provide an ontology and an API designed to also facilitate cross-community data integration of information related to media objects in the Web, such as video, audio, and images. The goal is to create a simple ontology containing a concise set of terms extracted from the prominent multimedia metadata standards. Additionally, mappings will be defined between a representative set of metadata standards and this ontology. As such, annotations of (news related) media objects can be accessed in a uniform way, disregarding the original format of the metadata. The challenge will be how to reconcile multimedia ontologies with news ontologies and domain specific vocabularies for solving these interoperability issues along the complete news workflow, as discussed in Chapter 2.

Furthermore, the origin of the news source is very important in the context of on-line news items. In this aspect, knowledge on the creator or issuer of the news gives more actual contextual information to the specific news item. Additionally, social links of the creator to other people or organisations are highly relevant when determining the context of a news item. *OpenSocial*[54] is a set of common APIs for building social applications across many Websites. However, to integrate this information according to the aforementioned LOD initiative, a formal representation is needed. A first initiative has been made to see how the Social Web can be integrated with the Semantic Web by the W3C's Social Web Incubator Group[55].

Finally, journalists often stress the absolute need of representing the provenance of all types of information in order to trigger confidence regarding the truthfulness of a news event report. The Open Provenance Model[56] is a model for provenance which meets the following requirements:

[54] http://www.opensocial.org/

[55] http://www.w3.org/2005/Incubator/socialweb/

[56] http://openprovenance.org/

- To allow provenance information to be exchanged between systems, by means of a compatibility layer based on a shared provenance model.
- To allow developers to build and share tools that operate on such a provenance model.
- To define the model in a precise, technology-agnostic manner.
- To support a digital representation of provenance for any 'thing', whether it is produced by computer systems or otherwise.
- To define a core set of rules that identify the valid inferences that can be made on provenance graphs.

Recently, the creators of the model have started the W3C Provenance Incubator Group[57] in which we participate and that will introduce the Open Provenance Model in the Semantic Web. As such, this issue within the complete news production workflow will get solved, as it will also be taken into account in the next Chapter 4, where provenance will prove to be an important topic too when news is to be archived for future-proof disclosure.

3.8 Conclusions and Original Contributions

We have presented a semantic version of the NewsML-G2 standard as a unifying (meta)datamodel dealing with dynamic distributed news event information. Using that ontology as a data communication interface within an end-to-end news distribution architecture, several services (aggregation, categorisation, enrichment, profiling, recommendation, and distribution) were hooked in the workflow engine giving broadcasters a tool to automatically recommend (developing) news stories 1-to-1 to the targeted customer for the first time.

At the same time, we provided the (inter)national (news) community with mechanisms to describe and exchange news events and profile information in a standardised way, thus contributing to the narrowing of the aforementioned '*technical semantic gap*'. We demonstrated the concepts of generic data portability of user profiles, and how to generate recommendations based on such a global profile –within which we integrated information fields from all the different social networks the user wanted to share. Our ideas were implemented

[57] http://www.w3.org/2005/Incubator/prov/

with open standards like *OpenID*, *OAuth*, and *OpenLike*, thus keeping the architecture open for other news event providers and profile providers. Point by point my own research contributions can thus be summarised as follows:

- Presented a semantic version of the NewsML-G2 standard as a unifying (meta)datamodel dealing with dynamic distributed news event information.
- Provided the (inter)national (news) community with mechanisms to describe, enrich and exchange news events and profile information in a standardised way.
- First to demonstrate the concepts of generic data portability of user profiles, and how to generate recommendations based on such a global profile.

The research that has led to this chapter is also described in the following publications:

1. E. Mannens, S. Coppens, T. De Pessemier, H. Dacquin, D. Van Deursen, R. De Sutter, and R. Van de Walle. Automatic News Recommendations via Profiling. Submitted to *Multimedia Tools and Applications – Special Issue on Automated Information Extraction in Media Production*, Springer-Verlag
2. E. Mannens, S. Coppens, T. De Pessemier, H. Dacquin, D. Van Deursen, and R. Van de Walle. Automatic News Recommendations via Profiling. In *the Proceedings of the ACM International Conference on Multimedia - the 3rd International Workshop on Automated Information Extraction in Media Production*, pages 45–50, October 2010, Florence, Italy
3. E. Mannens, S. Coppens, T. De Pessemier, K. Geebelen, H. Dacquin, D. Van Deursen, and R. Van de Walle. Unifying and Targeting Cultural Activities via Events Modelling and Profiling. In *the Proceedings of the ACM International Conference on Multimedia - the 1st International Workshop on Events in Multimedia*, pages 33–40, October 2009, Beijing, China
4. E. Mannens, R. Troncy, K. Braeckman, D. Van Deursen, W. Van Lancker, R. De Sutter, and R. Van de Walle. Automatic Information Enrichment in News Production. In *the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 61–64, May 2009, London, United Kingdom

5. S. Coppens, E. Mannens, T. De Pessemier, K. Geebelen, H. Dacquin, D. Van Deursen, and R. Van de Walle. Unifying and Targeting Cultural Activities via Events Modelling and Profiling. *Multimedia Tools and Applications – Special Issue on Events in Multimedia*, Online First (DOI: 10.1007/s11042-011-0757-6), Springer-Verlag, February, 2011

6. T. De Pessemier, S. Coppens, K. Geebelen, C. Vleugels, S. Bannier, E. Mannens, K. Vanhecke, and L. Martens. Collaborative Recommendations with Content-based Filters for Cultural Activities via a Scalable Event Distribution Platform. *Multimedia Tools and Applications*, Online First (DOI: 10.1007/s11042-011-0715-8), Springer-Verlag, January, 2011

7. D. Van Deursen, W. Van Lancker, W. De Neve, T. Paridaens, E. Mannens, and R. Van de Walle. NinSuna: a Fully Integrated Platform for Format-independent Multimedia Content Adaptation and Delivery based on Semantic Web Technologies. *Multimedia Tools And applications – Special Issue on Data Semantics for Multimedia Systems*, volume 46, number 2–3, pages 371–398, Springer-Verlag, February, 2010

8. D. Van Deursen, W. Van Lancker, S. De Bruyne, W. De Neve, E. Mannens, and R. Van de Walle. Format-independent and Metadata-driven Media Resource Adaptation using Semantic Web Technologies. *Multimedia Systems Journal*, volume 16, number 2, pages 85–104, Springer-Verlag, February, 2010

9. S. Coppens, E. Mannens, and R. Van de Walle. Disseminating Heritage Records as Linked Open Data. *International Journal of Virtual Reality*, volume 8, number 3, pages 39–44, IPI Press, September, 2009

10. C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. Personal Content Management System: a Semantic Approach. *Journal of Visual Communication and Image Representation – Special Issue on Emerging Techniques for Multi-media Content Sharing, Search and Understanding*, volume 20, number 2, pages 131–144, Elsevier, February, 2009

11. D. Van Rijsselbergen, B. Van De Keer, M. Verwaest, E. Mannens, and R. Van de Walle. Enabling Universal Media Experiences through Semantic Adaptation in the Creative Drama Production Workflow. In *the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 296–299, May 2009, London, United Kingdom

12. D. Van Deursen, W. Van Lancker, T. Paridaens, W. De Neve, E. Mannens, and R. Van de Walle. NinSuna: a Format-independent Multimedia Content Adaptation Platform based on Semantic Web Technologies. In *Proceedings of the 10th International Symposium on Multimedia*, pages 491–492, December 2008, Berkeley, United States

13. D. Van Deursen, C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. XML to RDF Conversion: a Generic Approach. In *Proceedings of the 4th International Conference on Automated Production of Cross Media Content for Multi-channel Distribution*, pages 138–143, November 2008, Florence, Italy

14. R. De Sutter, E. Mannens, D. Van Rijsselbergen, and R. Van de Walle. Automatic News Production. In *Proceedings of the International Broadcasting Conference*, pages 158–165, September 2008, Amsterdam, The Netherlands

Chapter 4

Making Disclosure of News Future Proof

Humor is a serious thing. I like to think of it as one of our greatest earliest natural resources, which must be preserved at all cost.

James Thurber

4.1 Introduction

If we just digitise our ‘physical’ multimedia news archives and safeguard these as electronic records, is this cultural heritage then without any doubt protected against all sorts of hazards: theft, loss, damage, fire, or other natural disasters? If our ‘paper’ archive would vanish into thin air, then one can indeed always revert to ‘a’ digitally secured version of the asset within the archive –preferably on distributed servers. On top of this safeguarding guarantee, a digital depot has a number of other advantages too: spatial borders are omitted, mobility is no longer a barrier, and searches provide enormous gaining in time as consultation of ‘physical’ documents/essence are no longer necessary. Nevertheless, such a digital archive is not invulnerable at all. Moreover, only a slight obsolescence of certain apparatus can have catastrophic consequences. As one can barely keep pace with the evolution in IT, preservation of our digital heritage, most notably multimedia in news items in our use case, is indeed a burning issue. Solving aforementioned issue is stringent as one definitely wants to avoid the loss of billions of terabytes of digital essence thereby also losing huge investments in both time, money, and irreplaceable memories. If we want to preserve digital multimedia content in the right way, archiving environments must comply to a number of specific requirements [89,116]. In one respect all software and hardware solutions should guarantee access to the information for a prolonged period of time. On the other hand, human contribution –be it in the form of archival descriptions, adoption of work processes, or the use of standards– vouches for information to be available and interpretable as long as possible for diverse groups of users, as digital information is also extremely brittle. As some hazards also threaten analogue documents, the following ones are distinctive for digital data though:

- Information in digital form is a conceptual object. Digital multimedia can be copied and altered very easily without any visual effects on the represented content. In contrast to analogue information, it is therefore much more difficult to preserve the authenticity of digital information. Both hardware, software, and human errors can induce data loss in short-term. Often these kinds of errors can be resolved by specialised correction methods. In other cases these data corruptions are only detected in a later phase, on a moment where all data is seemingly processed correctly neglecting both technical and visual aspects of the intellectual content.

- Technological changes can make data formats unusable over time as the life span of all storage techniques is finite. To guarantee durable access to digital multimedia information one needs to foresee migration and emulation plans. In addition, technical metadata should supply enough information on filed data to make timely interventions possible.

- In the long-term, changes occur in the domain knowledge of user groups, data specialists disappear, and organisations are reshuffled or just get new assignments. There's indeed always the danger of stale data being uninterpretable by the new user generation. The stored data must therefore be documented with enough contextual information to enable these new user groups with ever interpretable information.

Dependent on its nature, multimedia data is encapsulated in different container and file formats, all of which are solidly built to technical criteria to withstand obsolescence. Next to that, the specific knowledge domain stipulates which descriptive metadata are necessary, i.e., digital images in a library thus represent a scanned book which should be described by bibliographic metadata, whereas digital images in a museum merely represent an artwork, where other descriptive metadata needs to be taken into account. In the same way, a video resource owned by a broadcaster possibly depicts an episode of a (news) TV show, whereas a video resource being part of an installation of a video artist is to be interpreted in a totally different way. The former descriptive metadata picture the series and the episode, where the latter descriptive metadata cite the artist and the specifications of his video installation. As such, each sector or domain will embrace their own specific metadata requirements for describing multimedia resources. Nevertheless, generic archive consultation will benefit from having a common spanning descriptive metadata model which is able to search similar, but disjoint datasets completely.

Besides archiving the audio, video, photo, and text of news items, the repository has to be able to store aggregations of these objects too. Broadcasters should not only be able to disseminate their individual news items, but also introductory statements, interviews with stakeholders, related essence, etc. As such, these aggregations have to be stored, disseminated, and exchanged. Furthermore, this vast amount of metadata should be made easily available in bulk to other broadcasters/archival institutions too as a harvesting service.

The contributions of this chapter are therefore twofold. On one hand, Section 4.2 introduces the different levels for ‘active’ archiving, whereas in Section 4.3 we report on the necessity and the best practice of how to design a layered semantic metadata model for lasting ‘active’ archiving. On the other hand, in Section 4.4 we discuss these modelling decisions with respect to their consequences on the dissemination of these archived news events to an end-user, e.g., be it as a simple aggregation to be used in education, or be it as a complete harvesting service to be used in the Europeana[1] context. To do so, different strategies can be followed as will be elaborated on within the next few subsections therein. Finally, after contemplating on related work in Section 4.5, conclusions are drawn in Section 4.6.

4.2 Metadata Levels for ‘Active’ Archiving

Preservation of digital objects must be done on three conceptual levels [89]. One must preserve both the *medium*, the *technology*, and the *intellectual content*. As we want to be able to both exchange and archive multimedia news data, a semantic layered metadata model is necessary to describe all data fully and accurate on these three defined conceptual levels, as will be demonstrated in the subsequent sections.

[1] `http://www.europeana.eu/portal/`

4.2.1 Preserving the Medium

On the lowest level a digital file consists of bits and bytes saved on hardware systems which are liable to wear-and-tear. These hard disks and tapes have indeed a limited life span as their bit streams can be altered by external influences, e.g., corruption of these digital carriers. This level therefore needs error correcting hardware and software solutions. The authenticity of digital data is harder to guarantee and maintain than it is for analogue data. In the latter case it is sufficient to describe all features of the physical object, but for digital information the whole history of provenance and continuous processing must be archived too. 'Tenability' metadata, named *binary metadata* hereafter, are an answer to these issues.

4.2.2 Preserving the Technology

On a higher level, file formats and compression formats, e.g., MPEG-2 [85] (for video), MP3 [60] (for audio), and JPEG [96] (for images), describe the way the bits can be transformed to an interpretable multimedia representation. When a file format becomes obsolete, the archive has two options to preserve the stored data: migration, i.e., moving electronic files from one application to another, or emulation, i.e., designing software and/or hardware that will mimic a specific application. Both have pros and cons, as migration can cause data loss and emulation can become very complex. But in either case, metadata is needed to support these actions. At the same time, open standards are vital to foster future understandability and migratebility of file formats, as the interpretation of proprietary file formats needs proprietary software, which is again more difficult to archive in a sustainable manner. At this level, it is also very important to preserve the look and feel of the multimedia objects, as, e.g., resolution or color values might change when migrating file formats. Thus, a rich description of the look and feel is also necessary.

4.2.3 Preserving the Intellectual Content

On the highest level, the information should remain interpretable. Institution structures, terminologies, the designated community, and the rights of an object or institution might change over time. To keep that kind of information interpretable too, enough extra 'semantic' information should be included in the archived package. At this level, the archive not only needs descriptive metadata for a general description of the object, but also rights metadata and context metadata for describing the relations of the content information to information which is not packed in the information package itself. Examples of such context metadata are related datasets, references to documents in the original environment at the moment of publication, and helper files. This contextual information becomes indispensable overtime as the initial providers of the extra information on the datasets themselves are no longer available to explain why they archived the datasets in a certain way. As such, a dataset should provide enough contextual metadata to keep it interpretable for a designated community without the help of external experts.

4.3 Layered Semantic Metadata Model for 'Active' Archiving

According to the Open Archival Information System [89] (OAIS) an archive needs different types of metadata to fully describe the information of its assets for long-term preservation. As stated above, data can be lost on all three conceptual levels (medium, technology, and content). As such, the following six types of metadata are a guarantee that the data can be sufficiently described to fully satisfy these three levels:

- *Binary metadata* describe the data on bit level. Bitstreams are the actual data in a file. Binary metadata, e.g., file system information and file header information, keep the enclosed information accessible by pointing out how the bits should be transformed to a representation of the data, e.g., in a certain compression format.

- *Technical metadata* describe the data on file level. Data formats and their derivatives evolve quickly. As both container and compression formats age, it is hard to find software that is still able to interpret old formats. The only way to keep this kind of information accessible is to support migration and/or emulation in which the technical metadata, e.g. coder-decoder (codec) information, will be key in keeping that possible.

- *Structural metadata* describe the relationships between a set of files that correspond to a possible representation of the intellectual content of certain data. A news show might be an aggregation of a set of news items in a specific order identified by the rundown (as specified in Chapter 2). This structural metadata is necessary to fully describe the news show as a correct ordering of these news items.

- *Descriptive metadata* describe extra data, e.g., journalist, title, location, date, etc., to better find and locate the original data. When archiving digital multimedia content from different industries/institutions –

be it broadcasters, libraries, cultural institutions, and archives– an additional problem concerning descriptive metadata arises, i.e., a lot of industries/institutions already describe, control, and save their descriptive metadata according to their own standardised schemes. As such, some file these extra metadata as metadata, others file them as real data. Both strategies have their pros and cons. If a coordinating archive wants to file these extra descriptions as metadata, it means it is forced to choose one metadata standard to do so, which is not obvious, as most metadata schemes are domain specific. To guarantee lossless filing of all descriptive metadata our coordinating archive must opt for the lowest common multiple of all descriptive metadata schemes used by all partnering industries/institutions, which would lead to an enormous unmaintainable metadata scheme. We therefore opted to archive the descriptive metadata (in its original metadata format) together with their original data, thus being sure not to lose any information ever. On top of that, our coordinating archive foresees a generally excepted, descriptive exchanging metadata scheme (the greatest common divisor, see our *Upper Layer* in Section 4.3.1 and accompanying Figure 4.1) to be effectively used to search the complete, non-homogeneous archive. As such, the original metadata (saved as data itself) can be presented to the end-users, when the right (meta)data is *found* in the first place.

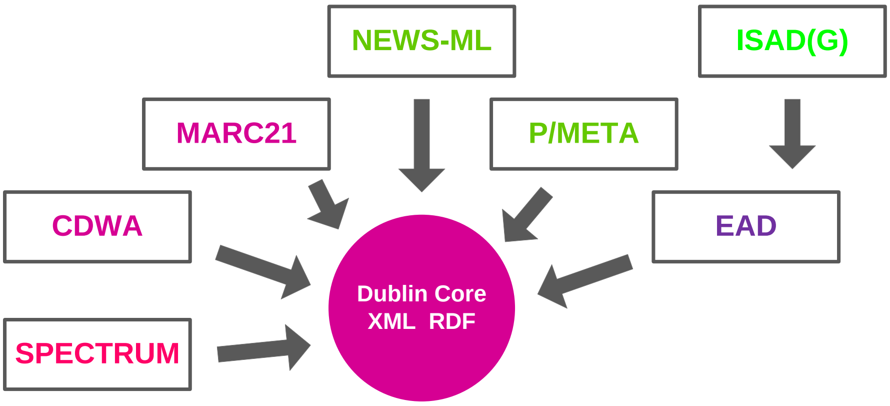


Figure 4.1: Greatest Common Divisor Mapping.

- *Preservation metadata* describe essential extra data that support and document the digital preservation process. No digital storage device is perfect and perpetually liable, as bit preservation is still an unsolved paradigm. The simplest model of these failures is analogous to the decay of radioactive atoms. Each bit in a data file independently is subject to a random process that has a constant small probability per unit time of causing its value to flip. The time after which there is a 50% probability that a bit will have flipped is the '*bit half-life*'. The requirement of a 50% chance that 1 petabyte of data will survive for a century translates into a required bit half-life of 8×10^{17} years. To put things into perspective, the current estimate of the age of the universe U is 1.4×10^{10} years, so this is a bit half-life of approximately only $6 \times 10^{7}U$ [146]. As stated earlier, information in a digital form is a conceptual object. This information can be altered and copied pretty easily without one notifying that in its visible representation. Opposed to analogue information, it is indeed much harder to preserve the authenticity of digital information. This too can be solved by adding tenability metadata to the preservation package of the archived essence. Such metadata have check sums, digital signatures, certificates, encryption, and cyclic redundancy control for indicating the data is not altered without it being documented. Furthermore, an archived dataset also needs its provenance documented. This type of preservation metadata (e.g., encoding software, version history, references to the original sources, etc.) describes the genesis of the intrinsic information, i.e., the original owners of the data, the processes determining the current form of that data, and all of its available, intermediate versions, as this information is vital in verifying all changes the data has experienced from genesis until date. Lastly, context-aware metadata (e.g., related data sets, help files, original language on first publication, etc.) must be retained, as these describe possible relationships of the intrinsic data with other data that is not embraced within its own information package.

- *Rights metadata* describe the rights on digital objects (e.g., rights metadata for describing copyright statements, (changing) licenses, and possible grants that are given), as this info is also vital to guarantee long-term access to the data, and thus must be archived too.

When developing a metadata schema for the long-term preservation of digital multimedia, metadata descriptions on all levels have to be taken into account, going from bit level descriptions to descriptions of the intellectual content. To realise this, we developed a layered semantic metadata schema, as will be shown in the next couple of subsections. Motivation and related work can be found in our book [116] written in cooperation with the BOM-Vl-project, as can be seen in Appendix E.7. The top layer offers an exchange layer of the descriptive metadata, which can be enriched with extra available information from the Web and published on-line if provisioned by the juridical conditions. The bottom layer offers an archive layer which takes care of the preservation metadata, rights metadata, binary metadata, technical metadata, and structural metadata necessary for active archiving. For the top layer, the RDFS [22] representation of DC [71] was chosen. For the bottom layer, an OWL representation of the preservation standard PREservation Metadata Implementation Strategies 2.0 [34] (PREMIS) is developed. This standard is based on the OAIS reference model and describes the data on all necessary levels.

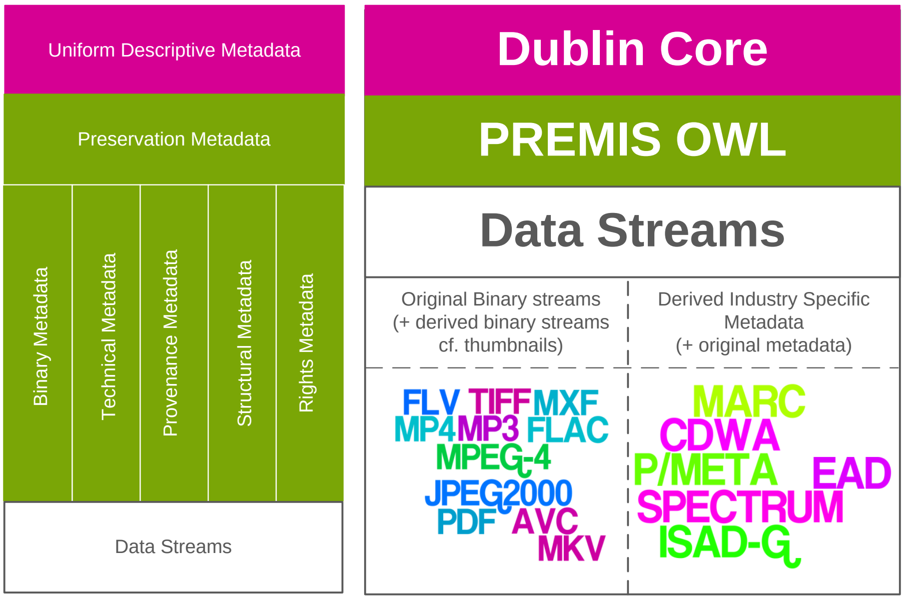


Figure 4.2: Layered Metadata Model.

4.3.1 Upper Layer: Dublin Core

Descriptive metadata describe the content of the data: subject, author, date of creation, file format, etc. This metadata makes it possible to manage and search the complete digital archive. DC was chosen to describe this top layer of descriptive metadata, as it is a broadly accepted descriptive scheme. The power of this schema is its simplicity and generality. It only consists of fifteen fields among which creator, subject, coverage, description, and date. It can answer to the basic W-questions: *Who*, *What*, *Where*, and *When*. All the fields in

DC are optional and repeatable. This makes it possible to map relatively easily almost all the descriptive metadata schemes to DC, as many institutions already support DC. As such, DC will be used as an upper ontology for linking all the descriptive metadata schemes with each other. For this, the RDFS schema of DC[2] is used. This RDFS schema allows linking the DC properties to data types, but also to other resources. With the RDFS schema, the upper ontology DC is most interoperable with the other models. There also exist OWL models of DC, but they designed the properties of DC as annotation properties, which makes them ineligible linking them to other resources.

4.3.2 Bottom Layer: PREMIS

For this layer, we developed an OWL schema of the preservation standard PREMIS 2.0 (as can be seen in Appendix C.1), which is based on the OAIS reference model. The data model of PREMIS 2.0 consists of five semantic units or classes important for digital preservation purposes:

- *Intellectual Entities*: a part of the content that can be considered as an intellectual unit for the management and the description of the content. This can be, e.g., a photo, or a database.
- *Object*: a discrete unit of information in digital form.
- *Event*: an action that has an impact on an object or agent.
- *Agent*: a person, institution, or software application that is related to an event of an object or is associated to the rights of an object.
- *Rights*: description of one or more rights, permissions of an object or agent.

Intellectual entities, events, and rights are directly related to an object. An agent can only be related to an object through an event or through rights. This way, not only the changes to an object are stored, but the event involved in this change is also described. These relationships offer the necessary tools to store the provenance of an object properly, as Figure 4.3 clarifies the data model of PREMIS 2.0.

[2] http://purl.org/dc/elements/1.1/

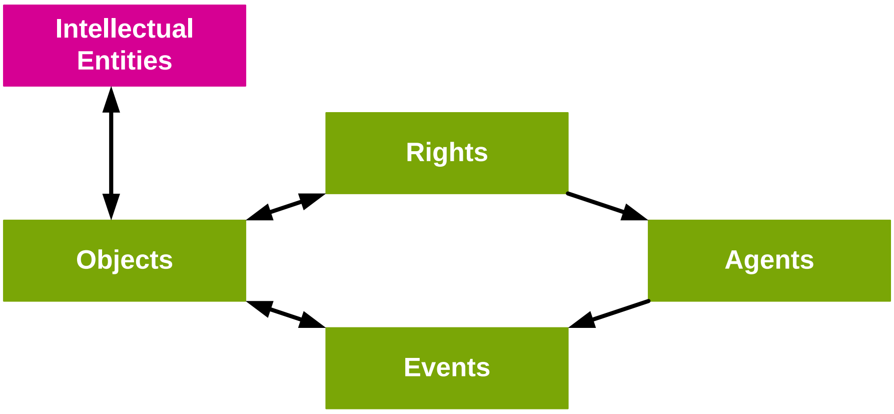


Figure 4.3: PREMIS Data Model.

4.3.2.1 Why PREMIS OWL?

Looking at the PREMIS 2.0 data model, one can notice it is dynamically relating the five entities to each other. Until now an XML schema[3] was available that implemented the PREMIS 2.0 data dictionary. This XML schema used the identifiers of the entities to relate those to one another. As a consequence, the relations between the entities are directed and not bi-directional. When using this XML schema as a data model for the long-term preservation platform, this has to be kept in mind when keeping the whole model consistent. Implementing the data dictionary using OWL, on the other hand, allows us to relate the entities directly to each other, without the need of referring to an identifier of the entity. Another advantage of using semantic Web technologies to implement the PREMIS 2.0 data dictionary, is that the relations can be made bi-directional using inverse properties, as was envisioned in the first place. As such, using a semantic model of the PREMIS 2.0 data dictionary helps keeping the whole archive more consistent in reusing information as much as possible.

For the implementation of the formalised ontology of the PREMIS 2.0 data dictionary, the XML schema of the data dictionary is used as a starting point. It needs to be stressed that PREMIS is a data model for the management of the archive, though on top of that, it can also be used to disseminate the preservation information, albeit it not being the primary concern. When designing the

[3] http://www.loc.gov/standards/premis/premis.xsd

OWL ontology of the PREMIS specification, we sticked as closely as possible to the original PREMIS 2.0 data dictionary specification, as that was a strict requirement from the PREMIS standardisation board. The OWL ontology of PREMIS 2.0 can thus be seen as a translation of the XML schema into an OWL schema. The data dictionary of PREMIS 2.0 was developed by experts in the domain of long-term preservation, where every element has its own clearly defined semantics. As such, any information loss was unacceptable according the PREMIS standardisation board.

4.3.2.2 Object Class

The *Object* class describes a unit of information in digital form and is related to the *intellectual entity* class. This *intellectual entity* is described by descriptive metadata which are very domain-specific. For this, there already exist a lot of descriptive metadata models. Therefore, the description of the *intellectual entity* is out of scope for PREMIS. In our implementation, the top layer describes this *intellectual entity*. The *Object* class contains three subclasses:

- *File*: a file is an ordered sequence of bytes that is known by the system.
- *Bitstream*: a bitstream is the actual data inside a file.
- *Representation*: a representation is a set of files with structural metadata needed for a complete description of an intellectual entity.

The *Object* class possesses all the necessary features to describe the object on the different levels, which is a recommendation of the OAIS reference model, as there are risks involved on every level. The minimum information for describing an object (*File*, *Bitstream*, or *Representation*) are:

- *objectIdentifier*, which gives the identifier of the object.
- *objectCharacteristics*, needed for the *Bitstream* subclass and the *File* subclass, which gives the necessary technical and binary metadata
- *storage*, necessary for describing a *File* or *Bitstream*, which indicates either the location the data is stored, or the medium the data is stored on.

An object can be described further in detail using:

- *preservationLevel*, because some repositories offer the opportunity to define a preservation level for an object.

- *significantProperties*, defining some significant properties of the object, which need to be preserved when, e.g., migrating the data.

- *originalName*, for indicating the original names of the packages delivered to the repository.

- *environment*, which describes the environment the user needs to render the content and interact with the content.

- *signatureInformation*, for storing digital signatures generated during ingest into the repository.

For linking object information to events, intellectual entities, or rights statements, the *Object* class offers four properties, i.e., *relationship*, *linkingEvent*, *linkingIntellectualEntity*, and *linkingRightsStatement*.

The *relationship* property is used to relate an object to one or more other objects. The relationship type can be structural (for relating parts of an object) or derivative (for relating objects, where one object is the result of a transformation performed on a related object). Structural relationships are used, e.g., to link a bitstream to a file. Derivative relationships are employed, e.g., to describe migrations of a file to another file format. For these relationships, no object property can be used relating an object directly to another object and/or event, because one is able to define the relationship type (e.g., structural or derivative), and the relationship sub-type (e.g., is part of, is source of, ...).

The other object properties, *linkingEvent*, *linkingIntellectualEntity*, and *linkingRightsStatement*, relate an object directly to an event, rights statement or intellectual entity. Furthermore, the *linkingEvent* property is not intended for structural or derivative relationships, but only for linking events to an object which do not alter the object, e.g., checksum checking.

4.3.2.3 Event Class

An *event* aggregates all the information about an action that involves one or more objects. This metadata is stored separately from the object metadata. Actions that modify objects should always be recorded as events. The *Event* class is described at least by an *eventIdentfier*, *eventType* (e.g., capture or creation), and an *eventDateTime*. This information can be extended using the *eventDetail* property, which gives a more detailed description of the event, and the *eventOutcomeInformation*, which describes the outcome of the event, in terms of success, failure, or partial success. These properties are able to describe any *event* altering an *object*. The *Event* class can be related to an *Agent* class or *Object* class via the respective properties *linkingAgent* and *linkingObject*.

4.3.2.4 Agent Class

This class aggregates information about attributes or characteristics of agents. *Agents* can be persons, organisations, or software. The minimum properties needed to describe the *Agent* class unambiguously are *agentIdentfier* and *agentType*. Optionally, an agent can also be described using the *agentName*. This is just enough to identify the agent. An agent can hold or grant one or more rights. It may carry out, authorise, or compel one or more events. Furthermore an agent can only create or alter an *object* through an *event* or with respect to a *rights* statement. The relationships between an *agent* and an *object* through an *event* or *rights entity* make it possible to describe the whole provenance of an *object*.

4.3.2.5 Rights Class

The minimum core rights information that a preservation repository must hold, is what rights or permissions a repository has to carry out related to objects within that repository. These may be granted by copyright law, by statute, or by a license agreement with the rights holder. *Rights* entities can be related to one or more *objects* and one or more *agents*. Every *Rights* class can be related to different *RightsStatements*. A *RightsStatement* encompasses three subclasses: the *Copyright* subclass, the *License* subclass, and the *Statute* subclass.

These three subclasses offer the necessary metadata for describing rights information, i.e., copyrights, licenses, and statutes. Every *RightsStatement* is described at least by a *rightsStatementIdentifier*, and has also the optional property *rightsGranted*, which describes the actions the granting agency has allowed the repository. The *RightsStatement* class can be related to an *Object* class or *Agent* class via the optional, repeatable object properties: *linkingObject* and *linkingAgent*.

This part of the PREMIS OWL schema is extended with a vocabulary that describes the roles *agents* can have concerning a *rights* statement. Our vocabulary is again based on the results of research performed within the BOM-Vl project. To fully describe the rights of an object, all the persons, involved in the production of the described object, should be taken into account which is impossible for many organisations. Therefore, a check-list was made with the most important rights and rights holders that should be described. Based on this check-list a vocabulary was made to describe the important legal roles of an agent, e.g., journalist, editor, director, actor, author, composer, conductor, etc.

4.4 Aggregating/Harvesting the News for Disclosure

4.4.1 Aggregation through OAI-ORE

4.4.1.1 OAI-ORE Protocol

Besides archiving the audio, video, photo, and text of news items, the repository has to be able to store aggregations of these objects too. Broadcasters should not only be able to disseminate their individual news items, but also introductory statements, interviews with stakeholders, related essence, etc. These aggregations have to be stored, disseminated, and exchanged too. For this, we use the Open Archives Initiative Object Reuse and Exchange [107] (OAI-ORE) protocol. Today, many information systems, like Content Management Systems (CMS), support the storage and identification of aggregations, and access to the aggregations and aggregated objects. In most systems, these objects vary in semantic type (e.g., article, book, video, dataset, etc.) and in metadata file format (e.g., PDF, XML, MP4, etc.). These objects can also be stored on different network locations, i.e., aggregated objects can be stored locally or externally. Information systems store, identify, and deliver access to these compound objects in an architecture-specific manner. Unfortunately, the way these information systems disseminate their compound objects is far from perfect and without any broadly accepted standard. In many cases, a lot of the advanced functionalities get lost when publishing the compound objects on the Web. Mostly, the publication is aimed at the end-users (humans) and not at agents (machines) such as Web crawlers. The structure of the object is often embedded in splash pages, user interface widgets, etc. This approach makes the structure of the compound object unclear for machine-based applications like browsers, Web crawlers, etc. Consider the example of a scanned book, where all the pages get an HTTP URI. A Web crawler can come across one of these pages and find links to the other pages of the book, to the chapter containing that page, or to the book itself. A Web crawler cannot distinguish between these links. For the Web crawler these are untyped links or links that do contain information, but this information remains unreadable to the Web crawler. Therefore, the order of the pages gets lost, etc.

The OAI-ORE standard tackles this problem by developing a standardised, interoperable and machine-readable mechanism that can express the information of compound objects. The standard makes sure that the logical boundaries of the aggregated objects and their mutual relations remain intact for machine agents when publishing the compound object on the Web. To achieve this, OAI-ORE makes use of resource maps. These resource maps are RDF (machine-readable) descriptions of the aggregation. They list the aggregated resources, their mutual relations and the Web context of the aggregation, together with the URI of the resource it is describing, i.e., the aggregation. In fact, these resource maps are named graphs [27] too, i.e., sets of triples, extended with a name, an URI, for the graph/resource map. The dereferencable graph is not the aggregation itself, but a representation of its description encoded in Atom or RDF/XML, as depicted in Figure 4.4. The ORE model demands that a resource map describes just one aggregation. An aggregation, on the other hand, can have multiple resource maps, each with its own representation. This makes it possible to describe the same aggregation, for instance, with an RDF description and a XHTML description.

Clients and applications need to determine the stable URI of the resource map from the URI of the aggregation, to get a description of that aggregation. This can happen in two ways:

- Firstly, every aggregation should get an URI, e.g., '`http://example.org/foo/`', just like any resource on the Web. From this URI, a Web agent is able to automatically get a machine-readable description of the aggregation, namely the resource map. Of course, this resource map also has an URI. This URI should be deducted from the URI of the aggregation. This is done, for instance, by using *Cool URIs* [8]. The Web agent adds '.rdf' or '.atom' to the URI of the aggregation and gets its machine-readable description, e.g., '`http://example.org/foo.rdf`' or '`http://example.org/foo.atom`'.
- The other (client-side) way is to append a fragment identifier ('#') to the URI of the resource map to get the aggregation, e.g., '`http://example.org/foo.rdf#aggregation`'

4.4.1.2 Semantic OAI-ORE Schema Implementation

The essence of the RDFS data model implementation is described here and is illustrated in Figure 4.4. The full details are available in the OAI-ORE Ab-

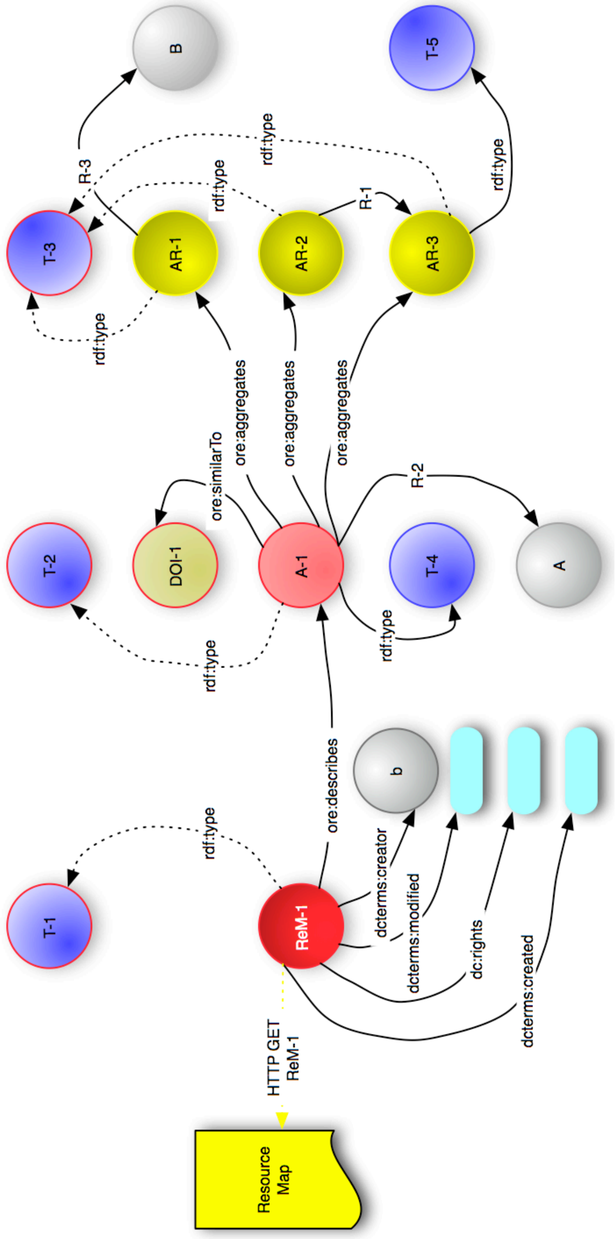


Figure 4.4: Schematic Representation of an OAI-ORE Aggregation.

stract Data Model specification [107]. In order to be able to unambiguously refer to an aggregation of Web resources, a new resource is introduced for a set or collection of other resources. This new resource, named an *Aggregation*, has an URI just like any other resource on the Web. And, since an *Aggregation* is a conceptual construct, it is a non-document resource that does not have a representation. Following the LOD guidelines, another resource is introduced to make information about the *Aggregation* available. This new resource, named a *ResourceMap*, has an URI and a machine-readable representation that provides details about the *Aggregation*. In essence, a *ResourceMap* expresses which *Aggregation* it describes (the *ore:describes* relationship in Figure 4.4), and it lists the aggregated resources that are part of the *Aggregation* (the *ore:aggregates* relationship, a subproperty of *dcterms:hasPart*). But, a *ResourceMap* can also express relationships and properties pertaining to all these resources, as well as metadata pertaining to the *ResourceMap* itself, e.g., who published it and when it was most recently modified (the *dcterms:creator* and *dcterms:modified* relationships in Figure 4.4). A *ResourceMap* can also express relationships of the *Aggregation*, aggregated resources, and the *ResourceMap* itself with any arbitrary other resource, as long as the resulting RDF graph is connected. In addition, for discovery purposes, the OAI-ORE data model allows a *ResourceMap* to express that an aggregated resource of a specific *Aggregation* is also part of another *Aggregation*. This is achieved by means of the *ore:isAggregatedBy* relationship (the inverse of *ore:aggregates*) between the aggregated resource and that other *Aggregation*. Be aware that an aggregated resource is itself an *Aggregation*, as nesting aggregations is supported. To that purpose, an *ore:isDescribedBy* relationship (the inverse of *ore:describes*, and a subproperty of *rdfs:seeAlso*) is expressed between the aggregated resource and a *ResourceMap* that describes it as being itself an *Aggregation*. Furthermore, the use of non-protocol-based identifiers (such as Digital Object Identifiers, i.e., DOIs) that can be expressed as URIs is quite common for referencing any multimedia (news) assets. In order to support this practice, the *ore:similarTo* relationship between an *Aggregation* and a somehow equivalent resource identified by a non-protocol-based URI is expressed. The specificity of *ore:similarTo* is situated between *rdfs:seeAlso* and *owl:sameAs*.

