

Interactieve dynamische

Tijdslijn

voor de nieuwe website van het

Rijksmuseum

Delft,
maart 2004,
Dhr. J.H. Companje

Afstudeerrapport

examinatoren:
Mevr. A.J. Warnink
Mevr. A.D. van Duuren

Fabrique

RIJKS MUSEUM
a m s t e r d a m



Haagse
Hogeschool



REFERAAT

Companje, JH

Interactieve dynamische tijdslijn voor de website van het Rijksmuseum

Delft, Fabrique, maart 2004

Afstudeerverslag van Rick Companje, in het kader van het afstuderen voor de opleiding Vormgeving en Ontwerp van Interactie (VIA) aan de Haagse Hogeschool te 's-Gravenhage.

Dit afstudeerverslag bevat de chronologische beschrijving van het afstuderen van Rick Companje bij Fabrique in Delft in de periode november 2003 tot en met maart 2004.

De afstudeeropdracht bestond uit het definiëren en ontwikkelen van een interactieve dynamische tijdslijn die deel uit maakte van de in 2004 door Fabrique en Q42 ontwikkelde website van het Rijksmuseum.

Descriptoren:

- Afstudeeropdracht
- Rijksmuseum
- Fabrique
- Q42
- Interactieve tijdslijn
- Macromedia Flash MX 2004
- Actionscript 2.0
- Object georiënteerd programmeren (OOP)
- UML

VOORWOORD

Met veel voldoening presenteer ik u mijn afstudeerrapport!

In dit rapport geef ik een beschrijving van de afstudeerperiode die ik heb genoten in het kader van de opleiding 'Vormgeving en Ontwerp van Interactie' aan de Haagse Hogeschool.

Het afstuderen heeft plaatsgevonden bij Fabrique [design, communicatie & nieuwe media] in Delft waar ik de kans kreeg een zeer uitdagende en veelzijdige afstudeeropdracht voor het Rijksmuseum te Amsterdam uit te voeren.

Ik wil graag van deze gelegenheid gebruik maken om Fabrique en met name Paul Stork en Matthijs Klinkert te bedanken die mij in staat stelden deze ontzettend uitdagende en veelzijdige afstudeeropdracht bij Fabrique te mogen uitvoeren.

Verder wil ik graag Kars Veling en Jeroen Pulles enorm bedanken voor hun betrokkenheid en technische ondersteuning bij het uitvoeren van mijn afstudeeropdracht.

De examinatoren van mijn project, mevrouw Warnink en mevrouw van Duuren, wil ik graag van harte bedanken voor de tijd die zij hebben willen steken in begeleiding en ondersteuning vanuit de Haagse Hogeschool.

Tenslotte wil ik graag alle personen, en met name Dora, die mij geholpen hebben bij het samenstellen en becommentariëren van mijn afstudeerrapport, bedanken voor hun inzet en kritische blik.

Ik wens u veel plezier toe bij het lezen van dit rapport!

Vriendelijke groet,

Rick Companje

INHOUDSOPGAVE

1.	INLEIDING	1
2.	PROJECTPLAN.....	2
2.1	Projectorganisatie	2
2.2	Afstudeeropdracht	4
2.3	Ontwikkelstrategie.....	5
2.4	Projectplanning.....	8
2.5	Bezoeken en overlegvormen.....	8
3.	ANALYSE EN DEFINITIE.....	10
3.1	Interne en externe doelen bepalen	10
3.2	Functionaliteiten van bestaande tijdslijnen analyseren	12
3.3	Functionele systeemeisen bepalen	15
3.4	Aanvullende specificaties bepalen.....	16
3.5	Functionaliteiten en omgeving analyseren.....	17
3.6	Technische analyse uitvoeren.....	21
3.7	Gegevens analyse uitvoeren	23
3.8	Resultaat van de fase analyse en definitie vaststellen.....	27
4.	FUNCTIONEEL ONTWERP	28
4.1	Plaats van de tijdslijn in website vaststellen.....	28
4.2	Klassendiagram opstellen	28
4.3	Functionaliteiten van de interface vastleggen.....	32
4.4	Resultaat van het functioneel ontwerp vaststellen	36
5.	TECHNISCH ONTWERP.....	37
5.1	Softwarepatronen onderzoeken	37
5.2	Softwareclasses technisch beschrijven.....	38
5.3	Dataverwerking technisch beschrijven	38
5.4	Meertalige ondersteuning technisch beschrijven.....	38
5.5	Resultaat van de fase technisch ontwerp vaststellen.....	38
6.	GRAFISCH ONTWERP	39
7.	PROGRAMMERING.....	40
7.1	Prototype ontwikkelen	40
7.2	Tussentijdse testen uitvoeren	41
7.3	Ontwikkeling van de tijdslijn doorzetten	42
7.4	Resultaat van de fase programmering vaststellen.....	42
8.	AFRONDENDE FASEN.....	43
8.1	Redactie	43
8.2	Implementatie	43
8.3	Documentatie.....	43
9.	EVALUATIE	44
9.1	Procesevaluatie	44
9.2	Productevaluatie.....	46
	LITERATUUROPGAVE.....	47
	BIJLAGEN.....	48

1. INLEIDING

In dit rapport geef ik, Rick Companje, een beschrijving van het door mij doorlopen proces en het opgeleverde product tijdens mijn afstudeerperiode.

Het afstuderen heeft plaatsgevonden bij Fabrique [design, communicatie & nieuwe media] gevestigd te Delft waar ik de kans kreeg een zeer uitdagende en veelzijdige afstudeeropdracht voor het Rijksmuseum te Amsterdam uit te voeren.

Zonder bij het project betrokken te zijn geweest moet het voor de lezer mogelijk zijn zich een goed beeld van de inhoud en diepgang van de afstudeeropdracht te vormen.

De hoofdstukindeling van dit rapport is gebaseerd op de fasering die ik gehanteerd heb tijdens het uitvoeren van de afstudeeropdracht.

In hoofdstuk 2 beschrijf ik de context waarin het uitvoeren van mijn afstudeeropdracht heeft plaatsgevonden. De betrokken partijen worden geïntroduceerd en hun rollen verduidelijkt waarna vervolgens wordt ingegaan op de inhoud en de aanpak van de afstudeeropdracht.

Hoofdstuk 3 beschrijft de uitgevoerde activiteiten binnen de fase 'analyse en definitie'. In dit hoofdstuk wordt beschreven hoe de reikwijdte van het project is onderzocht en afgebakend en worden een aantal uitgevoerde analyses beschreven.

Hoofdstuk 4 gaat in op de activiteiten die ik heb uitgevoerd binnen de fase functioneel ontwerp. In die fase werden alle functionaliteiten van het te ontwikkelen systeem beschreven. Vervolgens wordt in hoofdstuk 5 ingegaan op de activiteiten die betrekking hebben op de fase technisch ontwerp. In dit hoofdstuk wordt beschreven hoe de functionaliteiten uit de vorige fase technisch uitgewerkt zijn.

Hoofdstuk 6 gaat in op de fase grafisch ontwerp waarna vervolgens in hoofdstuk 7 de uiteindelijke programmering van het systeem wordt beschreven. Hoofdstuk 8 heeft betrekking op een aantal afrondende fasen.

Uiteindelijk worden in hoofdstuk 9 het doorlopen proces en het opgeleverde product geëvalueerd.

2. PROJECTPLAN

Dit hoofdstuk beschrijft de context waarin het uitvoeren van mijn afstudeeropdracht heeft plaatsgevonden. De betrokken partijen worden geïntroduceerd en hun rollen verduidelijkt. Vervolgens wordt ingegaan op de inhoud en de aanpak van de afstudeeropdracht.

2.1 Projectorganisatie

Eerst wordt de projectorganisatie geïntroduceerd waarmee ik tijdens het uitvoeren van mijn afstudeeropdracht te maken kreeg.

Rijksmuseum

Het Rijksmuseum gevestigd te Amsterdam is het grootste museum van Nederland. Er wordt veel aandacht besteed aan educatie en aan de inrichting en vormgeving van tentoonstellingen en andere uitingen. Hiervoor worden regelmatig vooraanstaande vormgevers en ontwerpers uitgenodigd.



De komende jaren wordt het Rijksmuseum verbouwd op een grootscheepse manier, zowel architectonisch als presentatietechnisch.

De huidige website, die inmiddels al een jaar of vijf mee gaat, is aan vervanging toe en wordt helemaal vernieuwd. Dit is een omvangrijk en complex project waarvoor het Rijksmuseum twee partijen heeft gecontracteerd: Fabrique en Q42.

Fabrique

Fabrique is een multidisciplinair ontwerp bureau, dat zich bezig houdt met het ontwerp en de realisatie van projecten op het gebied van industrieel ontwerp, grafisch ontwerp en nieuwe media.



Er zijn binnen de organisatie verschillende ontwerpteams aanwezig, elk met hun eigen soort activiteiten. Op dit moment bestaan er binnen Fabrique drie nieuwe media teams en twee grafische teams. Fabrique Industrial Design is een aparte BV en houdt zich bezig met productontwikkeling. De verschillende ontwerpdisciplines zijn echter nauw met elkaar verweven.

Fabrique is opgericht in 1992, telt momenteel in totaal achtenzestig medewerkers en heeft sinds het begin al veel bekende opdrachtgevers in zowel de entertainmentwereld als de zakenwereld. Grote namen als Mojo, Orange, Albert Heijn, Essent, Lowlands, Koninklijke Bibliotheek, Sweex, Gemeente Den Haag en nog vele anderen vullen het portfolio van Fabrique.

Fabrique is voor de nieuwe Rijksmuseum website middels een pitch gecontracteerd. Dat wil zeggen dat het Rijksmuseum meerdere partijen heeft uitgenodigd om met voorstellen te komen voor een nieuwe website. Op de pitch-presentatie van Fabrique werd positief gereageerd, toch twijfelde het Rijksmuseum aan technische expertise van Fabrique. Het Rijksmuseum gaf namelijk aan, naast een grote vernieuwing van vormgeving en structuur van de website, ook een technologische sprong te willen maken.

Om in deze behoefte te kunnen voorzien heeft Fabrique contact gezocht met technische partner, Q42, waarmee al eerder in samenwerkingsverband succesvol aan grote opdrachten werd gewerkt.

Fabrique en Q42 hebben al vaker bewezen in staat te zijn hun krachten te bundelen en op hoog niveau mee te denken en oplossingen te bieden.

Q42

Q42 is een klein en innovatief internet bureau dat zich richt op het creëren van gebruiksvriendelijke internet applicaties. Q42 is gevestigd in Den Haag en telt negen medewerkers.

De innovatieve DHTML toepassingen van Q42 zijn ingezet bij internet toepassingen van bekende landelijke organisaties zoals Cultureel erfgoed, Vara's Lagerhuis en SchoolTV.

De bekendheid van Q42 groeit tegenwoordig voornamelijk door hun populaire webbased XML Editor Xopus. Xopus is een zeer gebruiksvriendelijke en flexibele editor om content (inhoud/gegevens) van grote complexe websites te onderhouden.

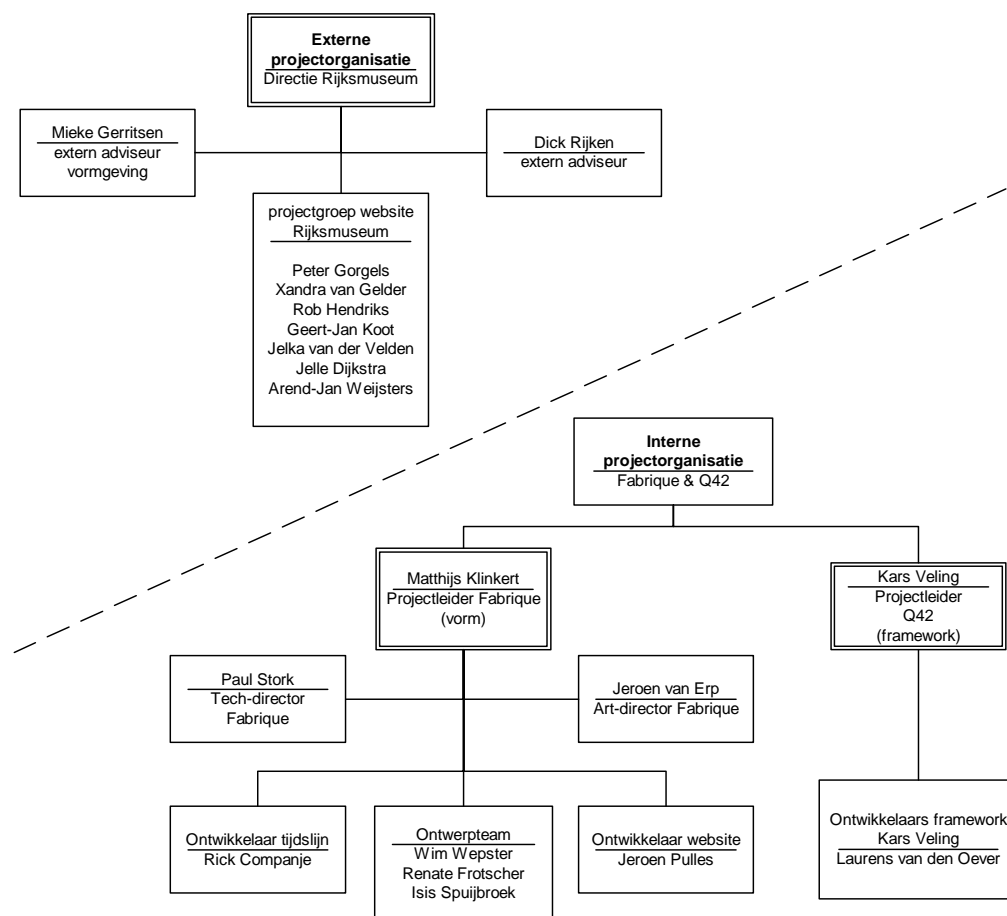
Ahold, één van Fabrique's grootste klanten heeft kort geleden Xopus van Q42 gekozen voor het beheren van hun XML content voor de Nederlandse commerciële websites van Albert Heijn, Etos, Gall & Gall, Albert en DeliXL. Samen met Fabrique ontwikkeld Q42 een geheel nieuwe management interface voor deze multinational.

Voor het omvangrijke content management systeem voor de website van het Rijksmuseum wordt Xopus eveneens toegepast.



Organogram van de projectorganisatie

In het onderstaande organogram is de projectorganisatie weergegeven. Er waren twee projectorganisaties te onderscheiden: een externe en een interne. De externe projectorganisatie was de projectgroep zoals deze bij het Rijksmuseum was samengesteld. De interne projectorganisatie bestond uit de samenwerkende partijen Fabrique en Q42.



Figuur 1: Interne en externe projectorganisatie

Q42 ontwikkelde als onderdeel van de interne projectorganisatie het technische framework, zorgde ervoor dat de grote hoeveelheid data aanspreekbaar werd en regelde zaken zoals webhosting.

Fabrique was verantwoordelijk voor de vorm en structuur van de nieuwe website voor het Rijksmuseum maar nam daarbij ook een aantal technische onderdelen voor haar rekening. Het schema van de interne projectorganisatie laat zien dat er bij Fabrique twee staff-leden betrokken zijn bij het project: Jeroen van Erp en Paul Stork, twee leden uit het management team van Fabrique. Jeroen van Erp vervult de art-directors rol bij dit project en coördineert daarbij de totstandkoming van de vormgeving van de nieuwe website.

Paul Stork is actief als tech-director. Hij levert adviezen op technisch gebied aan de externe projectorganisatie en fungeert als voornaamste tussenpersoon tussen Q42 en Fabrique.

Ook in de externe projectorganisatie bevinden zich twee staff-leden. Grafisch ontwerper Mieke Gerritsen en nieuwe media adviseur Dick Rijken. Zij vervullen de rol van adviseur bij de externe projectorganisatie. Mieke Gerrits leverde adviezen over vormgeving. Dick Rijken leverde algemene adviezen op het gebied van nieuwe media.

2.2 Afstudeeropdracht

Sinds september 2002 ben ik, Rick Companje, actief bij Fabrique als webontwikkelaar. Eerst als stagiair, later als werknemer in opleiding via het open-duale traject van de opleiding VIA van de Haagse Hogeschool.

In oktober 2003 werd mij bij Fabrique de kans geboden af te studeren op een bijzonder uitdagend project, 'de tijdslijn', een afgebakend onderdeel van de nieuwe website van het Rijksmuseum.

Na de opdracht grondig bestudeerd te hebben, ben ik direct contact gaan zoeken met de coördinatoren op school en zijn we gaan zoeken naar een mogelijkheid om vervroegd in te kunnen stromen in de afstudeerlichting november 2003-maart 2004.

Na aan een aantal voorwaarden voldaan te hebben kon ik een aantal weken later van start gaan met het uitvoeren van de afstudeeropdracht.

Het Rijksmuseum wilde de grote schat aan informatie over cultuurhistorische objecten uit haar collectie op de website presenteren. Daarbij was het belangrijk dat de verbanden tussen de objecten, periodes, gebeurtenissen en kunstenaars voor de bezoekers zichtbaar zouden zijn.

Bezoekers met verschillende achtergronden en interesses moesten de informatie waarnaar zij zochten kunnen vinden maar ook geïnspireerd raken en verleid worden tot het vergaren van extra kennis.

Het Rijksmuseum wilde om die reden de website voorzien van een dynamische interactieve tijdslijn. In mijn afstudeerperiode was ik verantwoordelijk voor de definiëring en ontwikkeling van de gehele tijdslijn.

Als afstudeerder heb ik verder naast de zorg voor de tijdslijn ook een algemene rol als projectlid vervuld en was ik actief betrokken bij diverse technische vergaderingen. Daarnaast dacht ik actief mee over de functionaliteit van de overkoepelende website.

Het was voor mezelf relevant me ook te betrekken bij de vormgeving van de overkoepelende website aangezien mijn tijdslijn deel van dat geheel zou gaan uitmaken. Door mijn betrokkenheid kon op het juiste moment de juiste keuzes maken.

Verder ben ik verantwoordelijk geweest voor het verlevendigen van één van de visuele concepten van de vormgevers door in Flash een interactieve demo te maken die gebruikt werd tijdens een presentatie bij het Rijksmuseum.

2.3 Ontwikkelstrategie

Tijdens het opstarten van het project heb ik een keuze gemaakt voor de te gebruiken ontwikkelstrategie. Hierbij heb ik rekening gehouden met de doelstelling van het project en met mijn ervaringen met methoden die ik eerder in praktijk had toegepast.

Bij VIA studenten is het evolutionair ontwikkelen van informatiesystemen volgens de IAD (Iterative Application Design) methode verre weg de meest gebruikte ontwikkelstrategie. Dat is niet zo vreemd want deze methode is tijdens de studie veelvuldig toegepast tijdens de practica. Iedere VIA student heeft dus ervaring met IAD, dus ik ook.

Bij Fabrique heb ik echter al enige tijd volgens een andere ontwikkelstrategie gewerkt. Deze strategie, die bij Fabrique de standaardfasering wordt genoemd (zie Figuur 2), leent zich net als IAD om software volgens een iteratief proces tot stand te laten komen.

De standaardfasering van Fabrique laat de systeemontwerper vrij te kiezen voor bepaalde modelleringstechnieken en het combineren daarvan.

Mede omdat ik al langer bij Fabrique werk en helemaal gewend ben aan deze ontwikkelstrategie was de standaardfasering voor mij een logische keuze. Een andere reden om te kiezen voor deze methode was dat ik deel uit maakte van de interne projectorganisatie en dat zij, voor de ontwikkeling van de overkoepelende website, dezelfde methode gebruikten.

Ik heb er voor gekozen de standaardfasering ook als 'kapstok' te gebruiken voor de structuur van dit rapport aangezien ik dit rapport het doorlopen proces op dezelfde chronologische wijze beschrijf.

Analyse en definitie

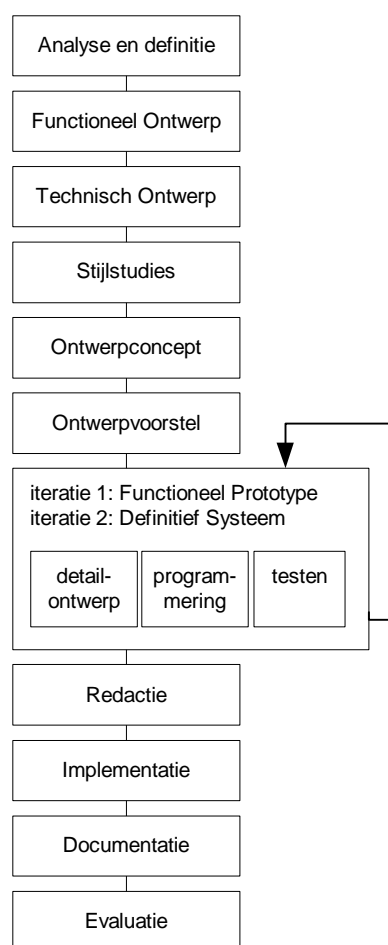
Zoals te zien is in het schema op deze pagina is 'analyse en definitie' de eerste fase van de standaardfasering. Tijdens deze fase wordt de reikwijdte van het project onderzocht en afgebakend. Er wordt gekeken naar de doelstellingen, doelgroepen, inhoud, vorm, techniek, beheer en organisatie van het project.

De gevonden randvoorwaarden, eisen en wensen worden vastgelegd. Het resultaat van deze fase is een projectdefinitie.

Functioneel ontwerp

In de fase functioneel ontwerp worden alle functionaliteiten van het te ontwikkelen systeem bepaald en vastgelegd.

Het resultaat van deze fase is een document genaamd 'functioneel ontwerp' waarin bovengenoemde functionaliteiten vastgelegd zijn.



Figuur 2: Standaardfasering van Fabrique

Technisch ontwerp

Het resultaat van deze fase is de eerste versie van het document 'technisch ontwerp'. Het technisch ontwerp bevat de uitgewerkte technische beschrijvingen van alle functionaliteiten uit de vorige fase. In latere fasen (bijv. tijdens de programmering) wordt het document indien nodig nog aangevuld en/of bijgewerkt.

Stijlstudies

Bij het uitvoeren van de stijlstudies wordt gezocht naar de juiste 'look & feel'. De juiste sfeer wordt gezocht door binnen de gestelde grenzen te spelen met kleur, vorm, beeldmateriaal, typografie, vlakverdeling, interface-elementen, etc.

Ontwerpconcept

Het concept ontwerp levert doorgaans een aantal verschillende concepten op. De opdrachtgever kiest het concept dat qua stijl het best bij de organisatie past.

Ontwerpvoorstel

Bij een ontwerpvoorstel worden aan de hand van het functioneel ontwerp een aantal gedetailleerde ontwerpen van representatieve schermen het systeem gemaakt. In deze ontwerpen wordt gebruik gemaakt van echte tekst en beeldmateriaal en definitieve naamgeving in de navigatie.

Functioneel prototype

In de standaardfasering van Fabrique is de fase functioneel prototype opgenomen. In deze fase wordt eerst een detailontwerp gemaakt op basis van het ontwerpvoorstel. Vervolgens wordt de keuze gemaakt of het prototype in de breedte of in de diepte wordt uitgewerkt. Bij een prototype van brede opzet worden alle functionaliteiten ondergebracht in de interface maar minder uitgewerkt.

Wanneer prototype in de diepte uitgewerkt wordt, worden maar een aantal functionaliteiten uitwerkt, maar deze zullen ook volledig operationeel zijn.

Na de ontwikkeling van het prototype wordt deze altijd getest. Afhankelijk van het soort project worden één of meerdere tests uitgevoerd. Meestal wordt er een usability test uitgevoerd, ook steeds vaker blijkt een performance test gewenst te zijn.

Definitief systeem

De resultaten van de tests die zijn uitgevoerd op het prototype zijn directe input voor de deze fase, de ontwikkeling van het definitieve systeem.

Eerst wordt het detailontwerp van het prototype verfijnd op basis van de testresultaten. Vervolgens worden de functionaliteiten waarmee het prototype nog niet uitgerust was ontwikkeld en geconstateerde problemen opgelost.

In sommige gevallen vindt na oplevering van het systeem nog een extra test plaats.

Redactie

In de fase 'redactie' wordt het systeem gevuld met reële data. De opdrachtgever zal bepaalde delen handmatig invoeren, andere delen zullen worden overgenomen of worden geconverteerd uit een bestaand systeem.

In deze fase vinden ook controles plaats op betrouwbaarheid en volledigheid van de data.

Implementatie

De fase 'implementatie' is het moment dat het volledig werkende systeem opgeleverd en geïnstalleerd wordt op de webserver.

In deze fase vindt ook de mondelinge kennisoverdracht plaats over het systeem.

Onduidelijkheden zullen worden meegenomen in de fase 'documentatie'.

De laatste controles vinden plaats en eventuele fouten die in een eerder stadium nog niet zichtbaar waren worden opgelost.

Documentatie

In de fase 'documentatie' wordt gezorgd dat de belanghebbenden voorzien worden van de juiste documenten. Handleidingen en eventuele herziene versies van documenten (zoals het technisch ontwerp) worden opgeleverd.

Evaluatie

In de fase evaluatie worden het product en het proces grondig geëvalueerd. Intern vinden er evaluerende gesprekken plaats, maar ook de opdrachtgever wordt gevraagd zijn oordeel te vellen over het uiteindelijke product en zijn tevredenheid wordt gepeild met betrekking tot het doorlopen proces.

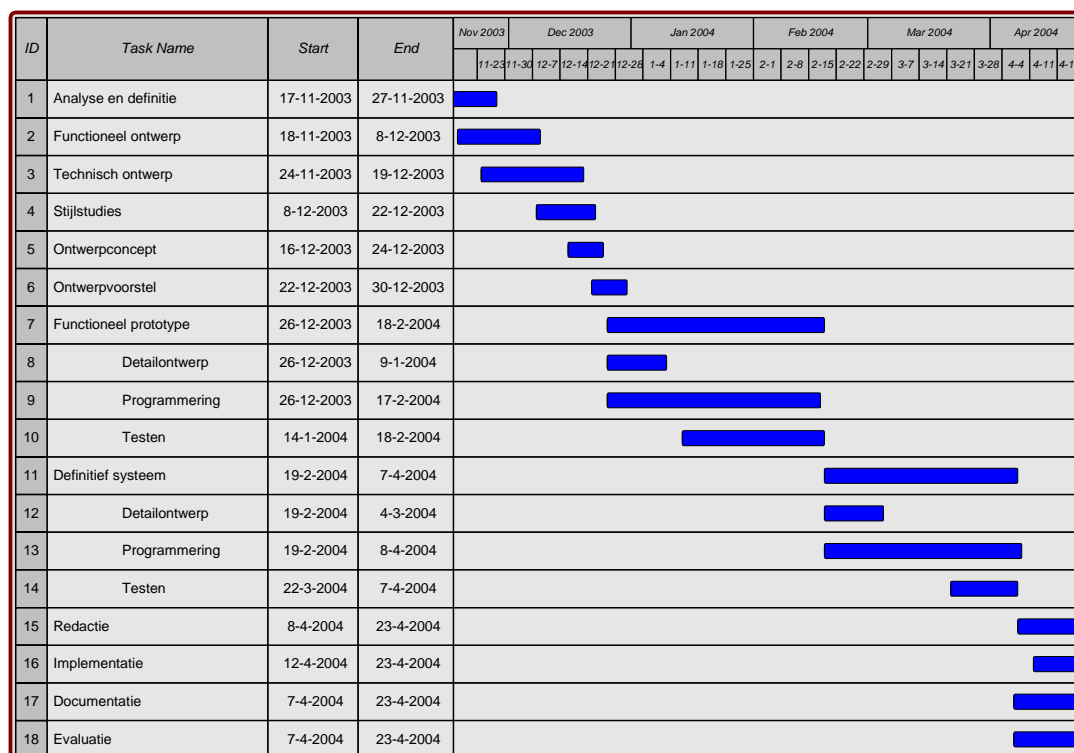
Zijn de gestelde doelen behaald, voldoet het product aan de voorafgestelde kwaliteitseisen, is de documentatie bruikbaar, blijft de gebruikte techniek toerijkend, waar zaten de problemen en op welke manier zijn oplossingen tot stand gekomen?

Er wordt bepaald of de gehanteerde ontwikkelstrategie voldoende heeft gewerkt en of de gehanteerde planning realistisch was.

De bevindingen uit deze fase dienen als input voor volgende ontwikkeltrajecten.

2.4 Projectplanning

De standaardfasering van Fabrique was een ideale basis voor de projectplanning. Alle fasen uit de standaardfasering heb ik terug laten komen in een zeer uitgebreide planning die ik aan het begin van het project heb opgesteld en telkens indien nodig bijstelde. In deze planning had ik, tot op de dag nauwkeurig, iedere te verwachten gebeurtenis vermeld alsmede de verschillende fasen die ik zou doorlopen. Het nadeel van deze gedetailleerde planning was dat het niet meer op één scherm of vel papier te tonen was waardoor de overzichtelijkheid verminderde. In de volgende Gantt-chart is echter de globale planning te zien welke ik aan het begin van het project voor ogen had.



Mijn derde bezoek aan het Rijksmuseum vond plaats in een later stadium en had ten doel mijn vorderingen betreffende de tijdslijn te tonen en feedback te verkrijgen voor het vervolgtraject. Dit was een erg spannend moment omdat het de eerste keer was dat ik de tijdslijn presenteerde aan de opdrachtgever.

Technisch overleg

Tijdens mijn afstudeerperiode werd er wekelijks vergaderd over de nieuwe website van het Rijksmuseum. Er vonden interne en externe vergaderingen plaats over functionaliteit, vorm en techniek.

Ongeveer één keer per drie weken werd er voor het project een technische vergadering belegd waar gesproken werd over technische keuzes en optredende problemen. Deze momenten waren voor mij erg geschikt om te overleggen hoe mijn tijdslijn qua techniek het beste in het geheel zou passen, welke databronnen ik wilde raadplegen, in welke vorm ik de data aangeleverd wenste te krijgen, hoe de resources (zoals afbeeldingen) benaderd konden worden etc.

De technische overlegmomenten hebben er voor gezorgd dat de andere ontwikkelaars goed op de hoogte waren van mijn voortgang en problemen die ik ondervond. Regelmatig loste het betrekken van een andere ontwikkelaar mijn probleem snel op.

Ook hebben de technische overlegmomenten er voor gezorgd dat de tijdslijn aan een stresstest onderhevig is gemaakt, daarover meer in paragraaf 7.2 (Tussentijdse testen uitvoeren)

Naast de technische overlegmomenten voor de nieuwe website van het Rijksmuseum vindt er bij Fabrique tweewekelijks een algemeen technisch overleg plaats waarbij alle ontwikkelaars/programmeurs bij Fabrique, waaronder ik zelf ook, ervaringen delen en kennis uitwisselen en waar technische problemen worden besproken. Onderwerpen die vaak ter tafel komen zijn nieuwe ontwikkelingen op het gebied van client- of serversided scripttalen, stylesheets en databases.

Tijdens mijn afstudeerperiode ben ik ook één maal aanwezig geweest bij het Flashbuddies overleg. Bij dit overleg houden Flash programmeurs en animators bij Fabrique elkaar op de hoogte van de ontwikkelingen op het gebied van deze 'ontwikkel omgeving' en worden onderling afspraken gemaakt over de gewenste werkwijze.

Mijn rol bij dit FlashBuddies overleg was het informeren van de andere leden over de mogelijkheden van de nieuwe versie Flash MX 2004 maar vooral welke grote gevolgen dit heeft voor de manier van werken wanneer naar deze versie wordt overgestapt. Ik ben betrokken bij dit overleg door de ervaring die ik tijdens het ontwikkelen van de tijdslijn op deed met Flash MX 2004. (zie paragraaf 3.6, Techniekselectie)

Volgens mijn verwachting werden er positieve en minder positieve signalen afgegeven. Een aantal mensen raakten enthousiast voor de 'volwassen' manier van programmeren in deze versie. Anderen reageerden met gemengde gevoelens en gaven aan de overstap (nog) niet te willen maken.

Begeleidingsmomenten

Naast alle inhoudelijke overlegmomenten had ik ook tweewekelijks een kort onderhoud over mijn proces met mijn begeleider Paul Stork. Hierbij hield ik hem op de hoogte van mijn vorderingen. Daarbij spraken we ook over bepaalde keuzes die ik maakte.

Persoonlijke technische begeleiding kreeg ik van Jeroen Pulles, een collega-ontwikkelaar waarvan ik heel veel heb geleerd op technisch gebied.

3. ANALYSE EN DEFINITIE

De eerste fase van de standaardfasering van Fabrique is analyse en definitie. Het doel van deze fase is de reikwijdte van het project te onderzoeken en af te bakenen. Onder andere zijn de interne- en externe doelen bepaald op basis waarvan de functionele systeemeisen in kaart zijn gebracht. Er is een technische analyse uitgevoerd, een gegevens analyse en er is gekeken welke algemene standaard in aanmerking kwamen om na te streven.

3.1 Interne en externe doelen bepalen

In deze paragraaf wordt beschreven hoe ik de interne en externe doelen heb bepaald. De interne doelen beschrijven de bedoelingen die het Rijksmuseum had met de tijdslijn. De externe doelen representeren de gebruikersbehoeften.

Interne doelen

De projectgroep binnen het Rijksmuseum (externe projectorganisatie) heeft een strategisch plan opgesteld waarin de visie en plannen van het Rijksmuseum voor de komende jaren staan vermeld. Dit rapport heet "Internet strategie 2004/2008" en was voor mij een goede bron van informatie.

Onder andere kon ik op basis van dit rapport een aantal interne doelen bepalen, de doelen die het Rijksmuseum intern met de tijdslijn voor ogen had.

Meer interne doelen heb ik verkregen uit de gesprekken die wekelijks gevoerd werden tussen de interne en externe projectorganisaties. Zelf ben ik ook bij één van die gesprekken aanwezig geweest. Per intern doel bepaalde ik de haalbaarheid en voegde ik een notitie bij waarin ik in sommige gevallen de relevantie aangaf, of vermeldde op welke manier aan de behoefte tegemoet zou worden gekomen. De volledige lijst met interne doelen is opgenomen in bijlage I. De onderstaande tabel is een deel daaruit.

Interne doelen	Haalbaarheid	Opmerking
1. Verbanden tussen elementen op de tijdslijn moeten zichtbaar zijn	hoog (maar zie opmerking)	De informatie op de tijdslijn wordt chronologisch horizontaal wordt geordend. Gelijktijdige gebeurtenissen en objecten worden onder elkaar getoond. Verbanden in tijd zijn op deze manier duidelijk zichtbaar. Andere verbanden (bijv. tussen kunstenaar en leermeester) lijken minder haalbaar en zullen een lagere prioriteit krijgen.
2. De tijdslijn zal tegelijkertijd op meerdere plaatsen op de website worden ingezet om informatie over verschillende thema's weer te geven.	hoog	Om dit te bereiken zal de tijdslijn een component zijn die zich in de presentatielaag bevindt en gegevens put uit verschillende informatiebronnen.
3. De vormgeving van de tijdslijn zal moeten aansluiten bij die van de rest van de nieuwe website.	hoog	De vormgeving van de tijdslijn zal worden bepaald wanneer er voldoende zekerheid bestaat over de vormgeving van de overkoepelende site.
4. ...		
5. ...		

Externe doelen bepalen

In het rapport *Internet strategie 2004/2008* stond vermeld dat de nieuwe website voor het Rijksmuseum zich richt op de volgende doelgroepen:

- Buitenlandse/Nederlandse eenmalige bezoeker
- Potentiële herhalingsbezoeker Buitenlands/Nederlands
- Gezin
- Kinderen/Jongeren
- Onderwijs
- Wetenschap
- Zakelijk (in mindere mate)

Omdat de tijdslijn een onderdeel is van de overkoepelende website moest ik mij voor de tijdslijn op dezelfde doelgroepen richten.

In bijlage I (analyse en definitie) is een matrix model opgenomen (afkomstig uit het bovengenoemde rapport) waarin de verschillende doelgroepen zijn uitgezet tegen verschillende onderdelen van de website.

De externe doelen, ook wel gebruikersbehoeften genoemd, heb ik vastgesteld door me te verplaatsen in de verschillende gebruikersgroepen en na te gaan welke externe doelen de uiteindelijke gebruikers zouden willen bereiken met de tijdslijn.

Ik heb geen echte gebruikers betrokken bij het bepalen van de externe doelen. Dit had ik achteraf zeker anders gedaan. Ondanks dat ik op deze manier toch behoorlijk veel gebruikersbehoeften heb weten te achterhalen, zouden bij een echt gebruikersonderzoek ongetwijfeld meer wensen naar voren zijn gekomen. De vraag is of ik die had kunnen uitwerken binnen deze korte periode, zo niet dan waren ze in ieder geval geïnventariseerd en zouden deze behoeften in een eventuele volgende versie van de tijdslijn verwerkt kunnen worden.

Net als voor de interne doelen bepaalde ik per gebruikersbehoefte de haalbaarheid en voegde ik een zelfde soort notitie toe.

De lijst met gebruikersbehoeften is eveneens opgenomen in bijlage I en bevat onder andere de gebruikersbehoeften uit de onderstaande lijst.

Gebruikersbehoeften	Haalbaarheid	Opmerking
1. ik wil weten welke kunstenaars tegelijk leefden	hoog	Deze informatie is zichtbaar doordat het leven van een kunstenaar als periode wordt weergegeven. Kunstenaars die tegelijkertijd leefden worden onder elkaar weergegeven waarbij de overlap duidelijk zichtbaar is.
2. ik wil snel van jaar X naar jaar Y kunnen springen zonder te hoeven uitzoemen	hoog	Er wordt een 'scrollbalk' toegevoegd die de totale tijdsperiode toont en waarmee direct naar een bepaald jaar gesprongen kan worden.
3. ik wil weten wie de maker is van een bepaald kunstwerk	hoog	Bij een muisklik op een kunstobject wordt een detailscherm getoond. Hierin wordt ook de naam van de maker vermeld en de mogelijkheid geboden door te klikken.
4. ...		
5. ...		

3.2 Functionaliteiten van bestaande tijdslijnen analyseren

In de fase analyse en definitie heb ik onderzoek gedaan naar de functionaliteiten van tijdslijnen die reeds op andere (museum) websites ingezet zijn.

Hierbij heb ik per tijdslijn de eigenschappen, voor- en nadelen in kaart gebracht.

Sterke punten kon ik verwerken in de tijdslijn die ik ging ontwikkelen. De zwakke punten van de andere tijdslijnen gaven mij een beeld met welke knelpunten ik ook te maken zou kunnen krijgen.

Ik heb twintig tijdslijnen geanalyseerd, variërend van zeer simpel, slecht uitgevoerd en statisch tot redelijk dynamisch en goed vormgegeven.

Alle bevindingen van deze analyse zijn opgenomen in bijlage II (Analyse van bestaande tijdslijnen). De tijdslijnen die voor mij het meest relevant waren zijn hieronder opgenomen.

Fabrique's pitch-demo

De eerste tijdslijn die ik heb geanalyseerd was een interactieve demo (throw-away prototype) van Fabrique die tijdens de pitch-presentatie bij het Rijksmuseum was gepresenteerd.

Deze demo had als doel een indruk te geven van de werking en het interactie-gevoel dat de tijdslijn zou krijgen en bevatte al een aantal functionaliteiten die het Rijksmuseum verwerkt zou willen zien in de tijdslijn zoals het filteren en het kunnen zien van verbanden tussen periodes en kunsthistorische objecten.

Deze demo was voor mij een zeer bruikbare bron van input voor het bepalen van de functionaliteiten die mijn tijdslijn zou gaan bevatten.

De demo deed een voorstel voor horizontaal scrollen door de tijd door middel van een scrollbar die de totale tijdsperiode voorstelde. Er werd zoom-functionaliteit gesuggereerd, waarmee de tijdslijn in verschillende detailniveau's bevraagd kon worden. Daarnaast was er ruimte gemaakt voor een filter waarmee de gebruiker bepaalde selecties zou moeten kunnen maken.

Op de tijdslijn was plaats voor afbeeldingen van kunstobjecten, levens van kunstenaars die evenals stijlperiodes werden getoond als horizontale balken. Daarnaast werd voorgesteld een aantal relevante historische gebeurtenissen en periodes op de tijdslijn te vermelden.

Het klikken op een kunstobject of periode maakte een detail-/informatievenster zichtbaar met daarin een beschrijving en een knop waarmee de verbanden met andere elementen op de tijdslijn zichtbaar werden.

De demo gaf goed het gevoel weer van de vloeiende bewegingen en geluiden die de tijdslijn zou maken wanneer er genavigeerd werd.

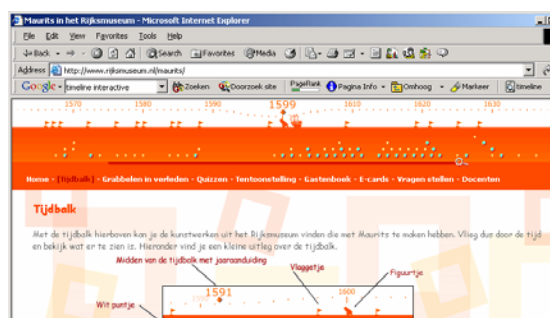


Maurits tentoonstelling

Een andere tijdslijn die ik heb geanalyseerd was die van de Maurits tentoonstelling die gehouden werd in 2000/2001 in het Rijksmuseum.

Deze tijdslijn was duidelijk gericht op kinderen en bevatte een uitgebreide legenda en uitleg.

De objecten en gebeurtenissen op de tijdslijn werden voorgesteld als kleine stipjes. Pas



wanneer de muiscursor boven een stipje werd gehouden verscheen een titel en een kleine afbeelding. Wanneer er geklikt werd op een object werd er in een nieuw venster de juiste informatiepagina over dit object op de Rijksmuseum website geopend.

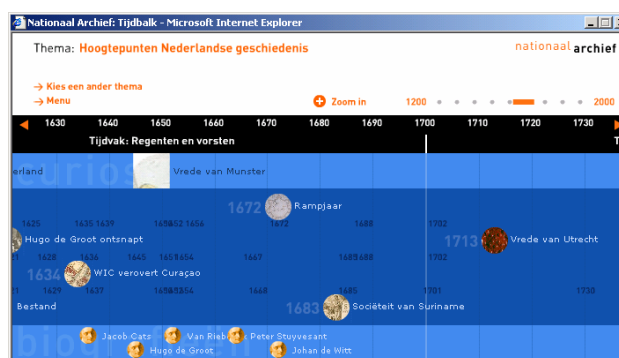
Een erg sterk punt van deze tijdslijn was de goede zoom functionaliteit. Door een vergrootglas te verschuiven kon van 500 jaar tot 10 jaar worden ingezoomd. Nadeel hierbij was dat de objecten nog steeds als stipjes getoond werden terwijl er wel plaats zou zijn om de kleine afbeeldingen direct te laten zien.

Nationaal Archief

De volgende tijdslijn die relevant leek om te onderzoeken was die van het Nationaal Archief.

Deze tijdslijn richt zich niet op kunsthistorische objecten en stijlperiodes, maar heeft betrekking op de geschiedenis van Nederland.

De tijdslijn werd geopend in pop-up, zag er grafisch goed uit en werd behoorlijk snel geladen.



De informatie op de tijdslijn werd gepresenteerd in een drietal horizontale lagen: curiosa, gebeurtenissen en biografieën.

De tijdslijn was uitgerust met zoom functionaliteit maar deze beperkte zich tot twee niveau's.

Ook bij deze tijdslijn was, in de vorm van een scrollbar, de totale tijdsperiode zichtbaar.

Als een zwaktepunt zag ik dat de teksten bij de gebeurtenissen elkaar regelmatig overlaptten waardoor deze niet meer leesbaar waren en dat er grote leegtes waren op de plaatsen waarvan geen informatie bekend was.

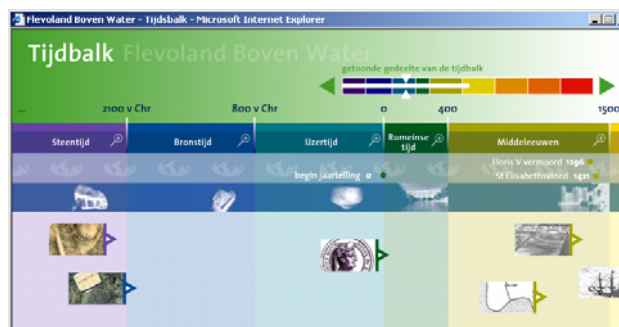
Een sterk punt van de tijdslijn van het Nationaal Archief was dat deze tijdslijn het gebruik van meerdere thema's ondersteunde.

Flevoland boven water

In tijdslijn van de provincie Flevoland wordt de geschiedenis beschreven van Flevoland in de prehistorie tot nu.

Ook bij deze tijdslijn was, in de vorm van een scrollbar, de totale tijdsperiode zichtbaar met als extra dat de scrollbar opgedeeld was in een aantal tijdvakken.

De tijdvakken op de scrollbar verschilden van kleur en kwamen



overeen met de kleurvlakken in de detailweergave onder de scrollbar.

Deze tijdslijn viel op doordat korte periodes met veel gebeurtenissen uitgerekt in de tijdslijn konden worden getoond. Hierdoor ontstaan geen grote leegtes op rustige tijden in de geschiedenis en wordt er tegelijkertijd ruimte gecreëerd voor grote hoeveelheden gebeurtenissen.

Ook was de tijdslijn van de provincie Flevoland in staat om gebeurtenissen van voor het begin van de jaartelling te tonen.

Verder werd er van bijna iedere periode een kleine afbeelding getoond, wat het geheel heel levendig maakte en liet de tijdslijn overlappende periodes onder elkaar zien.

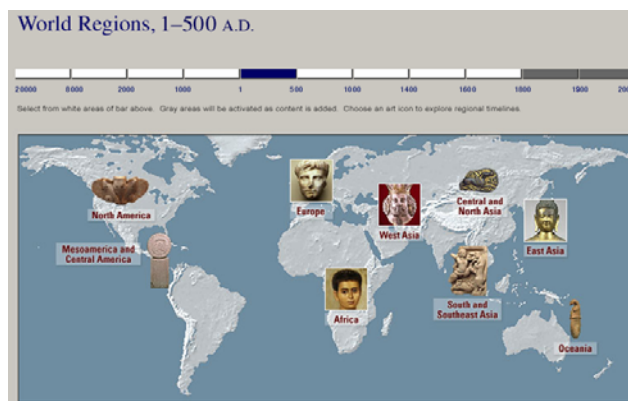
Metropolitan timeline of art

Een behoorlijk complexe tijdslijn is die van het Metropolitan Museum of Art in New York.

Tijdens het analyseren van deze tijdslijn ontdekte ik dat deze tijdslijn een ontzettend grote hoeveelheid kunsthistorische gegevens bezat die op een chronologische, geografische en thematische manier bevroegbaar waren. Vooral de geografische dimensie was ik nog niet eerder tegengekomen tijdens mijn zoektocht naar andere tijdslijnen.

Wat opviel was dat de tijdslijn uit allemaal losse informatiepagina's bestond. Bij de andere tijdslijnen die ik had onderzocht had je het gevoel dat je een losse applicatie gebruikte, deze tijdslijn was echter volledig geïntegreerd in de overkoepelende website.

Het feit dat de tijdslijn uit informatiepagina's bestond had als nadeel dat het geheel redelijk statisch overkwam. De tijdslijn was minder op amusement gericht maar voornamelijk op het verschaffen van informatie.



Conclusie

Het analyseren van bestaande tijdslijnen is erg nuttig geweest voor het bepalen van de functionaliteiten van de tijdslijn die ik ontwikkelde. Iedere zojuist besproken tijdslijn heeft wel iets bijgedragen.

De Metropolitan tijdslijn vormde bijvoorbeeld een goede eenheid met de website waarin hij gebruikt werd. De tijdslijn van het Nationaal Archief introduceerde thema's. Flevoland boven water bracht mij op het idee tijdsvakken te creëren met variabele lengtes waarmee drukke periodes op de tijdslijn meer ruimte kregen. De tijdslijn van de Maurits tentoonstelling was voorzien van subtiele geluidseffecten. Toch was de pitch-demo die Fabrique voor het Rijksmuseum gemaakt had nog het meest bruikbaar. Een groot aantal wensen van de opdrachtgever waren hierin al verwerkt, weliswaar niet uitgewerkt, maar er was wel goed over nagedacht.

3.3 Functionele systeemeisen bepalen

De analyse van bestaande tijdslijnen heeft samen met de interne- en externe doelen gediend als basis voor de functionele systeemeisen. Deze functionele eisen beschrijven de uiteindelijke mogelijkheden en eigenschappen van het systeem.

De functionele systeemeisen van de tijdslijn zijn onder te verdelen in twee categorieën: systeemeisen die gesteld zijn aan de interface en systeemeisen die betrekking hebben op de content/inhoud van de tijdslijn.

Interface eisen

De interface moest bestaan uit een aantal elementen, bedoeld voor het besturen en gebruiken van de interface waardoor de inhoud van de tijdslijn op een toegankelijke manier bevraagd kon worden.

Ik bepaalde voor de interface de volgende functionaliteiten:

- Scroll functionaliteit voor het (horizontaal) navigeren door de tijd.
- Zoom functionaliteit voor het instellen van de mate van detail of overzicht.
- Filter functionaliteit waarmee gebruikers de vooraf ingestelde selectie van data konden verfijnen. Meerdere filters moesten tegelijkertijd actief kunnen zijn.
- Zoek functionaliteit op basis van filters en indien mogelijk uitgebreid met de mogelijkheid te zoeken op steekwoorden.
- Variabel begin- en eindpunt van de tijdslijn.
- Variabele lengte van tijdvakken met als doel drukke periodes op de tijdslijn meer ruimte te geven en rustige periodes smaller af te beelden.
- Detailscherm binnen de tijdslijn welke zichtbaar wordt bij klikken op een content-element en informatie toont over dat content-element
- Inline hulpscherm met informatie over het gebruik van de tijdslijn
- Horizontale content-lagen waar de verschillende soorten content-elementen onder elkaar getoond worden. De filter is in staat één of meerdere content-lagen onzichtbaar te maken waardoor extra ruimte ontstaat voor de andere.
- Ondersteuning voor meertaligheid voor buitenlandse gebruikers. De actieve taal is afhankelijk van de taal waarin de overkoepelende website wordt bekeken
- Muis ondersteuning voor bediening van de interface elementen.
- Toetsenbord ondersteuning voor bediening van de interface elementen.
- Geluidseffecten als feedback ter ondersteuning van de navigatie.
- Transitities voor soepele overgangen tijdens het navigeren
- Kleurgebruik ter verduidelijking en verfraaiing van de interface.

Content eisen

De content-elementen tonen de daadwerkelijke inhoud die op de tijdslijn verschijnt. De volgende content-elementen zijn gedefinieerd:

- Kunstobjecten (artObjects) weer te geven als afbeelding/foto van het object. Deze moesten afhankelijk van het gekozen zoomlevel een andere vorm aankunnen nemen (globaal/gedetailleerd)
- Kunstenaars (artists) weer te geven als een periode op de tijdslijn. Periodes zijn op de tijdslijn gevisualiseerd als horizontale balkjes met daarin de naam van de kunstenaar.
- Stijlperiodes (stylePeriods) weer te geven als balk, met daarin de naam van de periode.
- Periodes uit de geschiedenis (historyPeriods) eveneens weer te geven als periode.
- Gebeurtenissen uit de geschiedenis (historyEvents) weer te geven als stip met titel ter verduidelijking.