4.4.2 Harvesting through OAI-PMH

4.4.2.1 OAI-PMH Protocol

The Open Archives Initiative Protocol for Metadata Harvesting [108] (OAI-PMH), on the other hand, is a protocol used for sharing and exchanging metadata. This protocol is very popular in the domain of digital libraries, thus ‘in extenso’ in the archiving domain. Currently more than 1700 repositories expose their metadata descriptions for several millions of items via the OAI-PMH protocol [65]. Client applications can use this protocol to harvest metadata coming from data providers using open standards like HTTP and XML. By implementing wrappers on top of their metadata repositories, broadcasters/institutions can easily expose their metadata via OAI-PMH. The number of OAI-PMH end-points is expected to grow, because many popular open source digital library systems, such as Fedora[4], DSpace[5], and EPrints[6], provide an OAI-PMH end-point by default. Another reason for the growth of these OAI-PMH end-points, is that major attempts to build union catalogues, e.g., The European Library project[7], rely on this protocol for indexing metadata originating from different remote sources. The OAI-PMH protocol disseminates the metadata about the items of a repository. These items can describe both digital and non-digital resources. An item is again identified by an URI. Each item can have multiple metadata records, each described by a (possibly) different metadata schema. These schemes are chosen by the data provider to suit their domain-specific demands. The most frequently used schemes are Bibliographic Records [75], MAchine Readable Cataloguing [112] (MARC) and MARC-21, Metadata Object Description Schema [114] (MODS), Metadata Encoding and Transmission Standard [113] (METS), and NewsML [80] in our specific use case. To guarantee a basic level of interoperability one of those metadata schemes must be unqualified DC as described in Section 4.3.1. The OAI-PMH protocol is based on HTTP. The request arguments are issued as GET or POST parameters, whereas the responses are encoded in XML syntax. Furthermore this OAI-PMH protocol only supports six request types (‘verbs’) as explanatory examples show in Appendix C.3):

- The *Identify* request retrieves administrative metadata about the repository, e.g., the name or owner of the repository.

- The *GetRecord* request retrieves metadata about an item in a certain metadata format.

- The *ListRecords* request harvests all metadata records in a certain metadata format for all items in the repository.

- The *ListIdentifiers* request lists all the identifiers of the available items.

[4] http://www.fedora-commons.org/pdfs/WhitePaper.10.28.05.pdf

[5] http://www.dspace.org/1_5_1Documentation/DSpace-Manual.pdf

[6] http://www.ariadne.ac.uk/issue50/eprints-v3-rpt/

[7] http://www.theeuropeanlibrary.org/

- The *ListMetadataFormats* request returns the available metadata formats used in the repository.

- The *ListSets* request gives the available sets in the OAI-PMH repository.

4.4.2.2 OAI-PMH Limitations

Although the OAI-PMH protocol is a widespread protocol for disseminating metadata records over primarily HTTP, it has some limitations. A first limitation is the use of possible non-dereferencable identities, i.e., stale URNs, to retrieve information from a OAI-PMH repository, a client must execute an HTTP request (only using one of the defined 6 verbs) on an OAI-PMH specific URI. Another limitation of the OAI-PMH protocol are its selection criteria. The client has only limited access to the metadata. The criteria to retrieve a record are the item identifier, the metadata formats, the sets, and the record creation date intervals. Retrieving a record that fulfils a certain condition, except the ones mentioned above, is alas not possible. For example, a request asking all the records about news items of *Paul Pierce*[8] in the repository is not possible. Publishing these records according to the LOD principles [16] (see Section 4.4.3.2 below) and providing a SPARQL end-point overcomes these problems. In order to search the whole repository, a common, upper layer metadata schema for initial enquiries has to be provided.

4.4.3 Interweaving OAI-ORE & OAI-PMH Data Sources with the Linked Data Cloud

4.4.3.1 LOD vs. OAI-ORE

Publishing resources as LOD conforms to the way OAI-ORE offers to publish aggregations [167, 168]. OAI-ORE demands that aggregations have to be identified by an URI, and have to be described using an RDF schema, i.e., a resource map, which also has an URI. When clients consume the URI of that aggregation, they should be able to automatically detect the URI of the resource map with the appropriate representation for the client. This principle totally conforms to the way LOD is published.

4.4.3.2 LOD vs. OAI-PMH

Interweaving OAI-PMH data sources, on the other hand, with the continuously evolving LOD could bring the following benefits [66]:

[8] http://en.wikipedia.org/wiki/Paul_pierce/

- Metadata that currently only can be harvested by OAI-PMH clients will become accessible for various clients and could then easily be integrated in various application scenarios.

- Semantic search engines and crawlers such as Sindice [131] will be able to index the exposed metadata, which in turn will increase the visibility of both the content they describe and the institution that provides the data.

- Clients will be able to follow links to other datasets and combine information that would not be related otherwise.

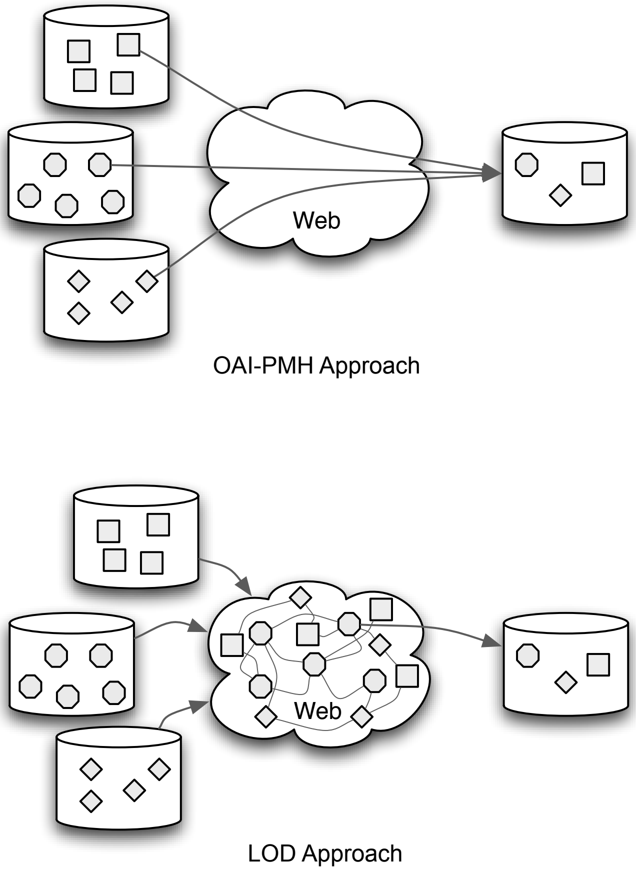


Figure 4.5: Conceptual Differences between OAI-PMH and LOD.

The current version of OAI-PMH does not allow for such a direct integration though. Although it uses Web technologies –in particular HTTP, XML, and URIs– for exchanging metadata, these have mainly the role of a transport layer between repositories. LOD, in contrast, follows the idea of the Web as being an information space in which the items of interest (resources) are identified by global identifiers (URI) and which allows embedded references to other URIs [99] as discussed in Section 3.3.1. Figure 4.5 illustrates and explains the conceptual differences between the OAI-PMH and the LOD approach. As a consequence, LOD requires that metadata are not only exchanged via the Web, but also exposed on the Web so that each described digital or non-digital item is accessible via a unique dereferencable URI, independent of any OAI-PMH specifics. In a second step, the exposed metadata originating from OAI-PMH data sources may be linked with related data from other sources so that applications can combine these different datasets.

OAI-PMH has been designed for transferring large amounts of metadata from a server to a client via the Web. From that perspective, it provides a reasonable solution for clients that need to aggregate or index metadata from remote repositories. The goal of the LOD initiative, however, is a different one: it aims at exposing metadata on the Web as machine and human-readable data that describe certain resources, which in turn are identified via dereferencable URIs. An additional goal of LOD is to provide structured query access to these data via SPARQL, which is both an RDF query language and a Web-based query protocol. As a result the following conceptual differences between OAI-PMH and LOD are identified:

- Both protocols use URIs for the identification of resources; in OAI-PMH, however, these URIs serve solely for identification purposes whereas in LOD, URIs take the role of dereferencable identities.

- OAI-PMH introduces protocol specific verbs (e.g., *GetRecord*) and a set of adjacent parameters. Clients must be aware of these verbs and parameters in order to be able to retrieve metadata from a remote repository. LOD, in contrast, builds on the functionality provided by existing Web technologies; e.g., standard HTTP methods. As a result, LOD data are accessible for any client that is aware of HTTP, URI, and HTML or RDF, respectively.

- LOD relies on the built-in HTTP content negotiation features in order to deliver data in various representations; OAI-PMH is restricted to XML as the only valid representation format.

- OAI-PMH provides batch retrieval functionality, which enables the transfer of a large amount of metadata descriptions within a single HTTP transaction. In LOD, such functionality is provided indirectly by the SPARQL query language and protocol. SPARQL allows the formulation of complex selection criteria and provides *LIMIT* and *OFFSET* clauses to return metadata that match certain criteria (e.g., all records describing items created by X) or even just a subset of the available metadata values (e.g., all authors of all news items in an archive).

- OAI-PMH can return metadata records for one and the same item in several metadata formats. When following the LOD design principles without tailoring them to OAI-PMH specific needs, one cannot request specific metadata formats from a LOD end-point. It is however possible to describe a resource with different vocabularies and to use the SPARQL query language to return metadata in certain formats only.

- OAI-PMH supports a kind of version control and allows clients to retrieve only those metadata records that were created or modified in a given date range, specified by *from* and *until* attributes in the *ListRecords* and *ListIdentifiers* requests. One possible approach in the context of LOD is to keep so called linked data update logs [6]. Another simple, straightforward solution is to introduce OAI-PMH specific vocabulary terms and use SPARQL to query for date ranges in order to retrieve the resources that have been created or modified within a specific date range. Lastly, the promising Memento project [168] tries to solve this issue on protocol level using the previously mentioned OAI-ORE paradigms.

Regarding these conceptual differences, we can observe that the LOD approach already subsumes a large fraction of the functionality provided by OAI-PMH, even though in a slightly different manner. This implies that if existing OAI-PMH data providers publish their metadata on the Web following the LOD principles, any client can fetch and process these metadata by simply crawling and resolving the exposed, dereferencable URIs in a certain URI domain space.

The proposed OAI2LOD Server [65] has some drawbacks, though. It serves records from an in-memory Jena RDF model[9] as the number of records a server can host, depends on the amount of memory assigned to the Java Virtual Machine. To overcome this problem, we adapted the OAI2LOD Server to serve records from an Openlink's Virtuoso triple store[10]. Another limitation

[9] http://jena.sourceforge.net/index.html

[10] http://virtuoso.openlinksw.com/wiki/main/Main/

of the OAI2LOD Server is that it can only serve records from one OAI-PMH repository. In order to solve this, we extended the OAI2LOD Server to expose not only sets and records from one OAI-PMH repository, but also collections, which collect the sets and records from other OAI-PMH repositories. This way, the OAI2LOD Server can harvest records from different OAI-PMH endpoints. Finally, we enhanced the OAI2LOD Server to serve records not only in DC, but also in our developed layered metadata schema (consisting of DC RDFS and PREMIS OWL as described in Section 4.3). Now, the harvested, transformed metadata records can be truly exposed as LOD.

4.5 Related Work

There is a lot of interest in digital preservation as can be seen by the multitude of projects in this area. *PLANETS*[11] (Preservation and Long-term Access through NETworked Services) was especially aimed at defining guidelines for preservation planning. However, it did not tackle how to integrate different existing metadata formats, or how to disseminate the metadata as LOD, as we proposed. Likewise, the *PRESTOSPACE*[12] (Preservation towards storage and access) project's objectives were to provide technical solutions and integrated systems for a complete digital preservation of all kinds of audio-visual collections. The project was especially focussed on the underlying technologies, e.g., automated generation of metadata or detection of errors in content [123], but is not using a standardised, semantic preservation model to support the archiving, as we developed. The *CASPAR*[13] project (Cultural Artistic and Scientific knowledge for Preservation, Access, and Retrieval) presented technologies for digital preservation. Here too, the OAIS Reference Model was chosen as the base platform, and the project was focused on implementing the different steps in the preservation workflow. They focus more on preservation services than on describing the preservation information though. Current trends are on integrating different media archives, as we also envisioned. *PrestoPRIME*[14], an ongoing project, researches and develops practical solutions for the long-term preservation of digital media objects, programmes and collections, and finds ways to increase access by integrating the media archives with European on-line digital libraries in a digital preservation framework.

[11] http://www.planets-project.eu/

[12] http://prestospace.org/project/index.en.html

[13] http://www.casparpreserves.eu/

[14] http://www.prestoprime.org/

Dappert et al. [39] define a data dictionary that can be used together with PREMIS. The usage of a common data dictionary in a digital archive is recommended, however this does not solve the interoperability problem of the different metadata formats available. Schmidt et al. [148] present a framework to create distributed preservation workflows. The framework provides a number of software components, including authorisation and authentication, workflow execution, service discovery, data and metadata management. However, the inclusion of Semantic Web languages like RDF is deemed to be future work. It is our opinion that our system can be integrated with the proposed framework, but further considerations are needed to see how our semantic layered metadata model can be included.

Our proposed architecture allows to include the metadata formats that were originally used to describe the data. However, the problem of utilising different metadata formats within one system is not trivial. Xing et al. [183] present a system for automating the transformation of XML documents using a tree matching approach. However, this method has an important restriction: the leaf text in the different documents has to be exactly identical. This is hardly the case when combining different metadata standards. Likewise, Yang et al. [186] propose to integrate XML Schema. They have a more semantic approach, using their Object-Relationship-Attribute model for Semi-Structured data (ORA-SS) to represent the information available in the XML Schemas and to provide mappings between the different documents. Their ORA-SS data model allows to define objects and attributes to represent hierarchical data, however more advanced semantic relationships cannot be represented. Poppe et al. [142] advocates a similar approach to deal with interoperability problems in CMSs using Semantic Web Technologies. An OWL upper ontology is created and the different XML-based metadata formats are represented as OWL ontologies and mapped to the upper ontology using OWL constructs and rules. However, the upper ontology is dedicated to the CMS and, as such, not suitable for this use case. Related to this, Stegmaier et al. [160] presented an overview of how to better combine and align metadata schemas, as is proposed by W3C's MAWG [179].

4.6 Conclusions and Original Contributions

When preserving digital (news) information for the long-term, different types of metadata are important to retain. Descriptive metadata are needed to describe the intellectual entities, whereas binary metadata, technical metadata, and structural metadata are essential for the description of the data on all lower levels (bitstream, file, and representation). Preservation metadata is also necessary to describe the provenance of the data, to guarantee the authenticity of its digital nature, and to provide a context. At last, rights metadata also need to be stored.

Our two-layered, semantic metadata schema described in this chapter offers the freedom to embrace all of these metadata types within a heterogeneous archiving context. We were the first to come up with this generic architecture for both open access and lasting archive, which were until recently considered orthogonal and not compatible. Our top layer (an RDFS representation of DC) takes care of the descriptive metadata and is also usable to initiate the exchange of multimedia data from different domains with non-homogeneous metadata. Our bottom layer, which became also the official OWL representation of PREMIS 2.0 (acknowledged by both the PREMIS standardisation board and Library of Congress), on the other hand, encompasses the needed binary metadata, technical metadata, structural metadata, preservation metadata, and the rights metadata for future-safe archiving. To describe the rights in a more detailed manner, the PREMIS OWL schema was extended with a vocabulary defining the different legal roles of persons, organisations and software. By describing the data with this layered metadata schema, all the risks that come with long-term preservation are minimised. By splitting up the semantic schema in two layers, the top layer with the descriptive metadata can be made public and weaved into the Web of data, if the rights permit it. The bottom layer can remain closed for the public –but a shift from mere 'access control' to 'transparent & accountable use/mis-use policies' should be encouraged– and is responsible for the long-term preservation of that data.

For (news) data providers –be it a certain local broadcaster or its coordinating European archival counterpart– that aim at integrating their OAI-PMH data end-points into the LOD cloud, we recommend from our experience to follow the OAI-PMH guidelines and expose their metadata also in other formats than DC, as we did using our layered semantic metadata model. Regarding the OAI-ORE developments, we can observe that the LOD principles already play an important role in the news use case domain. We have also seen that the conceptual *continuity semantic gap* (as mentioned in Chapter 1.2) between OAI-PMH and OAI-ORE is narrow and can easily be bridged by intermediate gateways like our enhanced OAI2LOD Server. Since the LOD approach actually subsumes a large fraction of the OAI-PMH functionalities, we believe that future releases of the OAI-PMH standard should even consider a shift towards the LOD principles, which would also enable a tighter integration with the newer

OAI-ORE protocol. Meanwhile, our enhanced OAI2LOD Server can be used for bridging this conceptual continuity semantic gap between all of these standards. Point by point my own research contributions can thus be summarised as follows:

- First to come up with a two-layered, semantic metadata schema for both open access and lasting archive, which were until recently considered orthogonal and not compatible.

- Official provider of OWL representation of PREMIS 2.0 (acknowledged by both the PREMIS standardisation board and Library of Congress).

The research that has led to this chapter is also described in the following publications:

1. E. Mannens, S. Coppens, L. Hauttekeete, T. Evens, and R. Van de Walle. Semantic Bricks for Performing Arts Archives and Dissemination. *IASA Journal*, number 35, pages 1–10, IASA, June, 2010

2. E. Mannens, S. Coppens, and R. Van de Walle. Een Gelaagd Semantisch Metadatamodel voor Langetermijnarchivering. *Bibliotheek- & Archiefgids*, number 5, pages 17–22, Vlaamse vereniging voor Bibliotheek-, Archief- en Documentatiewezen, September, 2009

3. E. Mannens, S. Coppens, and R. Van de Walle. A Network-centric Approach to Sustainable Digital Archives. In *the Proceedings of the 41th International Conference of the International Association of Sound and Audio-visual Archives*, pages 1–1, November 2010, Philadelphia, USA

4. E. Mannens, S. Coppens, L. Hauttekeete, R. De Sutter, and R. Van de Walle. Cloudcomputing Approach to Sustainable Media Archives. In *the Proceedings of the International Federation of Television Archives World Conference*, pages 1–1, October 2010, Dublin, Ireland

5. E. Mannens, D. Van Rijsselbergen, L. Hauttekeete, R. De Sutter, and R. Van de Walle. The Repurposing of Archive Content: The MEMENTO Project. In *the Proceedings of the International Federation of Television Archives World Conference*, pages 1–6, October 2009, Beijing, China

6. E. Mannens, S. Coppens, and R. Van de Walle. Semantic Bricks for Performing Arts Archiving & Dissemination. In *the Proceedings of the 40th International Conference of the International Association of Sound and Audiovisual Archives*, pages 85–86, September 2009, Athens, Greece

7. E. Mannens, S. Coppens, P. Bastijns, S. Corneillie, P. Hochstenbach, L. Van Melle, S. Van Peteghem, and R. Van de Walle. In *Book '(Meta)datastandaarden voor Digitale Archieven'*, pages 1–204, Universiteitsbibliotheek Gent, June, 2009

8. E. Mannens, T. Paridaens, L.Hauttekeete, T. Evens, and J. Gysels. In *Book 'Van Horen Zeggen III - Haalbaarheidsstudie naar een Innovatieve Applicatie voor de Ontsluiting van Mondelinge Bronnen'*, pages 1–172, Universiteitsbibliotheek Gent, September, 2007

9. L. Hauttekeete, K. De Moor, D. Schuurman, T. Evens, E. Mannens, and R. Van de Walle. Archives in Motion: Concrete Steps towards the Digital Disclosure of Audio-visual Content. *International Journal of Cultural Heritage*, Accepted for publication, Elsevier, March, 2011

10. T. Evens, L. De Marez, L. Hauttekeete, D. Biltereyst, E. Mannens, and R. Van de Walle. Attracting the Un-served Audience: The Sustainability of Long Tail-based Business Models for Cultural Television Content. *New Media & Society*, volume 12, issue 6, pages 1005–1023, SAGE Publications, September, 2010

11. S. Coppens, E. Mannens, J. Haspeslagh, P. Hochstenbach, I. Van Nieuwerburgh, and R. Van de Walle. Metadatastandaarden, Dublin Core en het Gelaagd Metadatamodel. In *Book Chapter in 'Bewaring en Ontsluiting van Multimediale Data in Vlaanderen'*, pages 46–62, Lannoo Campus, June, 2010

12. S. Coppens, E. Mannens, and R. Van de Walle. Disseminating Heritage Records as Linked Open Data. *International Journal of Virtual Reality*, volume 8, number 3, pages 39–44, IPI Press, September, 2009

13. T. Evens, L. Hauttekeete, E. Mannens, and R. Van de Walle. Surfen naar het Verleden. De Ontsluiting van Mondelinge Historische Bronnen in Vlaanderen. *FARO: tijdschrift over cultureel erfgoed*, volume 1, number 1, pages 26–32, FARO, March, 2008

14. S. Coppens, and E. Mannens. Workflow Engines PREMIS OWL Binding for Long-Term Preservation. In *the Proceedings of the 41th International Conference of the International Association of Sound and Audiovisual Archives*, pages 1–1, November 2010, Philadelphia, USA

15. S. Coppens, E. Mannens, and R. Van de Walle. Digital Long-Term Preservation using a Layered Semantic Metadata Schema of PREMIS

2.0. In *the Proceedings of the 2nd CULTURAL HERITAGE on line Conferenc*, pages 1–6, December 2009, Florence, Italy

16. L. Hauttekeete, T. Evens, E. Mannens, and R. Van de Walle. The Repurposing of Archive Content: The PokuMOn Project. In *the Proceedings of the International Federation of Television Archives World Conference*, pages 1–20, October 2009, Beijing, China

17. L. Hauttekeete, T. Evens, E. Mannens, and R. Van de Walle. Browsing through Memories: the Online Disclosure of Oral History in Flanders. In *Abstract Book of the 1st Global Conference on Digital Memories*, pages 22–22, March 2009, Salzburg, Austria

18. S. Notebaert, J. De Cock, S. Coppens, E. Mannens, M. Jacobs, J. Barbarien, P. Schelkens, and R. Van de Walle. Digital Recording of Performing Arts: Formats and Conversion. In *Book Chapter in 'Access to Archives of Performing Arts Multimedia'*, pages 95–119, VTi–IBBT, August, 2009

19. S. Coppens, E. Mannens, and R. Van de Walle. Semantic BRICKS for Performing Arts Archives and Dissemination. In *Book Chapter in 'Access to Archives of Performing Arts Multimedia'*, pages 121–141, VTi–IBBT, August, 2009

20. L. Hauttekeete, T. Evens, E. Mannens, and R. Van de Walle. Browsing through Memories: the Online Disclosure of Oral History in Flanders. In *Book Chapter in 'Digital Memories: Exploring Critical Issues'*, pages 139–147, Inter-Disciplinary Press, March, 2009

Chapter 5

Making Time Uniformly Identifiable in News Items

Percy, the colour of gold, is gold! Whatever substance you have discovered –if it has a name– would be called green! ... Oh Edmund, can it be true, that I hold in my mortal hands a nugget of purest GREEN?

Blackadder & Percy in Blackadder

5.1 Introduction

Media resources on the WWW used to be treated as 'foreign' objects as they could only be embedded using a plugin that is capable of decoding and interacting with these media resources. The HTML5 specification, however, is a game changer and all of the major browser vendors have committed to support the newly introduced `<video>` and `<audio>` elements [69][1]. However, in order to make audio/video clips accessible in a transparent way, it needs to be as easily linkable as simple HTML pages. In order to share or bookmark only the interesting parts of a video, we should be able to link into or link out of this time-linear media resource. If we want to further meet the prevailing accessibility needs of a video, we should be able to dynamically choose our preferred tracks that are encapsulated within this media resource, and we should be able to easily show only specific ROIs within this media resource. And last but not least, if we want to stroll through media resources based on (encapsulated) semantics, we should be able to

[1] At the time of writing, the following browsers support the HTML5 media elements: IE 9, Firefox 3.5, Chrome 4, Safari 4, Opera 10

master the full complexity of rich media by also enabling standardised media annotation [139, 165]. The mission of the W3C Media Fragments Working Group (MFWG) [180], which is part of W3C's Video in the Web activity[2], is to provide a mechanism to address media fragments on the Web using URIs [67, 79]. The objective of the proposed specification is to improve the support for the addressing and retrieval of sub-parts of media resources, as well as the automated processing of such sub-parts for reuse within the current and future Web infrastructure [119]. Example use cases are the bookmarking or sharing of media fragment URIs with friends via social network notifications by linking to specific regions of joint images, the automated creation of fragment URIs in search engine interfaces by having selective previews, or the annotation of media fragments when tagging audio and video spatially and/or temporally [117]. The examples given throughout this chapter to explain the Media Fragments URI specification are based on the following two scenarios. In scenario (a), Steve –a long-time basketball enthusiast– posts a message on his team's blog containing a Media Fragment URI that highlights 10 seconds of an NBA video clip showing the same nifty move that he himself performed in last Saturday's game. In scenario (b), Sarah –a news reporter by profession– quickly previews only the video footage in search of a particular high quality 10 seconds sub-clip to finalise editing today's headline story for the 'Evening News'.

Needless to say that this Media Fragments specification will have a major impact on the complete end-to-end news production chain. From the moment news footage is shot, edited and enriched with extra information down the news production workflow chain, until a specifically chosen news item is viewed by an interested end-user, one will be able to *uniquely identify, link to, display, browse, bookmark, re-composite, annotate, and/or adapt* spatial and/or temporal sub-clips of media resources, e.g., a camera might automatically annotate footage with the exact geo-coordinates the moment it is shot, a news editor might quickly browse through a months' footage by means of highlight captions in search of one particular item, whereas an end-user might create a video mash-up from his bookmarked video segments to share with his friends on a social platform.

The contributions of this chapter are the following. Firstly, we present the rationale for a Media Fragments specification in Section 5.2. In Section 5.3, we outline the boundaries and semantics of a Media Fragments URI and show how the syntax should look like, whereas Section 5.4 elaborates on how a

[2] http://www.w3.org/2008/WebVideo/Activity.html

media fragment specified as an URI fragment can be resolved stepwise using the HTTP protocol [78]. We then identify the influence of the current media formats on fragment extraction in Section 5.5. Finally, we outline our future work and give our conclusions in Section 5.7.

5.2 Related Work

Before video can become a 'first-class citizen' on the Web, one urgently needs to provide standardised ways to localise the temporal, spatial, and track sub-parts of audio-visual media content [67, 165]. Previous efforts to do so include both URI-based and non-URI-based mechanisms.

In the non-URI-based class of solutions, the Synchronised Multimedia Integration Language [26] (SMIL) specification over HTTP allows to play only a temporal fragment of the video by using the *clipBegin* and *clipEnd* attributes. However, current implementations have to get the complete media resource and then cut it up locally, which entails a terrible waste of bandwidth in case of large video resources. Using MPEG-7 [87], a video is divided into *VideoSegments* that can be described by a *MediaTimePoint* and *MediaDuration* corresponding to the starting time and shot duration, respectively. Using TV-Anytime [52], temporal intervals also can be defined, accessed, and manipulated through *segments* within an audio/video stream using MPEG-7 types for specifying the temporal boundaries. For images, one can use either MPEG-7 or a Scalable Vector Graphics [55] (SVG) code snippet to define the bounding box coordinates of specific regions. However, this approach implies an extra *indirection* since a semantic description of this region will actually be *about* a piece of an XML document just defining a multimedia fragment and not the fragment itself. As such, the identification and the description of the temporal fragment or region is intertwined (this use of indirection) and one needs to first parse and understand the metadata model in order to get access to the media fragment afterwards, which is not desirable at all. Finally, HTML ImageMaps [144] can also define spatial regions (a.o., rectangles) via the `<area>` and `<a>` elements. However, the complete media resource has first to be downloaded to the user agent too.

As for the URI-based mechanisms, SVG has a spatial URI mechanism using the '`#`' that specifies the region of an SVG image to be viewed, but having the same limitations as SMIL. The temporalURI draft specification [140] defines a temporal fragment of multimedia resources using the query parameter '`?`', thus creating a new resource which is not desirable as fragments should still have a direct link to their 'parent' resource. An in-depth analysis of '#' versus '?' is discussed in Section 5.3.3. MPEG-21 [94], on the other hand, specifies a normative syntax to be used in URIs for addressing parts of any resource using the '#', but the supported media types are restricted to the MPEG formats only. Furthermore, this specification has a very complex syntax which can be ambiguous. Hence, four schemes *–ffp()*, *offset()*, *mp()*, and *mask()–* are defined and both *mp()* and *ffp()* can be used for example to identify tracks. Since our rationale is to find a scheme that is easy enough to get the attention of developers and have a real impact on the Web, our impression is that MPEG-21 Part 17 was often over-designed even for the most simple use cases, thus preventing its adoption, as there are no real-world applications implementing this specification since it was launched in 2006. YouTube released a tool[3] to link to particular time points in videos and to annotate parts of those videos spatio-temporally. It alas uses the URI fragment marker '#' in a completely proprietary way. In contrast, the solution advocated by the MFWG is to only send the bytes corresponding to the media fragments requested and still be able to cache them, as for fragment-aware protocols, such as Real Time Streaming Protocol [149] (RTSP), we observed that this behaviour is possible. As a final requirement, we therefore want to be compatible with solutions widely deployed and enable for HTTP what is already possible with RTSP.

5.3 Media Fragments URIs

5.3.1 Media Resource Model

We assume that media fragments are defined on top of 'time-linear' media resources, which are characterised by a single timeline (see also Figure 5.1). Such media resources usually include multiple tracks of data, all parallel along this uniform timeline. These tracks can contain video, audio, text, images, or any other time-aligned data. Each individual media resource also contains control information in data headers, which may be located at certain positions within the resource, either at the beginning or at the end, or spread throughout the data tracks as headers for those data packets. There is also typically a general header for the complete media resource. To comply with progressive decoding, these different data tracks may be encoded in an interleaved fashion. Normally, all of this is contained within one single container file.

[3] http://www.youtube.com/t/annotations_about/

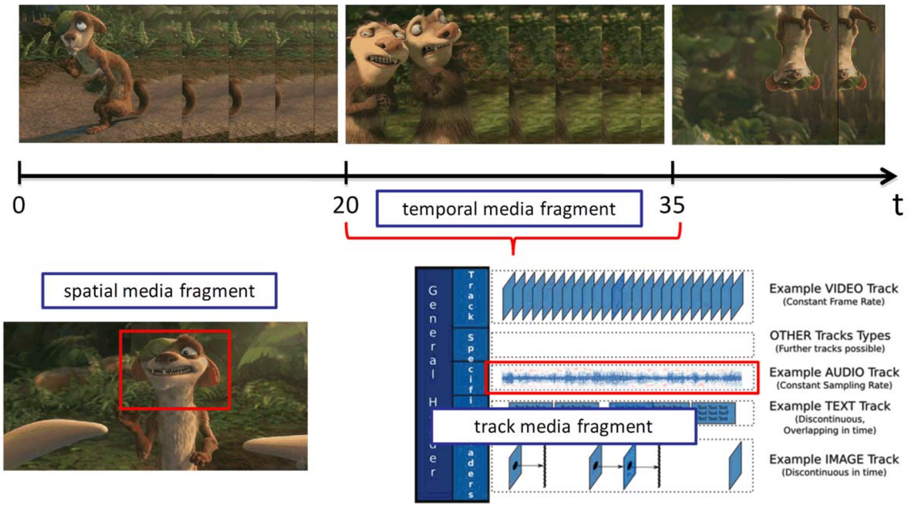


Figure 5.1: Example of Media Fragments and the Video Resource Model.

5.3.2 Requirements

We formally define a number of requirements for the identification and access of media fragments. Based on these requirements, we motivate a number of design choices for processing Media Fragment URIs.

- *Independent of media formats*. The Media Fragments URI specification needs to be independent of underlying media formats, such as MP4 [88] or Ogg [138].

- *Fragment axes*. Media fragments need to be accessed along three different axes: temporal, spatial, and track. Additionally, media fragments also could be identified through names.

- *Context awareness*. A media fragment must be a secondary resource of its parent resource. This way, the relationship between the media fragment and its parent resource is kept. Moreover, when accessing media fragments, user agents need to be aware of the context of the media fragment, i.e., where is the fragment located within the original parent resource.

- *Low complexity*. The Media Fragment URI specification should be kept as simple as possible. For instance, defining spatial regions is limited to the definition of rectangular regions which should be sufficient for most applications.

- *Minimise impact on existing infrastructure.* Necessary changes to all software components –be it user agents, proxies, or media servers– in the complete media delivery chain should be kept to a bare minimum. Furthermore, the access to media fragments should work as much as possible within the boundaries of existing protocols, such as FILE, HTTP(S), and RTSP [149].
- *Fragment by extraction.* Preferably, it should be possible to express media fragments in terms of byte ranges pointing to the parent media resource. This makes the fragments real sub-resources of the 'de facto' media resource. Therefore, we consider media segments obtained through a re-encoding process not as media fragments.

5.3.3 URI Fragments vs. URI Queries

According to the URI specification [79], URI fragment markers '#' point at so-called 'secondary' resources. Per definition, such a secondary resource may be "some portion or a sub-part of the primary resource, a particular representation view of the primary resource, or even another resource described by those representations." As such, it makes a viable syntax for Media Fragment URIs. A further consequence of the use of URI fragments is that the media type of the retrieved fragment should be the same as the media type of the primary resource, which means that an URI fragment pointing to a single video frame within a video results in a one-frame video, and not in a still image. To make these URI fragment addressing methods work, only byte-identical segments of a media resource can be considered, as we assume a simple mapping between the media fragment and its elementary byte-aligned building blocks. Where byte-identity cannot be maintained –and thus a form of transcoding of the resource is needed–, the user agent is not able to resolve the fragmentation by itself and an extra server interaction is required. In this case, URI queries have to be used, as they result in a server interaction to deliver a newly created transcoded resource.

The main difference between an URI query and an URI fragment is indeed that an URI query creates a completely new resource having no relationship whatsoever with the resource it is created from, while an URI fragment delivers a secondary resource that relates to the primary resource. As a consequence, URI query created resources cannot be mapped byte-identical to their parent resources –this notion does even not exist– and are thus considered re-encoded segments. As of the aforementioned requirements '*context awareness*' and '*fragment by extraction*' described in Section 5.3.2, the use of URI fragments is preferable over the use of URI queries. We discuss how to resolve media fragments using the '`#`' and the '`?`' in Section 5.4.

In the case of play-lists composed of media fragment resources, the use of URI queries –receiving a completely new resource instead of just byte segments from existing resources– could be desirable since it does not have to deal with the inconvenience of the original primary resources, i.e., its larger file headers, its longer duration, and its automatic access to the original primary resources. On top of that, URI queries have to take care of an extra caveat of creating a fully valid new resource. This implies typically a reconstruction of the media header to accurately describe the new resource, possibly applying a non-zero start-time or using a different encoding parameter set. Such a resource will then be cached in Web proxies as a different resource to the original primary resource. Current and future caching proxies will indeed also have to deal with Media Fragments URIs as bandwidth saving is one of the main use cases and issues.

5.3.4 Fragment Dimensions

5.3.4.1 Temporal Axis

The most obvious *temporal dimension* denotes a specific time range in the original media, such as 'starting at second 10, continuing until second 20'. Temporal clipping is represented by the identifier '`t`', and specified as an interval with a begin and an end time (or an in-point and an out-point, in video editing terms). If either or both are omitted, the begin time defaults to second 0 and the end time defaults to the end of the entire media resource. The interval is considered half-open: the begin time is part of the interval, whereas the end time on the other hand is the first time point that is not part of the interval. The time units that can be used are Normal Play Time (npt), real-world clock time (clock), and Society of Motion Picture and Television Engineers (SMPTE) time-codes [149, 159]. The time format is specified by name, followed by a colon, with '`npt:`' being the default. Some examples are:

Listing 5.1: Temporal Axis Example.

```
t=npt:10,20       # results in the time interval [10,20[
t=,20             # results in the time interval [0,20[
t=smpte:0:02:00, # results in the time interval [120,end[
```

5.3.4.2 Spatial Axis

The *spatial dimension* denotes a specific spatial rectangle of pixels from the original media resource. The rectangle can either be specified as pixel coordinates or percentages. A rectangular selection is represented by the identifier ‘`xywh`’, and the values are specified by an optional format ‘`pixel:`’ or ‘`percent:`’ (defaulting to pixel) and 4 comma-separated integers. These integers denote the top left corner coordinate (x,y) of the rectangle, its width and its height. If percent is used, x and width should be interpreted as a percentage of the width of the original media, and y and height should be interpreted as a percentage of the original height. Some examples are:

Listing 5.2: Spatial Axis Example.

```
xywh=160,120,320,240        # results in a 320x240 box
                              at x=160 and y=120
xywh=pixel:160,120,320,240  # results in a 320x240 box
                              at x=160 and y=120
xywh=percent:25,25,50,50    # results in a 50%x50% box
                              at x=25% and y=25%
```

5.3.4.3 Track Dimension

The *track dimension* denotes one or multiple tracks, such as ‘the English audio track’ from a media container that supports multiple tracks (audio, video, subtitles, etc). Track selection is represented by the identifier ‘`track`’, which has a string as a value. Multiple tracks are identified by multiple name/value pairs. Note that the interpretation of such track names depends on the container format of the original media resource as some formats only allow numbers, whereas others allow full names. Some examples are:

Listing 5.3: Track Dimension Example.

```
track=1&track=2           # results in only extracting
                            track '1' and track '2'
track=video               # results in only extracting
                            track 'video'
track=Kids%20Video        # results in only extracting
                            track 'Kids Video'
```

5.3.4.4 Named Dimension

The *named dimension* denotes a named section of the original media, such as ‘chapter 2’. It is in fact a semantic replacement for addressing any range along the aforementioned three axes (temporal, spatial, and track). Name-based selection is represented by the identifier ‘`id`’, with again the value being a string. Percent-encoding can be used in the string to include unsafe characters (such as a single quote). Interpretation of such strings depends on the container format of the original media resource as some formats support named chapters or numbered chapters (leading to temporal clipping), whereas others may support naming of groups of tracks or other objects. As with track selection, determining which names are valid requires knowledge of the original media resource and its media container format.