Alle content-elementen op de tijdslijn hebben een aanmaakdatum of beslaan een periode uit de geschiedenis. Aangezien datums uit het verleden nogal eens onnauwkeurig zijn of niet bekend moesten alle content-elementen in staat zijn grafisch, tekstueel of geanimeerd weer te geven wat de zekerheid is van hun plaats in de tijd.

Verbanden (in tijd) moesten zichtbaar zijn. In het functioneel ontwerp is beschreven op welke manier de elementen op de tijdslijn in horizontale richting zijn verdeeld en hoe er omgegaan wordt met overlappende periodes en gelijktijdige gebeurtenissen.

Net als de interface elementen ondersteunen ook de content-elementen meertaligheid voor buitenlandse gebruikers. De actieve taal is afhankelijk van de taal waarin de overkoepelende website wordt bekeken

Het beheer van de content vindt plaats door het maken van dataselecties uit de database. Aangezien de tijdslijn een objectklasse is binnen de presentatielaag van de website, zal deze als zodanig overal op de nieuwe website inzetbaar en kan per instantie van de tijdslijn een andere configuratie dus ook een andere dataselectie worden gekozen ter ondersteuning van een bepaald thema.

De dataselecties worden gemaakt met XOPUS, het content management systeem waarmee de totale website zal worden onderhouden. Dit valt buiten de grenzen van het afstudeerproject.

3.4 Aanvullende specificaties bepalen

Het bepalen van de aanvullende specificaties voor de tijdslijn was ook een activiteit binnen de fase analyse en definitie. De aanvullende specificaties, ook wel niet-functionele eisen genoemd worden moesten meetbaar en helder beschreven worden. Ten eerste om geen verwarring te laten ontstaan tijdens het ontwikkelen en ten tweede omdat de specificaties tijdens het testen worden gebruikt als meetcriteria.

In bijlage I (analyse en definitie) zijn de aanvullende specificaties opgenomen waarin onder meer op de volgende vragen antwoord wordt gegeven.

- Met hoeveel data moet de tijdslijn om kunnen gaan?
- Mag de tijdslijn een browser plug-in vereisen?
- Vanaf welke laadtijden iets als een progressiebalk zichtbaar moet zijn?
- etc.

3.5 Functionaliteiten en omgeving analyseren

Na de systeemeisen vastgesteld te hebben, heb ik de functionaliteiten en de omgeving van het systeem schematisch in kaart gebracht met behulp van UML. (Unified Modeling Language) UML is een notatiestandaard die gebruikt wordt in objectgeoriënteerde omgevingen. Het definieert de basis concepten van object georiënteerd analyseren en ontwerpen, en bevat een aantal diagrammen om te kunnen communiceren tussen deze concepten.

Use-case modellering

Het beschrijven van use-cases is een onderdeel van UML. Use-cases beschrijven hoe een systeem wordt gebruikt om de doelstellingen te bereiken. Er wordt in tekst een collectie van gerelateerde succesvolle en mislukkende scenario's uitgedrukt. Deze scenario's omvatten de belangrijkste taken die een persoon of systeem met de tijdslijn zou moeten kunnen uitvoeren. De collectie van alle use-cases samen vormen een use-case-model. Het use-case-model is een schematische weergave van alle use-case beschrijvingen samen. Het beschrijven en modelleren van use-cases kan afhankelijk van de behoefte gedetailleerd of heel globaal. Ik heb me beperkt tot use-cases die een tamelijk beknopte representatie van het systeem zijn, omdat ik naar mijn mening de functionaliteiten en omgeving van de tijdslijn al behoorlijk goed in kaart had gebracht tijdens het vaststellen van de functionele eisen.

Ik heb de volgende use-cases onderscheiden:

1. scrollen door de tijd
2. zoomen naar een ander detailniveau
3. filteren op bepaalde criteria
4. opvragen van details over een onderwerp
5. opvragen van de online help
6. definiëren van een thema/selectie
7. configureren van de tijdslijn voor een bepaald thema
8. tonen van de juiste content-elementen op de tijdslijn

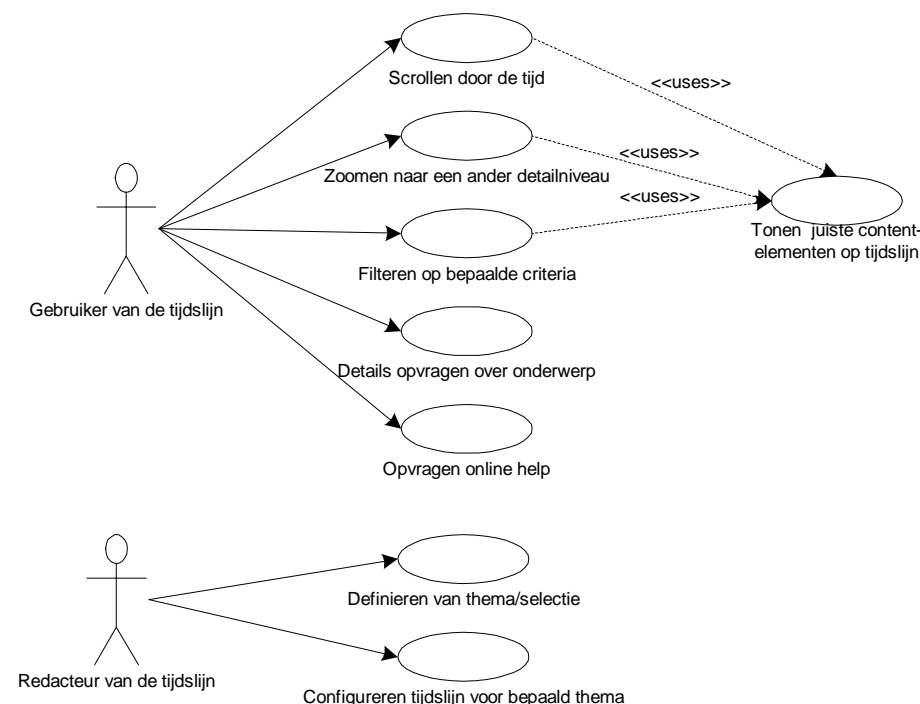
Van deze use-cases heb ik tekstuele beschrijvingen gemaakt waarin per use case het volgende wordt vermeld: naam, samenvatting, actoren, aannames, beschrijving, uitzonderen en het resultaat.

De complete lijst met use-case beschrijvingen is opgenomen in bijlage I (analyse en definitie – use-case beschrijvingen) Eén daarvan is hieronder weergegeven:

Naam	Scrollen door de tijd
Samenvatting	Er wordt een grote of kleine sprong door de tijd gemaakt.
Actoren	Gebruiker van de tijdslijn
Aannames	<ul style="list-style-type: none">- De tijdslijn is goed geconfigureerd- De tijdslijn is geladen- De tijdvakken (ElasticRange) zijn bekend (begin, eind, lengtes en intervallen van/ tussen periodes op de tijdslijn)- Het systeem bevat inhoud- De internetverbinding van de gebruiker voldoet aan de minimale specificaties zoals is beschreven in de systeemeisen en is tijdens het gebruik van het systeem actief.- De computer van de gebruiker voldoet aan de minimale specificaties zoals is beschreven in de systeemeisen.

Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker maakt met behulp van toetsenbord of muis een kleine of grote sprong in de tijd. 2. Het systeem controleert of er niet eerder dan [begin van de tijdslijn] of verder dan het [eind van de tijdslijn] wordt gegaan. 3. Er worden nieuwe gegevens getoond. <<uses: tonen van de juiste content-elementen op de tijdslijn>> 4. Afhankelijk van de grootte van de sprong worden extra gegevens/afbeeldingen bij geladen. Een kleine sprong zorgt voor buffering van het gebied dat in beeld dreigt te komen. Een grote sprong zal gegevens/afbeeldingen laden van het nieuwe zichtbare gebied. 5. Het systeem controleert of [te snel scrollen in kleine stapjes] plaatsvindt.
Uitzonderingen	<ol style="list-style-type: none"> a. [begin van de tijdslijn] de gebruiker probeert terug te gaan in de tijd, maar staat al aan het begin van de tijdslijn. b. [eind van de tijdslijn] de gebruiker probeert verder te gaan in de tijd, maar staat al aan het eind van de tijdslijn. c. [te snel scrollen in kleine stapjes] moet leiden tot het niet laden van gegevens/afbeeldingen die toch weer erg snel uit beeld verdwijnen, ivm performance
Resultaat	Er is een sprong gemaakt in de tijd. De nieuwe gegevens zijn zichtbaar gemaakt of 'druppelen' nog langzaam binnen.

Hoewel bij use-case de tekstuele beschrijvingen voorop staan heb ik voor het overzicht nog een use-case-model gemaakt die hieronder is weergegeven.



Figuur 4: Use-case model, alle use-case beschrijvingen in één model

In het use-case model zijn twee actoren opgenomen. De gebruiker van de tijdslijn is de bezoeker van de website die de tijdslijn gebruikt om informatie op te vragen. De rol redacteur wordt intern uitgevoerd bij het Rijksmuseum.

De taken van de redacteur worden niet rechtstreeks met de tijdslijn uitgevoerd maar met XOPUS, het content management systeem van Q42 welke buiten de grenzen van de afstudeeropdracht valt. Dat de twee use-cases toch zijn opgenomen is omdat er bij het ontwikkelen van de tijdslijn wel degelijk rekening moest worden gehouden met de wijze waarop de twee systemen op elkaar aangesloten konden worden.

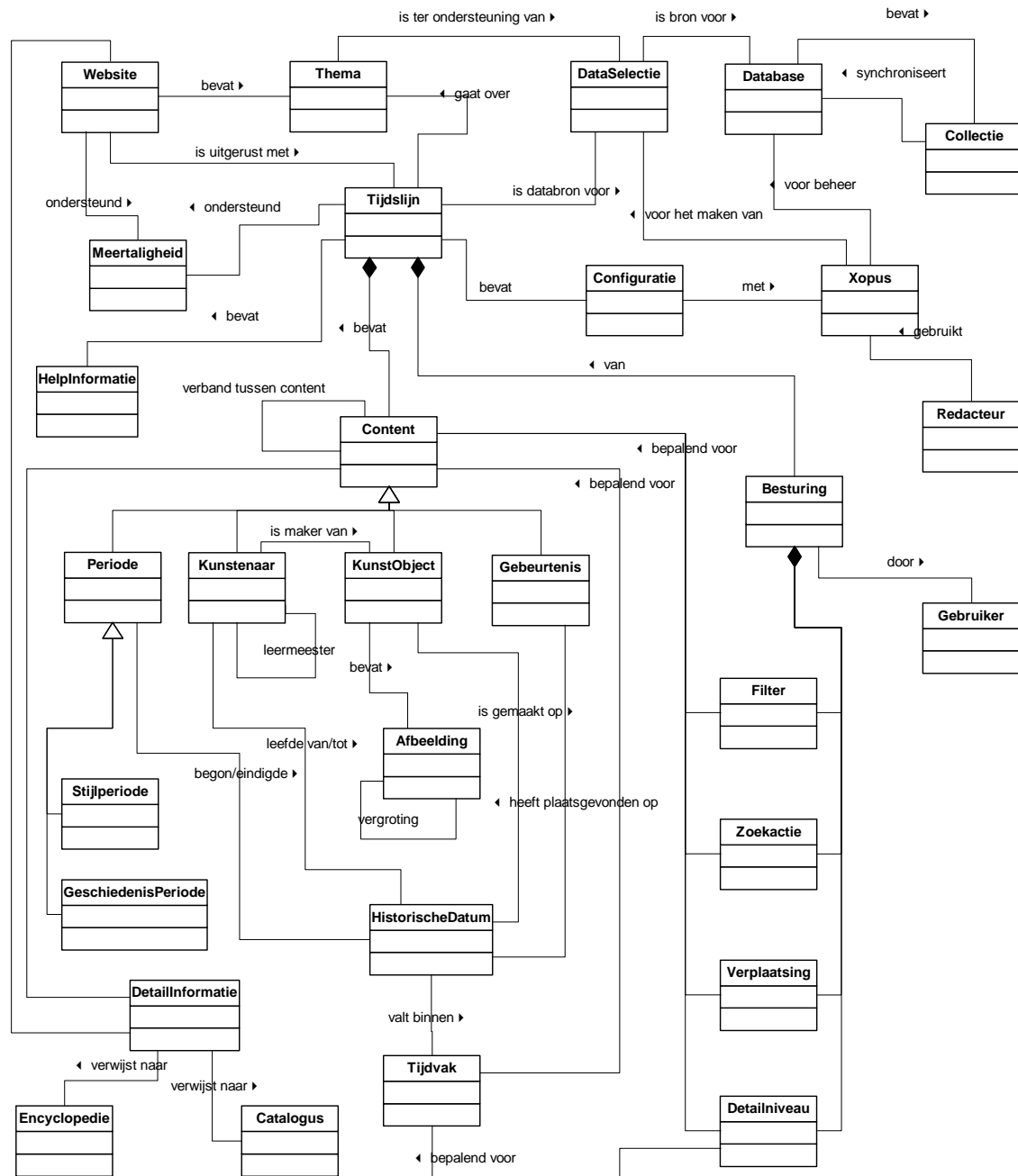
Domein modellering

Object georiënteerde ontwikkeling gaat uit van domeinmodellering. Hierbij wordt de omgeving, dus alles wat met de tijdslijn te maken krijgen, in kaart gebracht. Dit levert een domeinmodel op. Hierin staan de objecten uit het domein met hun kenmerken, gedrag en onderlinge relaties. (Zie Figuur 5 op de volgende pagina)

Het domeinmodel is het belangrijkste resultaat van de analysefase. Dit model zal uiteindelijk leiden tot het tot stand komen van het klassendiagram wat zal worden beschreven in het functioneel ontwerp in hoofdstuk 4.

Om de conceptuele klassen voor de tijdslijn te vinden heb ik gezocht naar zelfstandige naamwoorden binnen de geformuleerde systeemeisen en de use-case beschrijvingen. In bijlage I (analyse en definitie - domeinmodel) zijn de gevonden klassen opgenomen, alsmede een lijst met de klassen die op dat moment nog overbodig of dubbelop leken te zijn.

In het onderstaande domeinmodel zijn de conceptuele klassen weergegeven van de tijdslijn met hun associaties. Attributen van de klassen en multipliciteit van de associaties zijn in dit model nog buitenbeschouwing gelaten.



Figuur 5: Domeinmodel

3.6 Technische analyse uitvoeren

De technische analyse had als doel de meest geschikte techniek te kiezen voor de programmering van de tijdslijn en me op deze techniek te oriënteren.

Technische analyse van bestaande tijdslijnen

Een aantal tijdslijnen waarvan ik de functionaliteiten in kaart had gebracht, heb ik ook aan een technisch onderzoek onderhevig gemaakt om te bepalen van welke technieken gebruik gemaakt werd en op welke wijze de data werd bewaard. De meeste tijdslijnen waren ontwikkeld in Macromedia Flash, maar ik kwam er ook een aantal tegen die geprogrammeerd waren in Java, Macromedia Shockwave en zelfs vond ik er één, die geheel gebaseerd was op DHMTL. Het bleek dat een groot aantal tijdslijnen geen gebruik maakte van een database, maar de gegevens statisch bij zich droeg. Ook waren er een aantal die wel optimaal van de technieken gebruik maakten en die daardoor beter onderhoudbaar leken te zijn.

Het technisch analyseren van de tijdslijnen die in Macromedia Flash waren gemaakt was gemakkelijk. Door het bestand te openen met een decompiler kon de broncode opgevraagd worden. Mijn ervaring met Macromedia Flash was tijdens deze analyse nog gering, maar mijn programmeerkennis goed genoeg om te kunnen constateren dat de meeste tijdslijnen volgens het *quick 'n dirty* principe waren gebouwd. De broncode was onoverzichtelijk, slecht gedocumenteerd en uiteindelijk heeft geen van de geanalyseerde tijdslijnen als bruikbaar voorbeeld kunnen dienen. Daarentegen nam mijn inspiratie en motivatie toe, de gedachte werd steeds sterker: dat kan ik beter!

Techniekselectie

Hoewel uit de technische analyse van bestaande tijdslijnen bleek dat Flash verruit de meest gebruikte techniek was om een dergelijk project mee uit te voeren, vond ik het toch zeer belangrijk om op basis van goede argumenten voor een bepaalde techniek te kiezen. Daarom heb ik tijdens de fase analyse en definitie een techniekselectie-matrix opgesteld waarin ik de meest voor de handliggende ontwikkelomgevingen tegen elkaar uitzette en de voor- en nadelen ten opzichte van elkaar in kaart bracht. De techniekselectie-matrix is opgenomen in bijlage I (analyse en definitie – techniek selectie matrix).



De volgende technieken zijn met elkaar vergeleken: DHTML, Java (Applets), Macromedia Shockwave, Flash 5, Flash 6 en Flash 7. Al snel leek Flash 7 het meest in aanmerking te komen. In Flash 7 kon object georiënteerd geprogrammeerd worden, dat kon ook in Java, maar bij Java wordt de browser plug-in tegenwoordig minder gebruikt. De Flash plug-ins worden tegenwoordig volledig geaccepteerd door de gemiddelde website bezoeker en ook de gemakkelijke overstap naar een nieuwe versie van de Flash plug-in is positief. Macromedia Shockwave kwam in eerste instantie ook aanmerking wegens de goede rekenkracht bij het renderen van afbeeldingen. Maar net als bij Java was ook hier weer de plug-in de belemmerende factor. Deze is vele malen groter dan de Flash plugin.

Het verschil tussen de verschillende Flash versies zit vooral in het object georiënteerd kunnen programmeren, performance, het scheiden van de vormgeving en de code en het al dan niet ondersteunen van externe afbeeldingen.

De Flash 5 plug-in is zeer goed geaccepteerd bij de gemiddelde website bezoeker, maar viel al snel af wegens de slechte performance bij grote hoeveelheden data, en het niet ondersteunen van externe afbeeldingen.

De Flash 6 plug-in heeft al veel voordelen ten opzichte van de vorige versie maar wordt door de snelle opvolging door Flash 7 vaak gezien als voorproefje op de echte 'volwassen' Flash 7.

Flash 7, ook wel Flash MX 2004 genoemd, is uitgerust met ActionScript 2.0, een object georiënteerde programmeertaal die gebaseerd is op de ECMA-262 v4 standaard.

Naast mijn eigen afwegingen heb ik ook nog een aantal collega's geraadpleegd om helemaal zeker te weten dat ik de goede keus maakte. Toen ook zij aangaven voldoende vertrouwen te hebben in de nieuwste versie van Macromedia Flash besloot ik uiteindelijk. De tijdslijn zou gebouwd gaan worden in Flash MX 2004.

Oriëntatie op geselecteerde techniek

Ik had minimale ervaring met Flash en heb me tijdens deze fase ook erg moeten inlezen. Direct heb ik een aantal goede boeken aangeschaft over het Flash MX 2004 framework en Actionscript 2.0 en al lezende werd ik steeds enthousiaster. Ook op internet was al veel informatie te vinden over deze nieuwe Flash versie.

Tijdens het oriënteren op de geselecteerde techniek kwam ik er achter dat ik aan de voorbeelden uit de boeken niet genoeg had. Ik besloot als oefening al een aantal onderdelen van het te ontwikkelen prototype te bouwen. Hierbij dwong ik mezelf alvast na te denken welke classes ik zou gaan gebruiken.

Het bleek dat Flash MX 2004 goed om kon gaan met grote hoeveelheden XML data. Ik had vernomen dat bij de vorige Flash versies één van de bottlenecks was, bij deze versie was dat probleem opgelost.

Een groot probleem van vorige Flash versies was dat de programmacode nogal eens op onvindbare plaatsen door de applicatie verspreid was. Bij Flash MX 2004 wordt je bijna gedwongen om gestructureerd te programmeren.

3.7 Gegevens analyse uitvoeren

Tijdens de fase analyse en definitie zijn ook de te raadplegen databronnen geanalyseerd. De resultaten van die analyse worden in deze paragraaf beschreven.

Oude situatie: ADLIB en Aria

Er werd bij het Rijksmuseum gewerkt met een tweetal databases. De belangrijkste database voor het Rijksmuseum was de ADLIB Museum database. ADLIB Museum is een commerciële applicatie die gemaakt is voor collectie- en informatiebeheer in musea.



De tweede database waarmee gewerkt werd was de Aria database. De Aria database was een restant van een eerder informatiesysteem die onder andere gebruikt werd voor informatiezuilen binnen het museum en voor de oude website. Deze database bevatte gegevens die gebruikt werden ter verrijking van de ADLIB database. Een groot aantal informatiepagina's over kunstobjecten en kunstenaars kwam uit dit systeem.

De informatie uit de Aria database is geconverteerd en is samen met de gegevens uit de ADLIB database gemigreerd in een nieuwe XML Database.

Het Aria-systeem is hierdoor na de implementatie van de nieuwe website overbodig en wordt als zodanig opgeheven.

De ADLIB database blijft bestaan en wordt op gezette tijden gesynchroniseerd met de XML Database.

Nieuwe situatie: XML database

Hoewel het ontwerpen van de database buiten de grenzen van mijn afstudeeropdracht viel heb ik hiervan toch het een en ander meegekregen omdat ik voor de ontwikkeling van de tijdslijn op de hoogte moest zijn over de structuur en de inhoud van de database en de wijze waarop ik de database kon bevragen.

Q42 heeft voor de XML Database van het Rijksmuseum gebruik gemaakt van Oracle versie 9. De XML Database is opgedeeld in een aantal spaces. Spaces zijn te vergelijken met tabellen uit een relationele database. Een aantal spaces in de nieuwe database zijn artists, artobjects, screens, catalogs, dictionary en encyclopedia. Een space bestaat uit een collectie van XML documenten met per document een eigen unieke ID. Alle ID's binnen een space moeten uniek zijn, daarbuiten mogen dubbelingen voorkomen.

De XML documenten binnen een collectie zouden als records beschouwd kunnen worden met een ID als primaire sleutel. Een groot verschil is echter tussen een tabel en space dat een tabel een vaste definitie heeft met een vooraf bepaald aantal datavelden terwijl binnen de spaces vrij genest kan worden. Ook kunnen er referenties opgenomen worden naar documenten in andere spaces.

Het grote voordeel van het kunnen nesten is dat de database niet zal gaan bestaan uit talloze koppeltabellen zoals dat het geval zou zijn bij een relationele database. Een ander groot voordeel is dat er binnen spaces referenties naar elkaar opgenomen kunnen worden waardoor het bevragen van een XML database minder complexe queries vereist.

Een XML database kent net als een relationele database indexen waardoor snel in de database gezocht kan worden. Bij een relationele database worden indexen aangemaakt op tabelvelden. Bij een XML database worden de indexen aangemaakt op elementen en hun attributen.

Benodigde gegevens voor de tijdslijn

Voor het voeden van de tijdslijn met data wordt de XML database geraadpleegd.

Aan de hand van de bepaalde content-elementen, genoemd in de functionele eisen en die ook terugkwamen in domeinmodel, bepaalde ik de spaces waarvan de tijdslijn gebruikt ging maken.

De content van de tijdslijn werd verkregen uit de volgende spaces:

- artists (kunstenaars)
- artObjects (kunst objecten)
- stylePeriods (stijl periodes zoals bijv. Barok)
- historyPeriods (periodes uit de geschiedenis zoals bijv. WO II)
- historyEvents (gebeurtenissen uit de geschiedenis zoals bijv. het in gebruik nemen van het Rijksmuseum).

De spaces stylePeriods, historyPeriods en historyEvents bestonden nog niet en moesten aangemaakt worden. Het vullen van deze spaces met reële data hoorde thuis in de fase redactie.

ArtObjects en Artists konden geïmporteerd worden uit de ADLIB database. Over een aantal kunstenaars werden echter ook gegevens bijgehouden in de Aria database. Deze gegevens werden samengevoegd met die uit de ADLIB database.

Binnen de content-spaces waren de volgende elementen relevant voor de tijdslijn:

Space	Element	Datatype	Opmerking
artists	<u>ID</u>	String	uniek binnen space
	name	String	naam van kunstenaar
	birth	SpecialDate	geboortedatum van kunstenaar
	death	SpecialDate	overlijdensdatum van kunstenaar
	screenReference	String	referentie naar gegevens voor detailscherm
artObjects	<u>ID</u>	String	uniek binnen space
	name	String	naam van kunstwerk
	creationDate	SpecialDate	creëerdatum van kunstwerk
	collection	String	in welke collectie kunstwerk thuis hoort. Hierop moet gefilterd kunnen worden
	hasImage	Boolean	afbeelding beschikbaar ja/nee, de eventuele naam van de afbeelding wordt afgeleid van ID
	artistReference	String	referentie naar de maker
	screenReference	String	referentie naar gegevens voor detailscherm
stylePeriods	<u>ID</u>	String	uniek binnen space
	name	String	naam van de periode
	startDate	SpecialDate	begin van de periode
	endDate	SpecialDate	eind van de periode

Space	Element	Datatype	Opmerking
	screenReference	String	referentie naar gegevens voor detailscherm
historyPeriods	<u>ID</u>	String	uniek binnen space
	name	SpecialDate	naam van de periode
	startDate	SpecialDate	begin van de periode
	endDate	SpecialDate	eind van de periode
	screenReference	String	referentie naar gegevens voor detailscherm
historyEvents	<u>ID</u>	String	uniek binnen space
	name	SpecialDate	naam van de gebeurtenis
	date	SpecialDate	datum van de gebeurtenis
	screenReference	String	referentie naar gegevens voor detailscherm

Het datatype SpecialDate wordt in de volgende paragraaf toegelicht.

De space *screen* was nodig voor het verkrijgen van de data voor het opbouwen van de detailschermen. De uiteindelijke structuur van deze space was in deze fase nog niet bekend.

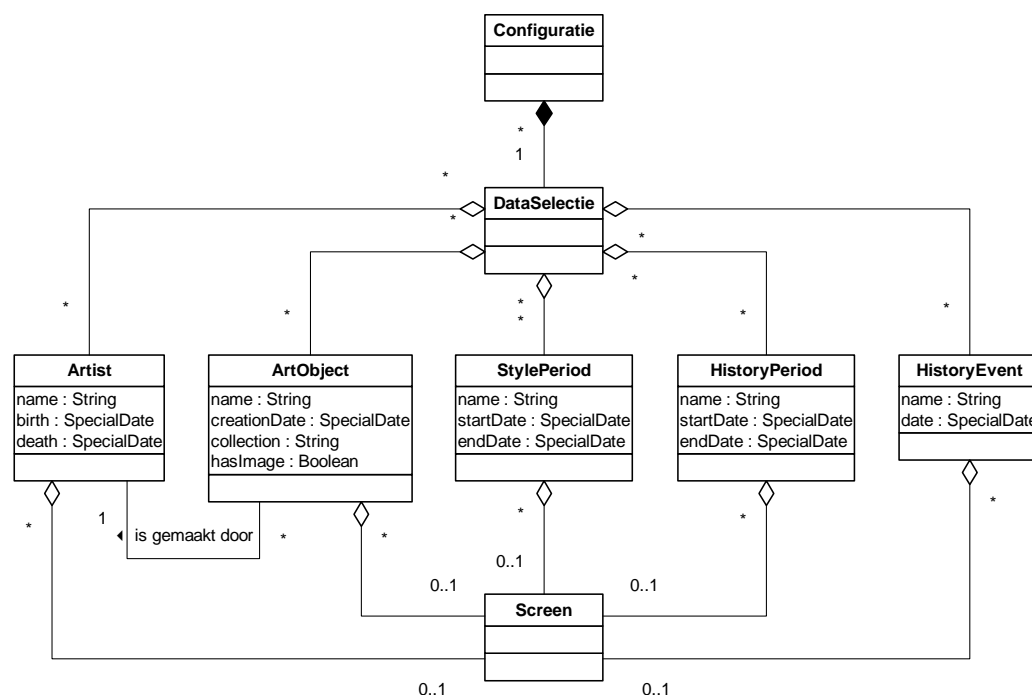
Tijdens de fase analyse en definitie werd al duidelijk dat er met grote hoeveelheden data gewerkt zou gaan worden. Er werd gesproken over 100.000 kunsthistorische entries in de database. Dit aantal is later bijgesteld naar 400.000.

Het is duidelijk dat een client-sided web-toepassing zoals de tijdslijn nooit in staat zal zijn al deze gegevens te verwerken, daarom heb ik samen met Q42 bepaald dat er van server-sided dataselecties gebruikt gemaakt wordt. Deze dataselectie is te zien in het reeds weergegeven domeinmodel en bevat database entries die betrekking hebben op een bepaald thema.

De dataselectie wordt bewaard in een aparte space zodat deze gecached kon worden.

Hierdoor zal de tijdslijn niet bij iedere request een complexe dataselectie query opleveren.

Het onderstaande schema toont de benodigde spaces voor de tijdslijn. Referenties naar elkaar zijn hier in opgenomen.



Figuur 6: benodigde spaces

Aangezien de gegevens niet in een relationele database worden bewaard maar in een XML database en de weergave van datastructuren hiertussen verschilt is hieronder ter illustratie een instantie van dataklasse ArtObject weergegeven:

```

<artObject id="AK-MAK-1117" hasImage="true" title="Tsuba"
.....artistID="AT-155" refID="SC-122">
  <creationDate value="1600" certainty="circa"/>
</artObject>

```

Standaarden voor datum notaties

Datums uit het verleden zijn vaak erg onnauwkeurig of onvolledig. De tijdslijn moest in staat zijn hier op een goede manier mee om te gaan. In de functionele eisen heb ik reeds beschreven dat alle content-elementen grafisch, tekstueel of geanimeerd moesten kunnen aangeven wat de zekerheid is van hun plaats in de tijd.

Zoals ook het datamodel te zien is, worden alle historische datums die door de tijdslijn gebruikt worden, opgeslagen als datatype SpecialDate. SpecialDate is gebaseerd op de ISO 8601 standaard. Deze standaard beschrijft een internationale datumnotatie volgens het volgende formaat:

```
YYYY-MM-DDThh:mm:ss.sTZD (bijv. 2004-07-16T19:20:30.45+01:00)
```

Aangezien de meeste datums waarmee gewerkt is onnauwkeurig zijn, wordt in de meeste gevallen alleen het jaar, en heel soms nog maand en dag gebruikt. De mogelijkheid tot het aanduiden van uren en minuten wordt niet gebruikt.

Om een zekerheidsfactor van een historische datum vast te leggen en eventueel een kalendereenheid toe te voegen heb ik gebruik gemaakt van de DublinCore standaard voor datumnotaties. Deze standaard beschrijft in XML formaat een aan te bevelen notatiewijze:

```
<date value="1551" certainty="circa" calender="AD" />
```

De zekerheidsfactor van een datum wordt vermeld in het *certainty* attribuut. Een aantal mogelijke waarden voor certainty zijn: exact, ca (circa), approx, before, after, early, late en mid.

De tijdslijn zal in eerste instantie alleen ingezet voor content-elementen met een datumzekerheid van exact of circa.

De DublinCore standaard voorziet ook in vastleggen van datums uit andere kalenders. Hoewel de tijdslijn in eerste instantie alleen hoeft om te gaan met data uit onze Gregoriaanse kalender (AD Anno Domini of in het Engels CE Common Era) zijn toch voorbereidingen getroffen om ook museale objecten uit andere kalenders te kunnen opnemen. Dit kan zeker nodig zijn wanneer het Rijksmuseum haar collectie uitbreidt met kunstobjecten van voor het jaar nul (BC - Before the Common Era / Before Christ).

3.8 Resultaat van de fase analyse en definitie vaststellen

De fase analyse en definitie heeft veel opgeleverd. Naast het in kaart brengen van alle systeemeisen en mogelijke functionaliteiten was na afloop van deze fase ook een keuze gemaakt voor de te gebruiken programmeeromgeving en was de te gebruiken data geanalyseerd.

Na afloop van de fase ging ik met een grote hoeveelheid bruikbare input de fase functioneel ontwerp in.

4. FUNCTIONEEL ONTWERP

In de fase functioneel ontwerp heb ik alle functionaliteiten van het te ontwikkelen systeem bepaald en vastgelegd. Het resultaat van deze fase is een document genaamd 'functioneel ontwerp' waarnaar in dit hoofdstuk regelmatig verwezen wordt. Het functioneel ontwerp is opgenomen als bijlage III.

4.1 Plaats van de tijdslijn in website vaststellen

Tijdens de fase analyse en definitie was mij al duidelijk geworden dat de tijdslijn op de website ter verrijking van thema's ingezet zou worden. De wijze waarop was echter nog niet bepaald. Het was nog niet zeker of de tijdslijn op de homepage kwam, of als pop-up, of ingebed (inline) op verschillende pagina's.

Ik had verwacht dat er in deze fase duidelijkheid zou komen over de plaats van de tijdslijn op de nieuwe website. Helaas was daar nog niet met zekerheid iets over te zeggen. Zolang het ontwerp van de overkoepelende website nog niet definitief was kon het nog alle kanten op gaan.

Hierom besloot ik dat de tijdslijn zo flexibel mogelijk opgebouwd moest worden zodat deze op iedere plaats op de website ingezet kon worden. Ik bepaalde dat het formaat (de afmetingen) van de tijdslijn configureerbaar moesten zijn en dat de vormgeving later toegepast moest kunnen worden.

4.2 Klassendiagram opstellen

In het domeinmodel uit de fase analyse en definitie is de omgeving, dus alles wat met de tijdslijn te maken krijgen, in kaart gebracht. Op basis van het domeinmodel heb ik in de fase functioneel ontwerp het klassendiagram opgesteld. In dit klassendiagram worden de benodigde softwareklassen beschreven, met hun associaties, attributen en methoden. Bepaalde conceptuele klassen uit het domeinmodel zijn hierin weggelaten, terwijl er een aantal computergerichte klassen zijn toegevoegd die bijvoorbeeld betrekking hebben op de user-interface of bedoeld zijn voor benadering van de XML database.

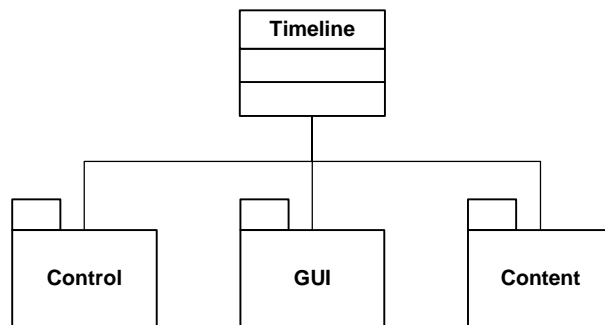
Hoewel ik de basisprincipes van object georiënteerd programmeren kende, en ook ervaring had met het modelleren van een (klein) systeem volgens UML, kostte het opstellen van een goed klassendiagram veel tijd en was het moeilijker dan ik had verwacht.

Dit had ik in eerste instantie nog niet door. Het klassendiagram dat uit het functioneel ontwerp voortvloeide vond ik al redelijk goed en heeft ook als basis gediend voor de programmering van de tijdslijn.

Regelmatig bekeek ik het klassendiagram grondig en constateerde ik dat het eigenlijk veel beter kon. Telkens gebruikte ik de nieuwe kennis die ik opdeed tijdens de volgende fasen om het klassendiagram te verbeteren en telkens had ik het gevoel dat het ook verbeterde. Maar het verbeteren bleef maar doorgaan en uiteindelijk ben ik nog steeds niet tevreden over het huidige klassendiagram. Ook tijdens het schrijven van dit afstudeerrapport las ik nog veel op internet en in boeken over UML en over de afspraken die er zijn om een systeem op heldere wijze op te bouwen. Maar omdat je niet door kunt blijven gaan met optimaliseren en het uiteindelijke diagram voldoende geschikt was om als basis te dienen voor de tijdslijn heb ik dit geaccepteerd.

Omdat dat het klassendiagram dat in de fase functioneel ontwerp tot stand is gekomen een minder goed beeld geeft van de functionaliteiten van de tijdslijn leek het me belangrijk om in dit hoofdstuk al de laatste versie van het klassendiagram op te nemen. Een van de voorlopers van het klassendiagram is opgenomen in bijlage III.

Ondanks de grootte van het klassendiagram, heb ik bewust gekozen het toch op te nemen in dit rapport. Ik ben namelijk van mening dat een goed klassendiagram een inzichtelijk beeld geeft van het systeem en ook voor de minder technische lezer verduidelijking kan bieden. Ik zie het als waardevolle toevoeging bij het beschrijven van de tijdslijn en de totstandkoming daarvan.

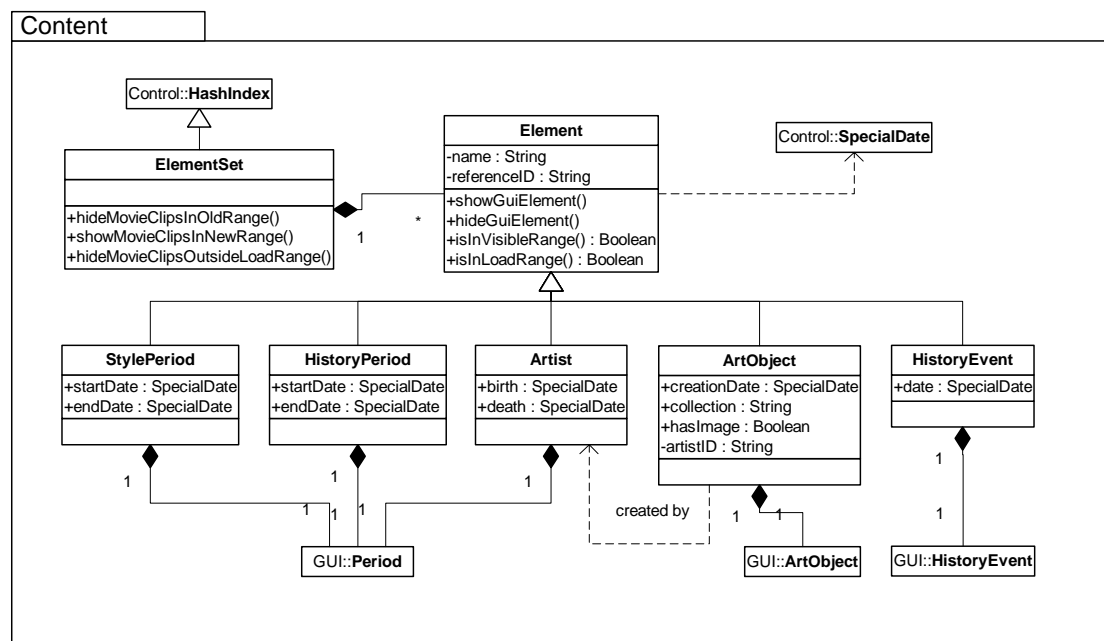


Figuur 7: Packages

Ik heb de tijdslijn opgedeeld in een drietal packages. De packages groeperen een aantal bij elkaar horende classes (klassen). Het opdelen in packages maakt de tijdslijn (en ook het model daarvan) stukken overzichtelijker.

Het klassendiagram uit de fase functioneel ontwerp bevatte nog geen packages en was daardoor minder overzichtelijk.

Ik heb de beschrijvingen bij de packages redelijk beknopt gehouden. De belangrijkste classes zullen in de volgende paragraaf worden toegelicht.



Figuur 8: classes binnen het content-package

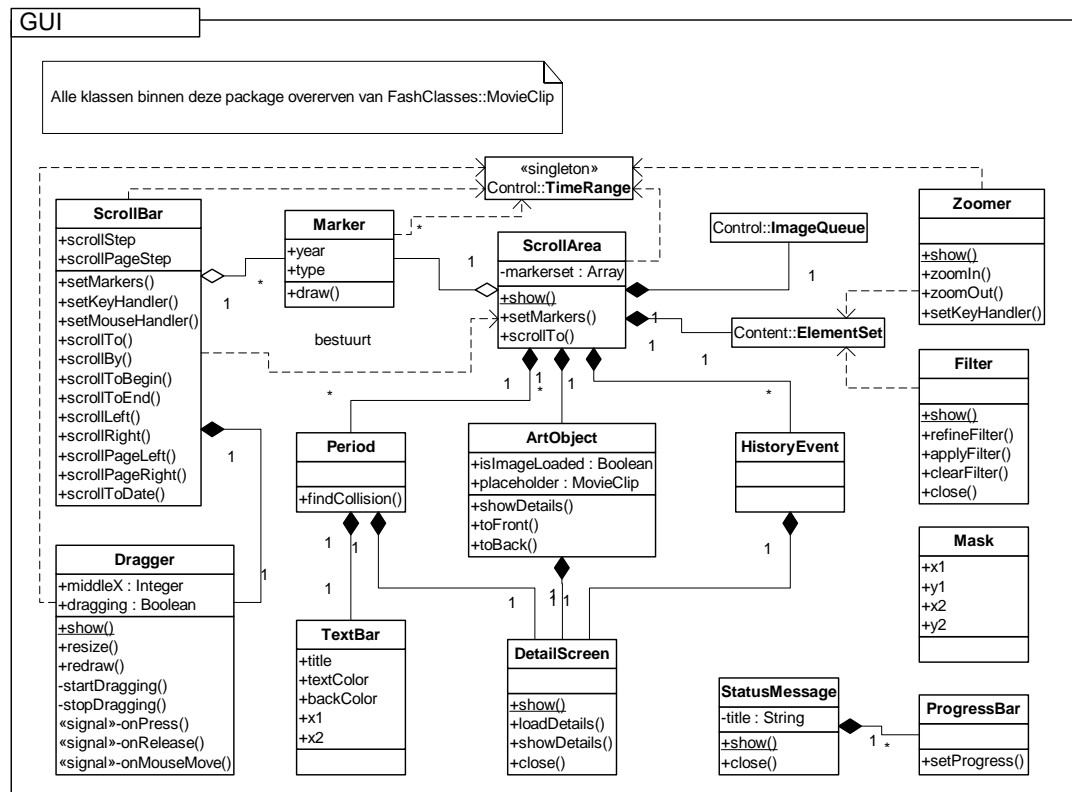
De bovenstaande afbeelding toont het *content* package. Dit package bevat de content, oftewel de inhoud of data van de tijdslijn.

De klasse *elementset* is gekoppeld aan *element* door een zwarte ruit. Dit betekent dat de elementset alle dataklassen *bevat*. Het driehoekje dat *element* koppelt aan de klassen eronder

staat voor overerving. *Artist*, *ArtObject* etc. overerven allemaal het gedrag en de eigenschappen van *element*.

In de fase analyse en definitie had ik al vastgesteld welke dataklassen de tijdslijn zou bevatten. In de fase functioneel ontwerp heb ik het gedrag van de dataklassen vastgelegd en hun plaats in het geheel. Zoals te zien is bevatten alle dataklassen een GUI-element (GUI = graphical user interface).

De onderstaande afbeelding toont de inhoud van het package *GUI*. Deze package bevat alle interface elementen waarmee de tijdslijn bestuurd kan worden.



Figuur 9: Classes binnen het GUI-package

Alle GUI elementen zijn overervingen van de class MovieClip die standaard in Flash aanwezig is.

De ScrollBar is een besturingselement die door de gebruiker van de tijdslijn bediend worden om te verplaatsen door de tijd.

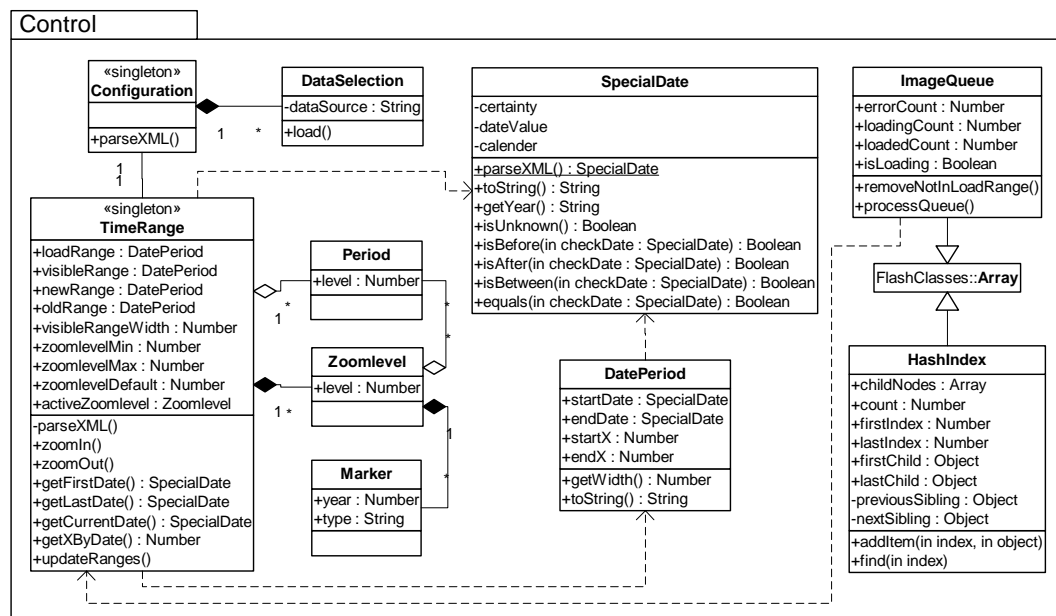
De scrollArea wordt bestuurd door de ScrollBar en bevat de elementen voor het tonen van de content: ArtObject, HistoryEvent en Period. Naar deze drie classes wordt verwezen door de dataklassen in de content-package.

De class `DetailScreen` is het venster dat verschijnt wanneer een gebruiker informatie over een bepaald content-element opvraagt.

De Zoomer is het besturingselement waarmee de gebruiker kan overschakelen naar een ander detailniveau.

Met het besturingselement Filter kan de gebruiker een eigen selectie maken van de aangeboden content. Content die voor hem of haar niet van belang is kan worden weggelaten.

Zoals in het schema te zien is, wordt er door een groot aantal classes verwezen naar ElasticRange, een class die zich in de control-package bevindt welke op de volgende pagina is afgebeeld. In het diagram wordt nog de naam TimeRange gebruikt, deze naam is gewijzigd in ElasticRange.



Figuur 10: Classes binnen het control-package

De classes binnen de control-package worden veelvuldig gebruikt door de classes in de andere packages.

De genoemde ElasticRange dient als laag tussen de jaartallen en x-coördinaten op de tijdslijn. De HashIndex wordt gebruikt door ElementSet om content geïndexeerd op jaartal te kunnen bewaren.

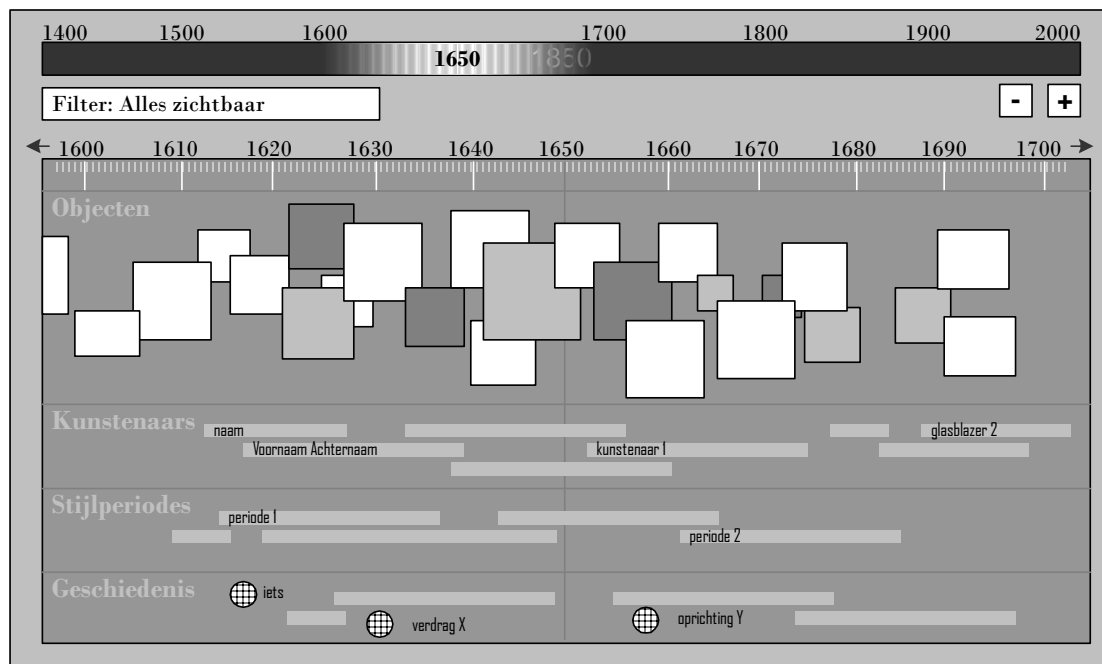
De ImageQueue handelt het laden van afbeeldingen af. Deze class wordt gebruikt om op een efficiënte manier de grote hoeveelheid afbeeldingen voor de ArtObjects te laden.

SpecialDate is de class die ingezet wordt voor het omgaan met datums volgens het formaat dat ik bepaald heb in de gegevensanalyse in paragraaf 3.7.

4.3 Functionaliteiten van de interface vastleggen

Een activiteit binnen de fase functioneel ontwerp was het bepalen van de functionaliteiten van de interface. In deze paragraaf beschrijf ik de totstandkoming daarvan.

Figuur 11 toont de globale indeling van de interface. Deze indeling toont de functionele lay-out en toont veel overeenkomsten met het pitch-ontwerp van Fabrique (zie paragraaf 3.2). De verschillende interface elementen zullen een voor een besproken worden.



Figuur 11: functionele lay-out van de tijdslijn

Scrollen door de tijd

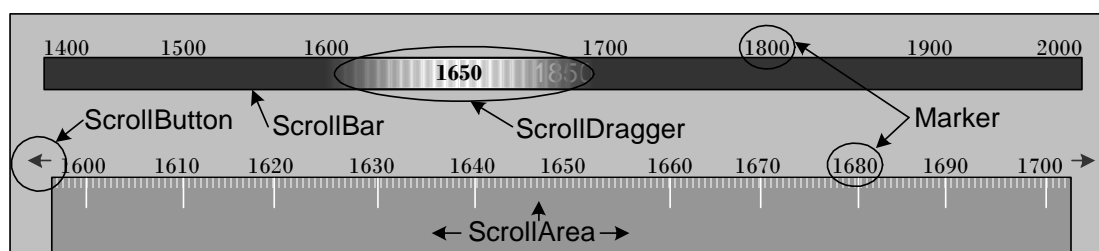
Zoals in het klassendiagram te zien was heb ik de tijdslijn voorzien van een ScrollBar (schuifbalk) en een ScrollArea. Deze twee GUI elementen maken het mogelijk dat er naar links en naar rechts door de tijd 'gescrolled' kan worden.

In de fase functioneel ontwerp heb ik het gedrag bepaald van deze twee elementen.

In de afbeelding is te zien dat de ScrollBar is voorzien van een Dragger (meesleper). De dragger kan met de muis of het toetsenbord naar links en naar rechts worden verplaatst. De ScrollArea volgt de bewegingen van de ScrollBar.

Ik heb er voor gekozen de ScrollBar voor de tijdslijn zoveel mogelijk het gedrag te laten imiteren van een ScrollBar zoals gebruikers die gewent zijn van een besturingssysteem zoals bijvoorbeeld Windows. Ik heb hiervoor gekozen zodat gebruikers de tijdslijn zo intuïtief mogelijk kunnen besturen.