Listing 5.4: Named Dimension Example.

```
id=1                          # results in only extracting
                                the section called '1'
id=chapter-1                  # results in only extracting
                                the section called 'chapter-1'
id=My%20Kids                  # results in only extracting
                                the section called 'My Kids'
```

As the temporal, spatial, and track dimensions are logically independent, they can be combined where the outcome is also independent of the order of the dimensions. As such, following fragments should be byte-identical:

Listing 5.5: Combined Dimensions Example.

```
#t=10,20&track=vid&xywh=pixel:0,0,320,240
#track=vid&xywh=0,0,320,240&t=npt:10,20
#xywh=0,0,320,240&t=smpte:0:00:10,0:00:20&track=vid
```

5.4 Resolving Media Fragments with HTTP

In the previous section we described the Media Fragments URI syntax, whereas in this section we present how a Media Fragments URI should be processed using the HTTP protocol. We foresee that the logic of the processing of a Media Fragments URI will be implemented within smart user agents, smart servers and sometimes in a proxy cacheable way. We observe that spatial media fragments are typically interpreted on the user agent side only (i.e., no spatial fragment extraction is performed on server-side) for the following reasons:

- Spatial fragments are typically not expressible in terms of byte ranges. Spatial fragment extraction would thus require transcoding operations resulting in new resources rather than fragments of the original media resource according to the semantics of the URI fragments defined in the corresponding RFC [79].
- The contextual information of extracted spatial fragments is not really usable for visualisation on client-side.

In the remainder of this section, we describe how to resolve Media Fragments URIs using the HTTP protocol focusing on the temporal and track dimensions.

5.4.1 User Agent mapped Byte Ranges

As stated in Section 5.1 (scenario (a)), Steve can now show off his awesome play using his smart-phone displaying the specific scene posted on his blog by using the following Media Fragments URI:

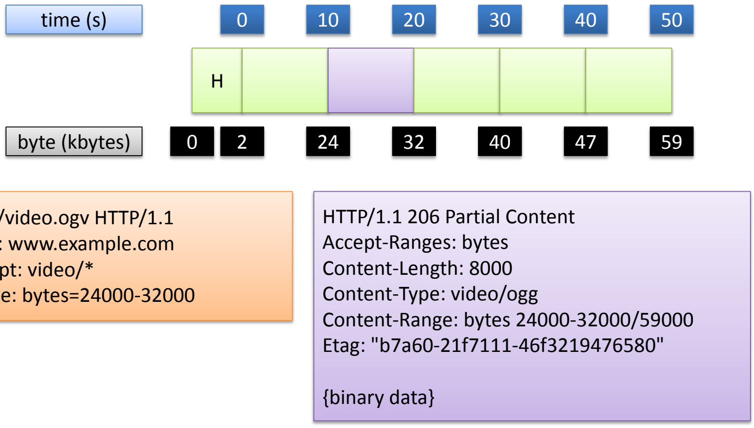
```
http://example.com/video.ogv#t=10,20
```

Since Steve does not want to blow his entire monthly operator fee with the unnecessary bandwidth cost of downloading the full movie, he uses a smart user agent that is able to interpret the URI, determine that it only relates to a subpart of a complete media resource, and thus requests only the appropriate data for download and playback. In this scenario, media fragments can be served by traditional HTTP Web servers. However, a smart user agent is necessary to parse the Media Fragments URI syntax. Furthermore, it requires knowledge about the syntax and semantics of various media codec formats.

5.4.1.1 User Agent requests Fragment for the first Time

We assume that Steve's smart-phone runs a smart user agent (e.g., an HTML5 compliant browser) that can resolve the temporal fragment identifier of the Media Fragments URI through an HTTP byte range request. A click on this URI triggers the following chain of events:

1. The user agent checks if a local copy of the requested fragment is available in its buffer, which is not the case.



```
GET /video.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
Range: bytes=24000-32000
```

```
HTTP/1.1 206 Partial Content
Accept-Ranges: bytes
Content-Length: 8000
Content-Type: video/ogg
Content-Range: bytes 24000-32000/59000
Etag: "b7a60-21f7111-46f3219476580"

{binary data}
```

Figure 5.2: User Agent requests Fragment for the first Time.

2. We consider that the user agent knows how time is mapped to byte offsets for this particular Ogg media format. It just parses the media fragment identifier and maps the fragment to the corresponding byte range(s).

3. The user agent sets up the decoding pipeline for the media resource at `http://example.com/video.ogv` by just downloading the first couple of bytes of the file that corresponds to the resource's header information.

4. The MIME-type of the requested resource is confirmed. The user agent can use the header information to resolve the fragment byte ranges. Based on the calculated time-to-byte mapping and the extracted information from the resource's header of where to find which byte, the user agent sends one or more HTTP requests using the *HTTP Range request* header (see Figure 5.2) for the relevant bytes of the fragment to the server. In this particular case, an HTTP 1.1 compliant Web server will be able to serve the media fragment.

5. The server extracts the bytes corresponding to the requested range and responds with an *HTTP 206 Partial Content response* containing the requested bytes.

6. Finally, the user agent receives these byte ranges and is able to decode and start playing back the initially requested media fragment.

Opera[4] already has implemented this recipe in its new alpha version and it is to be expected that all major browser vendors will include support for this Media Fragment's recipe in its upcoming versions. Also HTML5 already foresees support for this recipe within its media elements.

5.4.1.2 User Agent requests Fragment which is buffered

This time we also further assume that Steve already showed his clip to some of his friends, which means his smart user agent has a copy of it in its buffer. Therefore, a click on the aforementioned URI triggers the following chain of events:

1. The user agent checks if a local copy of the requested fragment is available in its buffer, which is the case.

2. The user agent knows how time is mapped to byte offsets for this particular Ogg media format. It just parses the media fragment identifier and maps the fragment to the corresponding byte range(s).

3. The resource could have changed on the server though, so the user agent needs to send a *conditional HTTP GET* using the *If-Modified-Since* and *If-None-Match* headers (see Figure 5.3), so it requests these byte ranges from the server under condition of it having changed.

4. The server checks if the resource has changed by checking the date. In this case the resource was not modified, so the server replies with a *304 HTTP response*. Note that an *HTTP If-Range header* cannot be used, because if the entity has changed, the entire resource would be sent.

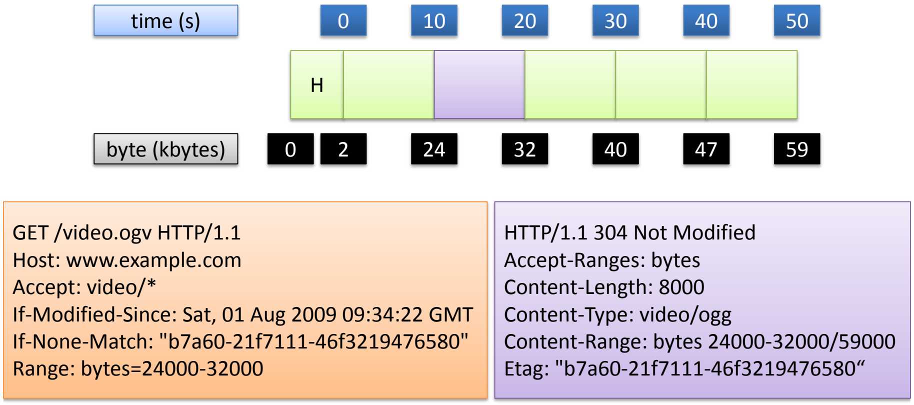
5. Finally, the user agent serves the decoded resource from its own existing buffer and starts playing back the initially requested media fragment.

5.4.1.3 User Agent requests Fragment of a changed Resource

In this particular case the aforementioned assumptions of the previous subsection still hold, but the resource on the server has changed. A click on that same URI now triggers the following chain of events:

1. The user agent checks if a local copy of the requested fragment is available in its buffer, which is the case.

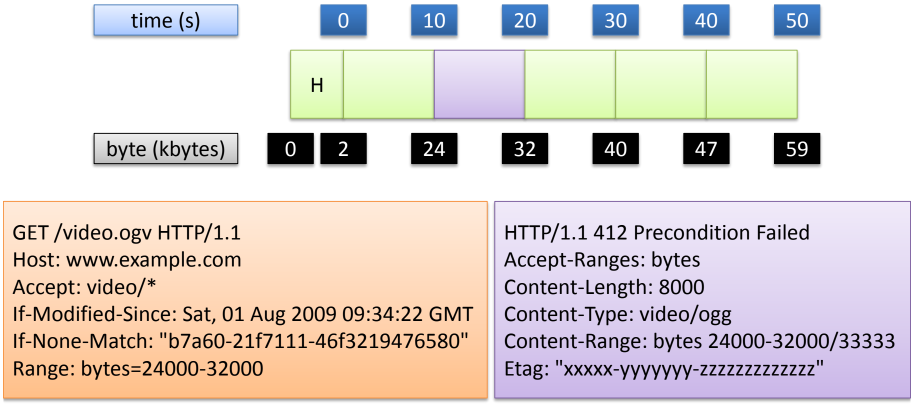
[4] http://www.opera.com/



```
GET /video.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
If-Modified-Since: Sat, 01 Aug 2009 09:34:22 GMT
If-None-Match: "b7a60-21f7111-46f3219476580"
Range: bytes=24000-32000
```

```
HTTP/1.1 304 Not Modified
Accept-Ranges: bytes
Content-Length: 8000
Content-Type: video/ogg
Content-Range: bytes 24000-32000/59000
Etag: "b7a60-21f7111-46f3219476580“
```

Figure 5.3: User Agent requests Fragment which is already buffered.



```
GET /video.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
If-Modified-Since: Sat, 01 Aug 2009 09:34:22 GMT
If-None-Match: "b7a60-21f7111-46f3219476580"
Range: bytes=24000-32000
```

```
HTTP/1.1 412 Precondition Failed
Accept-Ranges: bytes
Content-Length: 8000
Content-Type: video/ogg
Content-Range: bytes 24000-32000/33333
Etag: "xxxxx-yyyyyyy-zzzzzzzzzzzz"
```

Figure 5.4: User Agent requests Fragment of a changed Resource.

2. The user agent knows how time is mapped to byte offsets for this particular Ogg media format. It just parses the media fragment identifier and maps the fragment to the corresponding byte range(s).

3. The resource could have changed on the server though, so the user agent needs to send a *conditional GET*, so it requests these byte ranges from the server under condition of it having changed.

4. The server checks if the resource has changed by checking the date. In this case the resource has changed. Since the byte mapping may not be correct any longer, the server can only tell the user agent that the resource has changed and leave all further actions to the user agent, so it sends a *412 HTTP response* (see Figure 5.4).

5. Finally, the user agent can only assume the resource has changed and thus has to re-retrieve what it needs to get back of being able to serve this fragment. For most resources this may mean retrieving the header of the file, after which it is possible to again do a byte range retrieval, as is elaborated on within Subsection 5.4.1.1.

5.4.2 Server mapped Byte Ranges

We assume now that the user agent is able to parse and understand the Media Fragments URI syntax, but is unable to perform by itself the byte range mapping for a given request. In this case, the user agent needs some help from the media server to perform this mapping and deliver the appropriate bytes.

5.4.2.1 Server mapped Byte Ranges including Header Information

In a first case, the server has to help the client to set-up the initial decoding pipeline (i.e., there is no header info from the resource on client side yet). When Steve starts to play this by now ‘infamous’ 10 seconds of video, the following events occur:

1. The user agent parses the media fragment identifier and creates an *HTTP Range request* expressed in a different unit than bytes, e.g., a time unit expressed in seconds. Furthermore, it extends this header with the keyword ‘`include-setup`’ (see Figure 5.5) in order to let the server know that it also requires the initial header of the resource to initiate the decoding pipeline.

2. The server extracts the header information of the media resource as requested with ‘`include-setup`’.

3. The server, which also understands this time unit, resolves the *HTTP Range request* and performs the mapping between the media fragment time unit and byte ranges, and extracts the corresponding bytes.

4. Once the mapping is done, the server then wraps both the header information and the bytes requested in a multi-part *HTTP 206 Partial Content response*. As depicted in Figure 5.5, the first part contains the header

data needed for setting up the decoding pipeline, whereas subsequent parts contain the requested bytes of the needed fragment. Note that a new header, named '`Content-Range-Mapping`', is introduced to provide the exact mapping of the retrieved byte range corresponding to the original '`Content-Range`' request expressed with a time-unit. The decoder might need extra data, before the beginning and/or after the end of the requested sequence, since this initial period might not correlate to a random access unit of the clip to start/end with. In analogy with the '`Content-Range`' header, the '`Content-Range-Mapping`' header also adds the instance-length after the slash-character '`/`', thus providing context information regarding the parent resource in case the HTTP Range request contained a temporal dimension. More specifically, the header contains the start and end time of the parent resource, so the user agent is able to understand and visualise the temporal context of the media fragment.

5. Finally, the user agent receives the multi-part byte ranges and is able to setup the decoding pipeline (using the header information), decodes, and starts playing back the media fragment requested.

We have successfully developed a plugin named *Media Fragments 1.0* for the Firefox browser[5] that has implemented this recipe and is now available in the Mozilla add-ons library.

5.4.2.2 Server mapped Byte Ranges with Initialised Client

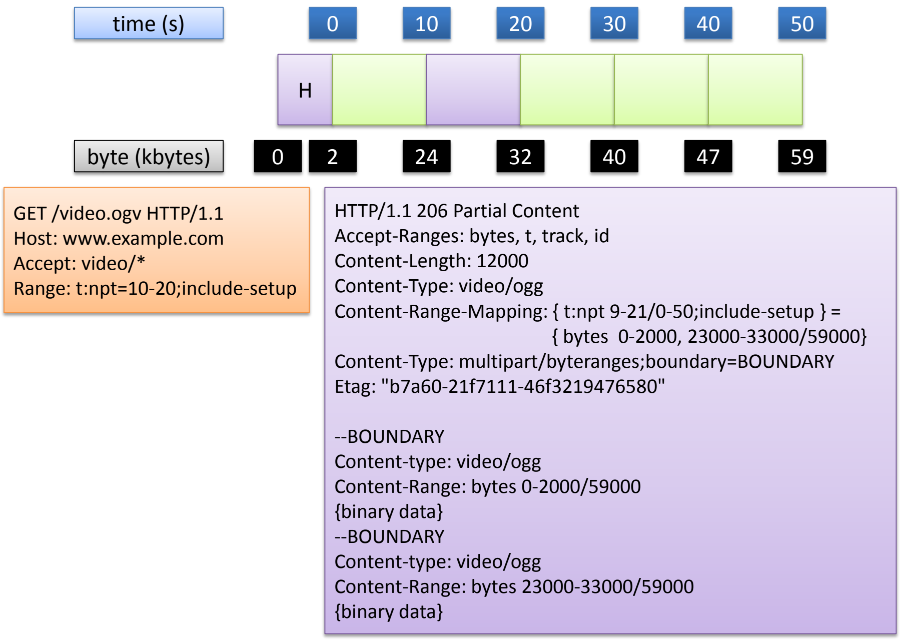
On the other hand, if the server can assume that the client already initiated the decoding pipeline, i.e., it knows about the header of the requested media resource, and no local copy is buffered on client side, then the following events occur:

1. The user agent parses the media fragment identifier and creates an *HTTP Range request* expressed in a different unit than bytes, e.g., a time unit expressed in '*npt*' seconds (Figure 5.6).

2. The server, which understands this time unit, resolves the *HTTP Range request* and performs the mapping between the media fragment time unit and byte ranges.

3. Once the mapping is done, the server extracts the proper bytes and wraps them within an *HTTP 206 Partial Content response*.

[5] `http://www.mozilla.com/`



```
GET /video.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
Range: t:npt=10-20;include-setup
```

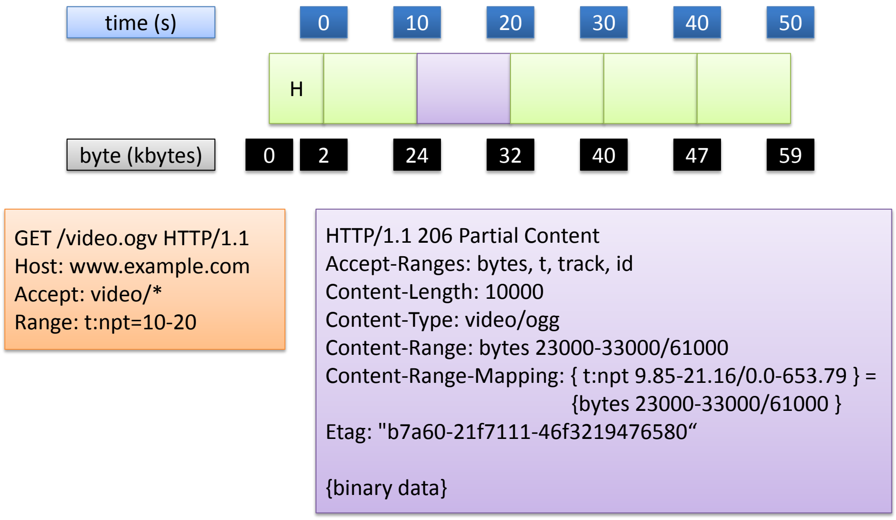
```
HTTP/1.1 206 Partial Content
Accept-Ranges: bytes, t, track, id
Content-Length: 12000
Content-Type: video/ogg
Content-Range-Mapping: { t:npt 9-21/0-50;include-setup } =
                        { bytes  0-2000, 23000-33000/59000}
Content-Type: multipart/byteranges;boundary=BOUNDARY
Etag: "b7a60-21f7111-46f3219476580"

--BOUNDARY
Content-type: video/ogg
Content-Range: bytes 0-2000/59000
{binary data}
--BOUNDARY
Content-type: video/ogg
Content-Range: bytes 23000-33000/59000
{binary data}
```

Figure 5.5: Server mapped Byte Ranges including Header Information.

As displayed in Figure 5.6, the response contains the additional '`Content-Range-Mapping`' header describing the content range in terms of time (ending with the total duration of the complete resource). Where multiple byte ranges are required to satisfy the *HTTP Range request*, these are transmitted as a multi-part message body (with media type '`multipart/byteranges`'), as can be seen in Figure 5.7 –note that for the sake of simplicity only 2 byte ranges are sent back, whereas interleaving normally means a lot more data chunks for one track.

4. Finally, the user agent receives these byte ranges and is able to decode and start playing back the media fragment requested.

Also note that if the server does not understand the *HTTP Range request* –which means it does not support media fragments–, it must ignore that header field –as in sync with the current HTTP specification– and deliver the complete resource. This also means we can combine both byte range and fragment range headers in one request, since the server will only react to the *HTTP Range header* it understands.



```
GET /video.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
Range: t:npt=10-20
```

```
HTTP/1.1 206 Partial Content
Accept-Ranges: bytes, t, track, id
Content-Length: 10000
Content-Type: video/ogg
Content-Range: bytes 23000-33000/61000
Content-Range-Mapping: { t:npt 9.85-21.16/0.0-653.79 } =
                        {bytes 23000-33000/61000 }
Etag: "b7a60-21f7111-46f3219476580“

{binary data}
```

Figure 5.6: Server mapped Byte Ranges with initialised Client.

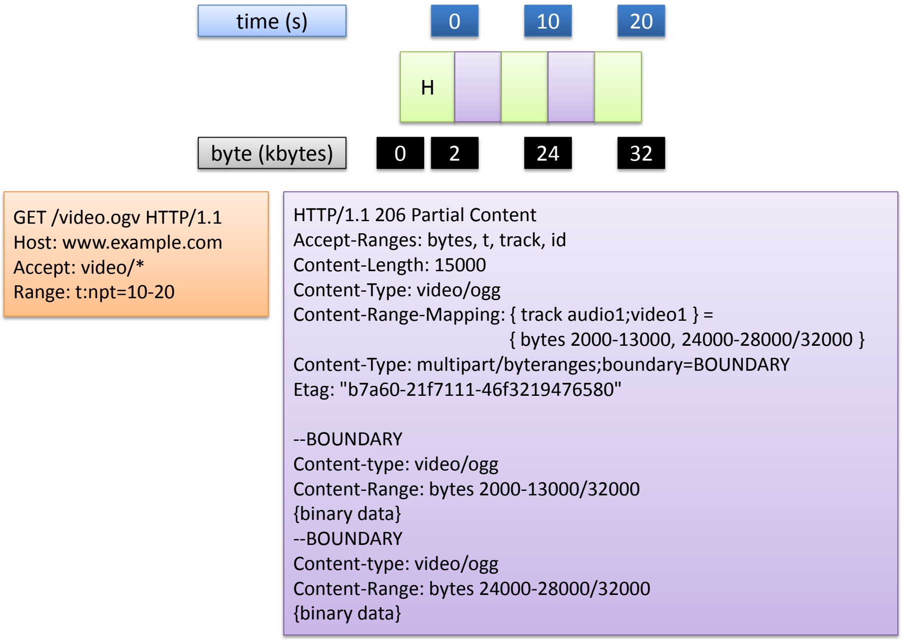
5.4.3 Proxy cacheable Server mapped Byte Ranges

To prevent wasting unnecessary bandwidth and to maximise throughput speeds, the current Internet architecture heavily relies on caching Web proxies. In the case of Media Fragments URIs that are sometimes resolved by servers as described in the previous section, existing Web proxies have no means of caching these requests as they only understand byte ranges.

In order to make use of the Web proxies, the user agent should therefore only ask for byte ranges. To be able to do so, we foresee an extra communication between the server and the user agent. In a first step, the server will just redirect the original request giving extra hints of the byte ranges to the request corresponding to the media fragment, instead of replying with the actual data. In a second step, the user agent will re-issue another range request, this time expressed in bytes which will therefore be cacheable for current Web proxies.

The chain of events for getting 10 seconds of Steve’s non-buffered basketball footage is then as follows:

1. The user agent parses the media fragment identifier and creates an *HTTP Range request* expressing the need for 10 seconds of the video track and further requests to retrieve just the mapping to byte ranges, hence



```
GET /video.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
Range: t:npt=10-20
```

```
HTTP/1.1 206 Partial Content
Accept-Ranges: bytes, t, track, id
Content-Length: 15000
Content-Type: video/ogg
Content-Range-Mapping: { track audio1;video1 } =
                        { bytes 2000-13000, 24000-28000/32000 }
Content-Type: multipart/byteranges;boundary=BOUNDARY
Etag: "b7a60-21f7111-46f3219476580"

--BOUNDARY
Content-type: video/ogg
Content-Range: bytes 2000-13000/32000
{binary data}
--BOUNDARY
Content-type: video/ogg
Content-Range: bytes 24000-28000/32000
{binary data}
```

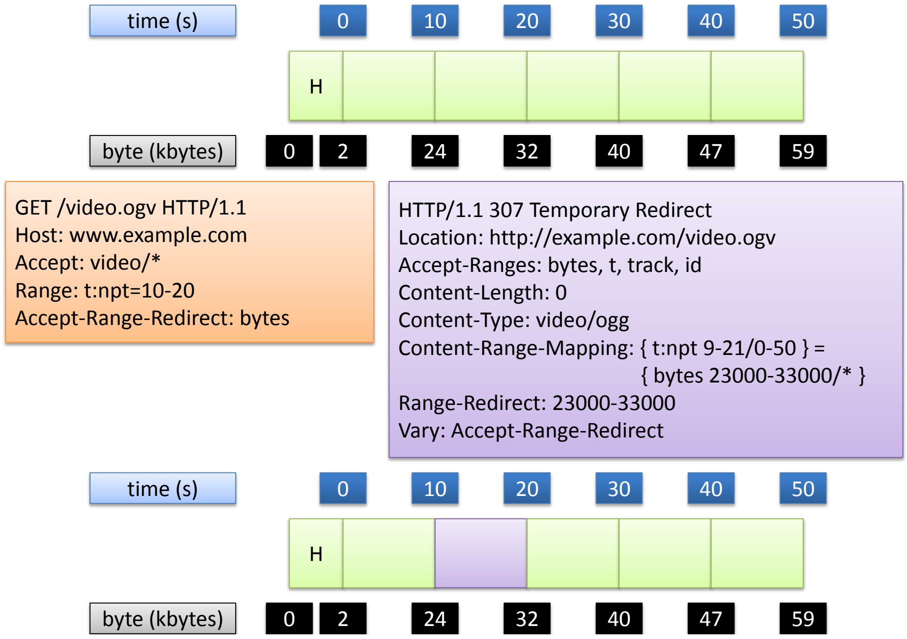
Figure 5.7: Server mapped Byte Ranges with multiple Byte Ranges Response.

the extra HTTP '`Accept-Range-Redirect`' header (see also Figure 5.8), which signals to the server that only a redirect to the correct byte ranges is necessary and the result should be delivered in the *HTTP Range-Redirect header*.

2. The server resolves the *HTTP Range request*, knows it only has to look for the correct byte ranges, and performs the mapping between the media fragment time unit and byte ranges.

3. Afterwards the server sends back an empty *HTTP 307 Temporary Redirect* that refers the user agent to the right byte range(s) for the previously requested correct time range.

4. After this initial indirection, the steps to follow by the user agent are the same as those we have discussed earlier in Section 5.4.1.



```
GET /video.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
Range: t:npt=10-20
Accept-Range-Redirect: bytes
```

```
HTTP/1.1 307 Temporary Redirect
Location: http://example.com/video.ogv
Accept-Ranges: bytes, t, track, id
Content-Length: 0
Content-Type: video/ogg
Content-Range-Mapping: { t:npt 9-21/0-50 } =
                        { bytes 23000-33000/* }
Range-Redirect: 23000-33000
Vary: Accept-Range-Redirect
```

Figure 5.8: Proxy cacheable Server mapped Byte Ranges.

5.4.4 Server triggered Redirect

The server can decide not to serve the requested media fragment in terms of byte ranges in particular cases where too many byte ranges would need to be extracted to satisfy the range request, i.e., because a track media fragment results in an awful lot of byte ranges which is often the case when using an interleaved file format. In this particular case, the server redirects the user agent to an alternate representation of this fragment, e.g., by transforming the Media Fragments URI into a Media Fragments Query URI.

In the scenario (b) described in the Section 5.1, Sarah wants to finalise the headline of today's 'Evening News'. Therefore, she scrolls through the video footage she has gathered, in order to find the appropriate final scene. Since she is only interested in finding the right frames to start editing the high quality footage, she would like to watch only 10 seconds of the video track in low-resolution clips, which can be translated into the following Media Fragments URI:

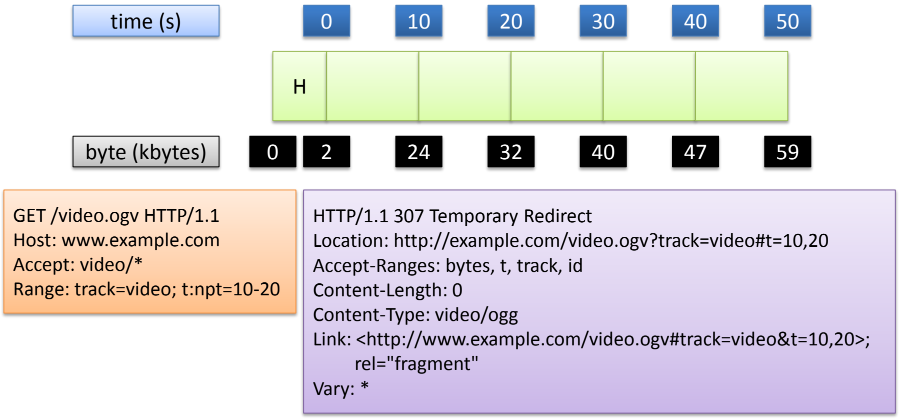
```
http://example.com/video.ogv#t=10,20&track=video
```

The chain of events for getting 10 seconds of the non-buffered video track of this footage is as follows:

1. The user agent parses the media fragment identifier and creates an *HTTP Range request* expressed in a different unit than bytes, e.g., a time unit expressed in '*npt*' seconds also stating it is only interested in the video track.

2. The server decides not to serve the requested media fragment in terms of byte ranges and redirects the user agent to an alternate representation. In this case, the server decides to handle the track fragment through URI Query resolution and the temporal fragment through an URI Fragment resolution. Further an *HTTP Link header* is added stating that the redirected location is a fragment of the originally requested resource, thus maintaining a direct link to that original resource, as can be seen in Figure 5.9.

3. Finally, the user agent follows the redirect, which in this case corresponds to the process specified in Section 5.4.5 hereafter for the track fragment, combined with the process specified in aforementioned Section 5.4.2 for the temporal fragment.



```
GET /video.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
Range: track=video; t:npt=10-20
```

```
HTTP/1.1 307 Temporary Redirect
Location: http://example.com/video.ogv?track=video#t=10,20
Accept-Ranges: bytes, t, track, id
Content-Length: 0
Content-Type: video/ogg
Link: <http://www.example.com/video.ogv#track=video&t=10,20>;
      rel="fragment"
Vary: *
```

Figure 5.9: Server triggered Redirect.

5.4.5 URI Query Resolution

As described in Section 5.3.3, a media fragment can also be resolved and delivered via an URI query, which can then be translated into the following Media Fragments URI:

```
http://example.com/video.ogv?t=10,20
```

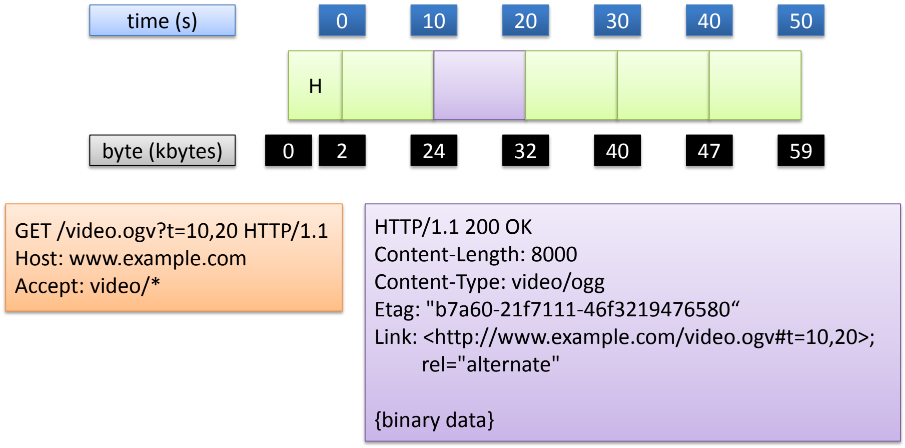
As this will deliver a new full resource, it is a simple HTTP retrieval process where the following chain of events occur:

1. The user agent has to check if a local copy of the requested resource is available in its buffer. If yes, it does a *conditional HTTP GET* with, e.g., an *If-Modified-Since* and *If-None-Match HTTP header*. Assuming the resource has not been retrieved before, the following simple *HTTP GET request*, as can be seen in Figure 5.10, is sent to the server.

2. The server has to create a complete new media resource for the URI query, which in the case of Ogg requires creation of a new resource by adapting the existing Ogg file headers and combining them with the extracted byte range that relates to the given fragment. Some of the codec data may also need to be re-encoded since, e.g., ‘t=10’ does not fall precisely on a decoding boundary, but the retrieved resource must match as closely as possible the initial URI query. This new resource is sent back as binary data within the reply.

3. Finally, the user agent is able to serve the decoded new resource to the user.

Note that if the server does not understand these query parameters, it typically ignores them and returns the complete resource. Furthermore, an *HTTP Link header* may be provided by the server indicating the relationship between the requested URI query and the original media fragment URI. This enables the user agent to retrieve further information about the original resource, such as its full length. In this case, the user agent is also able to display the dimensions of the primary resource or the ones created by the query. Also caching in Web proxies works as it has always worked –most modern Web servers and user agents implement a caching strategy for URIs that contain a query using one of the three methods for marking freshness: heuristic freshness analysis, the *HTTP Cache-Control header*, or the *HTTP Expires header*. In this case, many copies of different segments of the original resource video.ogv may end up in proxy caches. In the future an intelligent media proxy may devise a strategy to buffer such resources in a more efficient manner, where headers and



```
GET /video.ogv?t=10,20 HTTP/1.1
Host: www.example.com
Accept: video/*
```

```
HTTP/1.1 200 OK
Content-Length: 8000
Content-Type: video/ogg
Etag: "b7a60-21f7111-46f3219476580“
Link: <http://www.example.com/video.ogv#t=10,20>;
      rel="alternate"

{binary data}
```

Figure 5.10: URI Query Resolution.

byte ranges are stored differently. Furthermore, Media Fragments URI queries can be extended to enable user agents to use the *HTTP Range-Redirect header* to also revert back to a byte range request. This is analogous to Section 5.4.3. As a final note, a server that does not support media fragments through either URI fragment or URI query addressing will return the full resource in either case. It is therefore not possible to first try URI fragment addressing and when that fails to try URI query addressing.

5.5 Media Formats’ Influence on Fragment Extraction

The extraction of media fragments is dependent on the underlying media format. Moreover, media formats introduce restrictions regarding possibilities to extract media fragments along different axes. For instance, very few coding formats support the extraction of spatial fragments without the need to re-encode the media resource. As such, depending on the fragment axis, there are different requirements regarding the extraction of media fragments (i.e., obtaining media fragments without transcoding operations).

- *Temporal*: random access points need to occur in a regular way. Random access refers to the ability of a decoder to start decoding at a point in a media resource other than at the beginning and to recover an exact or approximate representation of the decoded resource [62]. In case of video, random access points typically correspond to intra-coded pictures.

- *Spatial*: independently coded spatial regions are required. More specifically, (mostly rectangular) regions of the picture need to be coded independently from each other. Typically, ROIs and a background are coded independently, which results in the possibility to extract these ROIs. Another, more advanced approach is 'interactive' ROI. Interactive ROI extraction can be used in applications in which the ROI cannot be defined during the encoding phase of the video sequence [46]. In order to support interactive (rectangular) ROI scalability, a video frame has to be divided into different tiles that can be selected on an individual basis. Each tile has to be coded such that the tiles can be decoded independently of other tiles. This way, the tiles belonging to a certain ROI can be selected on-the-fly during the extraction process. Scalable Video Coding [150] (SVC) is an example of a video coding format supporting such interactive ROI coding.

- *Track*: the way tracks are extracted from a media resource is dependent on the container format. In contrast to temporal and spatial fragment extraction, tracks are not 'encoded' within a container format and can thus always be extracted without low-level transcoding operations. Based on the headers of the container format, it is possible to locate the proper chunks corresponding to the desired track.

- *Named*: whether named fragments are supported or not depends again on the container format. Typically, named fragments are supported through container formats by using other metadata formats describing these named fragments. An instance of such a metadata format can, for example, correspond to a separate track or can be included in the headers of the container format. Examples of such metadata formats are MPEG-4 TimedText [93] and Rich Open multi-track media Exposition[6] (ROE).

In the following subsections, we provide more information regarding the extraction of media fragments applied to three frequently used media format combinations:

- H.264/AVC-encoded video [98] and Advanced Audio Coding (AAC) encoded audio [92] packed in an MP4 container;

- Ogg Theora-encoded video [184] and Ogg Vorbis-encoded audio [185] packed in an Ogg container.

[6] `http://wiki.xiph.org/ROE/`

- VP8-encoded video [161] and Vorbis-encoded audio packed in a WebM container.

For each combination, we discuss how media fragments along different axes can be extracted, based on the above defined requirements.

5.5.1 H.264/AVC and AAC in MP4

MPEG-4 Part 14 or MP4 file format, formally known as International Organisation for Standardisation/International Electro technical Commission (ISO/IEC) 14496-14:2003 [88], is a multimedia container format standard specified as a part of MPEG-4. It is able to store digital video and digital audio streams as well as other data, such as subtitles and still images. The MP4 file format is built on top of the ISO Base Media File Format [91] (known as MPEG-4 part 12), which is in its turn based on Apple's QuickTime file format[7]. H.264/AVC [98] is the latest block-oriented motion-compensation-based codec standard developed by the Joint Video Team[8] (JVT) and is used in applications, such as Blu-ray Disc, YouTube, and the iTunes Store. AAC is a standardised, lossy compression and encoding scheme for digital audio and was designed to be the successor of the popular MP3 format[9]. AAC has been standardised by ISO and IEC, as part of the MPEG-2[10] and MPEG-4[11] specifications and is the standard audio format for Apple's iPhone, Sony's PlayStation 3, Android based phones, etc.

Temporal fragment extraction is both supported by H.264/AVC and AAC. For H.264/AVC, random access points correspond to Instantaneous Decoding Refresh (IDR) frames, which are intra-coded frames that are not referenced by any frames outside the current Group Of Pictures (GOP). Thus, a temporal H.264/AVC fragment always starts with an IDR frame. An AAC audio stream consists of a stream of audio frames, each containing typically 1024 audio samples. Each audio frame corresponds to a random access point, resulting in a fine temporal granularity. The MP4 format provides support for headers containing information regarding random access points of the contained media streams. Hence, by interpreting the MP4 header, temporal fragments can be extracted on condition that random video access points occur in a regular way.

[7] http://www.quicktime.com/

[8] http://www.itu.int/ITU-T/studygroups/com16/jvt/

[9] http://www.iis.fraunhofer.de/EN/bf/amm/products/mp3/index.jsp

[10] http://mpeg.chiariglione.org/standards/mpeg-2/mpeg-2.htm

[11] http://mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm

Spatial fragments extraction is only supported by H.264/AVC in a limited way. More specifically, ROI coding can be realised in H.264/AVC using the Flexible Macroblock Ordering (FMO) tool. By using H.264/AVC FMO type 2, rectangular regions can be coded independently from each other (i.e., each region corresponds to a slice group). Extracting a particular ROI corresponds to the detection of coded P- and B-slices located in non-ROI slice groups and the substitution of these slices with place-holder slices [109].

Tracks are described in the headers of the MP4 format. More specifically, the track box contains general information about the track (e.g., name, creation time, width, height, etc.) as well as references to the location of the bytes representing the track (through the sample table box). Hence, by interpreting these boxes, track fragments can be extracted, based on the names provided by the MP4 container.

Named fragments for MP4 can, for instance, be obtained using MPEG-4 Timed Text, which is the text-based subtitle format for MPEG-4. It was developed in response to the need for a generic method for coding of text as one of the multimedia components within audio-visual presentations. As such, it can be used to give names to temporal fragments.

5.5.2 Theora and Vorbis in Ogg

The Ogg container format is a multimedia container format and the native file and stream format for the Xiph.org multimedia codecs [138]. It is an open format, free for anyone to use, and is able to encapsulate compressed audio and video streams. Ogg is a stream-oriented container, meaning it can be written and read in one pass, making it a natural fit for Internet streaming and use in processing pipelines. Note that this stream orientation is the major design difference over other file-based container formats, such as MP4. Theora is a video codec whose bitstream format was frozen in July 2004 by Xiph.org. This codec is frequently found on the Web and used by large sites complying with the Wikimedia Commons like Wikipedia[12]. Vorbis, also developed by the Xiph.Org Foundation, is an audio format specification for lossy audio compression and has proven popular among supporters of free software. Both Vorbis and Theora contain solely royalty-free and open technology.