De volgende afbeelding toont alle besturingselementen die gebruikt kunnen worden bij het scrollen door de tijd.



Figuur 12: besturingselementen voor het scrollen door de tijd

De ScrollBar kan op de volgende manieren bestuurd worden:

commando	acties ScrollArea en ScrollDragger
slepen van de dragger met linker muisknop	ScrollArea volgt de bewegingen die de gebruiker met de ScrollDragger maakt.
neerwaarts scrollen met het scrollwiel van de muis	naar rechts. Stap is afhankelijk van de afstand die het scrollwiel heeft afgelegd.
opwaarts scrollen met het scrollwiel van de muis	naar links. Stap is afhankelijk van de afstand die het scrollwiel heeft afgelegd.
muisklik ScrollButtonLeft	klein stapje naar links
muisklik ScrollButtonRight	klein stapje naar rechts
muisklik op ScrollBar	(grote) sprong naar de nieuwe plaats
muisklik op Marker	sprong naar dat jaartal
toets Links (←)	klein stapje naar links
toets Rechts (→)	klein stapje naar rechts
toets PageUp (PgUp)	grote stap naar links
toets PageDown (PgDn)	grote stap naar rechts
toets Home	naar het begin van de tijdslijn
toets End	naar het eind van de tijdslijn

Hulp en/of tips bij het scrollen door tijd kan de gebruiker vinden in de inline-help.

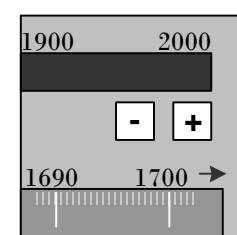
Zoomen naar verschillende detailniveau's

Tijdens de fase functioneel ontwerp heb ik beschreven hoe de tijdslijn om moest gaan met verschillende detailniveau's (zoomlevels).

Afhankelijk van het gekozen detailniveau's ziet de gebruiker een kortere of langere periode.

Op het laagste detailniveau moet het gehele tijdsbestek van de tijdslijn zichtbaar zijn. Hogere niveau's zoomen in naar bijvoorbeeld een periode van 100 jaar.

In het functioneel ontwerp heb ik de volgende detailniveau's voorgesteld:



Figuur 13: zoomfunctionaliteit

detailniveau	zichtbare periode
1	Alles zichtbaar
2	1000 jaar (millennium)
3	periode: romeinse tijd, middeleeuwen, gouden eeuw
4	500 jaar
5	100 jaar (eeuw)
6	50 jaar

7	10 jaar (decennium)
---	---------------------

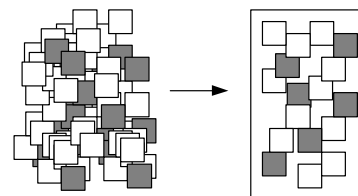
Op dat moment was ik mij blijkbaar nog niet bewust van de aard van de collectie van het Rijksmuseum. Deze collectie bevat (bijna) alleen maar kunstobjecten vanaf het jaar 1400. Detailniveau 2, 3 en 4 bleken hierdoor overbodig te zijn.

Het leek me belangrijk het aantal detailniveau's en de instellingen daarvan flexibel te maken, zodat de configuratie van de detailniveau's niet door mij maar door een redacteur van het Rijksmuseum gebeurt. Ik heb besloten de instellingen van de detailniveau's op te nemen in het configuratiebestand van de eerder genoemde ElasticRange.

Ik constateerde het detailniveau omgekeerd evenredig was met de hoeveelheid objecten op de tijdslijn die tegelijkertijd getoond moesten worden.

Het laagste detailniveau moest in staat zijn alle elementen uit het gehele tijdvak te tonen. Dit aantal ligt tussen 2000 en 4000 kunstobjecten, kunstenaars en periodes. Wanneer al deze elementen in één keer getoond zouden moeten worden, zou de tijdslijn overvol raken en totaal geen overzicht meer bieden. Ook kent in dit geval de infrastructuur haar beperkingen. Het is niet wenselijk duizenden afbeeldingen van kunstobjecten over de interverbinding van de gebruiker te versturen.

Om toch een reëel beeld te geven van de totale collectie besloot ik dat er een aggregatie van de gegevens getoond moest worden. Afbeeldingen van bijvoorbeeld een kunstobject zullen in dit geval niet allemaal één voor één geladen maar worden getoond als één afbeelding die een veelal kunstobjecten representeert zoals te zien is in Figuur 14.



Figuur 14: Aggregatie van kunstobjecten

Filteren op criteria

Tijdens de fase functioneel ontwerp heb ik de filterfunctionaliteit minder uitgebreid beschreven dan de rest van de onderdelen van de tijdslijn.

Ik wist al wel dat er een filter nodig was, maar de exacte criteria waren nog niet bekend. Ik beschreef een aantal scenario's, waarin mogelijke filter-criteria uit naar voren kwamen, maar die heb ik op dat moment niet teruggekoppeld met als gevolg dat de filter pas te laat de aandacht kreeg die hij verdiende.

Uit de scenario's bepaalde ik volgende filter-criteria:

- Kunstobjecten gemaakt door een bepaalde kunstenaar
- Kunstobjecten binnen een bepaalde collectie
- Alles binnen een bepaalde stijlperiode
- Periodes en gebeurtenissen uit de geschiedenis weglaten

Het leek me ook belangrijk de filter omgekeerd te kunnen passen. In plaats van: "Toon alle kunstwerken van een bepaalde kunstenaar" dacht ik aan "Toon alles, behalve de kunstwerken van die kunstenaar".

De filter dient als verfijning van de vooraf gemaakte dataselectie. De redacteur bepaald met welke dataselectie de tijdslijn gevuld wordt, gebruikers kunnen hier dan nog extra sub-selecties op toepassen.

Elastische tijdvakken (ElasticRange)

Een grote hoeveelheid van de collectie van het Rijksmuseum bestaat uit kunstobjecten uit de Gouden Eeuw (17^e eeuw). Op de tijdslijn is dit een drukke periode. Er zijn ook periodes uit de geschiedenis waarvan het Rijksmuseum minder kunstwerken bezit.

Om te voorkomen dat sommige plekken op de tijdslijn overvol raken, en andere plekken te leeg lijken heb ik in de fase functioneel ontwerp het principe elastische tijdvakken (ElasticRange) beschreven.

Bij dit principe wordt de tijdslijn onderverdeeld in een aantal periodes waarbij per periode een bepaalde hoeveelheid ruimte gereserveerd kan worden.

start	end	width (in pixels)
1400	1600	1500
1600	1700	2000
1700	1900	1500

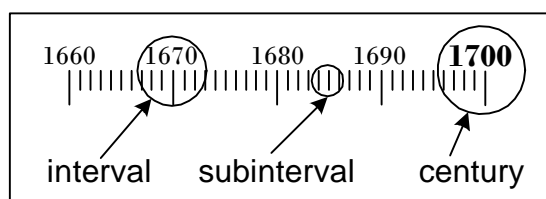
Om de tijdslijn meerdere detailniveau's te laten ondersteunen heb ik hieraan nog een extra dimensie toegevoegd: zoomlevel.

zoomlevel \ tijdvak	level 1	level 2	level 3	level 4
1400 / 1600	150	1000	2000	4000
1600 / 1700	300	2000	4000	6000
1700 / 2000	150	1000	2000	4000
Totale breedte	600	4000	8000	14000

De onderverdeling tussen de tijdvakken zou automatisch berekend kunnen worden. Ik heb echter, omdat de collectie van het Rijksmuseum niet vaak verandert, gekozen voor handmatige configuratie van de tijdvakken.

Bij het configureren van de tijdvakken kunnen ook een interval en een subinterval per tijdvak worden ingesteld. Het interval bepaalt om de hoeveel jaar er een streepjes (class Markers) wordt getoond op de tijdslijn (zie Figuur 15).

ElasticRange dient als tussenlaag tussen de jaartallen en de x-coördinaten op de tijdslijn.



Figuur 15: soorten intervallen

Verdeling van de content-elementen op de tijdslijn

De chronologische verdeling van periodes en gebeurtenissen op een tijdslijn wordt doorgaans horizontaal gevisualiseerd. Het zou voor de tijdslijn voor het Rijksmuseum onlogisch zijn om hiervan af te wijken, omdat een andere vorm bij iedere gebruiker tot verwarring zou leiden.

Periodes op de tijdslijn kunnen elkaar overlappen, bepaalde kunstenaars leefden op hetzelfde moment, diverse kunstobjecten zijn in hetzelfde jaar gemaakt etc. Kortom, de tijdslijn zou te maken krijgen met horizontale overlappingen.

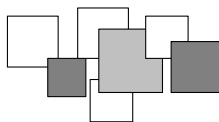
In de fase functioneel ontwerp bepaalde ik op welke manier omgegaan moest worden met verbanden en overlappingen in tijd.

Indien er bij periodes op de tijdslijn in horizontale richting overlapping plaatsvindt, wordt uitgeweken in verticale richting zoals is weergegeven in Figuur 16.



Figuur 16: weergave van periodes bij overlapping

Uit esthetische overwegingen besloot ik overlappingen bij kunstobjecten op een andere wijze te visualiseren. Ik beschreef een manier waarbij overlappen toegestaan werd, mits bij deze overlapping een ander object voldoende (voor minimaal 50%) zichtbaar bleef.



Figuur 17: weergave van kunstobjecten bij overlappingen

Visualisatie van elementen met onzekere datum

In de fase analyse en definitie bleek al dat veel kunsthistorische objecten een onzekere datum hebben. Ook stijlperiodes hebben in de meeste gevallen geen exact begin- en eindpunt. In het functioneel ontwerp heb ik vastgelegd hoe dit op tijdslijn worden zichtbaar wordt gemaakt en welke verschillende soorten onzekerheden er kunnen optreden. In bijlage III (functioneel ontwerp) zijn de verschillende verschijningsvormen opgenomen.

Overige functionaliteiten van de interface

De content-elementen die zich op de tijdslijn bevinden zijn allemaal klikbaar. Bij het aanklikken van bijvoorbeeld een kunstobject verschijnt een detailscherm waarin gedetailleerde informatie over het object wordt getoond. Tijdens de fase functioneel ontwerp stond nog niet geheel vast welke detailinformatie getoond zou worden, daarom moest het detailscherm dermate flexibel opgezet worden zodat met verschillende soorten detailinformatie omgegaan kon worden. Meer informatie over het detailscherm is te vinden in bijlage III.

4.4 Resultaat van het functioneel ontwerp vaststellen

In de fase functioneel ontwerp is het klassendiagram opgesteld en zijn de diverse interface elementen functioneel beschreven.

Helaas zijn niet alle interface elementen even uitgebreid beschreven. De filter bijvoorbeeld was wegens haast nog niet volledig uitgewerkt in het functioneel ontwerp. Dit had gevolgen voor de volgende fase.

Ook de eerste versie van het klassendiagram had beter kunnen zijn wanneer ik deze eerder met collega-ontwikkelaars had doorgenomen of me anderzijds in een vroeger stadium had verdiept in de software-patterns die in het volgende hoofdstuk aan bod komen.

Toch is de fase functioneel ontwerp één van de fasen die erg veel bruikbare input heeft opgeleverd voor volgende fasen. Alle beschreven functionaliteiten zoals het scrollen en het zoomen konden in de volgende fase technisch worden uitgewerkt.

5. TECHNISCH ONTWERP

Op basis van het klassendiagram uit het functioneel ontwerp heb ik in deze fase de technische werking van alle objectclasses uit het klassendiagram beschreven. Daarnaast ben ik dieper ingegaan op het benaderen van de XML database en heb ik vastgelegd op welke wijze externe resources (bijv. afbeeldingen) benaderd konden worden. Ook heb ik me tijdens deze fase verdiept in software-patterns. In dit hoofdstuk zijn verwijzingen opgenomen naar het technisch ontwerp in bijlage IV.

5.1 Softwarepatronen onderzoeken

Tijdens de fase technisch ontwerp heb ik mij ingelezen in standaarden die bestaan voor object georiënteerd programmeren (OOP).

Ik had gehoord over software-patterns (softwarepatronen) en het leek me zinvol om me daar in te verdiepen aangezien Flash MX 2004 (de geselecteerde ontwikkelomgeving) volledig voorbereid is op object oriëntatie.

Ik vond het belangrijk zoveel mogelijk volgens standaarden te werken. De volgende redenen speelden hierbij een rol: ik wilde zelf programmeer-technisch een grote groei doormaken. Daarnaast wilde ik de tijdslijn voorzien van goede en overzichtelijke programmacode die later gemakkelijk uitgebreid zou kunnen worden. Tenslotte was het project dermate complex dat niet-gestructureerd programmeren zou leiden tot het mislukken van het project en een product zou opleveren die niet aan de eisen zou voldoen.

Het duurde enige tijd voordat ik erachter was wat het precies betekende, werken volgens softwarepatronen, maar toen ik het begon te snappen, bleek ik ze zelf al diverse keren onbewust toegepast te hebben.

Een software patroon is een oplossingsstrategie voor een specifiek probleem in een bepaalde context oftewel een standaardoplossing die al vaker succesvol is toegepast voor een bepaald software-probleem.

Een patroon stelt een ontwikkelaar in staat om gebruik te maken van de ervaring van anderen. De patronen bieden bewezen oplossingen voor specifieke problemen. Parate kennis van patronen kan het ontwikkelproces versnellen en de uiteindelijke producten verbeteren. De oplossingen die de patronen beschrijven zijn niet nieuw, maar het feit dat ze systematisch benoemd en beschreven worden is dat wel. Een oplossing voor een software-probleem wordt pas een patroon wanneer deze meerdere malen in de praktijk toegepast is in bestaande software.

Ik heb me op verschillende manieren verdiept in softwarepatronen. Op internet heb ik een aantal documenten gevonden met bruikbare informatie. Onder andere het document 'software-patronen' van M.J.B. van Elswijk gaf veel verduidelijking. Ook de reader waarin J. Schramp (HHS) een bewerking van het boek 'Applying UML and Patterns' van Craig Larman levert was erg bruikbaar. In de literatuurlijst zijn verwijzingen opgenomen naar deze documenten.

Ik had me eigenlijk al eerder in softwarepatronen moeten verdiepen merkte ik tijdens deze fase. Er bestaan softwarepatronen voor de gehele doorlooptijd van een systeem-ontwikkelingsproces. Doordat ik daarvan niet op de hoogte was heb ik het klassendiagram regelmatig moeten verbeteren naarmate mijn ervaring met object georiënteerd programmeren groeide.

5.2 Softwareclasses technisch beschrijven

In het technisch ontwerp, dat opgenomen is in bijlage IV, zijn alle softwareclasses die tijdens het functioneel ontwerp zijn bepaald technisch beschreven. Tijdens het uitwerken en wijzigen van de classes werd het technisch ontwerp telkens bijgesteld.

In dit afstudeerrapport wordt bewust niet dieper ingegaan op de technische beschrijving van de classes en het berichtenverkeer ertussen, om zo ook de minder technische lezer geboeid te houden.

5.3 Dataverwerking technisch beschrijven

Binnen de fase technisch ontwerp heb ik een activiteit opgenomen die als doel had te onderzoeken op welke manier het beste gecommuniceerd kon worden met de XML database en hoe de data het best gestructureerd, aangeroepen en verwerkt kon worden.

Ik onderzocht de verschillende manieren hoe Flash met XML om kan gaan en koos uiteindelijk de manier die het makkelijkst in het gebruik leek, de meeste documentatie bevatte en die geen server-sided services vereiste.

Voor de datastructurering binnen de tijdslijn zelf, verwachtte ik tijdens het schrijven van de eerste versie van het technisch ontwerp genoeg te hebben aan een Array die de honderden content-elementen op de tijdslijn bevatte. Uit de stresstest die later gehouden is bleek dat toch gezocht moest worden naar een efficiëntere manier voor het omgaan met de grote hoeveelheid data op de tijdslijn. De stresstest wordt beschreven in hoofdstuk 7.2.

5.4 Meertalige ondersteuning technisch beschrijven

In de fase analyse en definitie werd duidelijk dat de tijdslijn in meerdere talen aangeboden moest worden. Zowel de content als de interface-elementen moesten dit ondersteunen.

Tijdens de fase technisch ontwerp heb ik vastgelegd hoe dit doel te bereiken. De taal waarin de content verschijnt is afhankelijk van de dataselectie. De dataselectie wordt voor iedere taal apart gecached op de server.

Tijdens het aanroepen van de tijdslijn wordt meegegeven in welke taal de website bekeken wordt. Aan de hand van dat gegeven wordt door de tijdslijn de juiste dataselectie uit de servercache opgevraagd. Alle interface-elementen zijn eveneens op de hoogte van de geselecteerde taal en presenteren zichzelf in de juiste vorm.

5.5 Resultaat van de fase technisch ontwerp vaststellen

Terugkijkend op de fase technisch ontwerp kan ik zeggen dat deze fase voor de totstandkoming van de tijdslijn veel heeft betekend. Het continu bijwerken van het technisch ontwerp document tijdens de volgende fasen dwong mij er toe zojuist gemaakte keuzes te beoordelen. Ik legde een soort verantwoording af voor mezelf. Het zou immers dom zijn om verkeerde keuzes vast te leggen terwijl wist dat ik die later weer zou moeten herroepen.

Ik heb in dit afstudeerrapport de gemaakte keuzes tijdens de fase technisch ontwerp niet uitvoerig toegelicht. Dit omdat de meeste keuzes zeer technische van aard waren en minder relevant voor de beeldvorming van het door mij doorlopen proces. In bijlage IV wordt dieper ingegaan op de materie.

6. GRAFISCH ONTWERP

De fase grafisch ontwerp in de standaardfasering van Fabrique bestaat uit het uitvoeren van stijlstudies, het maken van ontwerpconcepten en het komen met een ontwerpvoorstel.

Ik heb deze fase ook opgenomen in de planning van de tijdslijn. Hoewel ik deze fasen zelf niet zou doorlopen zou ik er wel degelijk mee in aanraking komen. Het was namelijk de bedoeling dat de ontwerpers die de website ontworpen ook zouden zorgdragen voor het interfaceontwerp van de tijdslijn.

De input die de ontwerpers van mij zouden krijgen had betrekking op de functionaliteiten en de globale opbouw van de tijdslijn. De output die ik van hen wenste te ontvangen waren één of meerdere schermafbeeldingen van de vormgegeven gebruikers interface van de tijdslijn in de stijl van de website. Daarnaast zou ik een aantal voorbeelden te zien krijgen hoe de tijdslijn verwerkt zou worden in het geheel.

Bij het uitvoeren van de stijlstudies voor de website werd door de ontwerpers gezocht naar de juiste look & feel voor het Rijksmuseum door vrij te spelen met kleur, vorm, beeldmateriaal, typografie, vlakverdeling etc.

Bruikbare resultaten uit de stijlstudie zijn meegenomen in het conceptontwerp. Dit ontwerp leverde een aantal verschillende geanimeerde concepten op. Op basis daarvan kon het Rijksmuseum bepalen welk concept qua stijl het best bij hen paste, oftewel welke ze het 'mooist' vonden.

Zelf was ik ook nog betrokken bij het uitwerken van een ontwerpconcept. Mijn hulp werd ingeroepen voor het maken van een interactieve demo van een van de grafische ontwerpen van de ontwerpers.

Doorgaans wordt één van de ontwerpconcepten gekozen en verder uitgewerkt tot een ontwerpvoorstel. Het Rijksmuseum nam echter het zekere voor het onzekere en wilde graag dat Fabrique meerdere ontwerpconcepten uitwerkte tot ontwerpvoorstel.

Bij een ontwerpvoorstel worden aan de hand van het functioneel ontwerp een aantal gedetailleerde ontwerpen van representatieve schermen van het systeem gemaakt. In deze ontwerpen wordt gebruik gemaakt van echte tekst en beeldmateriaal en definitieve naamgeving in de navigatie.

Aan de vormgeving van de tijdslijn zou begonnen worden door de ontwerpers nadat het Rijksmuseum een ontwerpvoorstel gekozen en goedgekeurd had. De tijdslijn moest immers qua vormgeving één geheel vormen met de overkoepelende website.

Helaas liet het goedkeuren van het ontwerpvoorstel eindeloos op zich wachten door communicatieproblemen zowel intern als onderling.

In dit rapport wordt uiteraard niet dieper ingegaan op de aard van die problemen, maar voor mezelf was het wel erg leerzaam om communicatie op zo'n hoog niveau te kunnen volgen.

Voor de tijdslijn had dit natuurlijk grote gevolgen. Aangezien de vormgeving voor de website (op het moment van schrijven) nog steeds niet definitief is, is er voor de tijdslijn tot nu toe nog geen lay-out bepaald. Ik heb er bewust voor gekozen om niet zelf met een vormgeving van de tijdslijn te komen, ten eerste omdat ik dat niet ingecalculeerd had in m'n planning en ten tweede om geen verkeerde suggesties te wekken bij het Rijksmuseum.

De tijdslijn is dus nog steeds voorzien van functionele lay-out. Dit is jammer omdat dit ik in dit rapport graag het definitieve ontwerp van de tijdslijn had willen presenteren.

7. PROGRAMMERING

Het ontbreken van de definitieve vormgeving stond de ontwikkeling van de tijdslijn niet in de weg. In de fase functioneel ontwerp had ik immers de functionaliteiten van de tijdslijn reeds bepaald waarna ik in de fase technisch ontwerp uitgebreid onderzocht had hoe de functionaliteiten technisch het best uitgewerkt kon worden. Deze basis was voldoende om een goede tijdslijn te ontwikkelen waar de vormgeving later als zijnde een schil over heen gelegd kon worden.

In eerste instantie was mijn bedoeling de fase ‘functioneel prototype tijdslijn’ en de fase ‘definitieve tijdslijn’ te verdelen over twee hoofdstukken. Ik had namelijk tijdens het uitstippelen van de ontwikkelstrategie een duidelijke scheiding tussen deze twee fasen voor ogen.

Ik heb deze hoofdstukken echter gecombineerd en verspreid over een aantal paragrafen omdat ik zelf de scheiding tussen prototype en definitieve tijdslijn minder als mijlpaal heb ervaren dan ik aanvankelijk verwachtte.

Dit hoofdstuk beschrijft de ontwikkeling van de het oorspronkelijke functionele prototype. Vervolgens wordt ingegaan op de tussentijdse testen waarna beschreven wordt hoe de ontwikkeling van de tijdslijn zich voortzette.

7.1 Prototype ontwikkelen

De programmering van de tijdslijn begon met het bouwen van een functioneel prototype. In dit prototype werden een aantal functionaliteiten in de diepte uitgewerkt. Het had vooral als doel om eventuele problemen vroegtijdig te kunnen constateren en lokaliseren.

Het prototype zou uitgebreid getest worden op verschillende criteria. Omdat de tijdslijn grote hoeveelheden data te verwerken zou krijgen, was een stresstest waarmee performance getest werd, geen overbodige luxe. Ook zou het prototype aan een usabilitytest onderhevig gemaakt worden. Daarnaast wilde ik graag code reviews uitvoeren. Hierbij wordt samen met een andere programmeur de totale programmacode doorlopen en beoordeeld op consistentie, het gebruik van softwarepatronen en het werken volgens de interne coding rules (programmeer regels) van Fabrique. Het uitvoeren van een code review zou kunnen betekenen dat ik mijn programmeerstijl na deze test drastisch zou moeten aanpassen indien nodig.

Eén van de eerste functionaliteiten die ik ontwikkeld heb voor de tijdslijn was de ScrollArea. De bijbehorende ScrollBar voor de besturing van de ScrollArea hoorde hierbij.

Ik besloot toetsenbordondersteuning zo vroeg mogelijk al te verwerken in het prototype zodat ik zelf tijdens het programmeren en het testen van het prototype niet continu de muis hoefde te gebruiken, anders had ik ongetwijfeld RSI opgelopen.

Het ElasticRange principe welke reeds beschreven is in het functioneel ontwerp was ook een onderdeel die al in een behoorlijk vroeg stadium veel aandacht kreeg. Veel van de andere softwareclasses zouden namelijk afhankelijk zijn van deze tussenlaag die zorgt voor de vertaalslag van jaartallen naar x-coördinaten.

Voor het opvragen van dummy data (testdata voor tijdens het programmeren) gebruikte ik in deze fase nog losse XML bestanden. De opbouw en de inhoud van deze bestanden verbeterde ik telkens en de wijzigingen verwerkte ik in het technisch ontwerp.

7.2 Tussentijdse testen uitvoeren

De tests waren opgezet om te bepalen of het prototype van de tijdslijn tot dan toe voldeed aan de specificaties ik had opgesteld in de fase analyse en definitie en geconcretiseerd had in het functioneel ontwerp.

Hierbij heb ik mezelf het volgende afgevraagd: waren de specificaties helder toen ik met programmeren begon en heb ik ze wel precies genoeg beschreven? De meeste specificaties voldeden aan deze criteria. Een aantal waren wat onduidelijker zoals de hoeveelheid dataverkeer die de tijdslijn mag genereren. In ieder geval waren ze bruikbaar om een aantal tests uit te kunnen voeren.

Stresstest

Het prototype van de tijdslijn werd tijdens de stresstest getest op performance bij grote hoeveelheden en verschillende soorten data. De test vond plaats bij Q42, de technische partij binnen de interne projectorganisatie.

De tijdslijn bleek al bij één van de eerste testsets (de tijdslijn 250 periodes laten tonen op een pc met een snelheid van 1GHz) al onnoemlijk traag te worden. De test is toen gestaakt omdat het zinloos was de nog druk verder op te voeren als het prototype bij deze hoeveelheid data al faalde.

We hebben toen direct met een team van drie programmeurs de knelpunten bepaald door de programmacode te debuggen en bepaalde instructies tijdelijk uit te schakelen.

De vertraging bleek veroorzaakt te worden door een drietal factoren:

1. De content-elementen overerden van de class MovieClip die standaard in Flash aanwezig is. Het plotseling aanmaken van de 250 van deze objecten vergde te veel van de testcomputer.
2. Collision detection tussen de verschillende periodes werd uitgevoerd met een recursieve functie die exponentieel groeide naarmate er meer periodes op de tijdslijn werden toegevoegd.
3. Het toenemen van het aantal MovieClips zorgde ervoor dat de tijdslijn steeds meer geheugen gebruikte.

We hadden verwacht dat de XML parser van Flash ook voor vertraging zou zorgen. Maar dit bleek zelfs bij grotere hoeveelheden data geen bottleneck te zijn.

Uit het analyseren van de programmacode voor de stresstest kwamen zeer bruikbare oplossingen naar voren. Ik kreeg goede tips om de data binnen de tijdslijn beter te structureren. Hiervoor is de HashIndex in het leven geroepen. Deze class is dan ook na de stresstest pas toegevoegd aan het klassendiagram. Verder werd mij aangeraden content-elementen niet te laten overerven van MovieClips, maar er een referentie naar op te nemen. Ik zou in dit rapport graag dieper ingaan op de gevonden technische oplossingen maar voor het inzichtelijk maken van het door mij doorlopen proces is dat niet relevant.

Code review

Een andere test die ik heb uitgevoerd is code review, een variant van whitebox-testing (structureel testen) waarbij samen met een andere programmeur de totale of belangrijkste programmacode wordt doorlopen en beoordeeld wordt op consistentie, het goed gebruiken van softwarepatronen, het werken volgens interne coding rules en dergelijke.

De code review die ik samen met Jeroen Pulles (ontwikkelaar binnen de interne projectorganisatie) heb uitgevoerd heeft veel opgeleverd.

Vooraf ontving ik tips hoe ik het beste met events (berichtenverkeer tussen de objecten) om kon gaan. Maar ook heb ik samen met hem de HashIndex verbeterd. Ook liet hij me zien dat ik op bepaalde plaatsen in de code het pattern 'factory method' had gebruikt zonder mezelf

daarvan bewust te zijn en ook constateerde we in enkele classes programmacode die eigenlijk behoorde tot de verantwoordelijkheid van een andere class.
De code review heeft de kwaliteit van de programmacode zeker verhoogd.

Usability testen

De usability test zoals ik me die aan het begin van het project had voorgesteld heeft een andere invulling gekregen dan ik had verwacht.
Het was de bedoeling externe gebruikers met het prototype te laten werken om op die manier hun tevredenheid te meten en te bepalen of met de tijdslijn op een efficiënte en effectieve manier gestelde doelen bereikt konden worden.
Ik heb er echter voor gekozen de usability test van het prototype uit te stellen. Dit omdat het verbeteren van de performance van de tijdslijn in dit stadium een grotere waarde had dan het testen op usability. Het prototype was nog niet in staat gebruikers een goed beeld te geven van hoe de tijdslijn zou worden.
Wanneer ik de tijdslijn toch al in dit stadium had willen testen op usability had ik dat kunnen doen door gebruikers het functionele ontwerp van de tijdslijn op papier te laten testen, maar omdat ook de vormgeving nog niet definitief was vastgesteld, leek het me toch zinvoller om de test uit te stellen.
Het is niet ondenkbaar dat ik u tijdens de zitting, waar dit rapport besproken wordt, meer informatie kan verschaffen over de invulling die de usability test uiteindelijk heeft gekregen.

7.3 Ontwikkeling van de tijdslijn doorzetten

Tijdens het doorontwikkelen van de tijdslijn hebben de resultaten uit de stresstest en de code review gezorgd voor een frisse kijk, nog meer motivatie en bovenal veel extra kennis.
Een groot aantal extra functionaliteiten die nog niet in prototype aanwezig waren zijn doorontwikkeld zoals bijvoorbeeld het zoomen.
Regelmatig stelde ik het technisch ontwerp nog bij en beschreef ik het resultaat van gemaakte keuzes.
Ik merkte ook dat mijn kennis van ActionScript 2.0 elke dag nog toenam door de ervaring die ik op deed tijdens het programmeren van de tijdslijn en door het blijven zoeken naar en analyseren van voorbeeldcode.
Op het gebied van performance begon de tijdslijn grote vooruitgang te boeken en ook interface-technisch bracht ik aantal interessante veranderingen aan. Het opnemen van zogenaamde tweens maakten de tijdslijn een stuk levendiger. Tweens zijn geanimeerde overgangen waarbij de plaats of verschijningsvorm van een MovieClip object in Flash wordt veranderd.

7.4 Resultaat van de fase programmering vaststellen

De fase programmering heeft veel opgeleverd, maar dat wil niet zeggen dat de tijdslijn nu helemaal klaar is. De tijdslijn moet programmeer-technisch nog verder geoptimaliseerd worden. Bepaalde onderdelen die tijdens het moment dat dit rapport geschreven werd al af hadden moeten zijn waren nog in ontwikkeling. De filterfunctionaliteit is hier een voorbeeld van.
Het duurt nog enige tijd voordat de tijdslijn in uiteindelijke vorm on-line geïmplementeerd kan worden, de overkoepelende website kent dit zelfde probleem aangezien de ontwikkeling hiervan ook nog niet geheel volbracht is.
Volgens de huidige verwachtingen zal de nieuwe website van het Rijksmuseum nog minimaal tot de maand mei of juni op zich laten wachten.

8. AFRONDENDE FASEN

De fasen redactie, implementatie en documentatie zijn in hoofdstuk ondergebracht. Deze fasen vinden alle drie plaats na oplevering van dit rapport. Ik beschrijf dan ook niet wat mijn activiteiten zijn geweest, maar wat mijn rol nog zal zijn tijdens fasen.

8.1 Redactie

In de fase redactie wordt de nieuwe database gevuld met reële data en worden controles uitgevoerd op de betrouwbaarheid en volledigheid van deze data.

Zoals in de gegevenanalyse is beschreven hebben Fabrique en Q42 reeds een aantal migraties en conversies uitgevoerd tijdens de overzetting van de Aria database naar de XML database. Ook heb ik al vermeld dat op gezette tijden de ADLIB database gesynchroniseerd zal worden met de XML database.

De taak van de redacteurs is om tijdens de fase redactie nieuwe gegevens aan de XML database toe te voegen.

De tijdslijn maakt gebruik van een aantal databronnen die nu nog niet in de XML database opgenomen zijn. Bijvoorbeeld de gegevens over stijlperiodes, gebeurtenissen en periodes uit de geschiedenis, maar ook de configuraties van de tijdslijn en de verschillende dataselecties zullen nog moeten worden aangemaakt.

Mijn rol tijdens deze fase is het ondersteunen van de redacteurs bij het invoeren van de gegevens voor de tijdslijn.

8.2 Implementatie

De fase implementatie vindt plaats zodra de gehele nieuwe website van het Rijksmuseum klaar is om gepubliceerd te worden. De tijdslijn zal op dat moment helemaal klaar zijn en een eenheid vormen met de website.

In deze fase vindt de bestandsoverdracht plaats naar de webserver en zullen nog een hoop technische acties ondernomen worden om de nieuwe website te configureren voor de on-line omgeving.

Voor de tijdslijn geldt hetzelfde, deze zal ook voorbereid worden op de on-line publicatie.

Naast bestandsoverdracht vindt in deze fase ook kennisoverdracht plaats. Alle belanghebbenden uit de externe projectorganisatie zullen worden ingelicht over het gebruik en het onderhoud van het totale systeem waar de tijdslijn een onderdeel van uit maakt.

Het is mogelijk dat er in deze fase nog fouten (bijv. bugs) worden geconstateerd die niet zichtbaar waren in de ontwikkelomgeving. Indien er problemen optreden wordt hier alle aandacht op gevestigd. Het is zaak deze fase zo spoedig mogelijk te doorlopen. Mijn rol hierbij is het opsporen en oplossen van fouten die betrekking hebben op de tijdslijn.

8.3 Documentatie

In de fase documentatie worden tenslotte de documenten opgeleverd die voor het Rijksmuseum relevant zijn. Het betreft hier technische documentatie, maar ook handleidingen bij de diverse onderdelen.

Ik zal nog verantwoordelijk zijn voor de technische handleiding die met de tijdslijn meegeleverd wordt.

Na het opleveren van de documentatie vind er altijd een evaluatie fase plaats waarbij het product en het proces geëvalueerd worden maar waar ook over de toekomst gesproken wordt.

9. EVALUATIE

In dit hoofdstuk evalueer het proces dat ik de afgelopen maanden heb doorlopen. Vervolgens beoordeel ik het product dat daaruit is voortgekomen, de tijdslijn.

9.1 Procesevaluatie

Bij het evalueren van mijn proces ga ik in op de gehanteerde ontwikkelstrategie en planning, vermeld ik mijn betrokkenheid bij het totale project en kijk ik naar mijn persoonlijke technische groei.

Ontwikkelstrategie en planning

Bij aanvang van het project moest ik een ontwikkelstrategie kiezen. Ik koos voor de standaardfasering van Fabrique omdat ik daar al enige tijd mee vertrouwd was en omdat ik deel uit maakte van de interne projectorganisatie die voor de ontwikkeling van de overkoepelende website, dezelfde methode gebruikte.

Voor mij was het de eerste keer dat ik alle fasen uit de standaardfasering zo bewust meemaakte en zelf verantwoordelijk was voor het plannen ervan. Zelf de verantwoording dragen voor een serieus project van dergelijke omvang heb ik als zeer spannend ervaren, en ook zeker als een grote uitdaging.

Terugkijkend op de invulling en het gebruik van de standaardfasering kan ik concluderen dat het mij heeft geboden wat ik er van verwacht had: een erg bruikbare houvast tijdens het doorlopen van de planning die ik opgesteld had op basis van deze fasering.

Alle fasen uit de standaardfasering heb ik terug laten komen in een planning die eigenlijk veel te gedetailleerd was en daardoor minder overzichtelijk.

Als ik terugkijk naar de eerste versie van de planning zie ik dat ik tijdens het op opstellen de fase analyse en definitie enigszins heb onderschat. Ik had door onwetendheid een aantal activiteiten uit deze fase ondergebracht bij latere fasen. Ook een aantal activiteiten die eigenlijk thuishoorden in het functioneel ontwerp waren terechtgekomen in het technisch ontwerp en anders om. Mezelf bijvoorbeeld verdiepen in software patterns was niet alleen nuttig voor het technisch ontwerp, maar had ook heel bruikbaar kunnen zijn voor het functioneel ontwerp. Tijdens het vorderen van het project bemerkte ik nog een aantal van dit soort dingen. Hierop stelde ik de planning tekens bij en bracht ik de activiteit alsnog onder bij de juiste fase.

In de standaardfasering is een duidelijke scheiding te zien tussen het prototype en het definitieve systeem met daar tussen een aantal tests.

Deze scheiding kwam tijdens het doorlopen mijn fasering minder sterk naar voren. Het prototype liep als het ware vanzelf over in de ontwikkeling van het definitieve systeem. Wel heb ik op dit omslagpunt een aantal tests uitgevoerd zoals de bedoeling was maar de usabilitytest heb ik helaas moeten uitstellen. De reden hiervoor was dat het oplossen van de problemen die uit de stresstest naar voren kwamen een hogere prioriteit hadden, daarbij was het prototype nog niet in staat gebruikers een goed beeld te geven van de tijdslijn. En omdat er nog geen grafische ontwerp van de interface beschikbaar was, was het ook niet zinvol om het prototype op papier op usability te testen.

Zoals reeds beschreven is in hoofdstuk 6 van dit rapport, heeft de fase grafisch ontwerp niet opgeleverd wat ik ervan verwacht had. Doordat de vormgeving van de overkoepelende website vertraging opliep was niet tijdig een definitieve lay-out voor de tijdslijn beschikbaar.

Ik had dit kunnen opvangen door zelf met een (tijdelijke) lay-out te komen voor de tijdslijn maar heb dit bewust niet gedaan, ten eerste omdat ik dit niet ingecalculeerd had in m'n planning en ten tweede om geen verkeerde suggesties te wekken bij het Rijksmuseum. De tijdslijn is uitgerust met de functionele lay-out waar de vormgeving later als zijnde een grafische schil over heen gelegd kan worden.

Ondanks de bovengenoemde problemen hebben toch de verschillende fasen zeer bruikbare resultaten geleverd voor de vervolgfases.

Hoe verder mijn proces vorderde des te beter zag ik hoe logisch de standaardfasering eigenlijk in elkaar zit. Nu ik eenmaal praktijkervaring heb met het toepassen van de totale standaardfasering zou ik bij een volgend project van deze omvang nog beter in staat zijn om volgens deze ontwikkelstrategie een goed ontwikkelproces te doorlopen.

Betrokkenheid bij het totale Rijksmuseum project

Als ik terugblik op mijn betrokkenheid bij het gehele project dan kan met recht concluderen dat deze hoog was. Ik wilde overal bij zijn en was vaak mee aan het kijken met de rest van de projectgroep. Zowel het proces dat de ontwerpers doormaakten als ook de totstandkoming van de technische onderdelen boeide mij enorm.

De bezoeken die ik tijdens de afstudeerperiode bracht aan het Rijksmuseum waren alle drie totaal verschillend van aard. De eerste was ter oriëntatie, de tweede feestelijk en de derde spannend. Maar alle drie hebben ze opgeleverd wat ik van verwacht had en waren bovenal erg leuk en interessant.

De diverse technische overlegmomenten die plaatsvonden hebben ook zeker hun vruchten afgeworpen. Ik bleef door deze momenten op de hoogte van de technische koers die gevaren werd en deelde de problemen die ik tijdens de ontwikkeling van de tijdslijn tegenkwam.

Ook het meewerken aan de interactieve demo die gebruikt werd voor het ontwerpconcept van de overkoepelende website toont mijn betrokkenheid. De presentatie van het ontwerpconcept vond namelijk gelijktijdig plaats met de geplande oplevering van de conceptversie van dit afstudeerrapport. De demo voor de presentatie gaf ik hierbij de hoogste prioriteit.

Technische groei

Op het technische vlak heb ik de afgelopen maanden een ontzettende groei doorgemaakt. Mijn jaren lange programmeerervaring wilde ik tijdens dit project tot een hoogtepunt brengen door volledig object georiënteerd volgens standaarden en regels te werken en daar het beste product uit voort te laten komen wat ik ooit gemaakt had.

Ik vind dat ik daar behoorlijk in geslaagd ben. Tijdens aanvang van het project ontbrak mij de kennis van Flash MX 2004 en ActionScript 2.0, maar het duurde niet lang voordat ik mijn weg gevonden had in deze programmeeromgeving en deze taal letterlijk kon lezen en schrijven!

9.2 Productevaluatie

Bij het evalueren van de tijdslijn voor het Rijksmuseum kijk ik of de gestelde doelen zijn bereikt, bepaal ik of de tijdslijn voldoet aan de eisen en specificaties, en werp ik een blik in de nabije toekomst.

Doelrealisatie

Het Rijksmuseum wilde in de vorm van een tijdslijn op de website verbanden aan kunnen geven tussen kunstenaars, kunstobjecten en periodes. Als afstudeerder was ik verantwoordelijk voor de definiëring en ontwikkeling van deze tijdslijn.

Ik heb interne en externe doelen in kaart gebracht en op basis daarvan functionele eisen en aanvullende specificaties voor de tijdslijn opgesteld.

Bijna alle functionele eisen waren na afloop van de afstudeerperiode verwerkt. Toch zullen de komende weken nog benut worden om de tijdslijn te optimaliseren voor gebruik op de website. Een aantal onderdelen verdient nog wat extra aandacht. Zo moet de vormgeving van de tijdslijn nog worden toegepast en moet een aantal functionaliteiten nog verder worden uitgewerkt om aan de eisen te voldoen. De filterfunctionaliteit is hier een voorbeeld van en ook het detailscherm was tijdens het schrijven van dit rapport nog niet operationeel. Deze worden in de komende weken nog door mij ontwikkeld.

De aanvullende specificaties hadden onder meer betrekking op performance in verband met de grote hoeveelheid data waarmee de tijdslijn te maken krijgt. In eerste instantie voldeed de tijdslijn niet aan een aantal belangrijke performance eisen. Deze problemen kwamen naar boven tijdens de stresstest en hebben gezorgd voor grote wijzigingen in de interne structuur van de tijdslijn. Dit heeft de performanceproblemen opgelost, maar heeft erg veel extra tijd gekost die niet ingecalculeerd was. Ik heb dit opgevangen door in die periode ook thuis, na werktijd en in de weekenden, aan de ontwikkeling van de tijdslijn te werken

Toekomst

Na afloop van de afstudeerperiode gaan mijn werkzaamheden bij Fabrique door. Niet meer als open-duale student, maar als fulltime webontwikkelaar. De eerste paar weken van mijn aanstelling zullen nog besteed worden aan de tijdslijn. De ontbrekende delen worden, nu ik na het opleveren van dit rapport weer tijd heb, door mij nog verder ontwikkeld.

In de weken vooraf gaande aan de zitting zal de tijdslijn nog een grote sprong doormaken in haar ontwikkeling.

Nadat de website voor het Rijksmuseum en daarmee ook de tijdslijn is opgeleverd vindt er met de alle betrokkenen een evaluatie plaats.

Tijdens deze evaluatie zal ook de definitieve tijdslijn besproken worden. Er zal worden ingegaan op de functionaliteit, usability, performance en ook op het slagen van de integratie van de tijdslijn in de overkoepelende website.

Het zal in ieder geval nog tot de maand mei of juni duren voordat de tijdslijn en de website voor publiek toegankelijk zijn. Tijdens de zitting kan ik u echter al een voorproefje geven van het uiteindelijke resultaat!

LITERATUUROPGAVE

Bell, D,

Java voor studenten
ISBN: 9043005150

Garrett, Jesse James, 2002

The elements of user experience
ISBN: 0735712026

ISO 8601, 2000

Data elements and interchange formats - Information interchange - Representation of dates and times

Craig Larman

Applying UML and patterns
ISBN: 0130925691

Elswijk, M.J.B. van, SERC

Software-patronen
<http://www.serc.nl/resources/publicaties/artikelen/Patterns.PDF>

Schmuller, J

UML in 24 uur
ISBN: 9039513449

Schramp, Jan, 2003

OOA & OOD, een bewerking van het boek Applying UML and Patterns van Craig Larman
<http://www.si.hhs.nl/~jan/OOAOD.htm>

Tolido, R.J.H., 1997

IAD - het evolutionair ontwikkelen van informatiesystemen
ISBN: 9039504016

BIJLAGEN

De bijlagen behorend bij dit afstudeerrapport zijn opgenomen in de bijlagenbundel. Deze bundel bestaat uit de volgende documenten. In deze documenten zijn een aantal onderdelen weggelaten dit ter bescherming van eigendomsrechten.

Bijlage I.	Analyse en Definitie
Bijlage II.	Analyse van bestaande tijdslijnen
Bijlage III.	Functioneel Ontwerp
Bijlage IV.	Technisch Ontwerp

‘De tijdslijn voor de nieuwe website van het Rijksmuseum’

Bijlage I

“Analyse en Definitie”

laatste wijziging:
24 maart 2004
Rick Companje

INHOUDSOPGAVE

1.	INLEIDING	3
2.	INTERNE DOELEN	4
3.	DOELGROEPEN.....	7
4.	EXTERNE DOELEN.....	8
5.	AANVULLENDE SPECIFICATIES	11
6.	USE-CASE BESCHRIJVINGEN.....	13
6.1	Scrollen door de tijd	13
6.2	Zoomen naar een ander detailniveau.....	14
6.3	Filteren op bepaalde criteria	14
6.4	Opvragen online help	14
6.5	Opvragen details over onderwerp.....	15
7.	USE-CASE MODEL	16
8.	DOMEINMODEL	17
8.1	Conceptuele klassen.....	17
8.2	Conceptueel model	19
9.	TECHNIEK SELECTIE MATRIX	20
10.	GEGEVENSANALYSE	22
10.1	Oude situatie: ADLIB en Aria	22
10.2	Nieuwe situatie: XML database	22
10.3	Benodigde gegevens voor de tijdslijn	23
11.	DATUM NOTATIES	25
11.1	Oude situatie	25
11.2	Datumnotaties volgens standaarden	26

1. INLEIDING

In de fase analyse en definitie uit de standaardfasering van Fabrique wordt de reikwijdte van het project onderzocht en afgebakend. Er wordt gekeken naar de doelstellingen, doelgroepen, inhoud, vorm, techniek, beheer en organisatie van het project. De gevonden randvoorwaarden, eisen en wensen worden vastgelegd.

Dit document is een bijeenraping van resultaten van bovengenoemde activiteiten.

2. INTERNE DOELEN

De externe projectorganisatie heeft een strategisch plan opgesteld waarin de visie en plannen van het Rijksmuseum voor de komende jaren staan vermeld. Dit rapport heet "Internet strategie 2004/2008".

Onder andere op basis van dit rapport zijn een aantal interne doelen bepaald. De interne doelen zijn de doelen die het Rijksmuseum met de tijdslijn voor ogen had.

Meer interne doelen zijn verkregen uit de gesprekken die wekelijks gevoerd werden tussen de interne en externe projectorganisaties.

Interne doelen (site objectives)	Haalbaarheid	Opmerking
1. De Tijdsbalk moet zijn toegespitst op de 17e Eeuw, de Philipsvleugel Presentatie.	hoog	Er wordt zelfs een stap verder gegaan. De tijdslijn wordt gevoed door selecties die door de redacteurs samengesteld worden. Deze selecties zullen meestal gaan over een bepaald thema, in dit geval de 17e eeuw
2. Verbanden tussen elementen op de tijdslijn moeten zichtbaar zijn	hoog	De informatie op de tijdslijn wordt chronologisch horizontaal wordt geordend. Gelijktijdige gebeurtenissen en objecten worden onder elkaar getoond. Verbanden in tijd zijn op deze manier duidelijk zichtbaar. Andere verbanden (bijv. tussen kunstenaar en leermeester) lijken lastiger en zullen een lagere prioriteit krijgen.
3. De gehele website dus ook de tijdslijn moet State-of-the-art zijn	hoog	Er wordt gebruik gemaakt van de nieuwste technieken
4. Goede interactiemogelijkheden voor gebruikers	hoog	Muis en toetsenbord zullen worden ondersteund
5. Gebruikersvriendelijkheid moet hoog zijn	hoog	bij de ontwikkeling van de tijdslijn wordt erg gelet op hoe de gebruikers zo efficiënt, effectief en naar tevredenheid de tijdslijn kunnen gebruiken. Het plan is een usabilitytest te organiseren waarbij het eerste prototype wordt getest.
6. Het verbreden en verdiepen van zowel zoekmachine-achtige als scenario-gebaseerde mogelijkheden van ontsluiting van de collectie		scenariobased: scrollen en zoomen door de tijd searchbased: filters verfijnen, zoeken op steekwoorden
7. De Tijdsbalk eventueel inzetten voor de Geschiedenis van het Gebouw/de Verbouwing.)	hoog	zie 1
8. Wenselijk om de tijdslijn printbaar laten zijn	lastig	Minder hoge prioriteit, wordt zeer waarschijnlijk niet uitgevoerd in deze versie
9. De tijdslijn moet inspireren, leren, verleiden en fysiek bezoek aan het museum stimuleren	hoog	De tijdslijn zal een schat aan cultuurhistorische informatie op een amuserende wijze in de tijd plaatsen.

10. De tijdslijn zal tegelijkertijd op meerdere plaatsen op de website worden ingezet om informatie over verschillende thema's weer te geven.	hoog	Om dit te bereiken zal de tijdslijn een component (format) zijn die zich in de presentatielaag bevindt en gegevens put uit verschillende informatiebronnen.
11. Er moet zoveel mogelijk bezoek worden gegenereerd voor de tijdens de verbouwing geopende Philipsvleugel.	hoog	zie 1
12. De bezoeker moet een idee krijgen waarom de Gouden Eeuw zo'n bijzondere periode was.	hoog	Ook voor het weergeven van informatie over dit thema zal de tijdslijn worden ingezet met als gegevens bron de selectie gegevens over de gouden eeuw
13. De door de tijdslijn weer te geven informatie moet door de contentmanagers (beheerders van de inhoud van de website) zelfstandig kunnen worden onderhouden.	hoog	de content zal met het content management systeem XOPUS door de redacteurs zelf worden onderhouden
14. De vormgeving van de tijdslijn zal moeten aansluiten bij die van de rest van de nieuwe website.	hoog	De vormgeving van de tijdslijn zal worden bepaald wanneer er voldoende zekerheid bestaan over de vormgeving van de overkoepelende site.
15. Thema's zullen betrekking hebben op gebeurtenissen in het verleden die verschillen in lengte van tijd. De tijdslijn moet in staat zijn dit soort informatie weer te kunnen geven.	hoog	Het begin, eind en lengte van weer te geven periodes zal volledig variabel zijn. Ook zullen intervallen en subintervallen (hoeveelheid stapjes tussen eeuwen, decennia, jaren etc.
16. Er is behoefte aan informatie op verschillende detailniveau's. De tijdslijn moet gebeurtenissen en objecten in de tijd van globaal tot zeer gedetailleerd kunnen weergeven	hoog	de tijdslijn wordt uitgerust met verschillende zoomlevels die variabel zullen zijn. Afhankelijk van de hoeveelheid informatie die op één scherm getoond wordt, wordt bepaald hoe gedetailleerd een object of periode gepresenteerd wordt.
17. De 45-minute tourist is een zeer belangrijke doelgroep en moet bediend worden. Aangezien deze doelgroep de Nederlandse taal niet beheerst zal de tijdslijn in meerdere talen aangeboden worden.	hoog	De tijdslijn zal de talen Nederlands en Engels ondersteunen. Er wordt rekening gehouden met meerdere talen in de toekomst.
18. De bezoeker wil de collectie van het Rijksmuseum goed kunnen doorzoeken. De tijdslijn moet hierbij een ondersteunende rol vervullen door de mogelijkheid te bieden te kunnen navigeren door (een gedeelte van) de collectie.	hoog	navigeren door de tijd, maar ook het verfijnen van de selectie door filters toe te passen. En eventueel het zoeken op steekwoorden.