Similar to H.264/AVC, random access points in Ogg Theora streams correspond to I-frames. Random access in Ogg Vorbis streams is obtained

[12] http://www.wikipedia.org/

through audio frames, similar to AAC (note that the number of samples is variable in case of Ogg Vorbis). Thus, temporal fragment extraction is both supported by Ogg Theora and Vorbis.

Spatial fragment extraction is not supported by Ogg Theora. More specifically, it does not provide support to encode spatial regions independent of one another.

An Ogg container is a contiguous stream of sequential Ogg pages. Ogg allows for separate tracks to be mixed at page granularity in an Ogg container. Tracks are identified by a unique serial number, which is located in each Ogg page header. Further, to link Ogg tracks to track names, again the ROE format can be used. ROE is a metadata format for describing the relationships between tracks of media in a stream. ROE descriptions are represented in an Ogg container by means of an Ogg Skeleton track. This way, track fragment extraction is supported by Ogg through fields containing the track names stored in Ogg Skeleton.

Named fragment extraction for temporal and track fragments in Ogg can be obtained by combining ROE, Skeleton, and Continuous Media Markup Language[13] (CMML). CMML allows a time-continuously sampled data file to be structured by dividing it into temporal sections (so-called clips) and provides these clips with some additional information. Thus, named fragment extraction can be done based on the CMML and ROE description of an Ogg container.

5.5.3 VP8 and Vorbis in WebM

The WebM container format, which is introduced by Google, is an open and royalty-free multimedia container format designed for the Web. WebM is based on the Matroska[14] media container format. It uses Vorbis for the representation of audio streams while VP8 [161] is used as video codec. VP8, released by Google –after acquiring On2 Technologies[15]–, is the latest open source video codec. Compression-wise, VP8 generally performs better than Theora.

The constraints implied by WebM for extracting temporal media fragments are very similar to the ones we discussed above for both H.264/AVC and Theora. More specifically, random access points in VP8 streams correspond

[13] http://wiki.xiph.org/CMML/

[14] http://www.matroska.org/

[15] http://www.on2.com/

to I-frames. Temporal fragment extraction is thus supported by VP8. Also, similar to Theora, spatial fragment extraction is not supported by VP8.

Tracks within WebM are identified by a track number and/or a track identifier. The track identifier can be used within a Media Fragment URI using the track axis, while the track number is used internally to indicate for each frame (audio or video) to which track it belongs. This way, we can address a track in a WebM container and also extract the track.

Further, named fragments within WebM containers can be obtained by defining chapters. A chapter in WebM is a combination of a time segment and one or more tracks. The name of the chapter could then be used as an identifier for the named fragment. Note that WebM also provides support for adding tags to chapters. However, these tags cannot always be used as identifier for a named fragment, because they could result in multiple time segments (e.g., two chapters having the same tag). Addressing multiple time segments within one Media Fragment URI is not supported, as discussed in Section 5.3.4.1.

5.6 Open Issues and Future Work

Currently, it is application dependent for all the specified media fragment axes, how their defined media fragments should be rendered to the end-user in a meaningful way. Temporal fragments could be highlighted on a timing bar whereas spatial fragments could be emphasised by means of bounding boxes or they could be played back in colour while the background is played back in grey-scale. Finally, track selection could be done via drop-down boxes or buttons. Whether the Media Fragments URIs should be hidden from the end-user or not is an application implementation issue.

There is currently no standardised way for user agents to discover the available named and track fragments. One could use ROE, which makes it possible to express the track composition of a media resource, in addition to naming these tracks and extra metadata info on language, role and content-type which could further help selecting the right tracks. Another candidate to use is the *Media Multi-track API*[16] from the HTML5 Working Group. This is a JavaScript API for HTML5 media elements that allows content authors to determine which data is available from a media resource. Not only does it expose the tracks that a media resource contains, but it also specifies the type

[16] http://www.w3.org/WAI/PF/HTML/wiki/Media_MultitrackAPI/

of data that is encapsulated within the resource –e.g., audio/vorbis, text/srt, video/theora, etc.–, the role this data plays –e.g., audio description, caption, sign language, etc.–, and the actual language –e.g., RFC3066[17] language code.

With such media fragments addressing schemes available, there is still a need to hook up the addressing with the actual bytes of the resource. For the temporal and the spatial dimension, resolving the addressing into actual byte ranges is relatively obvious across any media type. However, track addressing and named addressing need to be resolved too. Track addressing will become easier when we solve the above stated requirement of exposing the track structure of a media resource. The name definition, on the other hand, will require association of an *id* or *name* with temporal offsets, spatial areas, or tracks.

Finally, hyperlinking out of media resources is something that is not generally supported at this stage. Certainly, some types of media resources –QuickTime, Flash[18], MPEG-4, and Ogg– support the definition of tracks that can contain HTML marked-up text and thus can also contain hyperlinks. It seems to be clear that hyperlinks out of media files will come from some type of textual track. On 6 May 2010, the WHATWG version of the HTML5 draft specification introduced the Web Subtitle Resource Tracks format (WebSRT), a format intended for marking up timed track resources. WebSRT will have a means of addressing segments of a media resource by name and also a means to include text with hyperlinks, though the exact scope of WebSRT and its inclusion as part of HTML5 is still under debate.

5.7 Conclusions and Original Contributions

We have presented the rationale for a Media Fragment URIs specification to make video a 'first-class citizen' on the Web and pinpointed the drawbacks/inconsistencies of existing solutions. We outlined the boundaries, requirements and semantics of a Media Fragment URI. We clearly defined the syntax of Media Fragments over the different possible fragment dimensions (i.e., temporal, spatial, track, and named) and showed how the defined Media Fragments can be resolved stepwise using the HTTP protocol. We then identified the influence of current media formats on such Media Fragments extraction.

[17] `http://www.ietf.org/rfc/rfc3066.txt`

[18] `http://www.flash.com/`

We were the first to have developed an HTTP implementation for the W3C Media Fragments 1.0 specification [170] (both a Firefox client-side plugin and a Ninsuna[19] server-side implementation) in order to verify all the test cases defined by our working group. Furthermore, Opera already has implemented the time recipe in its new alpha version and it is to be expected that all major browser vendors will include support for the Media Fragments time recipe in its upcoming versions. Also HTML5 already foresees support for this time recipe within its media elements. In the near future, we also foresee reference implementations for the RTSP, Real Time Messaging Protocol[20] (RTMP), and File protocols.

In the meantime this W3C Media Fragments URI specification, which we edited and contributed to over the last couple of years, now really opens up time-related media to the Internet crowd and makes time-related annotation [117, 119] feasible for hyperlinking into the media, thus providing the necessary support for this already omnipresent 'third dimension' *time* into the Internet. Point by point my own research contributions can thus be summarised as follows:

- Co-chair and main contributor of W3Cs Media Fragments Working Group for the Media Media Fragments 1.0 specification.
- First to have developed an HTTP implementation for the W3C Media Fragments 1.0 specification (both a Firefox client-side plugin and a Ninsuna server-side implementation).

The research that has led to this chapter is also described in the following publications:

1. Editor of Working Draft on 'Media Fragments URI 1.0', June 2010, see `http://www.w3.org/TR/media-frags/`
2. Editor of Working Draft on 'Use Cases and Requirements for Media Fragments', March 2010, see `http://www.w3.org/TR/media-frags-reqs/`
3. E. Mannens, D. Van Deursen, S. Pfeiffer, C. Parker, R. Troncy, Y. Lafon, J. Janssen, M. Hausenblas, and R. Van de Walle. Universally Addressing Media Fragments. *Multimedia Tools and Applications – Special Issue on Multimedia Data Semantics*, Online First (DOI: 10.1007/s11042-010-0683-z), Springer-Verlag, December, 2010

[19] `http://ninsuna.elis.ugent.be/`

[20] Adobe's Real Time Messaging Protocol available at `http://www.adobe.com/devnet/rtmp/`

4. E. Mannens, S. Park, J. Soderberg, G. Adams, P. Le Hegaret, R. Van de Walle, and C. Seon Hong. Video in the Web: Technical Challenges and Accompanying Standardisation Activities. *IEEE Multimedia*, volume 17, issue 4, pages 90–93, IEEE Computer Society, October, 2010

5. E. Mannens, D. Van Deursen, and R. Van de Walle. Making Space and Time Uniformly Identifiable in the Web via Media Fragments. In *the Proceedings of the 8th IEEE Consumer Communications and Networking Conference – 1st International Workshop on Semantics to Enable Convergence for Consumer Communications and Applications*, pages 1023–1028, January 2011, Las Vegas, USA

6. E. Mannens, R. Troncy, J. Sendor, and D. Van Deursen. Implementing W3Cs Media Fragments URI Specification. In *the Proceedings of the Open Video Conference*, pages 1–1, October 2010, New York, USA

7. W. Van Lancker, D. Van Deursen, E. Mannens, and R. Van de Walle. Implementation Strategies for Efficient Media Fragment Retrieval. *Multimedia Tools and Applications – Special Issue on Recent Advances and Future Directions in Multimedia and Mobile Computing*, Accepted for publication, Springer-Verlag, March, 2011

8. D. Van Deursen, W. Van Lancker, E. Mannens, and R. Van de Walle. NinSuna: Metadata-driven Media Delivery. In *the Proceedings of the 3th International Service Wave Conference*, pages 1–2, December 2010, Ghent, Belgium

9. D. Deursen, W. Van Lancker, E. Mannens, and R. Van de Walle. NinSuna: a Server-side W3C Media Fragments Implementation. In *the Proceedings of the 11th IEEE International Conference on Multimedia & Expo*, pages 270–271, July 2010, Singapore, Singapore

10. D. Van Deursen, R. Troncy, E. Mannens, S.Pfeiffer, Y. Lafon, and R. Van de Walle. Implementing the Media Fragments URI Specification. In *the Proceedings of the 19th International World Wide Web Conference*, pages 1361–1364, April 2010, Raleigh, USA

Chapter 6

Proof of Concepts

I get very tense around apples ... Well, I get very tense generally.
I think I've fallen into the trap of blaming fruit.

Jeff Murdock in Coupling

6.1 Introduction

All what is described in previous chapters has (in some form or another) been put to the test in (at least) a proof-of-concept environment. Most of these were (partly) realised in research projects (see Appendix E for the projects I contributed to). To put things into perspective, I here broadly discuss an implemented use case per chapter.

- *The search use case*: Efficient media search applications can highly improve productivity in various domains. However, an important requirement for efficient media retrieval is the availability of high quality metadata documenting the archived media. This is also the case within a news production environment, where professional archive users spend considerable amounts of time searching the media archive in order to find useful media for reuse in news broadcasts. However, in a news production environment, where it is important to produce and distribute news as soon as possible to as many channels as possible, in an audiovisual quality as good as possible, metadata generation is often reduced to an absolute minimum. Consequently, dedicated archivists are responsible for the generation of high quality metadata as a last step in the news production chain in order to facilitate efficient media retrieval. Therefore, part of the PISA-project investigated how news-related audio and audio-visual content can be found in a more effective, efficient, and easy way. One of the goals of this research project was to increase the amount of quality metadata by capturing and structuring the available information during the news production by using a refined news data model originating from the concepts defined in Chapter 2. This automatically generated metadata can then be used by archivists as a starting point for further enrichment, leading to more accurate search results, which improves the overall efficiency and productivity of news programme editors.

- *The distribution use case*: While the initial impact of file-based production indeed mainly affected the work methods of the news production staff –journalists, anchors, editorial staff, etc–, the added-value for the end-user was still marginal, i.e., he might notice that news content is made available faster. Therefore, another part of the PISA-project & part of the CUPID-project investigated how the production of multiple versions of news bulletins for different consumption platforms can be automated, since news broadcasters generally aggregate and produce more material than is required for broadcast or on-line distribution. Furthermore, we aimed to demonstrate the possibility of dynamically digesting an up-to-date news bulletin by merging different news sources, assembled to match the *real* individual consumers' likings, by recommending his/her favourite topics. In order to do that, we used the Semantic Web technologies (LOD, profiling & recommendation), as discussed in Chapter 3 together with our own multimedia adaptation and distribution Ninsuna[1] platform.

- *The archiving use case*: The rapid rise of digital news & culture networks has a fundamental influence on the construction of our personal and social memory. The technologies used today to register, organise, find and share information have thoroughly changed our relation to past and present, but also the dynamics between remembering and forgetting. Consequently, the role and function of traditional 'memory' institutions such as the museum, the library and the broadcaster's news archive call for a reconsideration. Both the BOM-Vl-project and the Archipel-project try to find lasting preservation and dissemination strategies for diverse digital multimedia content. As such, a sustainable digital archive infrastructure is being elaborated on in Flanders in order to ensure a structural approach (a.o., primarily based on our work in Chapter 4) to the problem of sustainable digital archiving & dissemination.

[1] `http://ninsuna.elis.ugent.be/`

- *The Media Fragments use case*: The goal of the 'Video in the Web' activity is to make video a 'first class citizen' on the Web. In this context, video in the Web does not only include video, but also audio and still images. Because of the explosive growth of media on the Web, challenges such as content discovery, searching, indexing, and accessibility are appearing. Enabling users (from individuals to large organisations) to put media on the Web requires the development of a solid architectural foundation that enables people to create, navigate, search, link, consume, and distribute media resources. This way, media resources are effectively part of the Web instead of an extension that does not take full advantage of the Web architecture. The mission of W3C's MFWG is to address media fragments on the Web using URIs. Having global identifiers for arbitrary media fragments would allow substantial benefits, including linking, bookmarking, caching, and indexing. As we are actively participating in the standardisation of W3C's Media Fragments URI specification, we were the first to come up with a reference implementation. We hereafter show how our NinSuna multimedia content delivery platform can be used as a server-side implementation of W3C's Media Fragment URIs, which enables addressing media fragments on the Web using URIs.

6.2 The Search Use Case

6.2.1 Distributed News Production Process Information

As can be deducted from the discussion of the news production processes elaborated over in Chapter 2, (meta)data is generated and spread across the entire news production workflow chain. Unfortunately, during a media search operation within VRT, currently only information generated by an archivist at the end of this chain is used. Other, possibly valuable information, is not used at all when searching for relevant media. We identified the following information sources containing additional valuable information for media search within VRT's end-to-end news production workflow chain:

- *Ardome*: Ardome[2] is the central MAM system used by VRT, containing all produced news media. During ingest, metadata such as title, episode number, descriptive information of the audio tracks, aspect ratio, and video format can be inserted. However, this manual metadata insertion

[2] http://vizrt.com/products/article138.ece

is always limited due to time constraints. Once the media has been documented in *Basis*, metadata is copied from Basis to Ardome.

- *News Agency provided metadata*: Incoming media from news agencies, such as EBU Eurovision and Reuters, is accompanied by metadata documenting the media in the NewsML-G2 format. This metadata can be inserted into the MAM during ingest. Typical metadata contained in the NewsML-G2 document are titles and textual descriptions (in English) of the media content.

- *iNEWS*: iNEWS[3] contains important information as it is VRT's main tool for managing news productions. It is used to create the entire rundown of a news broadcast. Every news item from the rundown can be provided with anchor text, and open subtitles that must appear on the screen during broadcast by the *Character Generator* (CG).

- *Swift*: Closed subtitles are generated using the Swift[4] tool. In addition to the subtitle text, Swift contains subtitle layout, spoken language indication, and timestamps indicating the (dis)appearance time of a subtitle.

- *Basis*: Basis is considered to be the main tool for annotating and describing media resources at VRT. It has a relational database structure and currently contains over 700 000 records documenting media fragments spanning a period of over 20 years. A Basis record defines metadata fields such as title, duration, keywords, textual description, journalist, and identification number of the corresponding media in the MAM. Some fields, e.g., the keywords and journalist fields, can only contain controlled values, defined by a manually maintained thesaurus. The keywords thesaurus is the largest thesaurus containing over 300 000 terms. In addition to defining terms, relationships between terms are defined indicating '*narrower than*', '*related*' and '*used for*' relationships. Unfortunately, keywords are not categorised in terms of types, such as persons or locations.

By unifying all the information generated during news production, more efficient search applications can be realised. Also, by capturing information generated during news production, metadata can be generated and already filled in when creating an archival record, allowing an archivist to focus on the further enrichment of this metadata.

[3] http://www.avid.com/US/products/inews/

[4] http://www.softelgroup.com/

Subtitle information can significantly improve the media search operation [25, 43]. For example, when a certain keyword appears in a subtitle, time-stamp information can be used to start playback from the indicated time-stamp. Closed subtitles, generated with the Swift tool, are provided with timing information. However, the appearance and disappearance timestamps for open subtitles are not available, as journalists use the *Belle Nuit*[5] tool only to paste the subtitles onto the relevant frames and no related timing information is stored during this operation. However, this timing information can be obtained after media production through the use of speech analysis or Optical Character Recognition (OCR) tools.

Timing information indicating the appearance time of CG text can also improve media search operations. For example, when a user searches for media related to a person, timing information related to the CG text containing the name of this person can be used to start playback at this point. However, the exact moment CG text is displayed, is decided during live broadcast and this information is currently not stored. This timing information could be reconstructed through the application of feature extraction tools on the generated media.

It is clear that although much of the data generated during the news production process can be used for media search operations, not all this information is stored, and therefore introduces the need for post-production operations to reconstruct this information. In order to avoid the need for these post-production operations, this information should be stored.

In addition to the fact that not all information is stored, the current news production process does also not guarantee the preservation of inserted metadata. For example, when a news item from a news agency is ingested, an insertion of metadata provided in the accompanying NewsML-G2 document is also performed into the MAM. However, when an archival record is generated in Basis documenting the archived media, an update operation is performed which overwrites the original metadata inserted into the MAM. This introduces the loss of the metadata obtained from the NewsML-G2 document although this metadata can be important when searching for media and should therefore be preserved too.

[5] http://www.belle-nuit.com/subtitler/index.html

6.2.2 Data model of the News Production Information Sources

In order to represent the information present in the various identified information sources in a uniform way, we propose a data model specifically designed to preserve metadata along the news production process. The data model is based on the FIPA/PISA data model [178] (see Appendix E.20/ E.12). Note that this model is also compatible with the P/Meta data model [51] defined by the EBU and shares business objects with BBC's[6] Standard Media Exchange Framework (SMEF) model [23]. The core concepts of our data model are illustrated in Figure 6.1.

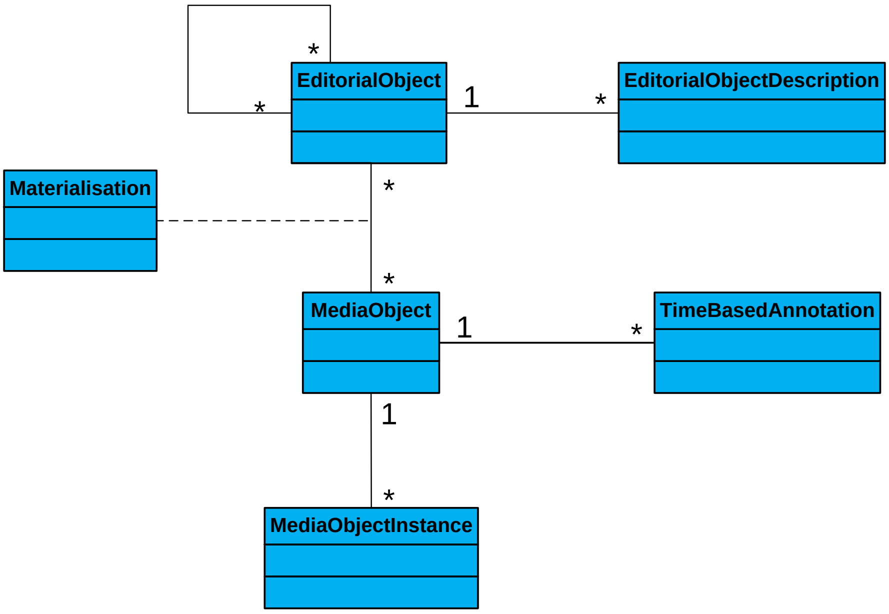


Figure 6.1: Core Concepts of the PISA Media Production Model.

The data model is centered around four elemental types of processes that contribute to media production [178]. The life cycle of an entire product (e.g, a news broadcast) starts with the product planning process. During product planning, initial product requirements are specified. Because this information is typically delivered from an external ERP system, the data model only represents an abstract notion of this process. The product planning process will, however, prepare a number of business objects for elaboration during the subsequent production processes.

[6] British Broadcasting Corporation, see http://www.bbc.co.uk/

The first of these processes is the PE process, in which the content of the product is determined by the editorial staff based on the initial product specifications. After the PE process, the product is defined as a composition of logically and editorially constituent parts, whereby each logical unit of creative or editorial work is represented by an editorial object. In the following manufacturing engineering process, information related to the manufacturing of a product is specified and represented in manufacturing objects. Finally, during the manufacturing process, the manufacturing objects are realised in the form of audio-visual material.

The following subsections describe how the news production process is represented using this generic media production (meta)data model. We introduce a number of extensions specific for news production and indicate how the corresponding information sources are mapped onto the model.

6.2.2.1 Product Engineering

An editorial object represents a unit of editorial work defined during the PE phase. For the news production process, we implemented three specialisation subtypes of editorial object: *News*, *NewsStory*, and *NewsItem*. A news story represents a topic that is to be covered during a news broadcast. A story can consist of several news items. A news item is the atomic editorial object for news production and can, e.g., correspond to an interview or a news anchor reading an introductory text from the auto-cue. Information contained in editorial objects is, e.g., the story title, broadcast, date, relevant keywords, etc.

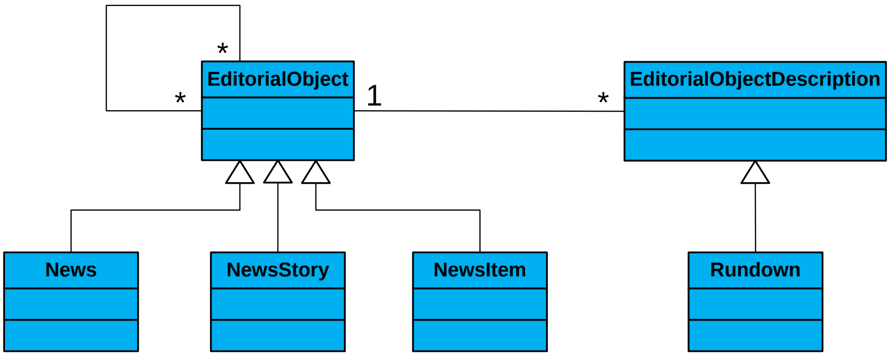


Figure 6.2: The EditorialObject Class.

Figure 6.2 depicts the UML class diagram of the introduced editorial objects. The *EditorialObject* class can be reflexively associated through a many-to-many relationship. As a result, a story can be part of a news broadcast. However, a story does not need to be related to a news editorial object, as some stories are developed without eventually being part of a news broadcast. Note that application logic is responsible for prohibiting the occurrence of invalid relations, such as the fact that a news broadcast cannot be part of a news item.

The *Rundown* class is used to specify the editorial content of a news broadcast, which includes the list of news items to be broadcast, and the order in which these items are scheduled. Hence, a *Rundown* is attached to the *News* editorial object as a *EditorialObjectDescription* subclass. Similarly, anchor texts and other metadata that define the editorial content of the more granular news item are also incorporated into an *EditorialObjectDescription*. The data model supports the representation of iterative versions of *EditorialObjectDescriptions* to model rundowns and news items that change over time. However, in VRT's current news production process, only the final versions are effectively stored.

6.2.2.2 Manufacturing Engineering

Considering the strict deadlines and well-established format of news production, there is little time and need for extended preparations and premeditation concerning the cinematography of news items. As such, the manufacturing engineering layer of the data model is of limited use, since the translation of editorial object semantics to a media object is straightforward and does not require, e.g., the accurate definition of camera positions by means of storyboarding or pre-visualisation, as is often the case with more elaborate media production processes, such as drama production [177].

6.2.2.3 Manufacturing

Due to the lack of manufacturing engineering in current day news production, objects from the PE layer can be related directly to objects in the lowest manufacturing layer. In the manufacturing phase, editorial objects are materialised into (audio-visual) *MediaObjects*. In news production, different versions of a media object can be generated and stored. For example, a HD-version and a downscaled version can be realised, containing the same audio-visual content. Every version is then represented by an instance of the *MediaObjectInstance* class and related to the corresponding *MediaObject*, as illustrated in Figure 6.3.

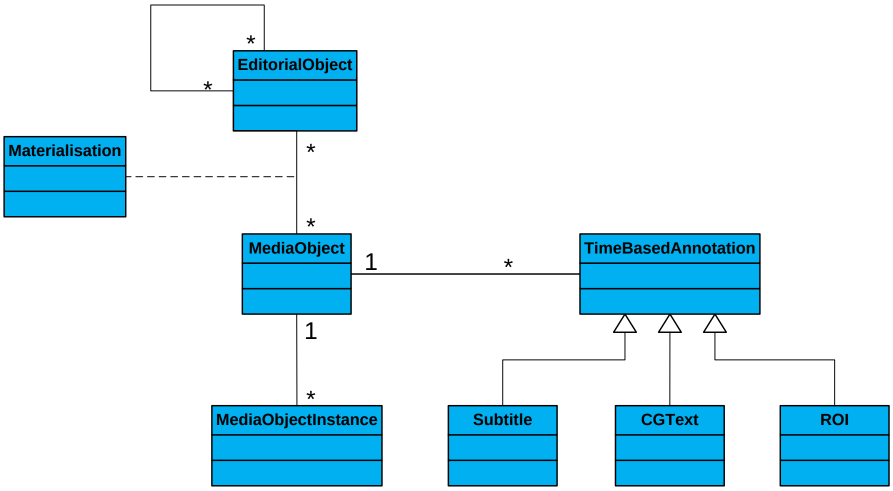


Figure 6.3: The MediaObject Class.

The *Materialisation* class associates an *EditorialObject* with a *MediaObject*, and contains information specifying how the editorial object was manufactured into the media object in question. An example of information that belongs to a materialisation object is the name of the camera operator. The *TimeBasedAnnotation* class provides media objects with time-based annotations. For example, a media object can be annotated with a subtitle or a CG text together with its associated appearance and disappearance time.

6.2.2.4 Data Source Mapping

The information from the selected data sources also needs to be mapped to our proposed data model. Note that we keep the sources of the different kinds of information, since the provenance of this information is crucial for future reuse[7].

- *iNEWS*: iNEWS –as already mentioned– is used as a management tool for news production. Therefore, iNEWS is used for defining instances of the *News*, *NewsStory* and *NewsItem* editorial objects. The order of the news items is defined through an instance of the *Rundown* object. Anchor text belongs to the *NewsItem* editorial object as this is generated during the PE process. Open subtitles and CG text are implemented

[7] http://www.w3.org/2005/Incubator/prov/wiki/Presentations_on_State_of_the_Art/

as instances of the corresponding subclasses of the *TimeBasedAnnotion* class, as this information has associated timing information.

- *Ardome*: Items stored in Ardome are represented through instances of the *MediaObjectInstance* class. Metadata present in Ardome belongs to the corresponding editorial objects that are related to these instances.

- *News Agency provided metadata*: Information taken from the NewsML-G2 documents, such as title and textual descriptions, is added to the corresponding instance of the *NewsItem* editorial object.

- *Swift*: Closed subtitles are represented as instances of the *Subtitle* class.

- *Basis*: Descriptive information present in a Basis record, such as title, description, and keywords, is associated with the corresponding editorial object. Information related to the realisation, such as the camera operator, director, and recording date, is provided as part of the *Materialisation* class. Technical information, such as the video coding format, is provided to the corresponding instance of the *MediaObjectInstance* class. Information that is common for all versions of a materialisation is represented as part of the *MediaObject* class. An example of such information present in the *MediaObject* class is the rights information.

6.2.3 Enabling Semantic Search

Unifying all generated information through the use of this common data model significantly improves the potential of a search application. However, by connecting this information with external data sets, even more intelligent search operations are enabled. The following sections describe the proposed architecture and give an overview of how external data sets are connected. The subsequent sections then illustrate some added functionality that is obtained by following this approach.

6.2.3.1 Search Architecture

The data model was formalised into an OWL ontology (see Appendix D.1) and all relevant information from the selected data sources, discussed in aforementioned Section 6.2.1, was converted into a representation according to this formalised data model and stored in an RDF triple store. The resulting RDF data present in the triple store can then be queried using the SPARQL query language. By formalising the data model and the corresponding information from the information sources, an unambiguous and machine processable representation of the available information is obtained. This in turn enables the use of reasoners, which can derive new facts based on the provided information.

The use of Semantic Web technologies also allows us to connect with other data sources using the LOD principles [11]. This again enables us to use information from external data sets in order to make more intelligent search applications. Therefore, the search application makes use of a query facade which in turn uses data from several data sets according to the LOD principles. The resulting architecture is illustrated in Figure 6.4.

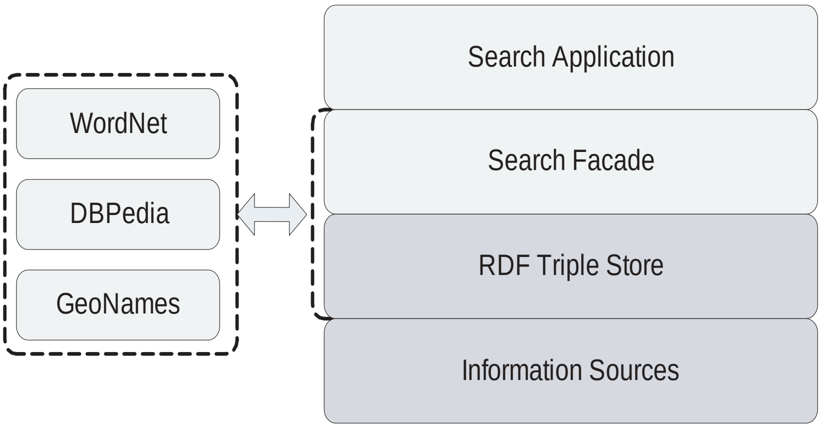


Figure 6.4: Semantic Search Architecture.

6.2.3.2 Connecting with the LOD Cloud

In order to be able to use information formalised in external data sets, concepts defined in our data model must be linked with the corresponding concepts defined in the external data sets. A valuable data set, which is already used by other broadcasters [104] and considered an important linking hub in the LOD cloud, is DBPedia [17]. We use DBPedia to link concepts representing persons and other concepts, except for locations. For locations, again the GeoNames data set is used, providing a formalised representation of geographic locations. As a starting point, we use the *Basis* keywords as interlinking mechanism with the other data sets. As already mentioned, every *Basis* record contains a number of relevant keywords taken from the keywords thesaurus. The keywords can directly be linked to concepts defined in other data sets. For other fields containing textual data such as the *Basis* description field and *Swift* subtitles, Named Entity Recognition [128] (NER) must first be applied. NER is part of future work that will be performed in order to extend the set of relevant entities.

We formalised the keywords thesaurus in SKOS [125]. Consequently, every concept defined in the thesaurus corresponds to an URI. Also, the relationships defined between different concepts are also present in the SKOS representation. The relations defined between concepts in the thesaurus are important in the disambiguation of concepts during mapping. For example, the concept 'apple' defined in the keywords thesaurus has the related concepts 'iMac'[8] and 'Taligent'[9]. These related concepts are then used to select the correct corresponding concept defined in DBPedia (`http://dbpedia.org/resource/Apple_Inc.` instead of `http://dbpedia.org/resource/Apple`).

As already mentioned, locations are mapped to the corresponding concept defined in GeoNames. Again, we make extensive use of the relations defined in the thesaurus. For example, the concept 'Parijs' (Eng. Paris), has a used-for-relationship with the concept 'Paris'. Also, 'Parijs' is defined as a narrower term of 'Frankrijk' (Eng. France). With this added information, it is possible to select the corresponding concept from GeoNames.

However, for many concepts little or even no relationships are defined in the thesaurus. In this case, selecting the corresponding concept from an external data set can be very difficult as there is no additional context present that can be used for disambiguation. For this reason, statistics are currently collected from the *Basis* records, in order to get a set of keywords that most frequently co-occur with another keyword. These keywords will then be used as additional context information in order to select the correct concept from the external data set. Note also that because of the fact that the majority of keywords are defined in Dutch, often an additional translation is needed for successful mapping with concepts from external data sets, e.g., DBPedia. By formalising the keywords thesaurus and connecting it's concepts with external data sets, new functionalities are obtained. Some of these are illustrated in the following sections.

[8] `http://en.wikipedia.org/wiki/Imac/`

[9] `http://en.wikipedia.org/wiki/Taligent/`

6.2.3.3 Suggesting alternative Queries

The result set of a query depends on the user input. When a user enters a general keyword, the result set can be too large. In order to limit this result set, other keywords having a *narrower-than* relationship with the entered keyword can be displayed as a suggestion. On the other hand, when a query results in an empty result set, an alternative keyword can be suggested which has results.

6.2.3.4 Lexical Information

As the thesaurus is currently maintained by hand, it is difficult to include all possible information related to an introduced keyword. Consequently, related information such as abbreviations and synonyms are often not included in the thesaurus. However, this makes it more difficult for a user to find results, as they have to know which words are included in the thesaurus. In order to provide a better search experience, a connection was made with a formalised version of the *Word-Net* lexical database[10]. Note that this data set is also linked with *Cornetto*[11], a lexical semantic database for the Dutch language.

When a user enters a keyword, additional information can then be retrieved and included in the query in order to optimise the search operation. For example, when a user enters ‘populatie’ (Eng. population) the result set of this query would be small as this concept is not included in the thesaurus. However, it can be found in *Cornetto* that the concept ‘populatie’ has similar meaning to ’bevolking’ (Eng. population), which is included in the thesaurus.

6.2.3.5 User Query Evaluation

When a user enters keywords, reasoning can be performed in order to try to find out what a user really searches for. For example, when a user enters the keywords ‘vice president Barack Obama’, it can be derived that the user possibly searches media related to ‘Joe Biden’ but maybe he does not recall his name. Although ‘Joe Biden’ is present as a concept in the thesaurus, no relations are present indicating that ‘Joe Biden’ is the ‘vice president’ for ‘Barack Obama’. In order to be able to show results having ‘Joe Biden’ as a keyword, we use information available in external data sets as follows. When a query with multiple keywords is performed, we try to find a subject (or object) for which a property and a related object (or subject) exists corresponding with the entered keywords as follows. From the entered keywords we retrieve the words

[10] `http://semanticweb.cs.vu.nl/lod/wn30/`

[11] `http://www2.let.vu.nl/oz/cltl/cornetto/`

'Barack Obama' and search for a corresponding concept in the formalised thesaurus having this as a label. This concept is present in the thesaurus and is already linked with the corresponding concept in DBPedia, thus allowing the use of that information available from DBPedia. In the following step, we try to map the other entered keywords with a property occurring in a triple having the concept of 'Barack Obama' either as its subject or object. As such, the property *dbpedia-owl:vicePresident* has as label 'vice president', allowing the identification of the needed property. Then we search for triples with the selected property and where 'Barack Obama' appears either as subject or object. Finally, we obtain the concept *dbpedia:Joe Biden*, which is also linked with the corresponding concept from the keywords thesaurus. This enables the inclusion of media provided with the keyword 'Joe Biden', as the user ultimately wanted.

6.3 The Distribution Use Case

6.3.1 Back-end News Gathering

As already mentioned in previous chapters, the main tool used for news production at VRT is Avid's iNEWS. It is used by directors and editors to create and manage the news rundowns, which consist of a list of items that will be covered during a news broadcast. An item can be a news anchor reading text from the auto-cue in the studio, a report made by a journalist, a live interview, etc.

There are multiple options to obtain content related to a news item, as can be seen in Figure 6.5:

- Audio-visual content captured by a news crew on location (typical for national news).

- Content from incoming wires shot by other news providers (e.g., Reuters, EBU Eurovision, etc.).

- News feeds from news agencies, such as EBU Eurovision and Reuters, containing media files and additional metadata (typically represented in NewsML-G2).

- Reuse of suitable audio-visual material obtained from the archive by a professional archive user.

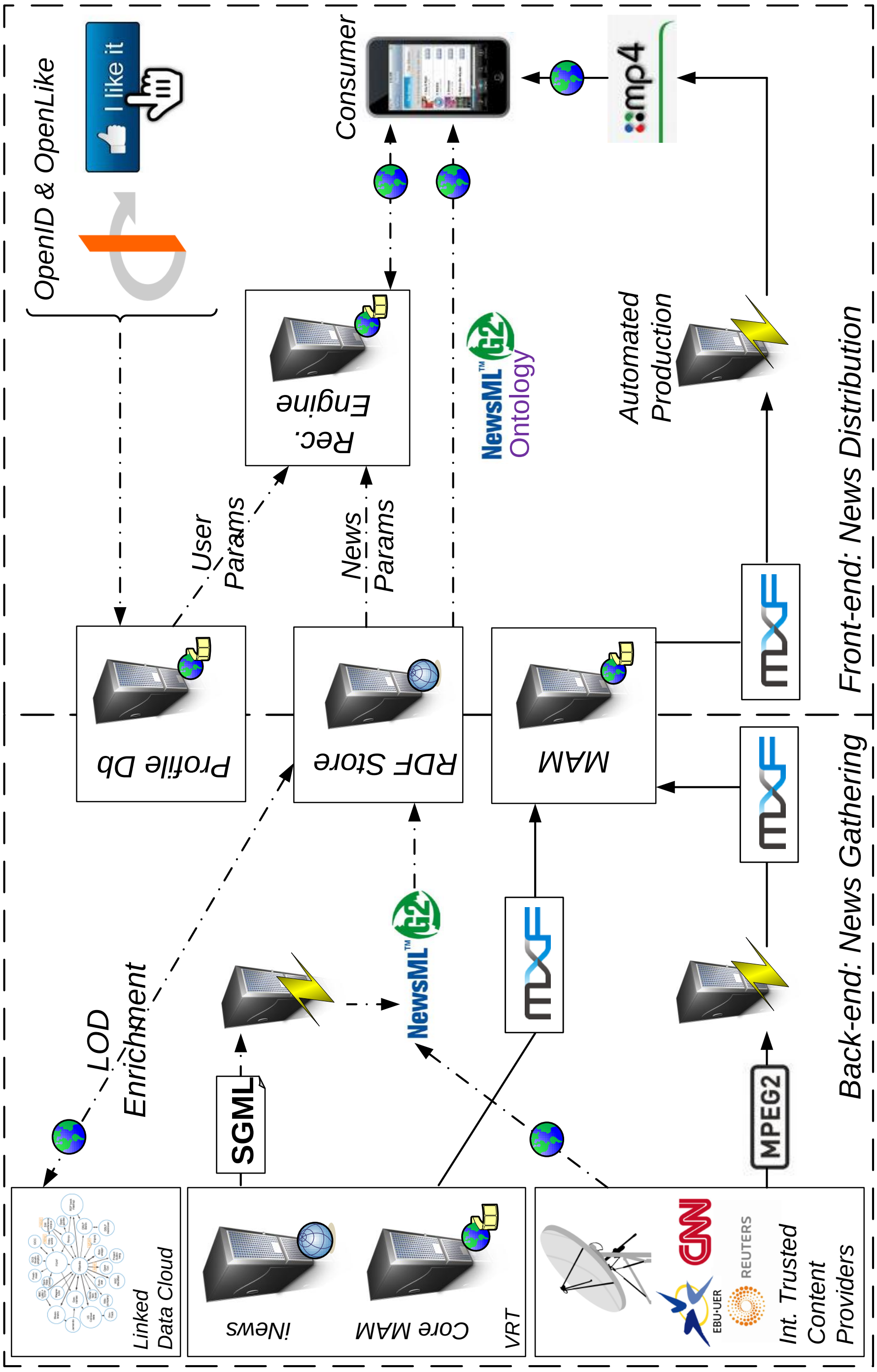


Figure 6.5: The News Distribution Architecture.