19. Alle formats, dus ook de tijdslijn zo maken dat ze op meerdere objecten en met meerdere presentaties kunnen werken.	hoog	zie 10
---	------	--------

3. DOELGROEPEN

In het rapport *Internet strategie 2004/2008* stond vermeld dat de nieuwe website voor het Rijksmuseum zich richt op de volgende doelgroepen:

- Buitenlandse/Nederlandse eenmalige bezoeker
- Potentiële herhalingsbezoeker Buitenlands/Nederlands
- Gezin
- Kinderen/Jongeren
- Onderwijs
- Wetenschap
- Zakelijk (in mindere mate)

Omdat de tijdslijn een onderdeel is van de overkoepelende website krijgt deze met dezelfde gebruikersgroepen te maken.

Het onderstaande matrix model is eveneens afkomstig uit het bovengenoemde rapport. De verschillende doelgroepen zijn uitgezet tegen verschillende onderdelen van de website.

	1e keer toeristen	Herhaling	Gezin	Wetenschap	Onderwijs	Jongeren	Zakelijk
Homepage	x	x	x	x	x	x	
Bezoekersinformatie plan your visit	x	x	x		x	x	
Nieuws		x		x	x		
De collectie		x		x	x		
Het Rijksmuseum is overall		x		x			
Evenementen	x	x	x		x	x	
De geschiedenis van het gebouw/ de verbouwing	x	x	x	x			
Onderzoek en wetenschap/ Wetenschappelijk magazine/		x		x	x		
Bibliotheek		x		x	x		
Jeugd			x			x	
Onderwijs			x		x		
Over het Rijksmuseum				x			x
Metazoekpagina		x		x	x		
Abonneren/Digitale Nieuwsbrief		x		x	x		x
E-card	x	x	x			x	

4. EXTERNE DOELEN

De externe doelen ook wel aangeduid met gebruikersbehoeften, heb ik vastgesteld door me te verplaatsen in de verschillende gebruikersgroepen en na te gaan welke externe doelen de uiteindelijke gebruikers zouden willen bereiken met de tijdslijn.

Gebruikersbehoeften	Haalbaarheid	Opmerking
1. ik wil alleen kunstenaars zien	hoog	kunstenaars worden op de tijdslijn opgenomen. Het verbergen van andere informatie kan met de filters worden gedaan
2. ik wil alleen objecten uit de prehistorie	zeer laag	niet in collectie
3. ik wil bij 0 beginnen en alles langslopen tot 2003	hoog	Er moet chronologisch langs de collectie gelopen kunnen worden.
4. ik wil snel weten wanneer de gouden eeuw was	hoog	gouden eeuw zal als periode uit de geschiedenis getoond worden
5. ik wil alles in het engels zien	hoog	taal van de tijdslijn is afhankelijk van de taal waarin de site op dat moment getoond wordt
6. ik wil in één keer zien wat er allemaal is (totaal uitgezoomed)	hoog	zoomfunctionaliteit zal beschikbaar zijn. In het meest uitgezoomde niveau is er echter geen mogelijkheid alle informatie gedetailleerd te tonen, daarom zullen er aggregaties getoond worden.
7. ik wil de belangrijkste dingen uit de geschiedenis zien	niet	relevantie van objecten en kunstenaars wordt niet bijgehouden in het systeem en zal ook niet automatisch berekend worden
8. ik wil combinaties van de bovenstaande	hoog	gebruikerselecties/filters moeten gecombineerd kunnen worden
9. ik wil uitgezoomed beginnen, een periode zoeken en dan inzoomen	hoog	zoomfunctionaliteit zal beschikbaar zijn
10. ik wil weten welke tijdperken er zijn geweest	hoog	tijdperken niet, maar wel tijdperiodes. Aan te geven als periodes in de laag geschiedenis
11. ik wil zien welke objecten uit jaar x zijn	hoog	inzoomen op een bepaald jaar
12. ik wil weten welke kunstenaars tegelijk leefden	hoog	deze informatie is zichtbaar doordat het leven van een kunstenaar als periode wordt weergegeven. Kunstenaars die tegelijkertijd leefden worden onder elkaar weergegeven waarbij de overlap zichtbaar is
13. ik wil snel van 0 naar 2000 zonder te hoeven uitzoemen	hoog	er wordt een 'scrollbalk' toegevoegd die de totale tijdsperiode toont en waarmee direct naar een bepaald jaar gesprongen kan worden
14. ik wil weten wie de maker is van een bepaald kunstwerk	hoog	bij een muisklik op een kunstobject wordt een detailscherm getoond. Hierin wordt ook de naam van de maker vermeld en de mogelijkheid geboden door te klikken.
15. ik wil alles weten over de maker van dit object	hoog	in de beschrijving van een object moet doorggeklikt kunnen worden naar de maker. Deze moet 'oplichten' in de tijdslijn waarna de gebruiker zal kunnen klikken op de maker zodat het detailscherm over de maker getoond wordt.
16. ik wil details over dit object	hoog	er zal in de vorm van een popup een detailscherm getoond worden

17. ik wil en beschrijving over een stijlperiode	hoog	zal tijdens de fase redactie moeten worden ingevoerd.
18. ik wil vanuit de beschrijving van een kunstenaar kunnen doorklikken naar zijn leermeester	niet	de detailbeschrijving zal geen hotspots bevatten.
19. ik wil weten hoeveel objecten er in totaal zijn	hoog	het is wel haalbaar om aan te geven bij een selectie of filter wat de hoeveel resultaten is, maar het is niet zeker of deze informatie relevant genoeg is.
20. ik wil dat ik meer zie over kunstenaars wanneer objecten, periodes en gebeurtenissen onzichtbaar zijn	hoog	reële wens, technisch haalbaar. Kan via een filter wordt of direct in de interface met een in- of uitklappijltje zal worden uitgevoerd. Nog niet bekend hoe kunstenaars worden getoond wanneer er meer ruimte beschikbaar is
21. ik wil zoeken op steekwoorden	hoog	reële wens, technisch haalbaar om bij de filter (of apart zoekvak) te zoeken op een steekwoord, en dan direct te scrollen naar het gevonden item.
22. ik wil kunnen filteren op steekwoorden	niet	er zullen vooraf bepaalde vaste filters worden aangeboden
23. ik wil zoeken binnen de resultaten ofwel de filter specifieker maken	hoog	bij de filter zullen meerdere argumenten opgegeven kunnen worden
24. ik wil een zoekwoord kunnen intypen	hoog	zie 21
25. ik wil plaatjes kunnen zien van voorwerpen	hoog	kunstobjecten worden getoond met afbeelding. Kunstobjecten die geen afbeelding hebben worden niet getoond op de tijdslijn.
26. ik wil dat het rijksmuseum van tevoren een thema kiest waar de tijdslijn over zal gaan.	hoog	Het Rijksmuseum zal de tijdslijn gaan inzetten ter verrijking van de thema's of onderwerpen die op hun site worden aangeboden. Hiervoor zullen zij vooraf selecties maken.
27. ik wil vanuit zo'n thema de filter verkleinen waardoor ik meer krijg te zien van de hele collectie	hoog	de tijdslijn wordt uitgevoerd met een filters die de bezoeker kan instellen.
28. als ik klik op een voorwerp wil ik eerst een korte tekst en evt daarna meer details	niet	Een klik zal direct alle detail informatie over het object laten zien. Wel worden er doorkliks opgenomen.
29. ik wil kunnen schuiven over de tijdslijn met een soort van target	hoog	de scrollbar wordt met een zoeker/dragger uitgevoerd.
30. ik wil het geboel hebben dat ik een zwiep aan de tijdbalk kan geven	hoog	de tijdslijn wordt voorzien van een aantal transities die het navigeren een 'beter gevoel' geven
31. ik wil precies een jaartal kunnen intypen	niet	via de scrollbalk kan naar een jaar genavigeerd worden
32. ik wil kunnen zien wanneer periodes elkaar overlappen	hoog	overlappende periodes worden onder elkaar getoond
33. ik wil dat de tijdslijn soepel loopt, zodat ik niet steeds moet wachten tot alles geladen is en dat ik daarna pas verder kan.	hoog	Er zal worden gebufferd. Afbeeldingen worden alvast geladen wanneer ze in beeld dreigen te komen. Ook tijdens het bufferen kan er gescrolled worden
34. Ik wil niet dat als er in een bepaald jaar geen objecten zijn dat dan m'n scherm leeg is	lastig	is een reële wens, lastig haalbaar. Een redacteur zal rustige periodes minder ruimte geven op de tijdslijn. Uit te sluiten valt het niet. Wanneer een gebruiker een filter in stelt kan het scherm als nog

		leeg worden.
35. Ik wil altijd een x aantal objecten op m'n scherm zien	niet	niet relevant
36. Ik wil dat de verhouding tussen periodes op de tijdslijn klopt	niet	De tijdslijn zal juist uitgerekte worden op drukke plaatsen en zal smaller zijn op rustige plaatsen
37. ik wil met het toetsenbord kunnen navigeren	hoog	het toetsenbord wordt ondersteund. Er zal in een inline/online hulpvenster worden uitgelegd hoe het toetsenbord gebruikt kan worden om optimaal te navigeren
38. ik wil alles met de muis kunnen doen	hoog	de muis wordt ondersteund. Er zal in een inline/online hulpvenster worden uitgelegd hoe de muis gebruikt kan worden om te navigeren. Bepaalde tekstinput (zoals trefwoorden) moeten met het toetsenbord ingevoerd worden.
39. er moet een help aanwezig zijn voor wanneer ik het niet snap of alle functionaliteiten optimaal wil benutten	hoog	de tijdslijn wordt voorzien van een inline of online help.

5. AANVULLENDE SPECIFICATIES

De aanvullende specificaties, ook wel niet-functionele eisen genoemd worden moeten meetbaar en helder beschreven worden. Ten eerste om geen verwarring te laten ontstaan tijdens het ontwikkelen en ten tweede omdat de specificaties tijdens het testen worden gebruikt als meetcriteria.

Wat zijn de minimum eisen die gesteld worden aan het computersysteem van de gebruiker?

De tijdslijn moet op een PC met 450Mhz (of computer met vergelijkbare snelheid) met een beschikbare geheugencapaciteit van 128MB gebruikt kunnen worden. Aanbevolen wordt echter een computer met de dubbele snelheid en geheugencapaciteit.

Wat is de minimumsnelheid van de internetverbinding van de gebruiker om de tijdslijn te kunnen gebruiken?

De tijdslijn moet gebruikt kunnen worden met een 56K modem. Aanbevolen wordt een ADSL of kabel verbinding.

Hoeveel mensen moeten de tijdslijn gelijktijdig kunnen gebruiken?

Het maximaal aantal gelijktijdige verbindingen hangt af van het aantal dat gekozen wordt voor de website. Dit aantal zal liggen rond honderd gelijktijdige bezoekers.

Mag de tijdslijn trager worden al naar gelang het aantal gelijktijdige gebruikers toeneemt?

De tijdslijn mag de webserver niet merkbaar vertragen.

Mag de gebruiker merken dat er data wordt geladen of moet het allemaal even soepel blijven lopen

Ja, er wordt bij het laden van de tijdslijn gebufferd. Hierbij zal de gebruiker merken dat de tijdslijn wordt geladen met data. Vervolg acties zoals scrollen en zoomen zullen

Vanaf welke laadtijden iets als een progressiebalk zichtbaar moet zijn?

Bij laadtijden waarbij de tijdslijn langer dan 3 seconden geen beweging of progressie vertoont dient een progressiebalk of statusvenster zichtbaar te zijn.

Mag de tijdslijn een browser plug-in vereisen?

Ja, mits dit een gangbare plug-in is zoals Flash, Java of Shockwave. Indien de plug-in niet aanwezig is op de computer van de gebruiker dient de gebruiker de mogelijkheid geboden te worden de plug-in zonder omslachtige handelingen te kunnen installeren.

Welke responsetijden worden er van de tijdslijn verwacht?

De tijdslijn zal bij navigatie door de gebruiker direct feedback moeten geven. Hetzij een beweging in gang zettend, hetzij door het tonen van een melding dat er iets gaat gebeuren.

Met hoeveel data moet de tijdslijn om kunnen gaan?

Er is een databron met 400.000 records. De tijdslijn moet in staat zijn al deze data te kunnen verwerken, maar niet gelijktijdig. Er worden thema's gedefinieerd waarin dataselecties worden gemaakt. Deze dataselecties beperking zich tot aantal tot 4000 items. Een dataselectie van deze grote moet de tijdslijn kunnen verwerken zonder dat daarbij aanzienlijk merkbare vertragingen optreden.

Welke eisen worden gesteld ten aanzien van de stabiliteit?

De tijdslijn mag er niet voor zorgen dat het systeem instabiel wordt. Het client-sided omgaan met grote hoeveelheden data kan problemen veroorzaken op dit gebied. Dit moet ten alle tijden voorkomen worden.

Welke eisen worden gesteld ten aanzien van voorspelbaarheid

De tijdslijn moet zich voorspelbaar gedragen. Gebruikers die enkele minuten met de tijdslijn hebben gewerkt moeten het gedrag van een volgende actie kunnen voorspellen.

Welke eisen worden gesteld ten aanzien van de onderhoudbaarheid?

Er moeten dataselecties gemaakt kunnen worden. Het realiseren van de techniek hiervoor valt buiten de opdracht.

De tijdslijn zelf moet ook onderhouden kunnen worden door de broncode op te leveren bij de implementatie. De tijdslijn dient volgens object oriëntatie standaarden te zijn opgezet.

Welke eisen worden gesteld ten aanzien van internationalisatie en meertaligheid?

De tijdslijn dient ook voor internationale bezoekers toegankelijk te zijn. De tijdslijn moet de talen Nederlands en Engels ondersteunen.

Van welke documentatie wordt de tijdslijn voorzien?

Er wordt een technische handleiding opgeleverd waarin de technische structuur van de tijdslijn wordt beschreven.

Daarnaast moet een inline gebruikershandleiding in de tijdslijn verwerkt.

Welke eisen worden gesteld ten aanzien van packaging?

De tijdslijn wordt door Fabrique en Q42 tijdens de implementatie op de website geïnstalleerd. Het Rijksmuseum krijgt een CD-ROM met de bronbestanden.

Zijn er licenties of andere juridische kwesties?

De tijdslijn wordt ontwikkeld met een ontwikkelomgeving waarvoor licenties verkregen zijn. Voor het opvragen van de tijdslijn zijn geen licenties nodig. Ook voor het installeren van de tijdslijn op de webserver zijn geen extra licenties vereist.

Wat is de maximale grootte van de gecompileerde tijdslijn?

De maximum grootte van de tijdslijn bedraagt 100KB. Dit is inclusief interface elementen maar exclusief te benaderen resources en data.

Hoeveel dataverkeer mag de tijdslijn genereren?

Deze hoeveelheid is afhankelijk van de tijdsduur dat de tijdslijn gebruikt wordt en de manier waarop (veel scrollen zal veel dataverkeer veroorzaken). Bij normaal gebruik mag de tijdslijn niet meer dan 1MB per minuut aan dataverkeer opleveren.

Welke eisen worden gesteld aan caching?

Dataselecties zullen worden gecached op de server zodat er niet bij iedere aanvraag een complexe query uitgevoerd hoeft te worden.

Clientsided worden ID's, jaartallen en namen van de totale dataselectie gecached.

Afbeeldingen worden geprebufferd en gecached, maar zodra ze buiten een te configureren range vallen worden ze weer verwijderd.

6. USE-CASE BESCHRIJVINGEN

Use-cases beschrijven tekstueel hoe een systeem wordt gebruikt om de doelstellingen te bereiken.

Een use-case is een collectie van gerelateerde succesvolle en mislukkende scenario's. Scenario's beschrijven een specifieke serie acties en interacties tussen een actor en het systeem. Een actor is een rol die uitgevoerd wordt door een persoon of ander systeem.

Voor alle use-case beschrijvingen zijn de volgende aannames van toepassing:

- De tijdslijn is goed geconfigureerd
- De tijdslijn is geladen
- De periodes binnen de ElasticRange zijn bekend (begin, eind, lengtes en intervallen van/tussen periodes op de tijdslijn)
- Het systeem bevat inhoud
- De internetverbinding van de gebruiker voldoet aan de minimale specificaties zoals is beschreven in de systeemeisen en is tijdens het gebruik van het systeem actief.
- De computer van de gebruiker voldoet aan de minimale specificaties zoals is beschreven in de systeemeisen.

6.1 Scrollen door de tijd

Naam	Scrollen door de tijd
Samenvatting	Er wordt een grote of kleine sprong door de tijd gemaakt.
Actoren	Gebruiker van de tijdslijn
Extra aannames	-
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker maakt met behulp van toetsenbord of muis een kleine of grote sprong in de tijd. 2. Het systeem controleert of er niet eerder dan [begin van de tijdslijn] of verder dan het [eind van de tijdslijn] wordt gegaan. 3. Er worden nieuwe gegevens getoond. 4. Afhankelijk van de grootte van de sprong worden extra gegevens/afbeeldingen bij geladen. Een kleine sprong zorgt voor buffering van het gebied dat in beeld dreigt te komen. Een grote sprong zal gegevens/afbeeldingen laden van het nieuwe zichtbare gebied. 5. Het systeem controleert of [te snel scrollen in kleine stapjes] plaatsvindt.
Uitzonderingen	<ol style="list-style-type: none"> a. [begin van de tijdslijn] de gebruiker probeert terug te gaan in de tijd, maar staat al aan het begin van de tijdslijn. b. [eind van de tijdslijn] de gebruiker probeert verder te gaan in de tijd, maar staat al aan het eind van de tijdslijn. c. [te snel scrollen in kleine stapjes] moet leiden tot het niet laden van gegevens/afbeeldingen die toch weer erg snel uit beeld verdwijnen.
Resultaat	Er is een sprong gemaakt in de tijd. De nieuwe gegevens zijn zichtbaar gemaakt of 'druppelen' nog langzaam binnen.

6.2 Zoomen naar een ander detailniveau

Naam	Zoomen naar een ander detailniveau
Samenvatting	De tijdslijn toont de inhoud op een globaler of gedetailleerder niveau. De hoeveelheid tegelijkertijd zichtbare gegevens verschilt.
Actoren	Gebruiker van de tijdslijn
Extra aannames	- Het aantal zoomlevels is bekend
Beschrijving	<ol style="list-style-type: none"> 1. Het systeem controleert op uitzonderingen [a. meest gedetailleerde zoomlevel] en [b. meest globale zoomlevel] 2. Op basis van de timeRange behorend bij dit zoomlevel wordt de schaal van de tijdslijn en de intervallen berekend en getoond. 3. Er wordt bepaald welke objecten nog/weer zichtbaar moeten zijn en welke verborgen kunnen worden. 4. Alle zichtbare gegevens worden hergepositioneerd en veranderen eventueel van verschijningsvorm indien dit voor dit zoomlevel nodig is.
Uitzonderingen	<ol style="list-style-type: none"> a. [meest gedetailleerde zoomlevel] is reeds actief. Er zal geen actie uitgevoerd worden b. [meest globale zoomlevel] is reeds actief. Er zal geen actie uitgevoerd worden
Resultaat	De tijdslijn toont de inhoud op een globaler of gedetailleerder niveau. De hoeveelheid tegelijkertijd zichtbare gegevens verschilt.

6.3 Filteren op bepaalde criteria

Naam	Filteren op bepaalde criteria
Samenvatting	De filter wordt verfijnd of verbreed
Actoren	Gebruiker van de tijdslijn
Extra aannames	- Filteren is mogelijk in deze configuratie
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker kan één of meerdere filters selecteren 2. De gebruiker kan indien gewenst zoeken op steekwoord 3. De gekozen filter(s) worden toegepast 4. Er wordt bepaald welke gegevens op de tijdslijn zichtbaar moeten/mogen zijn.
Uitzonderingen	<ol style="list-style-type: none"> a. [maximale hoeveelheid filters] is ingesteld. Er zal geen actie uitgevoerd worden b. [filter wordt niet gewijzigd]
Resultaat	Er is een filter toegepast waardoor meer of minder gegevens op de tijdslijn zichtbaar zijn.

6.4 Opvragen online help

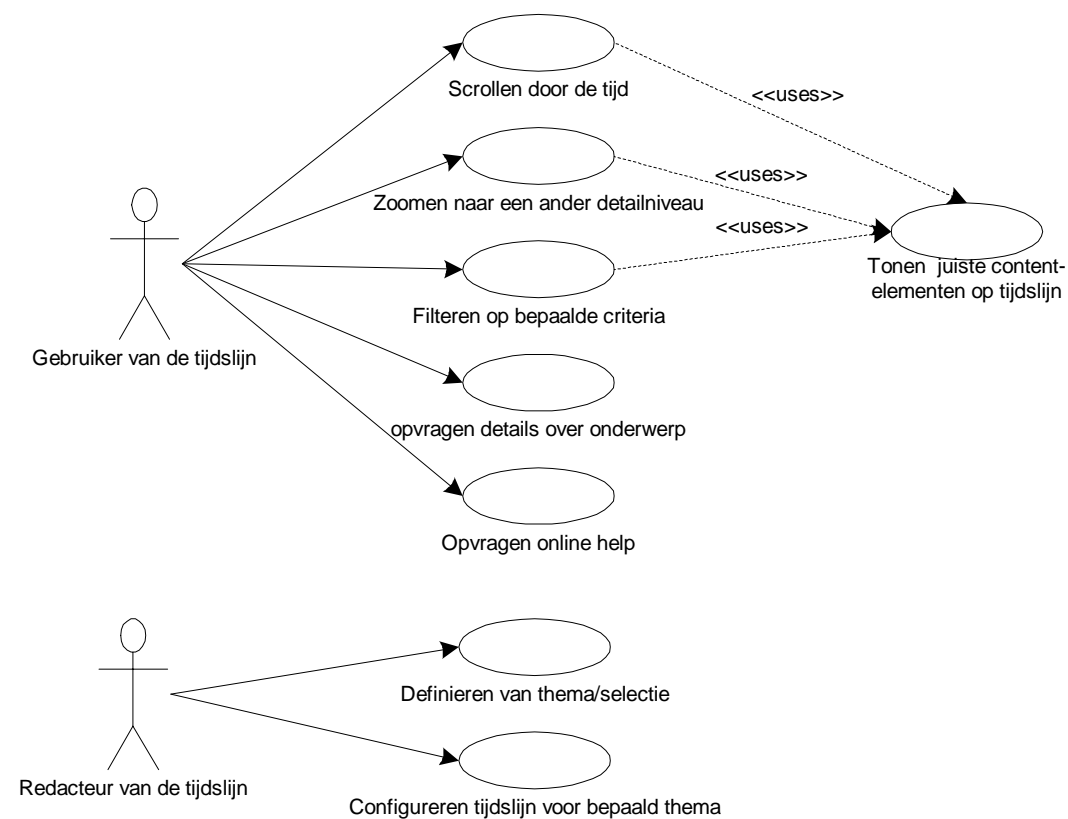
Naam	Opvragen online help
Samenvatting	De gebruiker vraagt het systeem om hulp
Actoren	Gebruiker van de tijdslijn
Extra aannames	- er zijn helpteksten aangemaakt in twee talen
Beschrijving	<ol style="list-style-type: none"> 1. De gebruiker vraagt de online help op over het gebruik van de tijdslijn 2. De online help toont informatie over het gebruik van de interface-elementen en over de content-elementen. 3. Indien de online help uit meerdere pagina's bestaat wordt de gebruiker binnen de online help navigatie aangeboden.
Uitzonderingen	
Resultaat	De gebruiker heeft de informatie gevonden waarnaar hij zocht of zijn vraag is niet beantwoord.

6.5 Opvragen details over onderwerp

Naam	Opvragen details over onderwerp
Samenvatting	De gebruiker vraagt gedetailleerde informatie op over een bepaald item op de tijdslijn
Actoren	Gebruiker van de tijdslijn
Extra aannames	-
Beschrijving	<ol style="list-style-type: none"> 1. Bij het opvragen van details zal een detailscherm getoond worden, items op de tijdslijn zullen dan niet te benaderen zijn. 2. Indien beschikbaar wordt de details getoond over dit onderwerp. (titel, jaartal/tijdsduur, beschrijving, afbeelding) zo niet, dan uitzondering [a. geen details beschikbaar] 3. Vanuit de detailinformatie kan genavigeerd worden naar een aantal gerelateerde items op de tijdslijn. 4. Na het sluiten van de detailinformatie worden de elementen tijdslijn weer zichtbaar.
Uitzonderingen	a. [geen details beschikbaar] dan wordt daarvan melding gedaan
Resultaat	De gebruiker heeft de informatie gevonden waarnaar hij zocht.

7. USE-CASE MODEL

Hoewel bij use-case de tekstuele beschrijvingen voorop staan heb ik voor het overzicht nog een use-case-model gemaakt die hieronder is weergegeven.



8. DOMEINMODEL

Object georiënteerde ontwikkeling gaat uit van domeinmodellering. Hierbij wordt de werkelijkheid waarin het systeem een rol speelt (het domein) in kaart gebracht. Dit levert een domeinmodel op. Hierin staan de concepten uit het domein met hun kenmerken, gedrag en onderlinge relaties.

Het domeinmodel is het belangrijkste resultaat van de analysefase. Het is een visuele representatie van conceptuele klassen van domeinobjecten uit echte wereld.

Het domeinmodel wordt weergegeven als een klassendiagram en bevat reeds attributen (weet-verantwoordelijkheden), associaties en multipliciteiten bij de associaties. De methoden van de klassen (de doe-verantwoordelijkheden) zijn nog niet in dit diagram opgenomen.