Captured media is ingested and stored on servers managed by VRT's core MAM Ardome. Consequently, it has many links with other information systems. Ardome contains all the produced media resources, both rough unfinished material as well as finished products. In addition to media storage, Ardome provides other functionalities such as browsing, rough-cut editing, and searching. In order to facilitate browsing and searching, low resolution versions of stored media are generated and metadata such as the title, audio track information and episode number can be inserted. However, during ingest, metadata insertion is kept to a minimum, as this is currently a manual operation and would otherwise take too much time. Therefore, it is often limited to the title, the owner/creator, a file name, and a path indicating the location where the media is to be stored.

Because captured content often needs further editing, content is also ingested in an editing server (Avid Unity ISIS[12]), which contains all media to be edited. As the editing server only hosts rough content that needs editing, the lifetime of this media is restricted to 72 hours. In order to save time, content from incoming wires is often simultaneously ingested in the MAM and the editing server, resulting in a dual ingest. If dual ingest is not possible, the media is first ingested in the MAM and is then sent from the MAM to the editing server. When content is captured from news feeds, the additional metadata is also inserted in the MAM. Note that the latter is currently also a manual operation.

After ingest, editing can be performed. The journalist who has been given the task to cover a news item retrieves the corresponding media from the editing server and starts editing using Avid NewsCutter[13]. Simultaneously, anchor text (appearing on the auto-cue during broadcast) is provided in iNEWS by the journalist and afterwards reviewed by the news anchor. Textual information that must appear as graphics on screen during the broadcast of a news item (e.g., the name of the interviewed person) is also provided in iNEWS.

An important task during editing is the generation of subtitles. Two types of subtitles can be considered. The first type, referred to as open subtitles, are subtitles that are 'burned' into the picture and therefore always appear on screen. A typical example of the use of open subtitles is when a translation is needed for an interviewed person speaking a foreign language. The second

[12] `http://www.broadcastautomation.com/products/unityISIS/index.asp`

[13] `http://www.avid.com/US/products/NewsCutter-Software/index.asp`

type of subtitles, referred to as closed subtitles, are by default not displayed but can be retrieved when requested (e.g., via Teletext). A journalist is responsible for creating open subtitles and inserting these in iNEWS. A copy of the edited media is rendered containing the open subtitles on the screen using the ‘Belle Nuit’ tool. The edited media with subtitles is transferred to the MAM and afterwards sent to the play-out server for broadcasting. Also a link is provided in iNEWS, relating the edited media with the corresponding item in the rundown. A version without subtitles is also sent to the MAM for later archiving. If there is time left, journalists also create the closed subtitles using Swift. If this is not the case, live subtitling is performed during broadcast by the subtitling department.

During broadcast, the edited media (with open subtitles) is available on the play-out server. iNEWS is used to follow the news rundown and to send the anchor text to the auto-cue. Textual information (e.g., the name of the interviewed person) is also displayed on screen when needed, using the CG. When necessary, closed subtitles are generated during broadcast. The broadcast is again captured on an incoming wire by the media management department in order to have a copy of the integral news broadcast (which includes the open subtitles and textual information displayed by the CG on the screen). The captured broadcast is then also marked for archival.

Archival is the last step in the news production process. An archivist who has been given the task to archive a news broadcast retrieves the rundown of the news broadcast from iNEWS and then searches for the corresponding media fragments in the MAM. Note that this is a manual operation. If a related fragment is found, the archivist watches it and generates a record containing relevant metadata using the archiving tool Basis. Basis is the main tool used to document archived media at VRT in order to facilitate media retrieval. It is also the main tool for media search, e.g., when searching for archived content for reuse. Typically, a user searches for a relevant media fragment in Basis and subsequently retrieves it from the MAM.

As the archival step is performed as the last one in the news media production process, it typically takes a few days before a media fragment is documented in Basis. When a Basis record is generated, some metadata fields that are also present in the MAM are transferred from Basis to the MAM, replacing previously entered metadata in the MAM (e.g., metadata taken from NewsML-G2 documents).

6.3.2 Front-end News Distribution

Sections 3.2 to 3.5 within Chapter 3 are targeted towards distributing personalised enriched news events to the end-user. Once, these recommendations are calculated and pushed to the user via a personal RSS feed, we can use our Ninsuna distribution platform to even further enhance the end-users' quality of experience through faceted browsing and on-the-fly content adaptation. Since Ninsuna's model-driven content adaptation [171, 172] is implemented using Semantic Web technologies such as RDF, OWL, and SPARQL, it is straightforward to use RDF-based annotations within our media delivery platform.

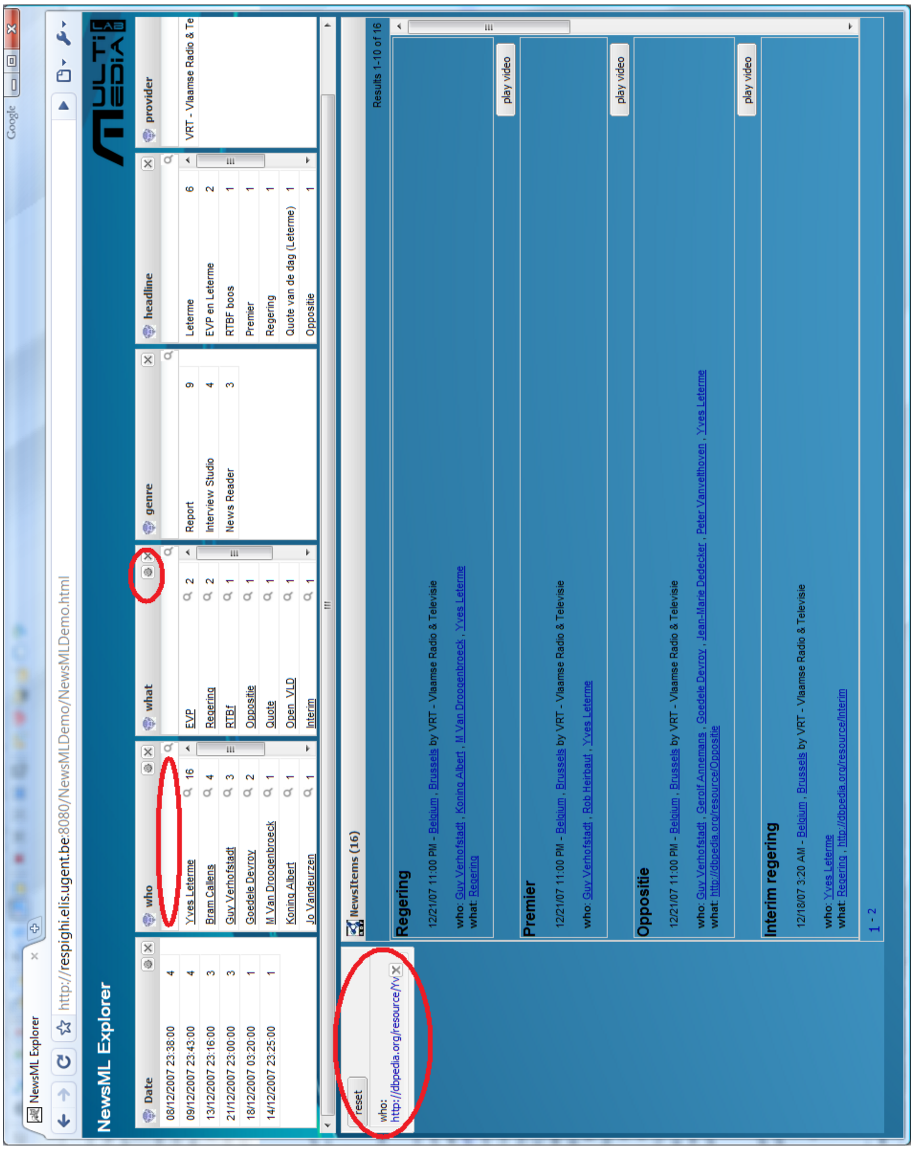
Hence, we built a faceted browser for our media delivery platform. More specifically, NinSuna Facets uses the faceted browsing paradigm to let the end-user obtain his/her personalised news fragments. The Web-based user interface uses Google's Web Toolkit[14] and connects to a SPARQL end-point where all RDF metadata of the news items is stored (see Figure 6.5). The top panel in Figure 6.6 shows examples of possible facets.

Similarly to [70], the facets are dynamically fetched and can be configured to match the users' needs. They correspond to the datatype and object properties of the class hierarchy that has the root *AnyNewsItem*. Within each facet (the predicate), a list (the object range) is composed of instances (for object properties) or literal values (for datatype properties) corresponding to the news items available (the subject domain). Selecting facets and their values enable to build complex queries made of multiple filters that further constrain the search of particular news stories.

The bottom panel in Figure 6.6 depicts the list of news items that match the filters currently selected with the facets. Some relevant item object properties and datatype properties are shown together with the extra records that were captured during the enrichment phase (as discussed in Section 3.3) and visualised as a tag cloud. This bottom panel is built using a Fresnel lens [141] which describes the formatting of the properties to be displayed.

Each facet is finally searchable either in plain text or for more complex objects (i.e., another Web resource with a type). Again similarly to [70], the faceted browser can pop up and display the local view for any objects (see Figure 6.7, our generic RDF browser). The RDF triples are then retrieved

[14] http://code.google.com/intl/en/webtoolkit/



NewsML Explorer
http://respighi.elis.ugent.be:8080/NewsMLDemo/NewsMLDemo.html
Google
NewsML Explorer
MULTiMEDIA LAB

Date	
08/12/2007 23:38:00	4
09/12/2007 23:43:00	4
13/12/2007 23:16:00	3
21/12/2007 23:00:00	3
18/12/2007 03:20:00	1
14/12/2007 23:25:00	1

who	
Yves Leterme	16
Bram Callens	4
Guy Verhofstadt	3
Goedele Devroy	2
M Van Droogenbroeck	1
Koning Albert	1
Jo Vandeurzen	1

what	
EVP	2
Regering	2
RTBf	1
Oppositie	1
Quote	1
Open VLD	1
Interim	1

genre	
Report	9
Interview Studio	4
News Reader	3

headline	
Leterme	6
EVP en Leterme	2
RTBF boos	1
Premier	1
Regering	1
Quote van de dag (Leterme)	1
Oppositie	1

provider
VRT - Vlaamse Radio & Te

who:
http://dbpedia.org/resource/Yv

Regering

12/21/07 11:00 PM - Belgium , Brussels by VRT - Vlaamse Radio & Televisie

who: Guy Verhofstadt , Koning Albert , M Van Droogenbroeck , Yves Leterme
what: Regering

Premier

12/21/07 11:00 PM - Belgium , Brussels by VRT - Vlaamse Radio & Televisie

who: Guy Verhofstadt , Rob Heirbaut , Yves Leterme

Oppositie

12/21/07 11:00 PM - Belgium , Brussels by VRT - Vlaamse Radio & Televisie

who: Guy Verhofstadt , Gerolf Annemans , Goedele Devroy , Jean-Marie Dedecker , Peter Vanvelthoven , Yves Leterme
what: http://dbpedia.org/resource/Oppositie

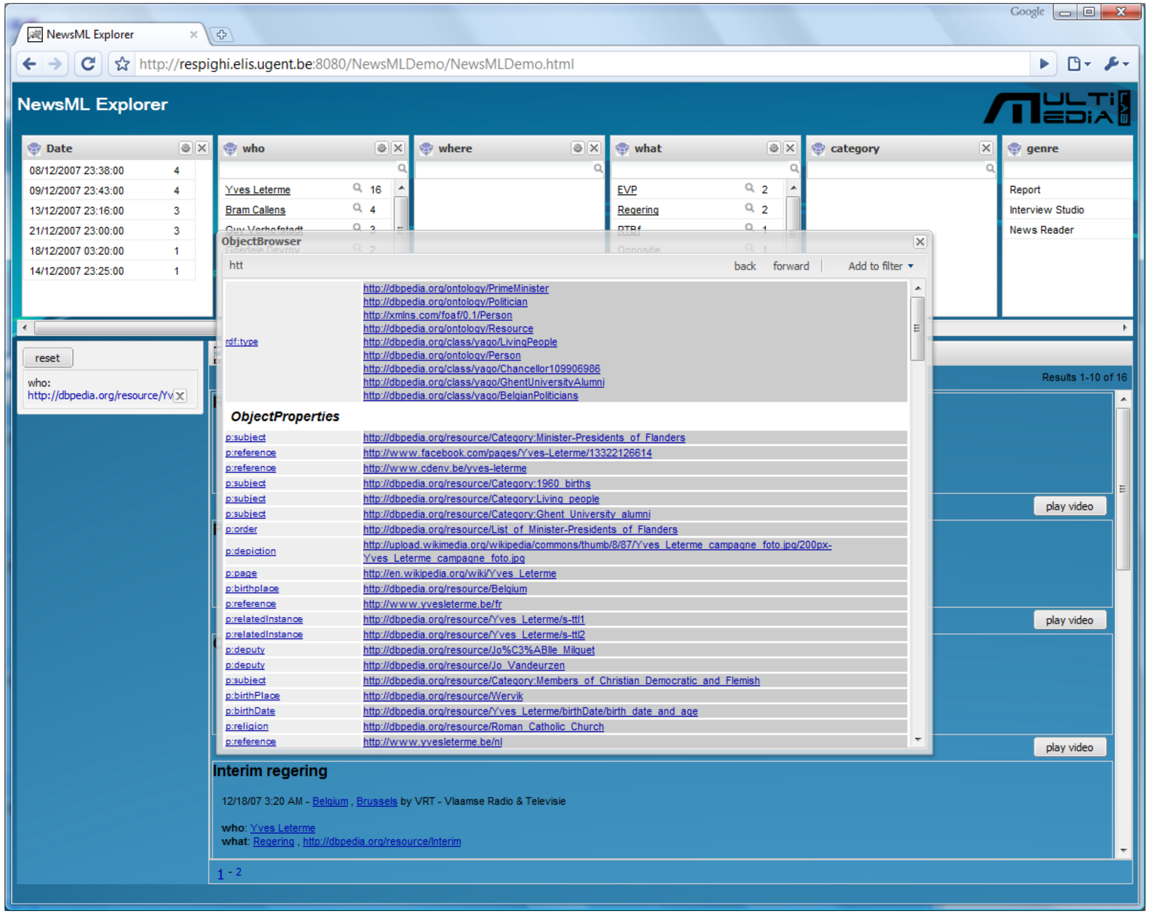
Interim regering

12/18/07 3:20 AM - Belgium , Brussels by VRT - Vlaamse Radio & Televisie

who: Yves Leterme
what: Regering , http://dbpedia.org/resource/Interim

Figure 6.6: The Ninsuna News Browser - Facets' view.

from the store and formatted according to an appropriate style-sheet. Any property can therefore be added as a filter on its own, thus re-iterating the cycle of selecting other resources within the pool of existing news items.



Date	
08/12/2007 23:38:00	4
09/12/2007 23:43:00	4
13/12/2007 23:16:00	3
21/12/2007 23:00:00	3
18/12/2007 03:20:00	1
14/12/2007 23:25:00	1

who	
Yves Leterme	16
Bram Callens	4
Guy Verhofstadt	3
Goedele Devroy	2

what	
EVP	2
Regering	2
RTBf	1
Oppositie	1

genre: Report, Interview Studio, News Reader

who: http://dbpedia.org/resource/Yv

ObjectBrowser

htt

rdf:type	http://dbpedia.org/ontology/PrimeMinister http://dbpedia.org/ontology/Politician http://xmlns.com/foaf/0.1/Person http://dbpedia.org/ontology/Resource http://dbpedia.org/class/yago/LivingPeople http://dbpedia.org/ontology/Person http://dbpedia.org/class/yago/Chancellor109906986 http://dbpedia.org/class/yago/GhentUniversityAlumni http://dbpedia.org/class/yago/BelgianPoliticians

ObjectProperties

p:subject	http://dbpedia.org/resource/Category:Minister-Presidents_of_Flanders
p:reference	http://www.facebook.com/pages/Yves-Leterme/13322126614
p:reference	http://www.cdenv.be/yves-leterme
p:subject	http://dbpedia.org/resource/Category:1960_births
p:subject	http://dbpedia.org/resource/Category:Living_people
p:subject	http://dbpedia.org/resource/Category:Ghent_University_alumni
p:order	http://dbpedia.org/resource/List_of_Minister-Presidents_of_Flanders
p:depiction	http://upload.wikimedia.org/wikipedia/commons/thumb/8/87/Yves_Leterme_campagne_foto.jpg/200px-Yves_Leterme_campagne_foto.jpg
p:page	http://en.wikipedia.org/wiki/Yves_Leterme
p:birthplace	http://dbpedia.org/resource/Belgium
p:reference	http://www.yvesleterme.be/fr
p:relatedInstance	http://dbpedia.org/resource/Yves_Leterme/s-ttl1
p:relatedInstance	http://dbpedia.org/resource/Yves_Leterme/s-ttl2
p:deputy	http://dbpedia.org/resource/Jo%C3%ABlle_Milquet
p:deputy	http://dbpedia.org/resource/Jo_Vandeurzen
p:subject	http://dbpedia.org/resource/Category:Members_of_Christian_Democratic_and_Flemish
p:birthPlace	http://dbpedia.org/resource/Wervik
p:birthDate	http://dbpedia.org/resource/Yves_Leterme/birthDate/birth_date_and_age
p:religion	http://dbpedia.org/resource/Roman_Catholic_Church
p:reference	http://www.yvesleterme.be/nl

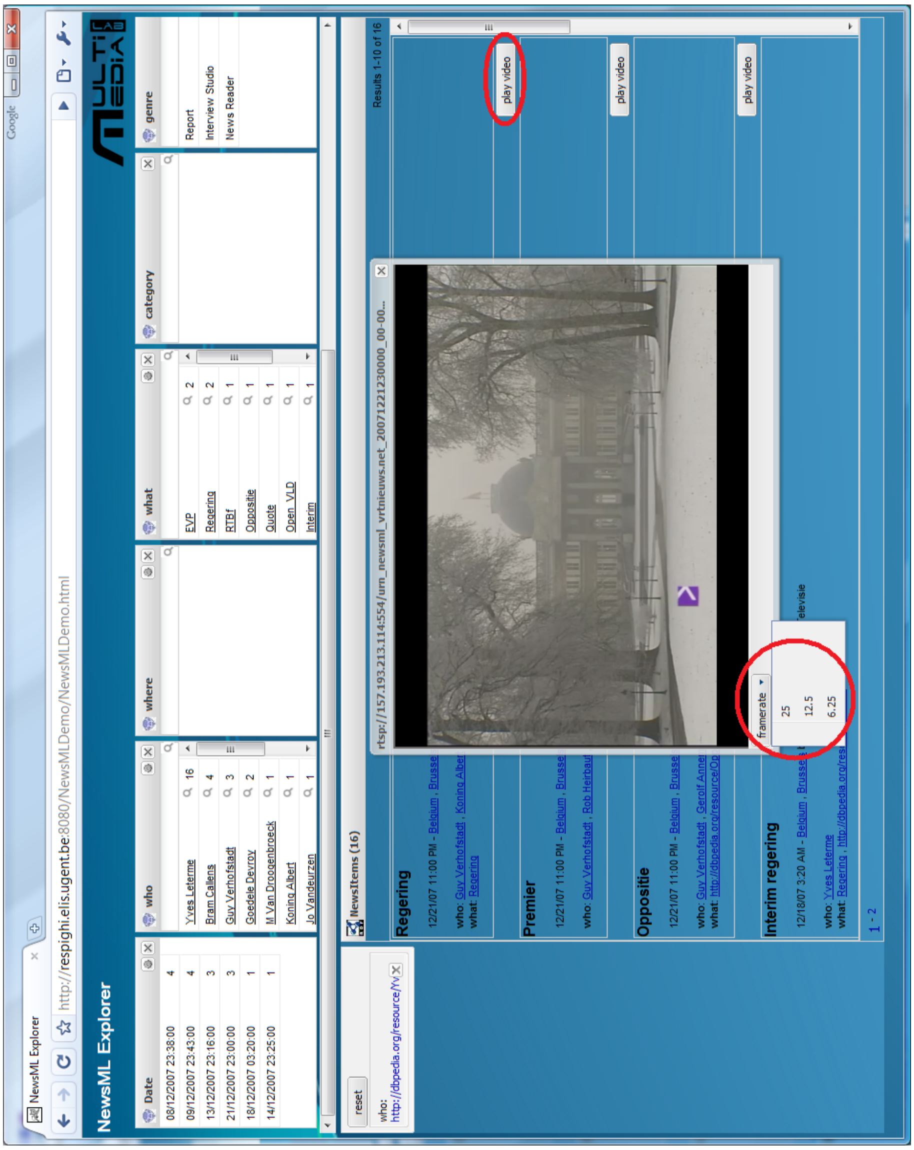
Interim regering

12/18/07 3:20 AM - Belgium , Brussels by VRT - Vlaamse Radio & Televisie

who: Yves Leterme
what: Regering , http://dbpedia.org/resource/Interim

Figure 6.7: The Ninsuna News Browser - RDF view.

Finally clicking on the play video button brings us to the video playback panel (see Figure 6.8). The selected news fragment immediately starts to play. During the playback, the frame rate can be adjusted in an on-the-fly fashion. The NinSuna Facets demo can be started by accessing the following link:



NewsML Explorer
http://respighi.elis.ugent.be:8080/NewsMLDemo/NewsMLDemo.html
NewsML Explorer
MULTIMEDIA LAB

Date	
08/12/2007 23:38:00	4
09/12/2007 23:43:00	4
13/12/2007 23:16:00	3
21/12/2007 23:00:00	3
18/12/2007 03:20:00	1
14/12/2007 23:25:00	1

who	
Yves Leterme	16
Bram Callens	4
Guy Verhofstadt	3
Goedele Devroy	2
M Van Droogenbroeck	1
Koning Albert	1
Jo Vandeurzen	1

what	
EVP	2
Regering	2
RTBf	1
Oppositie	1
Quote	1
Open VLD	1
Interim	1

Report
Interview Studio
News Reader

who:
http://dbpedia.org/resource/Yv

NewsItems (16)

Results 1-10 of 16

rtsp://157.193.213.114:554/urn_newsml_vrtnieuws.net_20071221230000_00-00...

Regering

12/21/07 11:00 PM - Belgium , Brusse

who: Guy Verhofstadt , Koning Alber
what: Regering

Premier

12/21/07 11:00 PM - Belgium , Brusse

who: Guy Verhofstadt , Rob Heirbau

Oppositie

12/21/07 11:00 PM - Belgium , Brusse

who: Guy Verhofstadt , Gerolf Anne
what: http://dbpedia.org/resource/Op

Interim regering

12/18/07 3:20 AM - Belgium , Brusse

who: Yves Leterme
what: Regering , http://dbpedia.org/res

Figure 6.8: The Ninsuna News Browser - Video view.

http://multimedialab.elis.ugent.be/NinSunaFacets/.

6.4 The Archiving Use Case

In this use case the distributed architecture of a digital long-term preservation archive is described. In this networked world, various resources are linked to each other. For this reason, we do not want to build yet another central e-depot, but a distributed network of storage components based on a Service Oriented Architecture (SOA). This SOA will make use of a central service hub, as depicted in Figure 6.10, which will offer the needed services for the platform. The objectives of the platform are twofold: disseminate the content as LOD and enable long-term preservation.

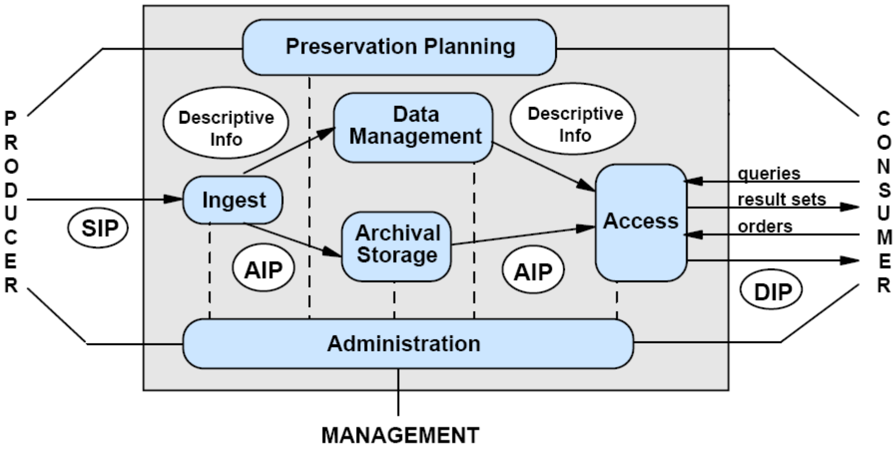


Figure 6.9: OAIS Functional Entities.

The first service the platform needs is a harvesting service. This service will harvest the content from Flanders' archival institutions –be it museums, broadcasters, and/or libraries. The protocol used for aggregating the data is OAI-PMH. This harvesting service has to be able to deal with multiple metadata formats, hence the different types of institutions. For this, we need a common ground on the metadata for search and retrieval. This common ground is DC RDF, the top layer of the layered metadata model, as explained in Chapter 4.3.1. This mapping will be done using a central mapping service (a second service), which is able to accept certain metadata records (for the moment DC, MARC21, ISAD(G), EAD, P/Meta, CDWA, and SPECTRUM are supported) and convert them to DC RDF. Before publishing these DC records as LOD, these records need to be enriched with data from external data sources. This third service will detect certain concepts in the triple store and convert them to resources. First of all these resources are institutions, which can have several collections on their own. These collections also get a URI, and become a second resource. Collections are filled with records or other collections. These records form a third resource, of course. Within these records, all persons, places and timing instances will be discovered, which in their turn will become extra other resources. These enriched DC records are stored in the Jena[15] triple store, which implements a SPARQL end-point using Joseki[16] –an HTTP engine that supports the SPARQL Protocol and the SPARQL RDF Query language.

The services discussed until now, are actually responsible for the dissemination of the harvested content as LOD, one of the two main objectives of the platform. The other main objective is enabling the long-term preservation, compliant with the OAIS reference model [89]. This OAIS model mandates that the archive not only stores the data for the long-term, but also keeps the data accessible for the long-term. For this it defines three sorts of packages: *Submission Information Packages* (SIP), *Archival Information Packages* (AIP), and *Dissemination Information Packages* (DIP). The SIPs are the packages as they are delivered to the Content Management System (CMS) that will fulfil the long-term preservation functions. These packages keep the metadata and the referenced files together in a package. For this package, the *BagIt* [129] package format is used. It allows storing the metadata together with the files referenced by the metadata. This package also allows storing some fixity data in it, e.g., ‘Message Digest Algorithm 5’ (MD5) checksums for validation. This SIP creation is a first service for implementing the long-term preservation functions. In our case, the SIP will consist of references to the multimedia files, stored in the cloud, together with the original metadata offered by the archival institution, and the mapped DC RDF record, for managing the CMS. The enrichment information does not become part of the SIP, because this information is not always persistent.

When an institution wants their harvested content preserved for the long-term, the digital representations the metadata records are describing must also be harvested. This is done in a separate service, as otherwise the harvesting process would take too long. This way, the metadata harvesting is separated from the essence harvesting. This service does only come into play for long-term archiving, not for dissemination. If the record would be harvested without being preserved for the long-term, these files won’t be harvested, but

[15] http://jena.sourceforge.net/

[16] http://joseki.sourceforge.net/

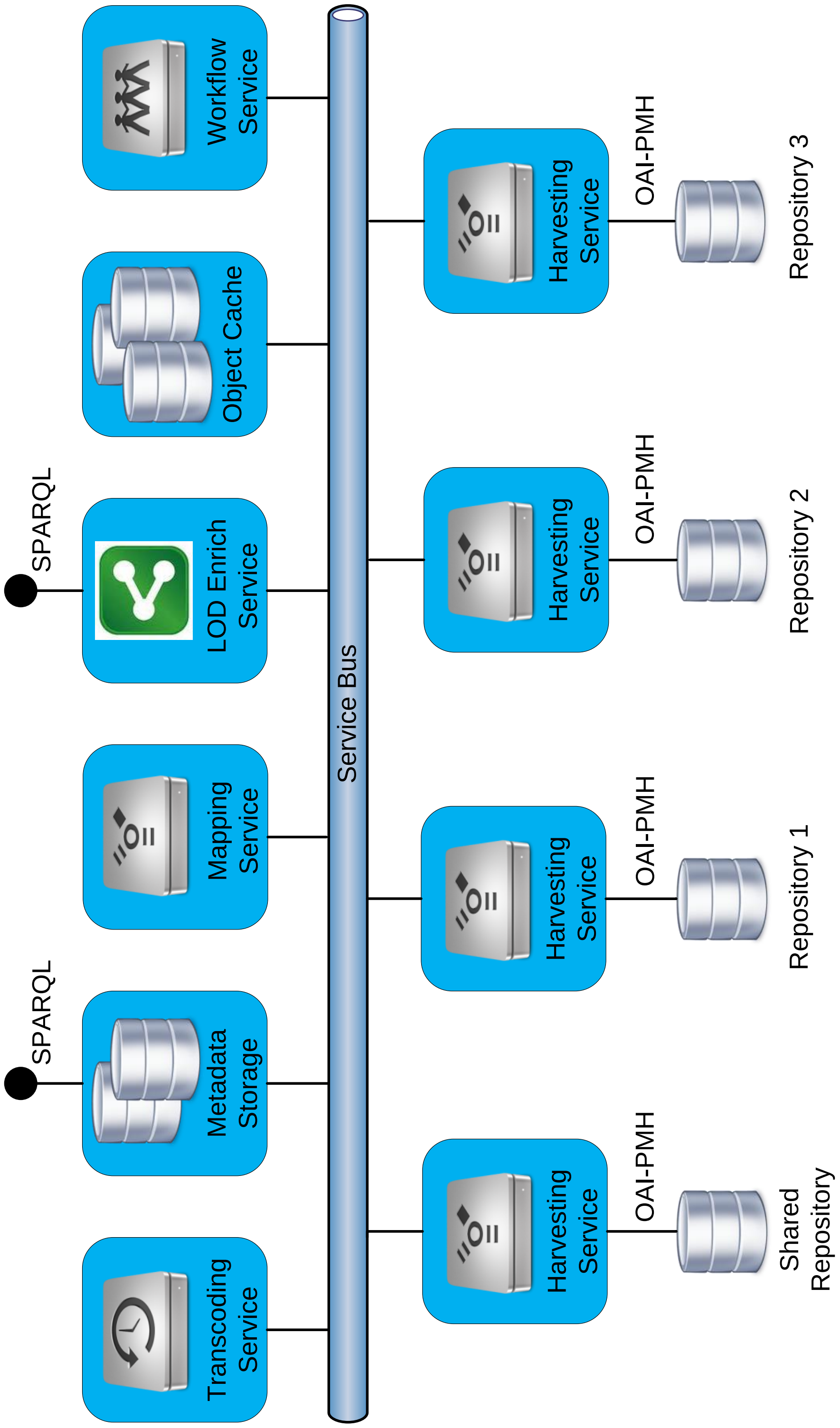


Figure 6.10: Archipel's Bus Architecture.

will only be referenced. The moment the institution wants to preserve its harvested records too, the files referenced in the records will ultimately be harvested and stored into the cloud, using *MogileFS*[17]. So, before creating SIP packages, we must first harvest the referenced multimedia files. This is yet another service offered by the service hub. Now the SIP becomes an AIP when it is stored into the CMS, as depicted in Figure 6.9. This is a package that also keeps the metadata and the multimedia files referenced by the metadata, together with the needed information for enabling the long-term preservation, as suggested by the OAIS reference model. This means the SIP packages are extended with preservation metadata, for which the PREMIS ontology from the layered metadata model (as discussed in Chapter 4.3.2) will be used. This preservation metadata is responsible for tracking the provenance of the data and keeping the data accessible for the long-term. The preservation metadata is created automatically at ingest of the SIP into the CMS. This means that when a multimedia file gets imported into the archive, it must first have a PREMIS *Intellectual Entity* description, which is our mapped DC RDF record. Linked to this intellectual entity, we have the multimedia files, which are described using the PREMIS *Object* description. This object describes the multimedia files in a technical manner. This will enable, e.g., future migration services. For this, we need characterisation services, which are able to detect the real file format and store the needed information about the file format in the *Object* description of the multimedia file. This characterisation service is also implemented on the service hub and the service ingesting the SIP into the CMS will make use of it. For the implementation of the characterisation service, we make use of the DROID[18], PRONOM[19], and JHOVE[20] tools.

Finally, the CMS must be able to disseminate DIPs. These are actually the same as SIPs, but they differ in the file format in which the referenced multimedia files are offered to the end-user, cf., the general OAIS Depiction 6.9. The file formats offered by the DIPs must be accessible, e.g., a document can be offered to the archive in *WordPerfect*[21] file format together with its metadata. This document is not accessible anymore. Therefore, the archive must migrate the file format to an accessible format, e.g., PDF/A-1 [90]. This PDF/A-1 file is the file that is referenced in the DIP. Of course this PDF/A-1 file will not replace the original *WordPerfect* file. It will become a new object linked to the metadata that will be offered in the DIP. For this DIP, the *BagIt* package for-

[17] http://danga.com/mogilefs/

[18] http://droid.sourceforge.net/

[19] http://www.nationalarchives.gov.uk/aboutapps/PRONOM/tools.htm

[20] http://hul.harvard.edu/jhove/index.html

[21] http://en.wikipedia.org/wiki/WordPerfect/

mat will also be used. Keeping the data accessible is a task of the preservation plans, defined by the OAIS reference model. They implement, e.g., migration services (e.g., transcoding) to keep the data accessible, validation services for virus checking, MD5 checking, digital signature checking, etc. Once a file format is not supported anymore, it must be migrated to another file format as defined by a preservation plan. These preservation plans, implemented as a workflow service, as can be seen in Figure 6.10, will actually implement the archives' policies. An example of such a policy is that all the images offered to the archive, will be stored using the lossless JPEG2000 [86] file format and will be disseminated using the lossy JPEG [96] file format. This means the long-term preservation archive needs services to migrate all the incoming image file formats to JPEG2000 and JPEG. These services will also be implemented into the service bus, which can be seen in Figure 6.10. Whenever a preservation plan comes to action, the AIP will be extended with preservation metadata, describing the preservation plan. For this, every preservation plan (a.o., migration service), will be described using PREMIS events, which are linked to the PREMIS objects. This way, the architecture, together with the developed, layered metadata model we proposed, is able to achieve long-term preservation.

6.5 The Media Fragments Use Case

As described in Chapter 5, three different approaches for media fragment retrieval over HTTP are feasible: retrieval without 'fragment-to-byte range' mapping, 'fragment-to-byte range' mapping calculation at the user agent, and 'fragment-to-byte range' mapping calculation at the server. These three approaches are compared to each other according to five criteria, as can be seen in Table 6.1. As already discussed, the approach where the full media resource is retrieved is not feasible for large media resources (i.e., the bandwidth cost and latency are too high), despite the minimal extensions needed within the current Web infrastructure.

Comparing the mapping calculation at the user agent and at the server, both approaches have their own pros and cons. Server-side mapping calculation is the most efficient approach in terms of bandwidth cost and latency, but requires extensions for existing Web caches and servers. User agent-side mapping calculation requires a large implementation cost at the user agent, and the mapping calculation itself may introduce an additional latency. In the next subsections, we propose two optimised approaches (i.e., one for user agent-side and one for server-side mapping calculation). More specifically, we present

Table 6.1: Evaluation of the different Media Fragment retrieval Strategies.

	Full resource retrieval	Mapping calculation at user agent	Mapping calculation at server
Bandwidth cost	full resource	media fragment + portions needed for mapping	media fragment
Latency	time to download bytes before start of the fragment	time needed for the mapping calculation	no extra latency introduced
User agent extensions	MF interpretation and rendering	MF interpretation, remote mapping module, and rendering	MF interpretation and rendering
Cache extensions	none	none	new Range units
Server extensions	none	none	MF request, mapping, module, and response

the following (orthogonal) contributions:

- *Media Fragments translation service*: a service assisting user agents to perform their mapping calculation (see subsection. 6.5.1).
- *Format-independent Media Fragments server*: a Media Fragments-aware server with a mapping calculation module that is independent of media formats (see subsection 6.5.2).

6.5.1 Media Fragments Translation Service

The main reason why we introduce a Media Fragments Translation Service (MFTS) is to preserve the existing Web infrastructure as much as possible. More specifically, our goal is that regular HTTP servers can serve media fragments, existing HTTP caches are able to cache media fragments, and user agents only require a minimal extension for media fragments retrieval (i.e., the minimal extension comes down to just parsing and interpretation of Media Fragments URIs). The role of the MFTS is illustrated in Figure 6.11. Suppose we want to retrieve the media fragment `video.ogg#t=10,20`, the following steps are taken:

1. The user agent parses and interprets the Media Fragment URI. Since the media is served by a regular HTTP server, fragments can only be retrieved in terms of byte ranges. However, there is no Media Fragments translation module available at the user agent. Therefore, the user agent makes use of the MFTS by asking which byte ranges correspond to the

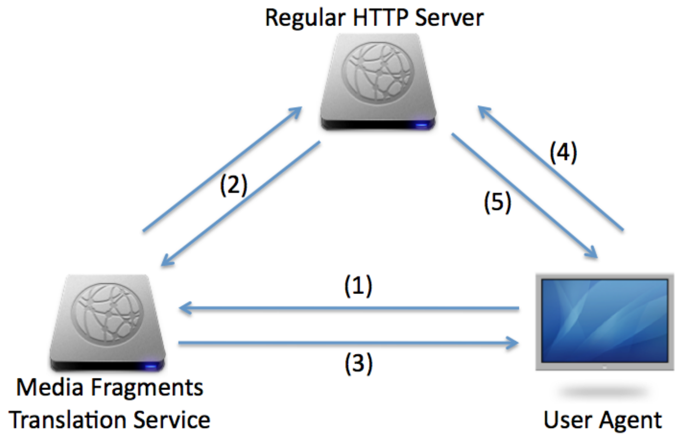


Figure 6.11: Retrieving Media Fragments with the Help from a Media Fragments Translation Service.

temporal range 10-20 seconds. The corresponding HTTP request message has been depicted in Figure 5.8.

2. The MFTS interprets the request from the user agent and calculates the mapping between the media fragment and its byte ranges. The latter is achieved in the same way as was discussed in Section 5.4.1, using so-called remote mapping calculation algorithms.

3. When the MFTS has found the mapping between the media fragment and its byte ranges (i.e., 23000-33000 bytes), it constructs an HTTP redirect response message (see Figure 5.8) indicating the relation between the media fragment the user agent requested and its location in terms of byte ranges.

4. The user agent follows the received HTTP redirect message. In other words, the user agent is now able to retrieve the media fragment using a regular HTTP byte range request (see Figure 5.2).

5. Finally, the Web server returns the requested bytes, which correspond to the media fragment. The user agent loads these bytes and can start playing the media fragment.

When the MFTS is unable to calculate the mapping (e.g., the MFTS is unaware of the underlying container format) or the mapping results in too many byte ranges (e.g., in case of interleaved tracks as mentioned in

Chapter 5), the MFTS redirects the user agent to the full media resource.

We implemented an MFTS in order to demonstrate the feasibility of this approach. The MFTS is available at `http://ninsuna.elis.ugent.be/MFProxy/` and can be used as follows:

```
GET /MFProxy?url=<MediaResourceURI>
Host: ninsuna.elis.ugent.be
Accept: video/*
Range: <Range>
Accept-Range-Redirect: bytes
```

Both time and track units are supported in the HTTP Range header. Note that for the moment, the MFTS only supports MP4 [88] and Ogg [138] media resources. The response times of the MFTS heavily depend on:

- The connection between the MFTS and the HTTP Web server: a slow link will naturally cause a higher response time.
- The remote mapping calculation algorithm (dependent on the underlying media format): index interpretation (e.g., in case of MP4) is more efficient than bi-sectional search over HTTP (e.g., in case of Ogg).

Given such an MFTS, every media resource on the Web can be served through a Media Fragment URI, on condition that the MFTS supports the underlying container format. Moreover, the MFTS could also be used as a kind of media resource information service, not only providing the mapping between media fragments and byte ranges, but also other information such as available tracks, duration, or bit rate.

6.5.2 NinSuna: a Format-independent Media Fragments Server

In the previous subsection, our goal was to preserve as much as possible the existing Web infrastructure with the introduction of media fragments. In this subsection however, we elaborate on the architecture of specialised media fragment servers. Moreover, we propose an architecture that is independent of the underlying media formats and that is able to efficiently interpret standardised Media Fragment requests.

The NinSuna platform, which the authors introduced in [172], is a format-independent media delivery platform that is able to perform high-level selection operations such as fragment extraction. Moreover, as elaborated on

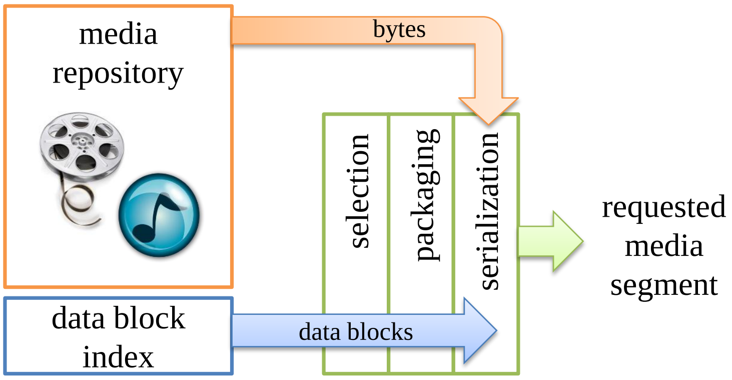


Figure 6.12: High-level Overview of Media Segment Extraction inside NinSuna.

in [173], NinSuna makes use of a format-independent packaging technique (i.e., to encapsulate media into a container format), independent of the high-level selection operations. Hereafter, we show how Media Fragments can be served through NinSuna, with at its core a format-independent selection and packaging engine.

6.5.2.1 High-level Architecture

A high-level overview of the architecture of NinSuna[22] is depicted in Figure 6.12. One of the keys in the design of our NinSuna platform is the data block-based index for the media repository. This index is based on a model for media resources describing the high-level structure of media resources in terms of Random Access Units (RAUs)[23] and corresponding data blocks [171]. More specifically, for each track of the media resource, we describe the list of RAUs. Subsequently, we describe each RAU in terms of data blocks, containing information such as display time and byte range. For a video track, a data block typically corresponds to a video frame or slice. By creating a format-independent index for every media resource in the repository, we can obtain a format-independent selection and packaging core.

The data block-based index is based on Semantic Web technologies. More specifically, our model for media resources is implemented in OWL [121]; thus the instances of the model (i.e., the index) are represented in RDF [103]. Additionally, data blocks are retrieved by means of SPARQL

[22] `http://ninsuna.elis.ugent.be/`

[23] Each RAU starts with a random access point and ends just before the next one

queries [143, 171]. A detailed workflow of NinSuna is depicted in Figure 6.13. If media resources need to be selected and packaged with our proposed method, metadata instances compliant to our model need to be generated during the index generation step (1); this way, the data block-based index is created in the form of RDF triples. The requested parts of the media streams are obtained during the data block selection step (2), where RDF graphs describing data blocks are queried using SPARQL. Based on the selected data blocks, a simple RDF-to-XML transformation is performed (3). The result of this transformation is an XML description of the selected data blocks, called a Bitstream Syntax Description (BSD). The latter can be used to create a packaged version of the adapted media bitstream. The classes and properties defined in our model, needed for the packaging process, are mapped to XML elements and attributes respectively.

The actual packaging process starts with the transformation of the BSD representing (part of) the elementary media bitstream (4). The resulting BSD represents an adapted and packaged media bitstream. The obtained BSD is compliant with MPEG-B BSDL [95], which implies that the BSDL framework can be used for further processing. The BSD transformation can be implemented using XSLT [31] or STX [30] (i.e., packaging filters), which enables the use of a format-independent transformation engine. Additionally, a Bitstream Syntax Schema (BS Schema, [95]) needs to be created, describing the high-level structures and syntax elements of the packaging format (5). Finally, the adapted and packaged media bitstream is created using BSDL's format-independent BSDtoBin parser [95], based on the BSD representing the adapted and packaged media bitstream, the BS Schema describing the delivery format, and the original media bitstream (6). Thus, given a media repository where each media resource is indexed according to the data block-based index, creating a media segment within NinSuna requires the following steps (see also Figure 6.12):

1. For each requested track and possibly given time range, the proper data blocks are selected. Note that the first returned data block always needs to correspond to a random access point.

2. The selected data blocks are packaged into a requested container format (e.g., MP4). For each container format, a packaging filter exists (e.g., a STX filter), taking as input data blocks and producing an XML description corresponding to a packaged version of the selected data blocks (e.g., put a header and syntax structures around the data blocks) [173].

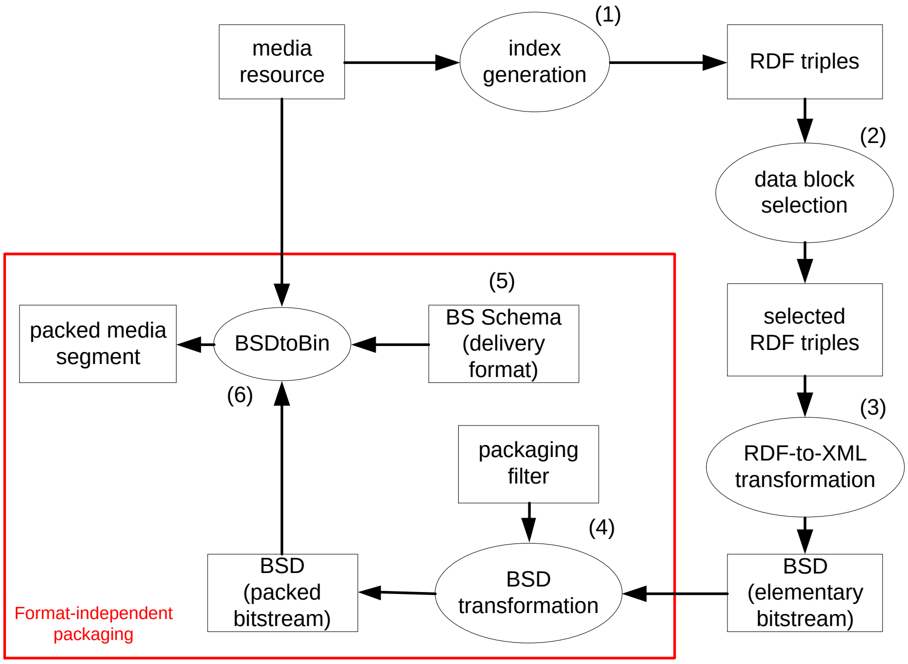


Figure 6.13: Detailed Workflow of NinSuna.

3. Finally, the selected and packaged data blocks are serialised. As already mentioned, the latter can be realised by making use of MPEG-B BSDL: based on the original media resource and an XML description of the selected and packaged data blocks, the desired media segment can be generated [95].

As described above, NinSuna enables the generation of a media resource corresponding to a segment of another media resource. Since the result of this operation is a new and playable media resource with a newly created header, we cannot call this a *media fragment*, as discussed in Section 5.3.3.

However, we can use URI queries as an interface for the NinSuna platform. For example, `http://ninsuna.elis.ugent.be/media.mp4?t=10,20` corresponds to a new media resource containing the frames between 10 and 20 seconds of media.mp4. Hence, in order to provide support for Media Fragment URIs in a format-independent way, additional provisions are necessary.

6.5.2.2 Extending NinSuna for Media Fragments Support

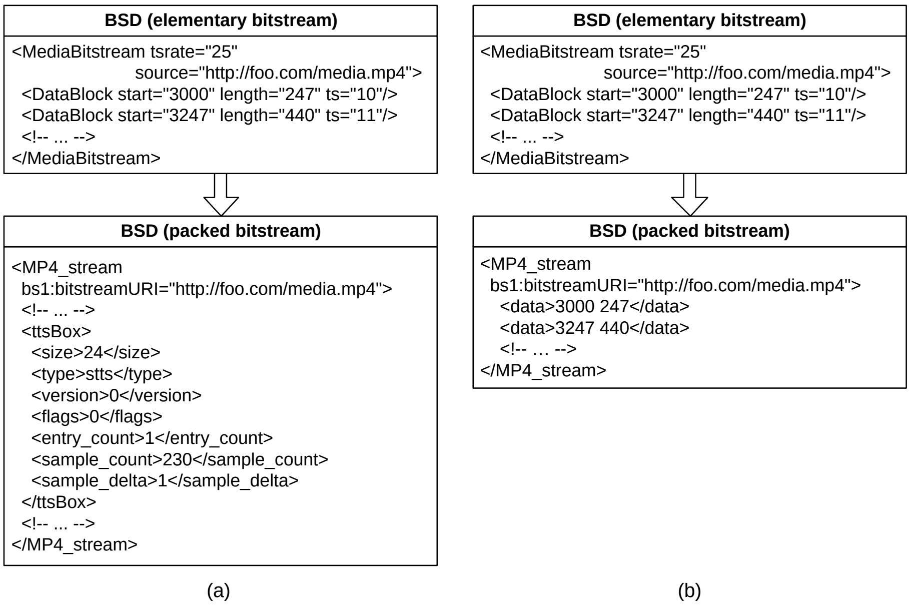
Extending a Web server with Media Fragments support is relatively easy. More specifically, if the server contains a fragment-to-byte range translation module for the requested media format, it can perform the mapping calculation and serve the corresponding bytes. However, in this scenario, the translation module is format-dependent: the server needs a different translation module for each media format that is being served.

Alternatively, NinSuna is designed to select and deliver media resources independent of the underlying coding formats. The latter is obtained by maintaining a generic index structure for each media resource; selection and packaging filters are based on this generic index structure (see subsection 6.5.2.1). As a consequence, each served media resource is selected and packaged on-the-fly, which means that requested media resources are only available at runtime. Therefore, mapping media fragments to byte ranges is more complex within NinSuna in comparison to regular Web servers, since the exact byte ranges are only available at runtime. Additionally, as a matter of optimisation, we do not want to generate the full media resource, if only a media fragment of that resource is requested.

In order to cope with these problems, we extended NinSuna so that it is able to efficiently interpret a media fragment request. Suppose we want to retrieve the media fragment `media.mp4#t=10,20` (comparable to Figure 5.6), the following steps are taken.

1. *Header generation*: the header (in this case the MP4 header) of the full media resource (i.e., media.mp4) is generated. More specifically, each packaging filter has two different modes: header and data. In header mode, the packaging filter only sends header XML structures to the serialisation module. An example is depicted in Figure 6.14(a), where only the header structures of an MP4 file are created. Additionally, the packaging header generates a map containing the relation between media fragments (i.e., tracks and time) and their byte offsets within the full media resource.

2. *Payload generation*: the data blocks corresponding to the requested temporal range (i.e., 10-20 seconds) are selected, packaged, and serialised. This time, the packaging filter operates in data mode and produces no header XML structures. For example, in Figure 6.14(b), no header structures are generated for the MP4 file, only pointers to the data are generated. Note that when the packaging filter operates in data mode, only



BSD (elementary bitstream)

```
<MediaBitstream tsrate="25"
                source="http://foo.com/media.mp4">
  <DataBlock start="3000" length="247" ts="10"/>
  <DataBlock start="3247" length="440" ts="11"/>
  <!-- ... -->
</MediaBitstream>
```

BSD (packed bitstream)

```
<MP4_stream
  bs1:bitstreamURI="http://foo.com/media.mp4">
  <!-- ... -->
  <ttsBox>
    <size>24</size>
    <type>stts</type>
    <version>0</version>
    <flags>0</flags>
    <entry_count>1</entry_count>
    <sample_count>230</sample_count>
    <sample_delta>1</sample_delta>
  </ttsBox>
  <!-- ... -->
</MP4_stream>
```

(a)

BSD (elementary bitstream)

```
<MediaBitstream tsrate="25"
                source="http://foo.com/media.mp4">
  <DataBlock start="3000" length="247" ts="10"/>
  <DataBlock start="3247" length="440" ts="11"/>
  <!-- ... -->
</MediaBitstream>
```

BSD (packed bitstream)

```
<MP4_stream
  bs1:bitstreamURI="http://foo.com/media.mp4">
    <data>3000 247</data>
    <data>3247 440</data>
    <!-- ... -->
</MP4_stream>
```

(b)

Figure 6.14: Illustrating Header and Data Mode in the MP4 packaging Filter.

data blocks are selected corresponding to the requested temporal range; when the packaging filter operates in header mode, all data blocks are selected corresponding to the requested resource in order to build the complete header and fragment-to-byte range map.

3. *Response generation*: based on the output of the serialisation module and the generated fragment-to-byte range mapping, an HTTP response is created (comparable to Figure 5.5).

For efficiency reasons, the generated header and fragment-to-byte range mapping are temporally cached. Further, NinSuna allows the combination of URI queries and URI fragments. For example, consider the media fragment `media.mp4?t=10,20#t=2,4`. Retrieving the latter from NinSuna would result in an MP4 header corresponding to the media resource `media.mp4?t=10,20` and payload data corresponding to 2-4 seconds (or 12-14 seconds in media.mp4). An implementation of Media Fragment support for NinSuna and additional examples are available at `http://ninsuna.elis.ugent.be/MediaFragmentsServer/`.

6.5.2.3 Application: Multiple Bit Rate Delivery

One of the scenarios where we deployed NinSuna and its server-side Media Fragments URI implementation is multiple bit rate delivery. Examples of multiple bit rate delivery on the Web are YouTube[24] or Dailymotion[25]. Typically, multiple bit rate versions of the same visual content are represented by different media resources. On the other hand, a media resource can also be characterised by a number of tracks, each combination representing the same audio-visual content but having a different bit rate. For example, consider the media resource media.mp4 having the following tracks:

1. High quality H.264/AVC video (1000 kbit/s).
2. Medium quality H.264/AVC video (500 kbit/s).
3. Low quality H.264/AVC video (200 kbit/s).
4. High quality AAC audio (192 kbit/s).
5. Low quality AAC audio (96 kbit/s).

Each combination of one audio and one video track results in the same audio-visual representation. However, different combinations will result in different bit rates/qualities. Having multiple bit rates at our disposal allows us to deliver media to a wide range of devices (from HD TVs to mobile phones) and to anticipate differing network conditions [135]. The following two subsections illustrate two approaches to deliver different versions towards end-users, using Media Fragment URIs.

6.5.2.4 HTTP Download

In the context of the IBBT MAPLE project (as can be seen in Section E.2 of Appendix E), NinSuna was used to deploy multiple bit rate delivery over HTTP. Therefore, we ingested Apple's iTunes Movie Trailers archive[26] into our NinSuna platform so that multiple versions of each movie trailer were available for the end-user[27]. The different versions of a media resource are represented by Media Fragment URIs. More specifically, URI queries are used to indicate the desired tracks (i.e., desired bit rate). Note that we use URI queries for track selection since track extraction from a track-interleaved

[24] http://www.youtube.com/

[25] http://www.dailymotion.com/

[26] http://trailers.apple.com/

[27] The trailers can be accessed at http://ninsuna.elis.ugent.be/DownloadServlet/apple/grasp.html

media resource results in a huge amount of byte ranges. Thus, the following combinations of media.mp4 could be created:

- `http://ninsuna/DownloadServlet/media.mp4?track=1;4`: high quality audio and video.
- `http://ninsuna/DownloadServlet/media.mp4?track=2;5`: medium quality audio and video.
- `http://ninsuna/DownloadServlet/media.mp4?track=3;5`: low quality audio and video.

6.5.2.5 HTTP Live Streaming

The downside of the approach presented in the previous subsection is that the quality of the media resource needs to be known beforehand and cannot be changed during the media delivery. Dynamic switching of tracks is interesting when the available bandwidth is rapidly changing. Recently, Apple proposed *HTTP Live Streaming*, which is a new open standard for live video streaming over HTTP, currently submitted as an IETF Internet-Draft [132]. It is able to send live or pre-recorded media to iPhones (and other devices). Any ordinary HTTP Web server can be used; for clients however, currently only QuickTime X (or later) and the player that comes with the iPhone OS 3.0 (or later) support HTTP Live Streaming.

In order to serve a media resource as an HTTP Live Streamable resource, the media resource needs to be segmented and stored as separate segments on the server. During this segmentation phase, an index file is created, containing a list of these media segment files and additional metadata. The format of the index file is an extension of the format used for MP3 play-lists (M3U[28]), i.e., M3U8. The workflow for delivering media using HTTP Live Streaming is then as follows:

1. The user agent fetches the index file of the desired media resource from the server.
2. Based on the index file, the user agent knows the location of the available media segment files.
3. The user agent starts downloading each available media segment file in sequence.

[28] `http://en.wikipedia.org/wiki/M3U/`

Listing 6.1: Describing alternative Versions of a Media Resource within an M3U8 Index File.

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1192
http://ninsuna/LiveStreamingServlet/media.m3u8?track=1;4
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=596
http://ninsuna/LiveStreamingServlet/media.m3u8?track=2;5
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=296
http://ninsuna/LiveStreamingServlet/media.m3u8?track=3;5
#EXT-X-ENDLIST
```

One of the features that comes with HTTP Live streaming is dynamic stream switching. To realise this, the server maintains multiple versions of the same multimedia content. Since the user agent requests small segments, the version of the multimedia content can be switched during the consumption of a particular media resource (e.g., due to varying network conditions). More specifically, the user agent can decide to request the next segment from an alternative version (which is then for instance characterised by a lower bit rate). We implemented HTTP Live streaming, inclusive dynamic stream switching, within NinSuna, using Media Fragment URIs. It is important to notice that we did not cut the media resources into different segments (as done in regular HTTP Live streaming set-ups). Instead, we use Media Fragment URIs to point to these (virtual) segments. Hence, it is not necessary to physically separate these segments [174]. In particular, the following extensions into NinSuna were necessary:

- MPEG-2 Transport Stream (MPEG-2 TS, [85]) packaging filter: currently, practical implementations of HTTP Live streaming user agents only support MPEG-2 TS as container format;
- M3U8 generation service: based on a requested media resource, a corresponding index file needs to be generated containing pointers to the different segments.

For our media.mp4 example, the generated M3U8 index file is depicted in Listing 6.1. As one can see, three alternative versions are specified for the media resource, each containing a different bit rate. As discussed above, the alternative versions can be represented by means of Media Fragment URIs, by making use of URI query parameters to specify the correct tracks. However, note that we do not point to a version of the media resource itself, but to the M3U8 index file of that media resource (e.g., `media.m3u8?track=1;4`). Given an M3U8 index file providing a number of alternative versions for a particular media resource, the user agent decides which version has to be retrieved, given the constraints of its usage environment (e.g., available bandwidth, available battery power). For the chosen version, the user agent retrieves the corresponding M3U8 index file; an example of the high quality version for our media.mp4 example is depicted in Listing 6.2. In the latter M3U8 index file, we see that the media resource is divided in temporal segments by means of Media Fragment URIs using the query parameter to indicate the correct tracks and using a fragment identifier to indicate the temporal range. More specifically, the media resource `media.ts?track=1;4` is divided in temporal ranges where each range is approximately 11s. Note that the boundaries of the media fragments occurring in the M3U8 index file are dependent on the random access points that occur in the media resource, which is the reason that the actual duration of each fragment differs from the target duration. The media fragments located in the M3U8 index file point to media resources packaged in an MPEG-2 Transport Stream container. These fragments can be requested and downloaded from the NinSuna server, as explained in subsection 6.5.2.2. As described extensively in above subsections, our model-driven media delivery platform NinSuna aims at being a server-side implementation of the Media Fragments URI 1.0 specification. More specifically, the platform is able to deliver temporal and track fragments by using efficient media segment extraction methods. Moreover, since the platform is independent of underlying media formats, it is also a generic solution to resolve legacy Media Fragments URIs, cf., our Media Fragments Translation Service being a Media Fragments proxy. Currently, two demos (available online at `http://ninsuna.elis.ugent.be/mediafragments/`) exist demonstrating the W3C Media Fragments Specification:

- Server-side Media Fragment extraction: both query- and fragment-based Media Fragments URIs are demonstrated.
- Client-side Media Fragment rendering: a Media Fragments media player/visualiser (both a Flash and HTML5 implementation are available).

Listing 6.2: Dividing a Media Resource into Packets by means of Media Fragment URIs.

```
#EXTM3U
#EXT-X-TARGETDURATION:11.0
#EXT-X-MEDIA-SEQUENCE:0
#EXTINF:10,
http://ninsuna/DownloadServlet/media.ts?track=1;4#t=0,10.1
#EXTINF:10,
http://ninsuna/DownloadServlet/media.ts?track=1;4#t
    =10.1,20.2
#EXTINF:10,
http://ninsuna/DownloadServlet/media.ts?track=1;4#t
    =20.2,30.3
#EXT-X-ENDLIST
```

Chapter 7

Conclusions

I want to be well known and famous and hundreds of years from now I want my life's story to be told weekly at 9:30.

Lord Blackadder in Blackadder

News is something nearly every European citizen reads, watches, or listens to on a daily basis, at home, while commuting to and from work, at work, and even as part of their work. As voting citizens, we need to understand local, national, and international politics to allow us to cast our vote. As company employees, we need to understand the state and development of local, national, and international economies to enable us to understand our markets. As part of our leisure time, we want to know about our favourite sports teams, the lives of our soap idols, or the most recent books available. Nowadays, this information is on-line, and hence accessible from anywhere.

In existing news workflow processes, news items are typically produced by news agencies, independent journalists or citizen media; afterwards consumed and enhanced by newspapers, magazines or broadcasters; then delivered to end-users; and finally perceived by these end-users that further leave a trace of what they have understood and felt facing these news events using blogs and tweets as means of expression. News items should therefore typically be accompanied by a set of metadata and descriptions that facilitate their storage, retrieval, and life cycle. However, much of the metadata is lost because of interoperability problems occurring along the production workflow. As such, opportunities for making use of the available metadata at the user interface level are often lost.

Consequently, users are overwhelmed by too many individual and disconnected pieces of information, and cannot situate the news in a proper context. A news event is defined as a cluster of statistically related news items, but no ontological notion of event is supported. In contrast, the organisation of news providers is centred on the notion of scheduled and breaking news events, but they currently lack the tools necessary to relate easily the news they produce to the events they manage on a daily basis. Semantic processing of news information can improve the clustering and organisation of individual news items –from heterogeneous sources, in multiple media types and in multiple languages– into meaningful events linked to appropriate background knowledge.

In this thesis, we state that *Interoperability of Semantics* is therefore key throughout the different workflow phases of news production. Firstly, within each media company all ranks of the organisation should be able to reason about a problem across levels given joint concepts, thereby closing the '*comprehensive semantic gap*' that historically exists between business management and technical engineering, thus smoothing the workflow between the organisation's entities and leading to a technical sound solution understood by everyone. Secondly, metadata should be identified, captured, and standard-based semantically described as soon as possible, thereby closing the '*technical semantic gap*' as different in-house systems down the workflow pipeline, as well as third party Internet Web services, will be able to harness this extra knowledge to their own favour. Finally, the gathered knowledge, i.e., the interlinked news items, should be archived in a future-safe way, thereby closing the '*continuity semantic gap*' as metadata, e.g., provenance data, on metadata will be key to long-lasting interoperability.

In Chapter 2, we were the first to align the end-to-end news production processes with the canonical processes, making all ranks within a broadcast organisation understand the process cycles of their systems in the context of the more generalised, standard process cycles of existing media systems, thus closing the aforementioned '*comprehensive semantic gap*'. The canonical processes help us in clarifying the complex interleaving of workflow processes on the different levels of responsibilities within an organisation. Furthermore, one can now easily envisage future scenarios where some of the processes within a system can be exchanged with those from other media production systems that adhered to aforementioned generic principles, e.g., production houses and local broadcasters. As such, we set up a framework to establish clear interfaces for the information flow across media processes among distinct news production phases so that compatibility across systems from different providers can be achieved. By identifying recurring and canonical functionality, process implementations are simplified and input and output from different processes are coordinated for better integration with these external systems. Remember that the workflow information generated and gathered during the production, the so-called metadata, must become an integral part of the production process and must be modelled and employed as soon as possible by production systems throughout all phases. This successful mapping then allows implementers of new manufacturing methods and of foreign integrating systems to better coordinate processes in terms of inputs, outputs, and functionality. Further work is also envisioned by us within W3C's MAWG for standardisation of (news) media applications' portability. Interdisciplinary research could also be done on identifying the different actors and their respective 'profiled' canonical processes within the whole end-to-end news production workflow chain, as production houses and local broadcasters are only covering certain parts of this end-to-end workflow chain.

In Chapter 3, we have presented a semantic version of the NAR/NewsML-G2 standard as a unifying (meta)data model dealing with dynamic distributed news event information. Using that ontology as a data communication interface within an end-to-end news distribution architecture, several services (aggregation, categorisation, enrichment, profiling, recommendation, and distribution) were hooked in the workflow engine giving broadcasters a tool to automatically recommend (developing) news stories 1-to-1 to the targeted customer for the first time. At the same first time, we provided the (inter)national (news) community with mechanisms to describe and exchange news events and profile information in a standardised way, thus contributing to the narrowing of the aforementioned '*technical semantic gap*'. We demonstrated the concepts of generic data portability of user profiles, and how to generate recommendations based on such a global profile –within which we integrated information fields from all the different social networks the user wanted to share. Our ideas were implemented with open standards like *OpenID*, *OAuth*, and *OpenLike*, thus keeping the architecture open for other news event providers and profile providers. Further work is again envisioned by us within W3C's MFWG & MAWG making annotations of (news related) media objects accessible in a uniform way, disregarding the original format of their metadata. The challenge will be how to reconcile multimedia ontologies with news ontologies and domain specific vocabularies for solving these interoperability issues along the complete news workflow.

Furthermore, the origin of the news source is very important in the context of on-line news items. In this aspect, knowledge on the creator or issuer of the news gives more actual contextual information to the specific news item. Additionally, social links of the creator to other people or organisations are highly relevant when determining the context of a news item. *OpenSocial* is a set of common APIs for building social applications across many Web sites. However, to integrate this information according to the aforementioned LOD initiative, a formal representation is needed. A first initiative has been made to see how the Social Web can be integrated with the Semantic Web by W3C's Social Web Incubator Group. Finally, journalists often stress the absolute need of representing the provenance of all types of information in order to trigger confidence regarding the truthfulness of a news event report. Recently, the creators of the Open Provenance Model have started the W3C Provenance Incubator Group in which we participate and that will introduce the Open Provenance Model in the Semantic Web. As such, this issue within the complete news production workflow will get solved, as provenance will proof to be an important topic too, when news is to be archived for future-proof disclosure.

In Chapter 4, we identified the necessary types of metadata that need to be retained when preserving digital (news) information for the long-term. Descriptive metadata are needed to describe the intellectual entities, whereas binary metadata, technical metadata, and structural metadata are essential for the description of the data on all lower levels (bitstream, file, and representation). Preservation metadata is also necessary to describe the provenance of the data, to guarantee the authenticity of its digital nature, and to provide a context. At last, rights metadata also need to be stored. We proposed a two-layered, semantic metadata schema that offers the freedom to embrace all of these metadata types within the archiving context. We were the first to come up with this generic architecture for both open access and lasting archive, which were until recently considered orthogonal and not compatible. Our top layer (the RDFS representation of DC) takes care of the descriptive metadata and is also usable to initiate the exchange of multimedia data from different domains with non-homogeneous metadata. Our bottom layer, which also became the official OWL representation of PREMIS 2.0 (acknowledged by both the PREMIS standardisation board and Library of Congress), on the other hand, encompasses the needed binary metadata, technical metadata, structural metadata, preservation metadata, and the rights metadata for future-safe archiving. By describing the data with this layered metadata schema, all the risks that come with long-term preservation are minimised. By splitting up the semantic schema in two layers, the top layer with the descriptive metadata can be made public and weaved into the Web of data, if the rights permit it. The bottom layer can remain closed for the public and is responsible for the long-term preservation of that data. For (news) data providers –be it a certain local broadcaster or its coordinating European archival counterpart– that aim at integrating their OAI-PMH data end-points into the LOD cloud, we recommend from our experience to follow the OAI-PMH guidelines and expose their metadata also in other formats than DC, as we did using our layered semantic metadata model. Regarding the OAI-ORE developments, we can observe that the LOD principles already play an important role in the news use case domain. We have also seen that the aforementioned conceptual *continuity semantic gap* between OAI-PMH and OAI-ORE is narrow and can easily be bridged by intermediate gateways like our enhanced OAI2LOD Server. Since the LOD approach actually subsumes a large fraction of the OAI-PMH functionalities, we believe that future releases of the OAI-PMH standard should even consider a shift towards the LOD principles, which would also enable a tighter integration with the newer OAI-ORE protocol. Meanwhile, our enhanced OAI2LOD Server can be used for bridging this conceptual *continuity semantic gap* between all of these standards.

In Chapter 5, we presented the rational for a Media Fragments specification. Needless to say that this Media Fragments specification will have a major impact on the complete end-to-end news production chain. From the moment news footage is shot, edited and enriched with extra information down the news production workflow chain, until a specifically chosen news item is viewed by an interested end-user, one will be able to uniquely identify, link to, display, browse, bookmark, re-composite, annotate, and/or adapt spatial and/or temporal sub-clips of media resources, e.g., a camera might automatically annotate footage with the exact geo-coordinates the moment it is shot, a news editor might quickly browse through a months' footage by means of highlight captions in search of one particular item, whereas an end-user might create a video mash-up from his bookmarked video segments to share with his friends on a social platform. As such, we outlined the boundaries and semantics of a Media Fragments URI and showed how the syntax should look like. We also elaborated on how a media fragment specified as an URI fragment & URI query can be resolved stepwise using the HTTP protocol, and finally, we identified the influence of current media formats on such fragment extraction. We were the first to have developed an HTTP implementation for the W3C Media Fragments 1.0 specification (both a Firefox client-side plugin and a Ninsuna server-side implementation) in order to verify all the test cases defined by our working group. Furthermore, Opera already has implemented the time recipe in its new alpha version and it is to be expected that all major browser vendors will include support for the Media Fragments time recipe in its upcoming versions. Also HTML5 already foresees support for this time recipe within its media elements. In the near future, we also foresee reference implementations for the RTSP, RTMP, and File protocols. In the meantime, this Media Fragments URI specification, which we edited and contributed to through our W3C standardisation activities over the last couple of years, now really opens up time-related media to the Internet crowd and makes time-related annotation feasible for hyperlinking into the media, thus providing the necessary support for this already omnipresent ‘third dimension’ *time* into the Internet.

In Chapter 6, we put into practise what has been elaborated on in all the previous chapters, i.e., the overall use case being end-to-end news production. Proof-of-concepts from the Archipel-project, the BOM-Vl-project, the CUPID-project, and the PISA-project (in alphabetical order, as I worked on all of these projects with the same eagerness and enthusiasm) were (partly) altered to satisfy this use case. As such, we discussed a ‘search use case’, a ‘distribution use case’, an ‘archiving use case’, and last but not least a ‘Media Fragments use case’ within end-to-end news production using all concepts covered from all the previous chapters.

To conclude this dissertation, we hope that we have convinced the reader that our work, although limited in scope, at least partly contributed to bridging the (semantic) gaps in end-to-end (both start-to-end, and top-to-bottom) news production. If you ask me, I am most proud of being the first to advocate Semantic Web technologies in our lab, IBBT in general, the Flemish broadcast industry, and the Flemish cultural & archival scene. Just seeing the eager uptake of this technology over the last couple of years in the aforementioned organisations, makes up for the numerous hours of sleep I have been deprived of over the last 4 years.

Appendix A

Summary of Canonical Processes of News

A.1 Summary of Canonical Processes of News

In this annex, a summary is made of the canonical processes of news production operations, as described in Chapter 2.

Table A.1: Canonical Processes of News Production Summary

Canonical process	**News production operations summary**
Premeditate	1) The *SOP* formally defines the news of a single day as a product, and specifies that each news bulletin on radio or television is an instance of this product. Broadband internet and mobile access offer a version of the news bulletin, which is continuously being modified. Through an explicit or implicit sales process, the sales order receives sales orders that trigger news bulletin composition, production, mastering, and eventually distribution processes. Input: the business strategy of the media production facility in the form of forecast and production portfolio; a sales order delivered by the sales process. Output: a generic BOM; production orders; distribution schedules. 2) The *Sales* process triggers the SOP by using explicit or implicit user feedback. Input: a buy transaction confirmation (explicit) from the consumer or a commissioning decision (implicit) from the producer. Output: a sales order that triggers the planning and production processes.

Table A.2: Canonical Processes of News Production Summary (continued)

Canonical process	News production operations summary
Construct message	*Story-editing* is a continuous process during the news day whereby the news reporter specifies the details of the content of each news item, including a specification of the media assets that will be used to illustrate the item. Input: external input in the form of news events. Output: partial or complete editorial objects; formal or informal descriptions of raw material and semi-finished components that needs to be produced to illustrate the news item.
Organise	*News bulletin composition* comprises the preparation of a news day. Using various sources such as newswires and dedicated calendar applications, the news reporters select content of the news day. Input: a generic BOM issued by the SOP; editorial objects of type news item. Output: a rundown is a composite artefact, which will be used as input for further production (create media asset) and mastering (publishing) processes. It is an ordered document structure that includes item descriptions that contain pointers to the underlying media objects.
Create media asset	1) *Goods receipt* is in the context of news production, audio-visual footage usually originating from correspondents abroad or news agencies. Input: audio-visual footage. Output: raw material media assets in the form of individual shots or newsfeeds; a dope sheet that describes the received material. 2) *Electronic news gathering* is the process of capturing material in the context of an anticipated news item. Input: a component description delivered by the story-editing process. Output: raw material media assets in the form of individual shots; creation metadata (camera, geographical position, time, etc.). 3) *Material-editing* is a production process, not necessarily news-related, whereby an editor uses an editing workstation to process existing audio-visual material in order to deliver new media assets which are here referred to as news clips. Input: a component description delivered by the story editing process; a material package issued by the material warehouse. Output: semi-finished material components in the form of news clips; creation metadata (e.g., refers to the original material).

Table A.3: Canonical Processes of News Production Summary (continued)

Canonical process	News production operations summary
Annotate	*Material annotation* is part of the material warehousing system, during which all available information related with the received material is collected, normalised, translated to match reference data, and potentially additional descriptive information is added by unsupervised analysis algorithms. Input: all available information related to the media asset that is being received by the material warehouse system, e.g., dope sheets, editorial object descriptions, component descriptions, creation metadata, and the logistic context of the finished product, i.e., a project or a programme. Output: a formal, multi-dimensional, and normalised description of a media asset, often using reference data or a thesaurus if possible.
Package	The *Material warehouse* system receives various types of media assets, including raw material, semi-finished components, or finished products. These are stored and uniquely identified, to be able to issue material at a later stage for further production, mastering, or redistribution. The material warehouse indexes the formal description delivered by the material annotation processes, based on which the warehouse can be efficiently managed and queried if applicable. By far the most important task of the material warehouse system is *material picking*, an implementation of *package*. Input: a material asset delivered by a media creation process, e.g., goods receipt, electronic news gathering, material editing; a formal description of the media asset delivered by the material annotation processes. Output: a material package, containing one or more media assets including embedded identification and annotation.
Query	Although being an important functionality of any one material warehouse/MAM system, the query process is not that visible in regular news production, unless archived material has to be searched for to be reused for documentary purposes. Of course, implicit queries are needed, for example, to search for and fetch the relevant material within a material package if it is needed within the material-editing process. In this particular case, identifiers of media assets and the formal description (which is in its turn an aggregate of identifiers to all kinds of annotations) are needed to build the underlying query.

Table A.4: Canonical Processes of News Production Summary (continued)

Canonical process	News production operations summary
Publish	*Mastering* is the process of packaging and labelling as function of the distribution channel, client capabilities, and/or user preferences. Input: a distribution order; an approved rundown issued by the news production process; a material package issued by the material warehouse. Output: one or more finished products (linear or non-linear assemblies) that are ready for distribution.
Distribute	The *distribution* processes are triggered by a distribution order issued by the sales and operations planning process. By transmission, physical distribution, or any other means of making it available, it issues the finished product delivered by the mastering process following the specifications of the distribution order (distribution channels, window of availability, conditional access, etc.). Input: a finished product (linear or non-linear assembly) delivered by the mastering process. Output: a product is delivered to the outside world.

Appendix B

NAR/NewsML and Global Profile Ontology

B.1 NAR/NewsML Ontology

B.1.1 NAR Ontology

Listing B.1: NAR Ontology.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix nar: <http://multimedialab.elis.ugent.be/ontologies
    /NAR/v0.1/NAR.rdf#>.

nar:AggregateComponent  a rdfs:Class;
  rdfs:subClassOf nar:Component.

nar:AnyItem             a rdfs:Class;

nar:AudioNewsItem       a rdfs:Class;
  rdfs:subClassOf nar:NewsItem.

nar:BasicComponent      a rdfs:Class;
  rdfs:subClassOf nar:Component.

nar:ComplexDatatype     a rdfs:Class;
  rdfs:subClassOf nar:Datatype.

nar:Component           a rdfs:Class;

nar:ConceptItem         a rdfs:Class;
```

```
  rdfs:subClassOf nar:AnyItem.

nar:Datatype            a rdfs:Class;
  rdfs:subClassOf nar:Component.

nar:Generic             a rdfs:Class;
  rdfs:subClassOf nar:ConceptItem.

nar:GeopoliticalArea    a rdfs:Class;
  rdfs:subClassOf nar:ConceptItem.

nar:GraphicNewsItem     a rdfs:Class;
  rdfs:subClassOf nar:NewsItem.

nar:KnowledgeItem       a rdfs:Class;
  rdfs:subClassOf nar:AnyItem.

nar:NewsItem            a rdfs:Class;
  rdfs:subClassOf nar:AnyItem.

nar:Organisation        a rdfs:Class;
  rdfs:subClassOf nar:ConceptItem.

nar:PackageItem         a rdfs:Class;
  rdfs:subClassOf nar:AnyItem.

nar:Person              a rdfs:Class;
  rdfs:subClassOf nar:ConceptItem.

nar:PhotoNewsItem       a rdfs:Class;
  rdfs:subClassOf nar:NewsItem.

nar:PointOfInterest     a rdfs:Class;
  rdfs:subClassOf nar:ConceptItem.

nar:RemoteContent       a rdfs:Class;

nar:SimpleDatatype      a rdfs:Class;
  rdfs:subClassOf nar:Datatype.

nar:TextNewsItem        a rdfs:Class;
  rdfs:subClassOf nar:NewsItem.

nar:VideoNewsItem       a rdfs:Class;
  rdfs:subClassOf nar:NewsItem.

nar:caption             a rdfs:Property;
  rdfs:subPropertyOf nar:description.
```

```
nar:captionWriter       a rdfs:Property;
  rdfs:subPropertyOf nar:contributor.

nar:channel             a rdfs:Property.
nar:contentClass        a rdfs:Property.
nar:contentCreated      a rdfs:Property.
nar:contributor         a rdfs:Property.
nar:creator             a rdfs:Property.
nar:date                a rdfs:Property.
nar:description         a rdfs:Property.
nar:filename            a rdfs:Property.
nar:headline            a rdfs:Property.
nar:height              a rdfs:Property.
nar:icon                a rdfs:Property.
nar:infoSource          a rdfs:Property.
nar:itemCreated         a rdfs:Property.
nar:language            a rdfs:Property.
nar:locCreated          a rdfs:Property.
nar:modified            a rdfs:Property.
nar:newsItem            a rdfs:Property.
nar:newsReader          a rdfs:Property.
nar:packageItem         a rdfs:Property.
nar:priority            a rdfs:Property.
nar:provider            a rdfs:Property.
nar:pubStatus           a rdfs:Property.
nar:remoteContent       a rdfs:Property.
nar:rendition           a rdfs:Property.

nar:schema              a rdfs:Property;
  rdfs:domain nar:AnyItem;
  rdfs:range xsd:string.

nar:sender              a rdfs:Property.
nar:service             a rdfs:Property.
nar:size                a rdfs:Property.
nar:slugline            a rdfs:Property.
nar:story               a rdfs:Property.
nar:subject             a rdfs:Property.
nar:title               a rdfs:Property.
nar:transmitId          a rdfs:Property.
nar:version             a rdfs:Property.
nar:width               a rdfs:Property.
nar:Datatype            a rdfs:Class.
nar:contributor         a rdfs:Property.
nar:description         a rdfs:Property.
```

B.1.2 NewsML Item Instance

Listing B.2: NewsML Item Instance.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:nar="http://multimedialab.elis.ugent.be/
        ontologies/NAR/v0.1/NAR.rdf#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:mmo="http://multimedialab.elis.ugent.be/
        ontologies/MMO/multimedia_model.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Description rdf:about="http://multimedialab.elis.
      ugent.be/ontologies/newsML/urn_newsml_vrtnieuws.
      net_20071208233800_00-00-23u Journaal.rdf#
      pi_20071208233800_09-02-Congo">
    <nar:subject rdf:resource="http://multimedialab.elis.
        ugent.be/ontologies/entities.rdf#Freek_Braeckman"/>
    <mmo:related>
      <mmo:TemporalSegment rdf:about="http://multimedialab.
          elis.ugent.be/ontologies/newsML/
          urn_newsml_vrtnieuws.net_20071208233800_00-00-23u
          Journaal.rdf#ts_54">
        <mmo:duration>PT6S170N1000F</mmo:duration>
        <mmo:start>T00:05:47:03F25</mmo:start>
      </mmo:TemporalSegment>
    </mmo:related>
    <nar:creator>VRT - Vlaamse Radio &amp; Televisie</
        nar:creator>
    <nar:genre>Graphic</nar:genre>
    <nar:subject rdf:resource="http://dbpedia.org/resource/
        Congo"/>
    <nar:locCreated rdf:resource="http://sws.geonames.org
        /2800866/"/>
    <nar:language>nl-BE</nar:language>
    <nar:title>Congo</nar:title>
    <nar:locCreated rdf:resource="http://sws.geonames.org
        /2802361/"/>
    <nar:itemCreated>2007-12-08T23:38:00+01:00</
        nar:itemCreated>
    <nar:headline>Congo</nar:headline>
    <nar:provider>VRT - Vlaamse Radio &amp; Televisie</
        nar:provider>
    <mmo:related>
      <mmo:TemporalSegment rdf:about="http://multimedialab.
          elis.ugent.be/ontologies/newsML/
          urn_newsml_vrtnieuws.net_20071208233800_00-00-23u
          Journaal.rdf#ts_53">
        <mmo:duration>PT2S889N1000F</mmo:duration>
        <mmo:start>T00:05:44:13F25</mmo:start>
      </mmo:TemporalSegment>
    </mmo:related>
  </rdf:Description>
</rdf:RDF>
```

B.2 Global Profile Ontology

Listing B.3: Global Profile Ontology.

```
@prefix globalprofile: <http://multimedialab.elis.ugent.be/
    ontologies/Profile/v1.0/GlobalProfile.rdf#>.
@prefix nar: <http://multimedialab.elis.ugent.be/ontologies
    /NAR/v0.1/NAR.rdf#>.

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/#>.
@prefix sioc: <http://rdfs.org/sioc/ns#>.

globalprofile:consumed a rdf:Property;
  rdfs:label    "attended";
  rdfs:comment  "links a person to a news event he/she
      consumed";
  rdfs:domain   foaf:Person;
  rdfs:range    nar:NewsItem.

globalprofile:goodRec a rdf:Property;
  rdfs:label    "good recommendation";
  rdfs:comment  "the persons thinks the news event is a
      good recommendation for him/her";
  rdfs:domain   foaf:Person;
  rdfs:range    nar:NewsItem.

globalprofile:badRec a rdf:Property ;
  rdfs:label    "bad recommendation";
  rdfs:comment  "the persons thinks the news event is a bad
       recommendation for him/her";
  rdfs:domain   foaf:Person;
  rdfs:range    nar:NewsItem.
```

Appendix C

Layered Semantic Metadata Model Instance Examples

C.1 PREMIS Ontology Instance Example

C.1.1 Object Instance Example

Listing C.1: PREMIS Ontology - Object Instance.

```
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns
    #> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix premisowl:  <http://multimedialab.elis.ugent.be/
    ontologies/Premis2.0/v1.0/premis.owl#>.

<object1> a   premisowl:File;
  premisowl:objectIdentifier
    <object1ID>;
  premisowl:preservationLevel
    <object1PreservationLevel>;
  premisowl:significantProperties
    <object1SignificantProperties>;
  premisowl:objectCharacteristics
    <object1ObjectCharacteristics>;
  premisowl:originalName            "0001h.tif";
  premisowl:storage
    <object1Storage>;
  premisowl:environment
    <object1Environment>;
  premisowl:relationship
    <object1Relationship1>;
  premisowl:linkingEvent
    <event1>;
  premisowl:linkingRightsStatement
    <rightsstatement1>;
  premisowl:linkingIntellectualEntity
    <dublinCoreDescription1>;
  .

<object1ID> a   premisowl:ObjectIdentifier;
  premisowl:identifierType           "hdl";
  premisowl:identifierValue          "archipel.music/
      gottlieb.09601";
  .

<object1PreservationLevel> a    premisowl:PreservationLevel
    ;
  premisowl:preservationLevelValue   "0";
  premisowl:preservationLevelRole    "master copy";
  premisowl:preservationLevelDateAssigned "2010-07-29
      T14:41:28";
  .

<object1SignificantProperties> a
    premisowl:SignificantProperties;
  premisowl:significantPropertiesType "behavior";
  premisowl:significantPropertiesValue "hyperlinks
      traversable";
  .

<object1ObjectCharacteristics> a
    premisowl:ObjectCharacteristics;
  premisowl:compositionLevel         "0";
  premisowl:fixity
    <object1Fixity>;
  premisowl:size                     "20800896";
  premisowl:format
    <object1Format>;
  premisowl:creatingApplication
    <object1CreatingApplication1>;
  premisowl:creatingApplication
    <object1CreatingApplication2>;
  premisowl:objectCharacteristicsExtension
    <object1CharacteristicsExtension>;
  .

<object1Fixity> a   premisowl:Fixity;
  premisowl:messageDigestAlgorithm   "MD5";
  premisowl:messageDigest            "36
      b03197ad066cd719906c55eb68ab8d";
  premisowl:messageDigestOriginator  "LocalDCMS";
  .

<object1Format> a   premisowl:Format;
  premisowl:formatDesignation
    <object1FormatDesignation>;
  premisowl:formatRegistry
    <object1FormatRegistry>;
  .

<object1FormatDesignation> a    premisowl:FormatDesignation
    ;
  premisowl:formatName               "image/tiff";
  premisowl:formatVersion            "6.0";
  .

<object1FormatRegistry> a   premisowl:FormatRegistry;
  premisowl:formatRegistryName       "PRONOM";
  premisowl:formatRegistryKey        "fmt/10";
  premisowl:formatRegistryRole       "specification";
  .

<object1CreatingApplication1> a
    premisowl:CreatingApplication;
  premisowl:creatingApplicationName  "ScandAll 21";
  premisowl:creatingApplicationVersion "4.1.4";
  premisowl:dateCreatedByApplication  "1998-10-30T08:28:32"
      ;
  .

<object1CreatingApplication2> a
    premisowl:CreatingApplication;
  premisowl:creatingApplicationName    "Adobe Photoshop";
  premisowl:creatingApplicationVersion "CS2";
  premisowl:dateCreatedByApplication   "2006-09-20T08:29:02
      ";
  .

<object1Storage> a    premisowl:Storage;
  premisowl:contentLocation
    <object1ContentLocation>;
  premisowl:storageMedium              "disk";
  .

<object1ContentLocation> a    premisowl:ContentLocation;
  premisowl:contentLocationType        "filepath";
  premisowl:contentLocationValue       "amserver";
  .
```

```
<object1Environment> a    premisowl:Environment;
  premisowl:environmentCharacteristic  "recommended";
  premisowl:environmentPurpose         "render";
  premisowl:environmentPurpose         "edit";
  premisowl:software
    <object1Software1>;
  premisowl:hardware
    <object1Hardware1>;
  .

<object1Software1> a    premisowl:Software;
  premisowl:swName                     "Adobe Acrobat";
  premisowl:swVersion                  "5.0";
  premisowl:swType                     "renderer";
  .

<object1Hardware1> a    premisowl:Hardware;
  premisowl:hwName                     "Intel x86";
  premisowl:hwType                     "processor";
  premisowl:hwOtherInformation         "60 mhz minimum";
  .

<object1Relationship1> a    premisowl:Relationship;
  premisowl:relationshipType           "derivation";
  premisowl:relationshipSubType        "is source of";
  premisowl:relatedObjectIdentifier
    <object1Relationship1ObjectIdentifier>;
  premisowl:relatedEventIdentifier
    <object1Relationship1EventIdentifier>;
  .

<object1Relationship1ObjectIdentifier> a
    premisowl:RelatedObjectIdentifier;
  premisowl:relatedObjectIdentificationType    "URL";
  premisowl:relatedObjectIdentificationValue "http://
      archipellod.demo.ibbt.be:8080/object/gottlieb/09601/
      ver01/0001v.jpg";
  premisowl:relatedObjectSequence            "0";
  .

<object1Relationship1EventIdentifier> a
    premisowl:RelatedEventIdentifier;
  premisowl:relatedEventIdentifierType  "LocalDCMS";
  premisowl:relatedObjectIdentifierValue "E002.1";
  premisowl:relatedObjectSequence       "1";
  .
```

C.1.2 Event Instance Example

Listing C.2: PREMIS Ontology - Event Instance.

```
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix premisowl:  <http://multimedialab.elis.ugent.be/ontologies/Premis2.0/v1.0/premis.owl#>.

<event1> a  premisowl:Event;
  premisowl:eventIdentifier          <event1ID>;
  premisowl:eventType                "dissemination migration";
  premisowl:eventDateTime            "2010-08-06T00:00:00.002";
  premisowl:eventDetail              "ImageMagick";
  premisowl:eventOutcomeInformation    <event1OutcomInformation>;
  premisowl:linkingAgent             <agent1>;
  premisowl:linkingObject            <object1>;
  premisowl:linkingObject            <object2>;
  .

<event1ID> a  premisowl:EventIdentifier;
  premisowl:identifierType           "LocalDCMS";
  premisowl:identifierValue          "E002.1";
  .

<event1OutcomeInformation> a premisowl:EventOutcomeInformation;
  premisowl:eventOutcome             "successful";
  .
```

C.1.3 Agent Instance Example

Listing C.3: PREMIS Ontology - Agent Instance.

```
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix premisowl:  <http://multimedialab.elis.ugent.be/ontologies/Premis2.0/v1.0/premis.owl#>.
```

```

<agent1> a  premisowl:Event;
  premisowl:agentIdentifier          <agent1ID>;
  premisowl:agentType                "person";
  premisowl:agentName                "Sarah Vandeputte";
  premisowl:linkingAgent             <agent1>;
  premisowl:linkingObject            <object1>;
  premisowl:linkingObject            <object2>;
  .

<agent1ID> a  premisowl:AgentIdentifier;
  premisowl:identifierType           "OpenID";
  premisowl:identifierValue          "http://svandeputte.
      archipelopenID.be";
  .
```

C.1.4 Rights Instance Example

Listing C.4: PREMIS Ontology - Rights Instance.

```
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns
    #> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix premisowl:  <http://multimedialab.elis.ugent.be/
    ontologies/Premis2.0/v1.0/premis.owl#>.

<rights1> a premisowl:License;
  premisowl:rightsStatementIdentifier <rights1ID>;
  premisowl:rightsBasis               "license";
  premisowl:licenseInformation        <licenseInformation1>
      ;
  premisowl:rightsGranted             <rightsGranted1>;
  premisowl:linkingObject             <object1>;
  premisowl:linkingObject             <object2>;
  premisowl:linkingAgent              <>;
  .

<rights1ID> a premisowl:RightsStatementIdentifier;
  premisowl:identifierType            "URL";
  premisowl:identifierValue           "http://archipellod.
      demo.ibbt.be:8080/rights/resource/dissemination";
  .

<licenseInformation1> a premisowl:LicenseInformation;
  premisowl:licenseIdentifier         <license1identifier>;
  premisowl:licenseTerms              "Here comes the
      actual text of the real license.";
  premisowl:licenseNote               "These objects may be
       disseminated.";
  .

<license1identifier> a  premisowl:LicenseIdentifier;
  premisowl:identifierType            "URL";
  premisowl:identifierValue           "http://archipellod.
      demo.ibbt.be:8080/license/resource/dissemination";
  .

<rightsGranted1> a  premisowl:LicenseInformation;
  premisowl:act                     <license1identifier>;
  premisowl:termOfGrant             <license1termofgrant>;
  .

<license1termofgrant> a premisowl:TermOfGrant;
  premisowl:startDate               "2009-09-01T08:30:00";
  .
```

C.2 OAI-ORE Aggregation Instance Example

Listing C.5: OAI-ORE Aggregation Instance.

```
<!-- About the Aggregation for the Archipel document -->
<rdf:Description rdf:about="http://archipel.org/aggregation
    /HugoClausCollection/01">
  <!-- The Resource is an ORE Aggregation  -->
  <rdf:type rdf:resource="http://www.openarchives.org/ore/
      terms/Aggregation"/>
    <!-- The Aggregation aggregates ... -->
    <ore:aggregates rdf:resource="http://archipel.org/tiff/
        HugoClaus/01"/>
    <ore:aggregates rdf:resource="http://archipel.org/tiff/
        HugoClaus/02"/>
    <ore:aggregates rdf:resource="http://archipel.org/tiff/
        HugoClaus/03"/>
    <!-- Metadata about the Aggregation: title and authors
        -->
    <dc:title>Aggregation aggregating all tiff images of
        Hugo Claus</dc:title>
    <dcterms:creator rdf:parseType="Resource">
      <foaf:name>Erik Mannens</foaf:name>
      <foaf:mbox rdf:resource="mailto:erik.mannens@ugent.be
          "/>
    </dcterms:creator>
</rdf:Description>

<!-- About the Resource Map (this RDF/XML document) that
    describes the Aggregation -->
<rdf:Description rdf:about="http://archipel.org/rem/
    HugoClausCollection/01">
  <!-- The Resource is an ORE Resource Map  -->
  <rdf:type rdf:resource="http://www.openarchives.org/ore/
      terms/ResourceMap"/>
    <!-- The Resource Map describes a specific Aggregation
          -->
    <ore:describes rdf:resource="http://archipel.org/
        aggregation/HugoClausCollection/01"/>
    <!-- Metadata about the Resource Map: datetimes, rights
        , and author -->
    <dcterms:modified>2009-08-12T07:30:34Z</
        dcterms:modified>
    <dcterms:created>2009-08-11T18:30:02Z</dcterms:created>
    <dc:rights>This Resource Map is available under the
        Creative Commons license</dc:rights>
    <dcterms:rights rdf:resource="http://creativecommons.
        org/licenses/by-nc/2.5/rdf"/>
    <dcterms:creator rdf:parseType="Resource">
      <foaf:page rdf:resource="http://archipel.org/"/>
      <foaf:name>Archipel Archive</foaf:name>
    </dcterms:creator>
</rdf:Description>

<!-- About the Tiff images of Hugo Claus-->
<rdf:Description rdf:about="http://archipel.org/tiff/
    HugoClaus/01">
  <dc:format>image/tiff</dc:format>
  <dc:title>Hugo Claus in Amsterdam</dc:title>
  <rdf:type>image</rdf:type>
  <dc:subject>Hugo Claus</dc:subject>
  <dc:subject>Amsterdam</dc:subject>
</rdf:Description>

<rdf:Description rdf:about="http://archipel.org/tiff/
    HugoClaus/02">
  <dc:format>image/tiff</dc:format>
  <dc:title>Hugo Claus and the COBRA movement</dc:title>
  <rdf:type>image</rdf:type>
  <dc:subject>Hugo Claus</dc:subject>
  <dc:subject>COBRA</dc:subject>
</rdf:Description>
```

```
<rdf:Description rdf:about="http://archipel.org/tiff/
   HugoClaus/03">
  <dc:format>image/tiff</dc:format>
  <dc:title>Funeral Hugo Claus</dc:title>
  <rdf:type>image</rdf:type>
  <dc:subject>Hugo Claus</dc:subject>
  <dc:subject>funeral</dc:subject>
</rdf:Description>
```

C.3 OAI-PMH Protocol Harvesting Example

An OAI-PMH example HTTP Request using the ‘GetRecord’ verb, e.g.,

`http://archipel1.demo.ibbt.be/omeka/oai-pmh-repository/request?verb=GetRecord&amp;metadataPrefix=oai_dc&amp;identifier=oai:archipel1.demo.ibbt.be:2`

results in an HTTP Response with an OAI-PMH payload as in Listing C.6 below:

Listing C.6: OAI-PMH Payload Instance.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
     http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2010-10-29T15:30:59Z</responseDate>
  <request verb="GetRecord" metadataPrefix="oai_dc"
      identifier="oai:archipel1.demo.ibbt.be:2">http://
      archipel1.demo.ibbt.be/omeka/oai-pmh-repository/
      request</request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:archipel1.demo.ibbt.be:2</
            identifier>
        <datestamp>2010-10-07T14:29:11Z</datestamp>
        <setSpec>1</setSpec>
      </header>
      <metadata>
        <oai_dc:dc xmlns:oai_dc="http://www.openarchives.
            org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/
  dc/elements/1.1/" xmlns:xsi="http://www.w3.org
  /2001/XMLSchema-instance" xsi:schemaLocation="
  http://www.openarchives.org/OAI/2.0/oai_dc/
  http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:title>Arne Sierens</dc:title>
<dc:creator>Thersites</dc:creator>
<dc:subject>vernieuwing, tekst, punk</dc:subject>
<dc:description>Interview fragment met Arne
    Sierens, in het kader van het project
    Toneelstof III - The wonder years.   E r  is
    een hele generatie opgestaan die -
    gedeeltelijk uit vrije keuze, gedeeltelijk
    gedwongen - ging repeteren buiten de grote
    huizen: in hangars, kelders en zolders.
    Gentenaar Arne Sierens studeerde in 1981 als
    regisseur af aan het RITCS en startte zijn
     carri re in Arca, NTG en Arena. Dat hij in
    deze huizen niet vond wat hij zocht, leidde in
     1982 tot de oprichting van een eigen
    gezelschap met Jan Leroy: De Sluipende Armoede
    . In 1984 waagde Sierens zich aan de creatie
    van een opera gebaseerd op het Nero-album Het
    Rattenkasteel, wat veel reactie uitlokte. Zijn
     eigen stijl ontplooide zich ten volle in De
    soldaat-facteur en Rachel, dat een volkse
    inspiratie combineerde met invloeden van de
    avant-garde. Vandaag realiseert Sierens zijn
    werk bij Compagnie Cecilia.</dc:description>
<dc:publisher>VTi</dc:publisher>
<dc:contributor>VTi</dc:contributor>
<dc:date>30/08/2010</dc:date>
<dc:type>digital video</dc:type>
<dc:format>digital video</dc:format>
<dc:identifier>http://archipel1.demo.ibbt.be/
    omeka/items/show/2</dc:identifier>
<dc:identifier>http://archipel1.demo.ibbt.be/
    omeka/archive/files/02
    _arne_sierens_1_1_b16ee911fc.mp4</
    dc:identifier>
<dc:identifier>http://archipel1.demo.ibbt.be/
    omeka/archive/files/02_arne_sierens_178137b536
    .mov</dc:identifier>
<dc:identifier>http://archipel1.demo.ibbt.be/
    omeka/archive/files/arnesierens_339e414f26.jpg
    </dc:identifier>
<dc:identifier>http://archipel1.demo.ibbt.be/
    omeka/archive/files/02_arne_sierens_92475f0949
    .flv</dc:identifier>
<dc:source>Thersites</dc:source>
```

```
          <dc:language>Dutch</dc:language>
        </oai_dc:dc>
      </metadata>
    </record>
  </GetRecord>
</OAI-PMH>
```

Appendix D

Proof-Of-Concepts' Implementation Details

D.1 News Data Model Ontology

Listing D.1: News Data Model Ontology.

```
#
# Prefixes
#
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.

@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

@prefix mdont: <http://medialoep.vrt.be/ontology#>.

mdont:
  a owl:Ontology;
  dc:title """News Multimedia ontology"""@en;
  dc:creator """MMLab"""@en;
  dc:publisher """VRT/Belgium, MMLab/Belgium"""@en;
  dc:description """Formal description of a multimedia
      ontology."""@en;
  #dce:format """OWL Full"""@en;
  #dce:identifier """"""@en;
  dc:language """English"""@en.

#----------
# Classes -
#----------
mdont:EditorialObject a owl:Class;
  dc:description """An editorial object represents a unit
      of work."""@en.

mdont:News a owl:Class;
  rdfs:subClassOf mdont:EditorialObject.

mdont:NewsStory a owl:Class;
  rdfs:subClassOf mdont:EditorialObject.

mdont:NewsItem a owl:Class;
  rdfs:subClassOf mdont:EditorialObject.

mdont:EditorialObjectDescription a owl:Class.

mdont:Rundown a owl:Class;
  rdfs:subClassOf mdont:EditorialObjectDescription.

#mdont:Component a owl:Class; # not used for now
# dc:description """A component represents a """@en.

#mdont:Composition a owl:Class; # not used for now
# rdfs:subClassOf mdont:Component.

mdont:MediaObject a owl:Class.

mdont:MediaObjectInstance a owl:Class.

#mdont:Realisation a owl:Class. # not used for now

mdont:Materialisation a owl:Class.

mdont:TimeBasedAnnotation a owl:Class.

mdont:Subtitle a owl:Class;
  rdfs:subClassOf mdont:TimeBasedAnnotation.

mdont:ROIInfo a owl:Class;
  rdfs:subClassOf mdont:TimeBasedAnnotation.

mdont:CGText a owl:Class;
  rdfs:subClassOf mdont:TimeBasedAnnotation.

mdont:Identifier a owl:Class.
```

```
mdont:TimelinePosition a owl:Class.

mdont:Element a owl:Class.

mdont:UsageRestriction a owl:Class.

#-------------
# Properties -
#-------------
#------------------------
# Data Model properties -
#------------------------
mdont:hasMaterialisation a owl:ObjectProperty;
  rdfs:domain mdont:EditorialObject;
  rdfs:range mdont:Materialisation.

# general property, currently used to link
# a materialisation instance with a mediaobject instance
mdont:hasRelatedMediaObject a owl:ObjectProperty;
  rdfs:range mdont:MediaObject.

mdont:hasInstance a owl:ObjectProperty;
  rdfs:domain mdont:MediaObject;
  rdfs:range mdont:MediaObjectInstance.

mdont:hasRundown a owl:ObjectProperty;
  rdfs:domain mdont:EditorialObject;
  rdfs:range mdont:Rundown.

#mdont:hasComposition a owl:ObjectProperty;
# rdfs:domain mdont:EditorialObject;
# rdfs:range mdont:EditorialObject.

# links editorial objects to a rundown
mdont:hasElement a owl:ObjectProperty;
  rdfs:domain mdont:Rundown;
  rdfs:range mdont:Element.

mdont:isPartOf a owl:ObjectProperty;
  rdfs:domain mdont:EditorialObject;
  rdfs:range mdont:EditorialObject.

# general property, used to reference stories
# in news rundowns from an element
mdont:reference a owl:ObjectProperty.

# used to enable ordering of stories in news rundowns
# and items in stories via element instances
```

```
mdont:position a owl:DatatypeProperty;
  rdfs:range xsd:integer.

#---------------
# BASIS FIELDS -
#---------------
mdont:alternativeTitle a owl:DatatypeProperty;
  rdfs:subPropertyOf dct:alternative;
  rdfs:label """alternative title"""@en;
  rdfs:label """andere titel"""@nl;
  rdfs:domain mdont:EditorialObject;
  rdfs:range xsd:string.

mdont:aspectRatio a owl:DatatypeProperty;
  rdfs:label """aspect ratio"""@en;
  rdfs:label """aspectverhouding"""@nl;
  rdfs:domain mdont:MediaObject;
  rdfs:range xsd:string.

mdont:broadcastDate a owl:DatatypeProperty;
  rdfs:label """broadcast date"""@en;
  rdfs:label """uitzenddatum"""@nl;
  rdfs:domain mdont:Materialisation;
  rdfs:range xsd:dateTime.

# thesaurus controlled
mdont:category a owl:ObjectProperty;
  rdfs:label """category"""@en;
  rdfs:label """categorie"""@nl;
  rdfs:domain mdont:EditorialObject;
  rdfs:range skos:Concept.

mdont:description a owl:DatatypeProperty;
  rdfs:subPropertyOf dc:description;
  rdfs:label """description"""@en;
  rdfs:label """omschrijving"""@nl;
  rdfs:domain mdont:EditorialObject;
  rdfs:range xsd:string.

mdont:episodeNumber a owl:DatatypeProperty;
  rdfs:label """episode number"""@en;
  rdfs:label """afleveringsnummer"""@nl;
  rdfs:domain mdont:EditorialObject;
  rdfs:range xsd:integer.

# thesaurus controlled
mdont:format a owl:ObjectProperty;
  rdfs:label """format"""@en;
  rdfs:label """formaat"""@nl;
  rdfs:domain mdont:MediaObjectInstance;
  rdfs:range skos:Concept.

# thesaurus controlled
mdont:keyword a owl:ObjectProperty;
  rdfs:subPropertyOf dc:subject;
  rdfs:domain mdont:EditorialObject;
  rdfs:range skos:Concept.

# DC terms also defines rightsholder, with an Agent range
mdont:rightsholder a owl:DatatypeProperty;
  rdfs:label """rightsholder"""@en;
  rdfs:label """rechthebbende"""@nl;
  rdfs:domain mdont:MediaObject;
  rdfs:range xsd:string.

mdont:recordingDate a owl:DatatypeProperty;
  rdfs:label """recording date"""@en;
  rdfs:label """opnamedatum"""@nl;
  rdfs:domain mdont:Materialisation;
  rdfs:range xsd:dateTime.

# owl:ObjectProperty, when location are mapped to e.g.
    geoNames
mdont:recordingLocation a owl:DatatypeProperty;
  rdfs:label """recording location"""@en;
  rdfs:label """opnameplaats"""@en;
  rdfs:domain mdont:Materialisation;
  rdfs:range xsd:string. #rdfs:range skos:Concept.

# thesaurus controlled
mdont:series a owl:ObjectProperty;
  rdfs:label """series"""@en;
  rdfs:label """reeks"""@nl;
  rdfs:domain mdont:EditorialObject;
  rdfs:range skos:Concept.

# thesaurus controlled
mdont:serviceCode a owl:ObjectProperty;
  rdfs:label """service code"""@en;
  rdfs:label """dienstcode"""@en;
  rdfs:domain mdont:EditorialObject;
  rdfs:range skos:Concept.

mdont:title a owl:DatatypeProperty;
  rdfs:subPropertyOf dc:title;
  rdfs:label """title"""@en;
  rdfs:label """titel"""@nl;
  rdfs:domain mdont:EditorialObject;
  rdfs:range xsd:string.

mdont:usageRestriction a owl:ObjectProperty;
  rdfs:label """usage restriction"""@en;
  rdfs:label """gebruiksbeperking"""@nl;
  rdfs:domain mdont:MediaObject;
  rdfs:range mdont:UsageRestriction.

# thesaurus controlled
mdont:usageRestrictionType a owl:DatatypeProperty;
  rdfs:label """usage restriction note"""@en;
  rdfs:label """toelichting gebruiksbeperking"""@nl;
  rdfs:domain mdont:UsageRestriction;
  rdfs:range skos:Concept.

# general property, currently used here additional notes
# in a usage restriction. and for BASIS opm field
mdont:note a owl:DatatypeProperty;
  rdfs:label """note"""@en;
  rdfs:label """nota"""@nl;
  rdfs:range xsd:string.

# - END BASIS FIELDS -

#---------------
# iNEWS FIELDS -
#---------------

mdont:anchorText a owl:DatatypeProperty;
  rdfs:range xsd:string.

# - END iNEWS FIELDS -

mdont:timelinePosition a owl:ObjectProperty;
  rdfs:domain mdont:Materialisation;
  rdfs:range mdont:TimelinePosition.

mdont:duration a owl:DatatypeProperty;
  rdfs:label """duration"""@en;
  rdfs:label """duur"""@en;
  rdfs:domain mdont:TimelinePosition;
  rdfs:range xsd:time.

mdont:offset a owl:DatatypeProperty;
  rdfs:label """offset"""@en;
  rdfs:label """offset"""@nl;
  rdfs:domain mdont:TimelinePosition;
  rdfs:range xsd:time.
```

```
#mdont:trackDescription a owl:DatatypeProperty;
# rdfs:label """audio track information"""@en;
# rdfs:label """informatie audiospoor"""@nl;
# rdfs:domain mdont:MediaObject;
# rdfs:range xsd:string.

mdont:timeBasedAnnotation a owl:ObjectProperty;
  rdfs:domain mdont:MediaObject;
  rdfs:range mdont:TimeBasedAnnotation.

mdont:annotationText a owl:DatatypeProperty;
  rdfs:domain mdont:TimeBasedAnnotation;
  rdfs:range xsd:string.

mdont:appearanceTime a owl:DatatypeProperty;
  rdfs:domain mdont:TimeBasedAnnotation;
  rdfs:range xsd:time.

mdont:disappearanceTime a owl:DatatypeProperty;
  rdfs:domain mdont:TimeBasedAnnotation;
  rdfs:range xsd:time.

# not supported yet
#mdont:relatedTrack a owl:ObjectProperty;
# rdfs:domain mdont:TimeBasedAnnotation;
# rdfs:range mdont:Track.

mdont:identifier a owl:ObjectProperty;
  rdfs:range mdont:Identifier.

# ARD = ardome id, BASISBBNR = bandnummer
mdont:identificationSystem a owl:DatatypeProperty;
  rdfs:domain mdont:Identifier;
  rdfs:range xsd:string.

mdont:identificationCode a owl:DatatypeProperty;
  rdfs:domain mdont:Identifier;
  rdfs:range xsd:string.

mdont:playsRole a owl:ObjectProperty;
  skos:definition """Specifies the fact an agent takes part
       in an act."""@en;
  rdfs:domain foaf:Agent.

mdont:assistant a owl:Class;
  rdfs:label """assistant"""@en;
  rdfs:label """assistent"""@nl;
  rdfs:subPropertyOf mdont:playsRole.
```

```
mdont:audioRecorder a owl:ObjectProperty;
  rdfs:label """audio recorder"""@en;
  rdfs:label """verantwoordelijke geluidsopname"""@nl;
  skos:definition """Person responsible for audio recording
     ."""@en;
  rdfs:subPropertyOf mdont:playsRole.

mdont:cameraOperator a owl:ObjectProperty;
  rdfs:label """camera operator"""@en;
  rdfs:label """cameraman"""@nl;
  rdfs:subPropertyOf mdont:playsRole.

mdont:commentator a owl:ObjectProperty;
  rdfs:label """commentator"""@en;
  rdfs:label """commentator"""@nl;
  rdfs:subPropertyOf mdont:playsRole.

mdont:director a owl:ObjectProperty;
  rdfs:label """director"""@en;
  rdfs:label """regisseur"""@nl;
  rdfs:subPropertyOf mdont:playsRole.

mdont:editor a owl:ObjectProperty;
  rdfs:label """editor"""@en;
  rdfs:label """monteur"""@nl;
  rdfs:subPropertyOf mdont:playsRole.

mdont:headProduction a owl:ObjectProperty;
  rdfs:label """head production"""@en;
  rdfs:label """productieleider"""@nl;
  rdfs:subPropertyOf mdont:playsRole.

mdont:journalist a owl:ObjectProperty;
  rdfs:label """journalist"""@en;
  rdfs:label """journalist"""@nl;
  rdfs:subPropertyOf mdont:playsRole.

mdont:presenter a owl:ObjectProperty;
  rdfs:label """presenter"""@en;
  rdfs:label """presentator"""@nl;
  rdfs:subPropertyOf mdont:playsRole.

mdont:producer a owl:ObjectProperty;
  rdfs:label """producer"""@en;
  rdfs:label """producer"""@nl;
  rdfs:subPropertyOf mdont:playsRole.

mdont:scenarioWriter a owl:ObjectProperty;
  rdfs:label """scenario writer"""@en;
```

```
    rdfs:label """scenarist"""@nl;
    rdfs:subPropertyOf mdont:playsRole.

  mdont:audioProducer a owl:ObjectProperty;
    rdfs:label """audio producer"""@en;
    rdfs:label """uitvoerder sonorisatie"""@nl;
    rdfs:subPropertyOf mdont:playsRole.
```

Appendix E

Summary of Contributing Projects

E.1 ARCHIPEL

- *Project title*: Network-centric approach to sustainable digital archives.
- *Project type*: IBBT project (1 Oct 2009 - 30 Sep 2011).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/archipel`.
- *Author's role*: Participating as overall leader on semantic metadata modelling, aggregating, harvesting, exchanging, and archiving of Flanders' valuable multimedia assets.

E.2 MAPLE

- *Project title*: Mobile, Adaptive & Personalised Learning Experience.
- *Project type*: IBBT project (1 Oct 2009 - 30 Sep 2011).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/maple-2`.
- *Author's role*: Participating as work package leader on semantic metadata modelling, enrichment, and reasoning on eLearning multimedia assets.

E.3 LLINGO

- *Project title*: Language Learning in an Interactive Game Environment.
- *Project type*: IBBT project (1 Oct 2009 - 30 Sep 2011).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/llingo-2`.
- *Author's role*: Participating as overall leader & work package leader on infinite 3D-gaming texture modelling.

E.4 EPICS

- *Project title*: eLearning Platform in the Cultural heritage Sector.
- *Project type*: IBBT project (1 Oct 2009 - 30 Sep 2011).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/epics-2`.
- *Author's role*: Participating as work package leader on semantic metadata modelling, and didactisation of digital heritage content in a eLearning platform.

E.5 Share4Health

- *Project title*: Healthcare professionals collaboration Space.
- *Project type*: IBBT project (1 May 2008 - 30 Apr 2010).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/share4health`.
- *Author's role*: Participating as work package leader on patient ontology building and patient data exchange.

E.6 CHF

- *Project title*: Chronic Heart Failure Disease Management.
- *Project type*: IBBT project (1 May 2008 - 30 Apr 2009).

- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/chf`.
- *Author's role*: Participating as work package leader on CHF ontology building.

E.7 BOM-Vl

- *Project title*: Archiving & Disclosure of Flanders' valuable Multimedia assets.
- *Project type*: IBBT project (1 Jan 2008 - 30 Jun 2009).
- *Project URL*: `https://projects.ibbt.be/bom-vl/`.
- *Author's role*: Participating as overall leader & work package leader on semantic metadata modelling, and exchanging & archiving of Flanders' valuable multimedia assets.

E.8 CUPID

- *Project title*: Cultural Profile & Information Database.
- *Project type*: IBBT project (1 Jan 2008 - 31 Dec 2009).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/cupid`.
- *Author's role*: Participating as work package leader on events and profile ontology building, and data aggregation and enrichment.

E.9 IFIP

- *Project title*: Independent Films In Progress.
- *Project type*: IBBT project (1 Jan 2008 - 31 Dec 2009).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/ifip`.
- *Author's role*: Participating as overall leader & work package leader on multimedia workflow and exchange platform.

E.10 CoCoNut

- *Project title*: Communication & Collaboration Network Utilities.
- *Project type*: IBBT project (1 Jan 2008 - 31 Dec 2009).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/coconut`.
- *Author's role*: Participating as consultant on Web2.0 collaboration tool set.

E.11 Heritage 2.0

- *Project title*: Social interactive location-based Cultural Heritage Experience.
- *Project type*: IBBT project (1 Apr 2007 - 31 Mar 2009).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/heritage-2-0`.
- *Author's role*: Participating as work package leader on semantic metadata modelling, data aggregation and data exchange of Cultural Heritage Records.

E.12 PISA

- *Project title*: Production, Indexing and Search of Audio-visual material.
- *Project type*: IBBT project (1 Mar 2007 - 30 Sep 2008).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/pisa`.
- *Author's role*: Participating as overall leader & work package leader on metadata modelling of media production tools.

E.13 PokuMOn

- *Project title*: Performing arts Multimedia Dissemination.
- *Project type*: IBBT project (1 Apr 2007 - 31 Mar 2009).

- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/pokumon`.
- *Author's role*: Participating as overall leader & work package leader on semantic metadata modelling, on-line distribution and archiving of multimedia of performing arts.

E.14 GEISHA

- *Project title*: Grid Enabled Infrastructure for Service Oriented High Definition Media Applications.
- *Project type*: IBBT project (1 Mar 2007 - 28 Feb 2009).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/geisha`.
- *Author's role*: Participated as work package leader on HD-based media infrastructure and corresponding multimedia compression and container formats.

E.15 PeCMan

- *Project title*: Personal Content Management Platform.
- *Project type*: IBBT project (1 Feb 2007 - 31 Jan 2009)..
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/pecman`.
- *Author's role*: Participated as work package leader on metadata modelling for personal content management systems.

E.16 Video QSAC

- *Project title*: Video To The Home - Quality Sensing, Aggregation & Control.
- *Project type*: IBBT project (1 Jan 2007 - 31 Dec 2009).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/video-q-sac`.

- *Author's role*: Participating as work package leader on DRM business modelling and multimedia encoding & bit stream adaptation.

E.17 MADUF

- *Project title*: MAximizing Dvb Usage in Flanders.
- *Project type*: IBBT project (1 Jan 2006 - 31 Mar 2008).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/maduf`.
- *Author's role*: Participating as work package leader on deploying interactive services for mobile television using DVB-H.

E.18 MultiGov

- *Project title*: Multichannel strategy for e-Government.
- *Project type*: IBBT project (1 Sep 2005 - 31 Oct 2007).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/multigov`.
- *Author's role*: Participated as consultant on how multimedia information flows can be optimally delivered.

E.19 MCDP

- *Project title*: Multimedia Content Distribution Platform.
- *Project type*: IBBT project (1 Jan 2005 - 28 Feb 2007).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/mcdp`.
- *Author's role*: Participating as work package leader on a multi-modal multimedia content distribution system.

E.20 FIPA

- *Project title*: File based Integrated Production Architecture.
- *Project type*: IBBT project (1 Jan 2005 - 31 Dec 2006).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/fipa`.
- *Author's role*: Participating as overall leader & work package leader on digital media production.

E.21 IPEA

- *Project title*: Innovative Platform for Electronic Archiving.
- *Project type*: IBBT project (1 Jan 2005 - 31 Dec 2006).
- *Project URL*: `http://www.ibbt.be/en/projects/overview-projects/p/detail/ipea`.
- *Author's role*: Participating as overall leader & work package leader on exchanging and archiving of multimedia broadcast material.

References

[1] B. Adida and M. Birbeck, editors. *RDFa Primer – Bridging the Human and Data Webs*. W3C Note. World Wide Web Consortium, October 2008. Available at http://www.w3.org/TR/xhtml-rdfa-primer/.

[2] B. Adida, M. Birbeck, S. McCarron, and S. Pemberton, editors. *RDFa in XHTML: Syntax and Processing*. W3C Recommendation. World Wide Web Consortium, October 2008. Available at http://www.w3.org/TR/rdfa-syntax/.

[3] Adobe. Extensible Metadata Platform (XMP) - Specification Part 1 - Data and Serialization Model, 2010. Available at http://www.adobe.com/devnet/xmp/.

[4] A. M. W. Association. Advanced Authoring Format (AAF) – AAF Object Specification, AAF Stored Format Specification, and AAF Low-Level Container Specification, 2005. Available at http://www.aafassociation.org/downloads.shtml.

[5] J. E. I. D. Association. Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.2, 2002. Available at http://www.exif.org/Exif2-2.PDF.

[6] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller. Triplify: Light-weight Linked Data Publication from Relational Databases. In *Proceedings of the 18th International Conference on World Wide Web*, pages 621–630, Madrid, Spain, April 2009.

[7] D. Austerberry. *Digital Asset Management*. Focal Press - Elsevier, October 2006.

[8] D. Ayers and M. Vlkel, editors. *Cool URIs for the Semantic Web*. W3C Note. World Wide Web Consortium, December 2008. Available at http://www.w3.org/TR/cooluris/.

[9] T. Beckers, N. Oorts, F. Hendrickx, and R. Van de Walle. Multichannel Publication of Interactive Media Documents in a News Environment. In *Proceedings of the 14th International Conference on the World Wide Web*, pages 1088–1089, Chiba, Japan, May 2005.

[10] D. Beckett and B. McBride, editors. *RDF/XML Syntax Specification (Revised).* W3C Recommendation. World Wide Web Consortium, February 2004. Available at `http://www.w3.org/TR/REC-rdf-syntax/`.

[11] T. Berners-Lee. Design Issues: Linked Data. Available at `http://www.w3.org/DesignIssues/LinkedData.html`.

[12] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[13] D. Berrueta and J. Phipps, editors. *Best Practice Recipes for Publishing RDF Vocabularies.* W3C Note. World Wide Web Consortium, August 2008. Available at `http://www.w3.org/TR/swbp-vocab-pub/`.

[14] M. Birbeck and S. McCarron, editors. *CURIE Syntax 1.0 – A Syntax for Expressing Compact URIs.* W3C Recommendation. World Wide Web Consortium, January 2009. Available at `http://www.w3.org/TR/curie/`.

[15] P. V. Biron and A. Malhotra, editors. *XML Schema Part 2: Datatypes – Second Edition.* W3C Recommendation. World Wide Web Consortium, October 2004. Available at `http://www.w3.org/TR/xmlschema-2/`.

[16] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data – The Story So Far. *International Journal on Semantic Web & Information Systems*, 5(3):1–22, September 2009.

[17] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia – a Crystallisation Point for the Web of Data. *International Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–156, September 2009.

[18] S. Bocconi, F. Nack, and L. Hardman. Using Rhetorical Annotations for Generating Video Documentaries. In *Proceedings of the 6th IEEE International Conference on Multimedia & Expo*, pages 1–4, Amsterdam, The Netherlands, July 2005.

[19] T. Bray, D. Hollander, A. Layman, and R. Tobin, editors. *Namespaces in XML 1.0 (Second Edition).* W3C Recommendation. World Wide Web Consortium, August 2006. Available at `http://www.w3.org/TR/REC-xml-names/`.

[20] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, and F. Yergeau, editors. *Extensible Markup Language (XML) 1.0 (Fourth Edition).* W3C Recommendation. World Wide Web Consortium, August 2006. Available at `http://www.w3.org/TR/2006/REC-xml-20060816/`.

[21] J. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, July 1998.

[22] D. Brickley and R. Guha, editors. *RDF Vocabulary Description Language 1.0: RDF Schema.* W3C Recommendation. World Wide Web Consortium, February 2004. Available at `http://www.w3.org/TR/rdf-schema/`.

[23] British Broadcasting Corporation. Standard Media Exchange Framework (SMEF) – Data Model, V1.10, 2005. Available at `http://www.bbc.co.uk/guidelines/smef/`.

[24] A. Brown. Understanding Technological Change: the Case of MRPII. *International Journal of Operations & Production Management*, 13(12):25–35, December 1993.

[25] M. Brown, J. Foote, G. Jones, K. Sparck-Jones, and S. Young. Automatic content-based retrieval of broadcast news. In *Proceedings of the 3rd ACM International Conference on Multimedia*, pages 35–43, San Francisco, USA, November 1995.

[26] D. Bulterman, G. Grassel, J. Jansen, A. Koivisto, N. Layaïda, T. Michel, S. Mullender, and D. Zucker, editors. *Synchronized Multimedia Integration Language (SMIL 2.1)*. W3C Recommendation. World Wide Web Consortium, December 2005. Available at `http://www.w3.org/TR/SMIL2/`.

[27] J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named Graphs, Provenance and Trust. In *Proceedings of the 14th International Conference on World Wide Web*, pages 613–622, Chiba, Japan, May 2005.

[28] P. Castells, F. Perdrix, E. Pulido, M. Rico, R. Benjamins, J. Contreras, and J. Lores. Neptuno: Semantic Web Technologies for a Digital Newspaper Archive. *Lecture Notes in Computer Science – The Semantic Web: Research and Applications*, 3053:445–458, May 2004.

[29] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana, editors. *Web Services Description Language (WSDL) Version 2.0 Part 1 – Core Language*. W3C Recommendation. World Wide Web Consortium, June 2007. Available at `http://www.w3.org/TR/wsdl20/`.

[30] P. Cimprich, O. Becker, C. Nentwich, H. Jirousek, M. Batsis, P. Brown, and M. Kay. Streaming Transformations for XML (STX) – Version 1.0, 2004. Available at `http://stx.sourceforge.net/documents/spec-stx-20040701.html`.

[31] J. Clark, editor. *XSL Transformations (XSLT) – Version 1.0*. W3C Recommendation. World Wide Web Consortium, November 1999. Available at `http://www.w3.org/TR/xslt/`.

[32] K. Clark, L. Feigenbaum, and E. Torres, editors. *SPARQL Protocol for RDF*. W3C Recommendation. World Wide Web Consortium, January 2008. Available at `http://www.w3.org/TR/rdf-sparql-protocol/`.

[33] I. E. Commission. Digital Video (DV) Specification – Codec and Tape Format at 25Mb/s), 1998. IEC 61834.

[34] P. E. Committee. PREMIS Data Dictionary for Preservation Metadata – Version 2.0, 2008. Available at `http://www.loc.gov/standards/premis/v2/premis-2-0.pdf`.

[35] D. Connolly, editor. *Gleaning Resource Descriptions from Dialects of Languages (GRDDL).* W3C Recommendation. World Wide Web Consortium, September 2007. Available at `http://www.w3.org/TR/grddl/`.

[36] S. Corcoran. Using Social Applications in Ad Campaigns, 2009. Available at `http://www.forrester.com/Research/Document/Excerpt/0,7211,54050,00.html`.

[37] C. Cornelis, X. Guo, J. Lu, and G. Zhang. Clustering Methods for Collaborative Filtering. In *Proceedings of the 15th National Conference on Artificial Intelligence – Workshop on Recommendation Systems*, pages 114–129, Madison, USA, July 1998.

[38] C. Cornelis, X. Guo, J. Lu, and G. Zhang. A Fuzzy Relational Approach to Event Recommendation. In *Proceedings of the 1st Indian International Conference on Artificial Intelligence*, pages 2231–2242, Pune, India, December 2005.

[39] A. Dappert and A. Farquhar. Implementing Metadata that Guides Digital Preservation Services. In *Proceedings of the 6th International Conference on Preservation of Digital Objects*, pages 50–58, San Francisco, USA, October 2009.

[40] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.-K. Papastathis, and M. Strintzis. Knowledge-assisted semantic video object detection. *IEEE Transactions on Circuits, Systems & Video Technolologies*, 15(10):1210–1224, October 2005.

[41] M. Davis. Active Capture: Integrating Human-Computer Interaction and Computer Vision/Audition to Automate Media Capture. In *Proceedings of the 4th IEEE International Conference on Multimedia & Expo*, pages 185–188, Baltimore, USA, July 2003.

[42] S. De Bruyne, W. De Neve, K. De Wolf, D. De Schrijver, P. Verhoeve, and R. Van de Walle. Temporal Video Segmentation on H.264/AVC Compressed Bitstreams. In *Proceedings of the 13th International Multimedia Modeling Conference*, pages 1–12, Singapore, Republic of Singapore, January 2007.

[43] F. De Jong, T. Westerveld, and A. De Vries. Multimedia Search without Visual Analysis: the Value of Linguistic and Contextual Information. *IEEE Transactions on Circuits & Systems for Video Technology*, 17(3):365–371, March 2007.

[44] T. De Pessemier, S. Dooms, T. Deryckere, and L. Martens. Time Dependency of Data Quality for Collaborative Filtering Algorithms. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 1–4, Barcelona, Spain, September 2010.

[45] T. De Pessemier, K. Vanhecke, S. Dooms, T. Deryckere, and L. Martens. Probability-based Extended Profile Filtering, an Advanced Collaborative Filtering Algorithm for User-Generated Content. In *Proceedings of the 6th International Conference on Web Information Systems and Technologies*, Valencia, Spain, April 2010.

[46] D. De Schrijver, W. De Neve, D. Van Deursen, S. De Bruyne, and R. Van de Walle. Exploitation of Interactive Region of Interest Scalability in Scalable Video Coding by Using an XML-driven Adaptation Framework. In *Proceedings of the 2nd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution*, pages 223–231, Leeds, United Kingdom, December 2006.

[47] R. De Sutter, K. Braeckman, E. Mannens, and R. Van de Walle. Integrating Audiovisual Feature Extraction Tools in Media Annotation Production Systems. In *Proceedings of the 13th IASTED International Conference on Internet and Multimedia Systems and Applications*, pages 76–81, Honolulu, USA, August 2009.

[48] R. De Sutter, S. Notebaert, and R. Van de Walle. Evaluation of metadata standards in the context of digital audio-visual libraries. *Lecture Notes in Computer Science – Research and Advanced Technology for Digital Libraries*, 4172:220–223, September 2006.

[49] C. Dorai and S. Venkatesh. Bridging the Semantic Gap in Content Management Systems: Computational Media Aesthetics. In *Proceedings of the 1st Conference on Computational Semiotics for Games and New Media*, pages 94–99, Amsterdam, The Netherlands, September 2001.

[50] G. Doyle. Broadcast Media Management in a Data-Centric Workflow. Available on `http://www.sgi.com/pdfs/3477.pdf`.

[51] EBU. P/Meta Metadata Exchange Scheme v1.1, Technical Report 3295, 2005. Available at `http://www.ebu.ch/en/technical/metadata/specifications/`.

[52] EBU/ETSI. ETSI TS 102 822-3-1: TV-Anytime – Part 3: Metadata, 2008. Available at `http://www.tv-anytime.org/`.

[53] N. Fernandez, J. Blazquez, J. Fisteus, L. Sanchez, M. Sintek, A. Bernardi, M. Fuentes, A. Marrara, and Z. Ben-Asher. NEWS: Bringing Semantic Web Technologies into News Agencies. *Lecture Notes in Computer Science – The Semantic Web: ISWC 2006*, 4273:778–791, November 2006.

[54] N. Fernandez, L. Sanchez, J. Blazquez, and J. Villamor. The News Ontology for Professional Journalism Applications. *Integrated Series in Information Systems – Ontologies A Handbook of Principles, Concepts and Applications in Information Systems*, 14:887–919, April 2007.

[55] J. Ferraiolo and D. Jackson, editors. *Scalable Vector Graphics (SVG) 1.1 Specification*. W3C Recommendation. World Wide Web Consortium, April 2009. Available at `http://www.w3.org/TR/SVG11/`.

[56] R. Garcia, F. Perdrixa, R. Gila, and M. Oliva. The Semantic Web as a Newspaper Media Convergence Facilitator. *Journal of Web Semantics – Semantic Multimedia*, 6:151–161, April 2008.

[57] D. Gasevic, D. Djuric, and V. Devedzic, editors. *Model Driven Architecture and Ontology Development*. Springer-Verlag, March 2006.

[58] D. I. Group. DIG35 Specification - Metadata for Digital Images - Version 1.1, 2001. Available at `http://www.i3a.org/technologies/metadata/`.

[59] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon, editors. *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. W3C Recommendation. World Wide Web Consortium, April 2007. Available at `http://www.w3.org/TR/soap12-part1/`.

[60] S. Hacker and S. Hayes. *MP3, The Definitive Guide*. O'Reilly, March 2000.

[61] M. Haller, H.-G. Kim, and T. Sikora. Audiovisual Anchorperson Detection for Topic-oriented Navigation in Broadcast News. In *Proceedings of the 7th IEEE International Conference on Multimedia & Expo*, pages 1–4, Toronto, Canada, July 2006.

[62] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj. Isolated Regions in Video Coding. *IEEE Transactions on Multimedia*, 6:259–267, April 2004.

[63] L. Hardman, e. Obrenovic, F. Nack, B. Kerherv, and K. Piersol. Canonical Processes of Semantically Annotated Media Production. *Multimedia Systems – Special Issue on Canonical Processes of Media Production*, 14(6):327–340, December 2008.

[64] I. Hargreaves and J. Thomas. *New News, Old News*. ITC and BSC Research Publication, October 2002.

[65] B. Haslhofer and B. Schandl. The OAI2LOD Server: Exposing OAI-PMH Metadata as Linked Data. In *Proceedings of the 1st International Workshop on Linked Data*, pages 1–5, Beijing, China, April 2008.

[66] B. Haslhofer and B. Schandl. Interweaving OAI-PMH Data Sources with the Linked Data Cloud. *International Journal of Metadata, Semantics and Ontologies*, 5:17–31, April 2010.

[67] M. Hausenblas, R. Troncy, T. Burger, and Y. Raimond. Interlinking Multimedia. In *Proceedings of the 2nd Workshop on Linked Data on the Web*, pages 1–9, Madrid, Spain, April 2009.

[68] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, Berkeley, USA, August 1999.

[69] I. Hickson and D. Hyatt, editors. *HTML5 – A Vocabulary and associated APIs for HTML and XHTML*. W3C Working Draft. World Wide Web Consortium, 4 March 2010. Available at http://www.w3.org/TR/html5/.

[70] M. Hildebrand, J. Van Ossenbruggen, and L. Hardman. /facet: A Browser for Heterogeneous Semantic Web Repositories. In *Proceedings of the 5th International Semantic Web Conference*, pages 272–285, Athens, USA, November 2006.

[71] D. C. M. Initiative. DCMI Metadata Terms, 2008. Available at http://dublincore.org/specifications/.

[72] International Press Telecommunications Council. EventsML-G2 Specification – Version 1.1, 2009. Available at http://www.iptc.com/std/EventsML-G2/EventsML-G2_1.1.zip.

[73] International Press Telecommunications Council. NewsML-G2 Specification – Version 2.2, 2009. Available at http://www.iptc.com/std/NewsML-G2/NewsML-G2_2.2.zip.

[74] Internet Engineering Task Force. RFC 1738: Uniform Resource Locators (URL), 1994. Available at http://www.ietf.org/rfc/rfc1738.txt.

[75] Internet Engineering Task Force. RFC 1807: A Format for Bibliographic Records, 1995. Available at http://www.ietf.org/rfc/rfc1807.txt.

[76] Internet Engineering Task Force. RFC 2045: Multipurpose Internet Mail Extensions (MIME) – Part One: Format of Internet Message Bodies, 1996. Available at http://tools.ietf.org/html/rfc2045/.

[77] Internet Engineering Task Force. RFC 2141: Uniform Resource Name (URN) – Generic Syntax, 1997. Available at http://tools.ietf.org/html/rfc2141/.

[78] Internet Engineering Task Force. RFC 2616: HyperText Transfer Protocol – HTTP/1.1, 1999. Available at http://www.ietf.org/rfc/rfc2616.txt.

[79] Internet Engineering Task Force. RFC 3986: Uniform Resource Identifier (URI) – Generic Syntax, 2005. Available at http://tools.ietf.org/html/rfc3986/.

[80] IPTC. NewsML v1.2, 2003. Available at http://www.newsml.org/.

[81] A. Isaac and E. Summers, editors. *SKOS Simple Knowledge Organization System – Primer*. W3C Note. World Wide Web Consortium, August 2009. Available at http://www.w3.org/TR/skos-primer/.

[82] A. Iskold. The Art, Science and Business of Recommendation Engines, 2007. Available at http://www.readwriteweb.com/archives/recommendation_engines.php.

[83] ISO/IEC. ISO/IEC 9075:1992, Database Language SQL, 1992. Available at http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt.

[84] ISO/IEC. Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s – Part 2: Video, 1993. ISO/IEC 11172-2:1993.

[85] ISO/IEC. Information Technology – Generic Coding of Moving Pictures and associated Audio Information – Systems, 2000. ISO/IEC 13818-1:2000.

[86] ISO/IEC. JPEG2000 Core Coding System – Part 1, 2000. ISO/IEC 15444-1:2000.

[87] ISO/IEC. Information Technology – Multimedia Content Description Interface – Part 1: Systems, 2002. ISO/IEC 15938-1:2002.

[88] ISO/IEC. Information technology – Coding of Audio, Picture, Multimedia and Hypermedia Information – Part 14: MP4 file format, 2003. ISO/IEC 14496-14:2003.

[89] ISO/IEC. Space Data and Information Transfer Systems – Open Archival Information System – Reference Model. ISO/IEC 14721:2003, 2003.

[90] ISO/IEC. Document Management – Electronic Document File Format for Long-term Preservation – Part 1: Use of PDF 1.4 (PDF/A-1), 2005. ISO/IEC 19005-1:2005.

[91] ISO/IEC. Information Technology – Coding of Audio-visual Objects – Part 12: ISO Base Media File Format, 2005. ISO/IEC 14496-12:2005.

[92] ISO/IEC. Information technology – Coding of audio-visual objects – Part 3: Audio. ISO/IEC 14496-3:2005, 2005.

[93] ISO/IEC. Information Technology – Coding of Audio-visual Objects – Part 17: Streaming Text Format, 2006. ISO/IEC 14496-17:2006.

[94] ISO/IEC. Information Technology – MPEG-21, Part 17: Fragment Identification of MPEG Resources, 2006. ISO/IEC 21000-17:2006.

[95] ISO/IEC. Information technology – MPEG Systems Technologies – Part 5: Bitstream Syntax Description Language, 2008. ISO/IEC 23001-5:2008.

[96] ITU-T. Information Technology - Digital Compression and Coding of continuous-tone Still Images - Requirements and Guidelines. ITU-T Recommendation T.81, 1992.

[97] ITU-T. ITU-T Rec. X.667 — ISO/IEC 9834-8. Generation and Registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier Components, 2004. Available at http://www.itu.int/ITU-T/studygroups/com17/oid/X.667-E.pdf.

[98] ITU-T and ISO/IEC. Advanced Video Coding for Generic Audiovisual Services, 2003. ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC.

[99] I. Jacobs and N. Walsh, editors. *Architecture of the World Wide Web – Volume One*. W3C Recommendation. World Wide Web Consortium, December 2004. Available at `http://www.w3.org/TR/webarch/`.

[100] Joint US/EU ad hoc Agent Markup Language Committee. DAML+OIL, March 2001. Available at `http://www.daml.org/2001/03/daml+oil-index.html`.

[101] G. Karypis. Evaluation of Item-Based Top-N Recommendation Algorithms. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 247–254, Atlanta, USA, October 2001.

[102] M. Klein. Xml, rdf, and relatives. *IEEE Intelligent Systems*, 16(2):26–28, March 2001.

[103] G. Klyne and J. Carroll, editors. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation. World Wide Web Consortium, February 2004. Available at `http://www.w3.org/TR/rdf-concepts/`.

[104] G. Kobilarov, T. Scott, Y. Raimond, S. Oliver, C. Sizemore, M. Smethurst, C. Bizer, and R. Lee. Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections. *Lecture Notes in Computer Science – The Semantic Web, Research and Applications: ESWC 2009*, 5554:723–737, May 2009.

[105] S. Kopf, F. Lampi, T. King, and W. Effelsberg. Automatic Scaling and Cropping of Videos for Devices with Limited Screen Resolution. In *Proceedings of the 14th ACM International Conference on Multimedia*, pages 957–958, Santa Barbara, USA, October 2006.

[106] H. Kosch, L. Bszrmnyi, M. Dller, M. Libsie, P. Schojer, and A. Kofler. The life cycle of multimedia metadata. *IEEE Multimedia*, 12(1):80–86, January 2005.

[107] C. Lagoze and H. Van de Sompel. The Open Archives Initiative – Object Reuse and Exchange – Version 1.0, 2008. Available at `http://www.openarchives.org/ore/1.0/toc/`.

[108] C. Lagoze and H. Van de Sompel. The Open Archives Initiative – Protocol for Metadata Harvesting – Version 2.0, 2008. Available at `http://www.openarchives.org/OAI/openarchivesprotocol.html`.

[109] P. Lambert, D. De Schrijver, D. Van Deursen, W. De Neve, Y. Dhondt, and R. Van de Walle. A Real-Time Content Adaptation Framework for Exploiting ROI Scalability in H.264/AVC. *Lecture Notes in Computer Science – Advanced Concepts for Intelligent Vision Systems*, 4179:442–453, September 2006.

[110] A. Le Hors, P. Le Hégaret, L. Wood, G. Nicol, J. Robie, M. Champion, and S. Byrne, editors. *Document Object Model (DOM) – Level 3 Core Specification*. W3C Recommendation. World Wide Web Consortium, April 2004. Available at `http://www.w3.org/TR/DOM-Level-3-Core/`.

[111] S. Lerouge, R. De Sutter, and R. Van de Walle. Personalizing Quality Aspects in Scalable Video Coding. In *Proceedings of the 6th IEEE International Conference on Multimedia & Expo*, pages 1–4, Amsterdam, The Netherlands, July 2005.

[112] Library of Congress. MARC & MARC 21: MAchine-Readable Cataloging, 2010. Available at `http://www.loc.gov/marc/`.

[113] Library of Congress. METS: Metadata Encoding and Transmission Standard, 2010. Available at `http://www.loc.gov/standards/mets/`.

[114] Library of Congress. MODS: Metadata Object Description Schema, 2010. Available at `http://www.loc.gov/standards/mods/`.

[115] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[116] E. Mannens, S. Coppens, P. Bastijns, S. Corneillie, P. Hochstenbach, L. Van Melle, S. Van Peteghem, and R. Van de Walle. *(Meta)datastandaarden voor Digitale Archieven.* Universiteitsbibliotheek Gent, June 2009.

[117] E. Mannens and R. Troncy, editors. *Use Cases and Requirements for Media Fragments*. W3C Working Draft. World Wide Web Consortium, December 2009. Available at `http://www.w3.org/TR/media-frags-reqs/`.

[118] E. Mannens, R. Troncy, K. Braeckman, D. Van Deursen, W. Van Lancker, R. De Sutter, and R. Van de Walle. Automatic Information Enrichment in News Production. In *Proceedings of the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 61–64, London, United Kingdom, May 2009.

[119] E. Mannens, R. Troncy, S. Pfeiffer, and D. Van Deursen, editors. *Media Fragments URI 1.0.* W3C Working Draft. World Wide Web Consortium, December 2009. Available at `http://www.w3.org/TR/media-frags/`.

[120] E. Mannens, M. Verwaest, and R. Van de Walle. Production and Multi-channel Distribution of News. *Multimedia Systems – Special Issue on Canonical Processes of Media Production*, 14(6):359–368, December 2008.

[121] D. McGuinness and F. van Harmelen, editors. *OWL Web Ontology Language: Overview*. W3C Recommendation. World Wide Web Consortium, February 2004. Available at `http://www.w3.org/TR/owl-features/`.

[122] O. W. G. Members, editor. *OWL 2 Web Ontology Language – Document Overview*. W3C Recommendation. World Wide Web Consortium, October 2009. Available at `http://www.w3.org/TR/owl2-overview/`.

[123] A. Messina, L. Boch, G. Dimino, W. Bailer, P. Schallauer, W. Allasia, and R. Basili. Creating Rich Metadata in the TV Broadcast Archives Environment: the PrestoSpace Project. In *Proceedings of the 2nd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution*, pages 193–200, Leeds, UK, December 2006.

[124] A. Miles and S. Bechhofer, editors. *SKOS Simple Knowledge Organization System – Reference*. W3C Recommendation. World Wide Web Consortium, August 2009. Available at http://www.w3.org/TR/skos-primer/.

[125] A. Miles, B. Matthews, M. Wilson, and D. Brickley. SKOS core: Simple Knowledge Organisation for the Web. In *Proceedings of the 5th International Conference on Dublin Core and Metadata Applications*, pages 1–9, Madrid, Spain, September 2005.

[126] F. Nack and W. Putz. Designing Annotation Before Its Needed. In *Proceedings of the 9th ACM International Conference on Multimedia*, pages 251–260, Ottawa, Canada, September 2001.

[127] F. Nack and W. Putz. Saying what it means: Semi-automated (news) media annotation. *International Journal of Multimedia Tools & Applications*, 22(3):263–302, March 2004.

[128] H. Nguyen and T. Cao. Named Entity Disambiguation: A Hybrid Statistical and Rule-based Incremental Approach. In *Proceedings of the 3rd International Asian Semantic Web Conference*, pages 420–433, Bangkok, Thailand, December 2008.

[129] L. of Congress. The BagIt File Packaging Format – v0.96, 2009. Available at http://www.digitalpreservation.gov/library/resources/tools/docs/bagitspec.pdf.

[130] T. O'Reilly. What is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software, 2005. Available at http://oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=1.

[131] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, 3:37–52, January 2008.

[132] R. Pantos. HTTP Live Streaming. Available at http://tools.ietf.org/html/draft-pantos-http-live-streaming-01/.

[133] M. Papagelis, D. Plexousakis, and T. Kutsuras. Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. *Lecture Notes in Computer Science – Trust Management*, 3477:224–239, May 2005.

[134] F. Pereira. Content and Context: two Worlds to bridge. In *Keynote talk to the 4th International Workshop on Content-Based Multimedia Indexing*, pages 1–6, Riga, Latvia, June 2005.

[135] F. Pereira and I. Burnett. Universal multimedia experiences for tomorrow. *IEEE Signal Processing Magazine*, 20(2):63–73, March 2003.

[136] F. Pereira, A. Vetro, and T. Sikora. Multimedia retrieval and delivery: Essential metadata challenges and standards. *Proceedings of the IEEE*, 96(4):721–744, April 2008.

[137] A. Perkis, Y. Abdeljaoued, C. Christopoulos, T. Ebrahimi, and J. Chicharo. Universal multimedia access from wired and wireless systems. *IEEE Transactions on Circuits, Systems & Signal Processing - Special Issue on Multimedia Communication*, 20(3):387–402, March 2001.

[138] S. Pfeiffer. RFC 3533: The Ogg Encapsulation Format Version 0, 2003. Available at `http://www.ietf.org/rfc/rfc3533.txt`.

[139] S. Pfeiffer. Architecture of a Video Web - Experience with Annodex. In *Proceedings of the W3C Video on the Web Workshop*, pages 1–5, Brussels, Belgium, December 2007.

[140] Pfeiffer, Silvia and Parker, Conrad and Pang, A. Internet Draft: Specifying Time Intervals in URI Queries and Fragments of Time-based Web Resources, 2005. Available at `http://annodex.net/TR/draft-pfeiffer-temporal-fragments-03.html`.

[141] E. Pietriga, C. Bizer, D. Karger, and R. Lee. Fresnel: a Browser-Independent Presentation Vocabulary for RDF. In *Proceedings of the 5^{th} International Semantic Web Conference*, pages 158–171, Athens, USA, November 2006.

[142] C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. Personal Content Management System: a Semantic Approach. *Journal of Visual Communication and Image Representation – Special Issue on Emerging Techniques for Multimedia Content Sharing, Search and Understanding*, 20(2):131–144, February 2009.

[143] E. Prud'hommeaux and A. Seaborne, editors. *SPARQL Query Language for RDF*. W3C Recommendation. World Wide Web Consortium, January 2008. Available at `http://www.w3.org/TR/rdf-sparql-query/`.

[144] D. Raggett, A. Le Hors, and I. Jacobs, editors. *HyperText Markup Language (HTML) 4.01 – Specification*. W3C Recommendation. World Wide Web Consortium, December 1999. Available at `http://www.w3.org/TR/html4/`.

[145] D. Rayers. Metadata in tv production: Associating the tv production process with relevant technologies. *SMPTE Motion Imaging Journal*, 112(9):287–292, September 2003.

[146] D. Rosenthal. Bit Preservation: A Solved Problem? In *Proceedings of the 5^{th} International Conference on Preservation of Digital Objects*, pages 1–7, London, UK, September 2008.

[147] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of Recommendation Algorithms for E-commerce. In *Proceedings of the 2^{nd} ACM Conference on Electronic Commerce*, pages 158–167, Minneapolis, USA, October 2000.

[148] R. Schmidt, R. King, A. Jackson, C. Wilson, F. Steeg, and P. Melms. A Framework for Distributed Preservation Workflows. *International Journal of Digital Curation*, 5:205–217, July 2010.

[149] H. Schulzrinne, A. Rao, and R. Lanphier. RFC 2326: Real Time Streaming Protocol, 1998. Available at `http://www.ietf.org/rfc/rfc2326.txt`.

[150] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, September 2007.

[151] A. Seaborne, G. Manjunath, and C. Bizer, editors. *SPARQL Update – A Language for Updating RDF Graphs*. W3C Member Submission. World Wide Web Consortium, 2008. Available at `http://www.w3.org/Submission/SPARQL-Update/`.

[152] B. Selic. The pragmatics of model-driven development. *IEEE Software*, 20(5):19–25, October 2003.

[153] N. Shadbolt, W. Hall, and T. Berners-Lee. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, May 2006.

[154] M. Smith, C. Welty, and D. McGuinness, editors. *OWL Web Ontology Language Guide*. W3C Recommendation. World Wide Web Consortium, February 2004. Available at `http://www.w3.org/TR/owl-guide/`.

[155] SMPTE. Unique Material Identifier (UMID), 2000. SMPTE 330M.

[156] SMPTE. Type D-10 Stream Specification – MPEG-2 4:2:2P@ML for 525/60 and 625/50 (IMX-50), 2001. SMPTE 356M.

[157] SMPTE. Material Exchange Format (MXF) – File Format Specification, 2004. SMPTE 377M.

[158] SMPTE. Material eXchange Format (MXF) – Operational Pattern 1a (Single Item, Single Package), 2004. SMPTE 378M.

[159] Society of Motion Picture and Television Engineers. SMPTE RP 136: Time and Control Codes for 24, 25 or 30 Frame-Per-Second Motion- Picture Systems, 2004. Available at `http://www.smpte.org/standards/`.

[160] F. Stegmaier, W. Bailer, T. Buerger, M. Dller, M. Hffernig, W. Lee, V. Malais, C. Poppe, R. Troncy, H. Kosch, and R. Van de Walle. How to Align Media Metadata Schemas? Design and Implementation of the Media Ontology. In *Proceedings of the 4th International Conference on Semantic and Digital Media Technologies – Workshop on Semantic Multimedia Database Technologies*, pages 1–14, Graz, Austria, December 2009.

[161] The WebM Project. VP8 Data Format and Decoding Guide, 2010. Available at `http://www.webmproject.org/code/specs/`.

[162] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, editors. *XML Schema Part 1: Structures – Second Edition*. W3C Recommendation. World Wide Web Consortium, October 2004. Available at `http://www.w3.org/TR/xmlschema-1/`.

[163] R. Troncy. Bringing the IPTC News Architecture into the Semantic Web. In *Proceedings of the 7th International Semantic Web Conference*, pages 483–498, Karlsruhe, Germany, 2008.

[164] R. Troncy, O. Celma, S. Little, R. Garcia, and C. Tsinaraki. MPEG-7 based Multimedia Ontologies: Interoperability Support or Interoperability Issue? In *Proceedings of the 1st International Workshop on Multimedia Annotation and Retrieval enabled by Shared Ontologies*, pages 1–14, Genova, Italy, 2007.

[165] R. Troncy, L. Hardman, J. Van Ossenbruggen, and M. Hausenblas. Identifying Spatial and Temporal Media Fragments on the Web. In *Proceedings of the W3C Video on the Web Workshop*, pages 1–6, Brussels, Belgium, December 2007.

[166] B. Van De Keer, D. Van Rijsselbergen, M. Verwaest, E. Mannens, and R. Van de Walle. Extending a Data Model for a Drama Product Manufacturing System with Re-purposing Support. In *Proceedings of the 13th IASTED International Conference on Internet and Multimedia Systems and Applications*, pages 105–110, Honolulu, USA, August 2009.

[167] H. Van de Sompel, C. Lagoze, M. Nelson, S. Warner, R. Sanderson, and P. Johnston. Adding eScience Assets to the Data Web. In *Proceedings of the 2nd International Workshop on Linked Data*, pages 1–10, Madrid, Spain, April 2009.

[168] H. Van de Sompel, R. Sanderson, M. Nelson, L. Balakireva, H. Shankar, and S. Ainsworth. An HTTP-Based Versioning Mechanism for Linked Data. In *Proceedings of the 3rd International Workshop on Linked Data*, pages 1–10, Raleigh, USA, April 2010.

[169] D. Van Deursen, C. Poppe, G. Martens, E. Mannens, and R. Van de Walle. XML to RDF Conversion: a Generic Approach. In *Proceedings of the 4th International Conference on Automating Production of Cross Media Content for Multi-channel Distribution*, pages 138–143, Florence, Italy, November 2008.

[170] D. Van Deursen, R. Troncy, E. Mannens, S. Pfeiffer, Y. Lafon, and R. Van de Walle. Implementing the Media Fragments URI Specification. In *Proceedings of the 19th International World Wide Web Conference (WWW)*, pages 1361–1363, Raleigh, USA, April 2010.

[171] D. Van Deursen, W. Van Lancker, S. De Bruyne, W. De Neve, E. Mannens, and R. Van de Walle. Format-independent and Metadata-driven Media Resource Adaptation using Semantic Web Technologies. *Multimedia Systems Journal*, 16(2):85–104, January 2010.

[172] D. Van Deursen, W. Van Lancker, W. De Neve, T. Paridaens, E. Mannens, and R. Van de Walle. NinSuna: a Fully Integrated Platform for Format-independent Multimedia Content Adaptation and Delivery based on Semantic Web Technologies. *Multimedia Tools and Applications – Special Issue on Data Semantics for Multimedia Systems*, 46(2):371–398, January 2010.

[173] D. Van Deursen, W. Van Lancker, P. Debevere, and R. Van de Walle. Format-independent Media Delivery, Applied to RTP, MP4, and Ogg. In *Proceedings of the 4th International Conference on Multimedia and Ubiquitous Engineering*, pages 1–6, Cebu, Philippines, August 2010.

[174] D. Van Deursen, W. Van Lancker, and R. Van de Walle. On Media Delivery Protocols in the Web. In *Proceedings of the 11th IEEE International Conference on Multimedia & Expo*, pages 1028–1033, Singapore, July 2010.

[175] D. Van Rijsselbergen, B. Van De Keer, and R. Van de Walle. The Canonical Expression of the Drama Product Manufacturing Processes. *Multimedia Systems – Special Issue on Canonical Processes of Media Production*, 14(6):395–403, December 2008.

[176] D. Van Rijsselbergen, B. Van De Keer, M. Verwaest, E. Mannens, and R. Van de Walle. On the Implementation of Semantic Content Adaptation in the Drama Manufacturing Process. In *Proceedings of the 10th IEEE International Conference on Multimedia & Expo*, pages 822–825, New York, USA, June 2009.

[177] D. Van Rijsselbergen, M. Verwaest, E. Mannens, and R. Van de Walle. On how Metadata enables Enriched File-based Production Workflows. In *Proceedings of the SMPTE Annual Tech Conference and Expo*, pages 1–19, Hollywood, USA, October 2009.

[178] D. Van Rijsselbergen, M. Verwaest, B. Van De Keer, and R. Van de Walle. Introducing the Data Model for a Centralized Drama Production System. In *Proceedings of the 8th IEEE International Conference on Multimedia & Expo*, pages 615–618, Beijing, China, July 2007.

[179] W3C Media Annotations Working Group. Media Annotations Working Group, 2010. Available at `http://www.w3.org/2008/WebVideo/Annotations/`.

[180] W3C Media Fragments Working Group. Media Fragments Working Group, 2010. Available at `http://www.w3.org/2008/WebVideo/Fragments/`.

[181] W3C Multimedia Semantics Incubator Group . Multimedia Semantics Incubator Group, 2007. Available at `http://www.w3.org/2005/Incubator/mmsem/`.

[182] J. Weng, C. Miao, A. Goh, Z. Shen, and R. Gay. Trust-based agent community for collaborative recommendation. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1260–1262, Hakodate, Japan, May 2006.

[183] G. Xing, Z. Xia, and A. Ernest. Building Automatic Mapping between XML Documents Using Approximate Tree Matching. In *Proceedings of the 22nd International ACM Symposium on Applied Computing*, pages 525–526, Seoul, South Korea, March 2007.

[184] Xiph.org Foundation. Theora Specification, 2009. Available at `http://theora.org/doc/Theora.pdf`.

[185] Xiph.org Foundation. Vorbis I Specification, 2010. Available at `http://xiph.org/vorbis/doc/Vorbis_I_spec.pdf`.

[186] X. Yang, M. L. Lee, and T. W. Ling. Resolving Structural Conflicts in the Integration of XML Schemas: A Semantic Approach. *Lecture Notes in Computer Science – Conceptual Modeling – ER 2003*, 2813:520–533, October 2003.