8.1 Conceptuele klassen

Om de conceptuele klassen voor de tijdslijn te vinden heb ik gezocht naar zelfstandige naamwoorden binnen de geformuleerde systeemeisen en de use-case beschrijvingen.

- website
- gebruiker
- redacteur
- tijdslijn
- kunstenaar / artist
- kunstobject / art object
- periode
- stijlperiode / style period
- geschiedenis periode / history period
- gebeurtenis / geschiedenis gebeurtenis / history event
- filter
- detailniveau / zoomlevel
- detail informatie / detailscherm
- help informatie / helpscherm
- thema
- selectie / dataselectie
- content / inhoud / content-elementen
- verplaatsing / scrollfunctionaliteit
- zoekactie / zoekfunctionaliteit
- muis
- toetsenbord
- webserver
- database
- beginpunt tijdslijn
- eindpunt tijdslijn
- tijdvak / timeRange
- meertaligheid
- geluidseffecten
- transitities
- kleurgebruik
- afbeelding / foto
- stip / target
- periodebalk
- verbanden
- Xopus
- configuratie
- internetverbinding

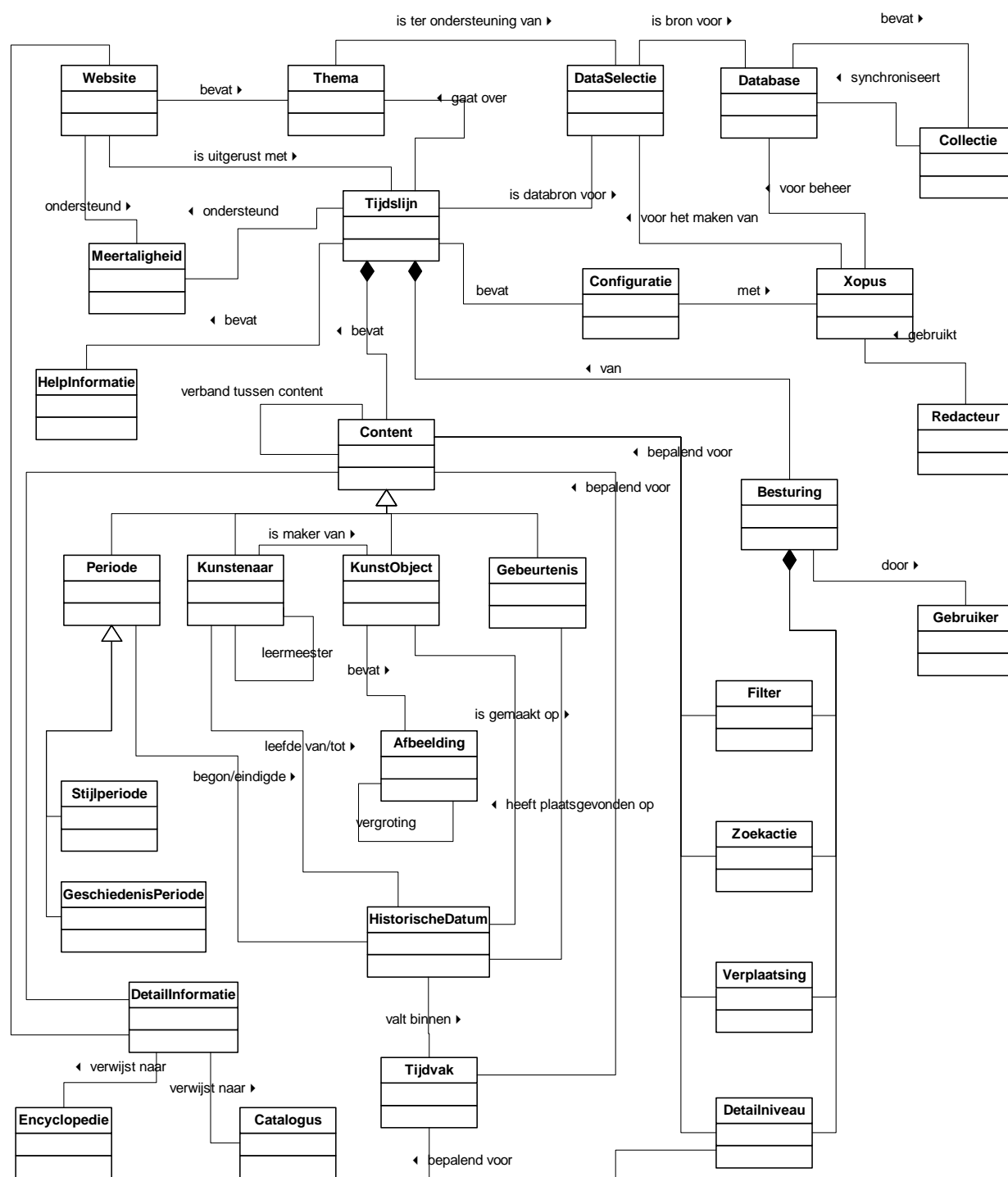
- computer
- jaartal
- decennium
- eeuw
- collectie
- vergroting
- encyclopedie
- leermeester
- catalogus
- besturing / userinterface

Afgevalen conceptuele klassen

- beheer, viel in één met configuratie, en maken van selecties
- leermeester werd associatie van kunstenaar class naar zichzelf
- beginpunt en eindpunt werden attributen van tijdvak
- verbanden werd associatie van content class naar zichzelf
- computer, ICT voorzieningen op dit moment minder relevant
- jaar, eeuw, decennium hoort bij HistorischeDatum
- periode en stip, te veel interface spec.
- vergroting werd associatie van afbeelding class naar zichzelf
- webserver, ICT voorzieningen op dit moment minder relevant
- internetverbinding, ICT voorzieningen op dit moment minder relevant
- geluidseffect, te veel interface spec.
- transitie, te veel interface spec.
- muis, te veel interface spec.
- toetsenbord, te veel interface spec.
- kleurgebruik, te veel interface spec.

8.2 Conceptueel model

In het onderstaande conceptuele model zijn de conceptuele klassen weergegeven van de tijdslijn met hun associaties. Attributen van de klassen en multipliciteit van de associaties zijn op dit moment nog buitenbeschouwing gelaten.



9. TECHNIEK SELECTIE MATRIX

In deze techniek selectie matrix worden de meest voor de handliggende ontwikkel-omgevingen tegen elkaar uitgezet en de **voordelen van A ten opzichte van B** in kaart gebracht.

B \ A	DHTML	JAVA APPLET	SHOCK WAVE	FLASH 5	FLASH 6	FLASH 7
DHTML		geen plugin vereist	geen plugin vereist	geen plugin vereist externe jpg's	geen plugin vereist	geen plugin vereist
JAVA APPLET	sneller in renderen & rekenen sneller in volledig OO browser en OS onafhankelijk		?	volledig OO externe jpg's	volledig OO	?
SHOCK WAVE	grafisch sterker	grafisch sterker		betere rendering	?	?
FLASH 5	grafisch sterker	grafisch sterker	plug-in gangbaar en kleiner		plug-in gangbaar en kleiner	plug-in gangbaar en kleiner
FLASH 6	grafisch sterker	grafisch sterker	?	externe jpg's betere XML ondersteuning		plug-in gangbaar en kleiner
FLASH 7	grafisch sterker volledig OO	grafisch sterker	volledig OO	volledig OO externe jpg's betere XML ondersteuning	volledig OO	

De volgende technieken zijn met elkaar vergeleken: DHTML, Java (Applets), Macromedia Shockwave, Flash 5, Flash 6 en Flash 7.

Al snel leek Flash 7 het meest in aanmerking te komen. In Flash 7 kon object georiënteerd geprogrammeerd worden, dat kon ook in Java, maar bij Java was de algehele beschikbaarheid van de browser plug-in een belemmerende factor. De Flash plug-ins worden tegenwoordig volledig geaccepteerd door de gemiddelde website bezoeker en ook de gemakkelijke overstap naar een nieuwe versie van de Flash plug-in is positief.

Macromedia Shockwave kwam in eerste instantie ook in aanmerking wegens de goede rekenkracht bij het renderen van afbeeldingen. Maar net als bij Java was ook hier weer de plug-in de belemmerende factor. Deze is vele malen groter dan de Flash plugin.

Het verschil tussen de verschillende Flash versies zit vooral in het object georiënteerd kunnen programmeren, performance, het scheiden van de vormgeving en de code en het al dan niet ondersteunen van externe afbeeldingen.

De Flash 5 plug-in is zeer goed geaccepteerd bij de gemiddelde website bezoeker, maar viel al snel af wegens de slechte performance bij grote hoeveelheden data, en het niet ondersteunen van externe afbeeldingen.

De Flash 6 plug-in heeft al veel voordelen ten opzichte van de vorige versie maar wordt door de snelle opvolging door Flash 7 vaak gezien als voorproefje op de echte 'volwassen' Flash 7. Flash 7, ook wel Flash MX 2004 genoemd, is uitgerust met ActionScrip 2.0, een object georiënteerde programmeertaal die gebaseerd is op de ECMA-262 v4 standaard.

10. GEGEVENSANALYSE

Tijdens de fase analyse en definitie zijn de te raadplegen databronnen geanalyseerd.

10.1 Oude situatie: ADLIB en Aria

Er werd bij het Rijksmuseum gewerkt met een tweetal databases. De belangrijkste database voor het Rijksmuseum was de ADLIB Museum database. ADLIB Museum is een commerciële applicatie die gemaakt is voor collectie- en informatiebeheer in musea.

De tweede database waarmee gewerkt werd was de Aria database. De Aria database was een restant van een eerder informatiesysteem die onder andere gebruikt werd voor informatiezuilen binnen het museum en voor de oude website. Deze database bevatte gegevens die gebruikt werden ter verrijking van de ADLIB database. Een groot aantal informatiepagina's over kunstobjecten en kunstenaars kwam uit dit systeem.

De informatie uit de Aria database is geconverteerd en is samen met de gegevens uit de ADLIB database gemigreerd in een nieuwe XML Database.

Het Aria-systeem is hierdoor na de implementatie van de nieuwe website overbodig en wordt als zodanig opgeheven.

De ADLIB database blijft bestaan en wordt op gezette tijden gesynchroniseerd met de XML Database.

10.2 Nieuwe situatie: XML database

Q42 heeft voor de XML Database van het Rijksmuseum gebruik gemaakt van Oracle versie 9. De XML Database is opgedeeld in een aantal spaces. Spaces zijn te vergelijken met tabellen uit een relationele database. Een aantal spaces in de nieuwe database zijn artists, artobjects, screens, catalogs, dictionary en encyclopedia. Een space bestaat uit een collectie van XML documenten met per document een eigen unieke ID. Alle ID's binnen een space moeten uniek zijn, daarbuiten mogen dubbelingen voorkomen.

De XML documenten binnen een collectie zouden als records beschouwd kunnen worden met een ID als primaire sleutel. Een groot verschil is echter tussen een tabel en space dat een tabel een vaste definitie heeft met een vooraf bepaald aantal datavelden terwijl binnen de spaces vrij genest kan worden. Ook kunnen er referenties opgenomen worden naar documenten in andere spaces.

Het grote voordeel van het kunnen nesten is dat de database niet zal gaan bestaan uit talloze koppeltabellen zoals dat het geval zou zijn bij een relationele database. Een ander groot voordeel is dat er binnen spaces referenties naar elkaar opgenomen kunnen worden waardoor het bevragen van een XML database minder complexe queries vereist.

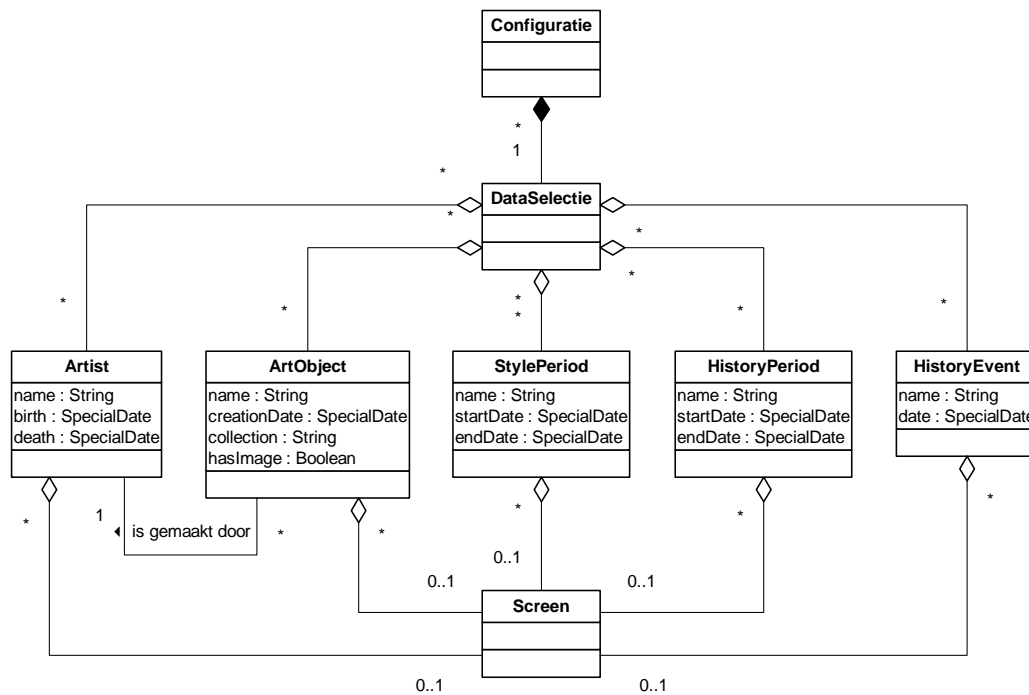
Een XML database kent net als een relationele database indexen waardoor snel in de database gezocht kan worden. Bij een relationele database worden indexen aangemaakt op tabelvelden. Bij een XML database worden de indexen aangemaakt op elementen en hun attributen.

10.3 Benodigde gegevens voor de tijdslijn

Binnen de content-spaces waren de volgende elementen relevant voor de tijdslijn:

Space	Element	Datatype	Opmerking
artists	<u>ID</u>	String	uniek binnen space
	name	String	naam van kunstenaar
	birth	SpecialDate	geboortedatum van kunstenaar
	death	SpecialDate	overlijdensdatum van kunstenaar
	screenReference	String	referentie naar gegevens voor detailscherm
artObjects	<u>ID</u>	String	uniek binnen space
	name	String	naam van kunstwerk
	creationDate	SpecialDate	creëerdatum van kunstwerk
	collection	String	in welke collectie kunstwerk thuis hoort. Hierop moet gefilterd kunnen worden
	hasImage	Boolean	afbeelding beschikbaar ja/nee, de eventuele naam van de afbeelding wordt afgeleid van ID
	artistReference	String	referentie naar de maker
	screenReference	String	referentie naar gegevens voor detailscherm
stylePeriods	<u>ID</u>	String	uniek binnen space
	name	String	naam van de periode
	startDate	SpecialDate	begin van de periode
	endDate	SpecialDate	eind van de periode
	screenReference	String	referentie naar gegevens voor detailscherm
historyPeriods	<u>ID</u>	String	uniek binnen space
	name	SpecialDate	naam van de periode
	startDate	SpecialDate	begin van de periode
	endDate	SpecialDate	eind van de periode
	screenReference	String	referentie naar gegevens voor detailscherm
historyEvents	<u>ID</u>	String	uniek binnen space
	name	SpecialDate	naam van de gebeurtenis
	date	SpecialDate	datum van de gebeurtenis
	screenReference	String	referentie naar gegevens voor detailscherm

Het onderstaande schema toont de benodigde spaces voor de tijdslijn. Referenties naar elkaar zijn hier in opgenomen.



11. DATUM NOTATIES**11.1 Oude situatie**

In de Aria database worden van een kunstenaar o.a. de 'leefde_van' en 'leerde_tot' datum bewaard.

De notatie van deze data is niet geheel volgens standaarden waardoor de data verkeerd geïnterpreteerd kunnen/zullen worden en dus niet gebruikt kan worden voor de tijdslijn.

Leefde_van	Leefde_tot
1857	1923
werkzaam 1511	1544
ca. 1561	1596
16de eeuw	
voor 1387	na 1401
werkzaam tussen 1573	en 1596
werkzaam 17 ^{de}	eeuw
3de kwart 17de eeuw	na 1637
laatste kwart 17de eeuw	
1620/21	1690
na 1622	1669 of later
16de	eeuw
ca.	1500
werkzaam	1663-1672
begin	17de eeuw
20ste	eeuw
c. 1556	1626
c. 1550?	
1755?	1846
1589/90	1633/38
1606	1683/4
	1603/1604
?	1774
ca. 1450	ca. 1531/1533
geboren 1955	
(lege string)	(lege string)
<NULL>	<NULL>
14th	century
3rd quarter 17th century	after 1637
mid	17th century
active	17th century
c.	1500
	1663-1672
after 1622	1669 or later
c. 1525?	before 1588?
active between 1573	and 1596
b. 1955	<NULL>
active 1622	1654-before 1679

11.2 Datumnotaties volgens standaarden

De data zullen volgens de standaarden van Dublin Core (Date en Date Certainty) en ISO 8601 worden genoteerd. Omdat het vaak om data gaat die niet exact zijn moet een bepaalde zekerheidsgraad (certainty) worden bijgehouden. Een aantal mogelijke waarden voor certainty zijn: exact, ca, approx, before, after, early, late, mid.

NB. Approx betekent in het Nederlands: "dicht bij". Ca. betekent "omstreeks, ongeveer". Het is niet duidelijk wanneer voor approx gekozen wordt en wanneer voor circa.

van ... (evt. tot)	Certainty	DC / ISO 8601	Display
1857	exact	1857	1857
16de eeuw	exact	1500/1599	16 ^e eeuw
3 ^{de} kwart 17 ^{de} eeuw	exact	1675/1699	3 ^e kwart 17 ^e eeuw
3rd quarter 17th century	exact	1675/1699	3 ^e kwart 17 ^e eeuw
1620/21	exact	1620/1621	1620/1621
16 ^{de} ... (eeuw)	exact	1500/1599	1500/1599
20 ^{ste} ... (eeuw)	exact	1900/1999	1900/1999
1603/1604	exact	1603/1604	1603/1604
1589/90	exact	1589/1590	1589/1590
1683/4	exact	1683/1684	1683/1684
geboren 1955	exact	1955	1955
1589/90	exact	1589/1590	1589 of 1590
14th ... (century)	exact	1300/1399	1300/1399
1663-1672	exact	1663/1672	1663/1672
ca. ... (1500)	ca	1500	ca. 1500
ca. 1561	ca	1561	ca. 1561
c. 1556	ca	1556	ca. 1556
c. 1550?	ca	1550	ca. 1550
1755?	ca	1755	ca. 1755
? ... (1774)	ca	1774	1774
c. ... (1500)	ca	1500	ca. 1500
c. 1525?	ca	1525	ca. 1525
b. 1955	before	1955	voor 1955
voor 1387	before	1387	voor 1387
after 1637	after	1637	na 1637
na 1401	after	1401	na 1401
begin ... (17 ^e eeuw)	early	1600/1699	begin 17 ^e eeuw
mid ... (17th century)	mid	1600/1699	mid 17 ^e eeuw

probleemdatums			
1654-before 1679	exact	1654/ 1678	ca. 1654/1678
1669 or later	after	1668	na 1668
before 1588?	before, ca	1588	voor ca. 1588
active 1622			
active ... (17th century)			
active between 1573 ... (and 1596)			
werkzaam 1511			
werkzaam tussen 1573 ... (en 1596)			
werkzaam 17 ^{de} ... eeuw			
werkzaam ... (1663-1672)			

De data worden standaard genoteerd in de Gregoriaanse kalender (Common Era: CE of AD). Mogelijke kalenders zijn:

CE	Gregorian Calender (Common Era: CE of AD).
BC	Before the Common Era (Before Christ)
AH	Islamic Calender (Anno Hegira)
AM	Jewish Calender (Anno Mundi)
FR	French Revolutionary Calender
JU	Julian Calender

Datumgegevens van objecten, personen, periodes, of gebeurtenissen zullen doorgaans in termen van de Gregoriaanse kalender worden vermeld. Wellicht zullen er ook data uit de BC kalender voorkomen.

De data kunnen als volgt in het XML schema worden opgenomen:

```
<date calendar="CE" certainty="ca" value="1551"/>
```

of

```
<period>
  <start value="1663/1672"/>
  <end certainty="circa" value="1737"/>
</period>
```

‘De tijdslijn voor de nieuwe website van Rijksmuseum’

Bijlage II

“Analyse van bestaande tijdslijnen”

laatste wijziging:
24 maart 2004
Rick Companje

INHOUDSOPGAVE

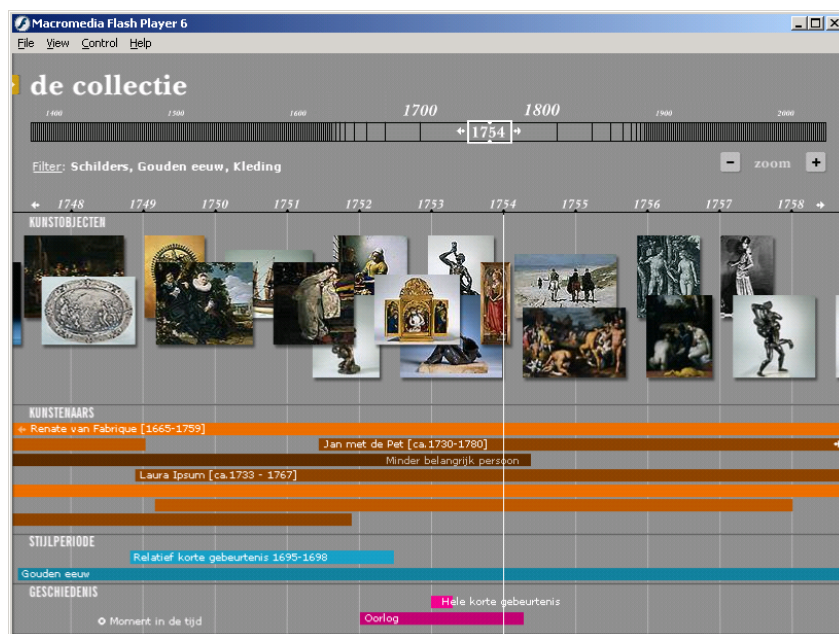
1.	ANALYSE VAN BESTAANDE TIJDSLIJNEN	3
1.1	Pitch ontwerp Fabrique	3
1.2	Australisch parlement	3
1.3	Carriere tijdlijn.....	4
1.4	BGW Timeline.....	4
1.5	Who what when	6
1.6	Deep space timeline	7
1.7	Olive Tree Bible Software.....	8
1.8	Rijksmuseum Maurits tentoonstelling	9
1.9	Time line tool	10
1.10	MuslimHeritage.com.....	11
1.11	Business Law Timeline.....	12
1.12	Mary baker eddy.....	13
1.13	Classic Motown.....	14
1.14	Nationaal archief.....	15
1.15	art timeline.....	16
1.16	Flevoland boven water.....	17
1.17	U2.com	19
1.18	Worcester art museum.....	20
1.19	The Metropolitan Museum of Art	21
1.20	Van Gogh museum.....	22
2.	CONCLUSIE.....	23

1. ANALYSE VAN BESTAANDE TIJDSLIJNEN

In de fase analyse en definitie heb ik onderzoek gedaan naar de functionaliteiten van tijdslijnen die reeds op andere (museum) websites ingezet zijn. Hierbij heb ik per tijdslijn de eigenschappen, voor- en nadelen in kaart gebracht. Sterke punten kon ik verwerken in de tijdslijn die ik ging ontwikkelen. De zwakke punten van de andere tijdslijnen gaven mij een beeld met welke knelpunten ik ook te maken zou kunnen krijgen. Ik heb twintig verschillende tijdslijnen geanalyseerd, variërend van zeer simpel, slecht uitgevoerd en statisch tot redelijk dynamisch en goed vormgegeven.

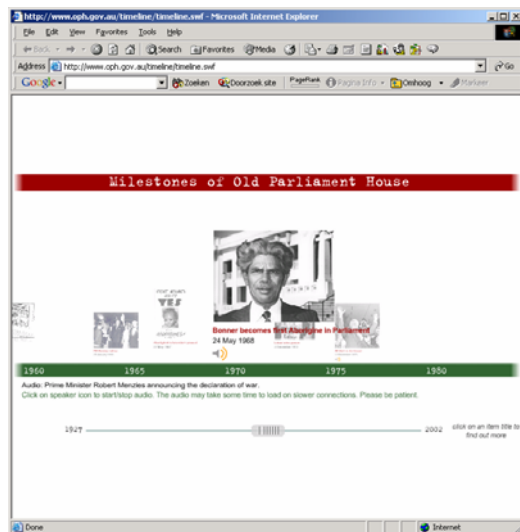
1.1 Pitch ontwerp Fabrique

- DEMO!
- Flash
- Statisch
- Goed geluid
- Objecten, kunstenaars, stijlperiodes, geschiedenis/gebeurtenissen
- Scroll is goed
- Zwiep gevoel kan meer



1.2 Australisch parlement

<http://www.oph.gov.au/timeline/timeline.swf>



1.3 Carrière tijdslijn

- Flash
- Per jaar
- Irritant geluid
- Soepele beweging bij scroll
- Geen zwiep mogelijk aan tijdslijn

<http://www.smart.net/~jworsley/portfolio/flash/caltimeline.html>



1.4 BGW Timeline

- Flash
- Per jaar
- Geluidseffecten (irritante)
- Soepele beweging bij scroll
- Geen zwiep mogelijk aan tijdslijn

http://www.smart.net/~jworsley/portfolio/pages/flash_bgwtimeline.html

BGW:TIMELINE

Client: Inversion Studios (Personal Project)

Media: Flash Interactive

Size: 600x350

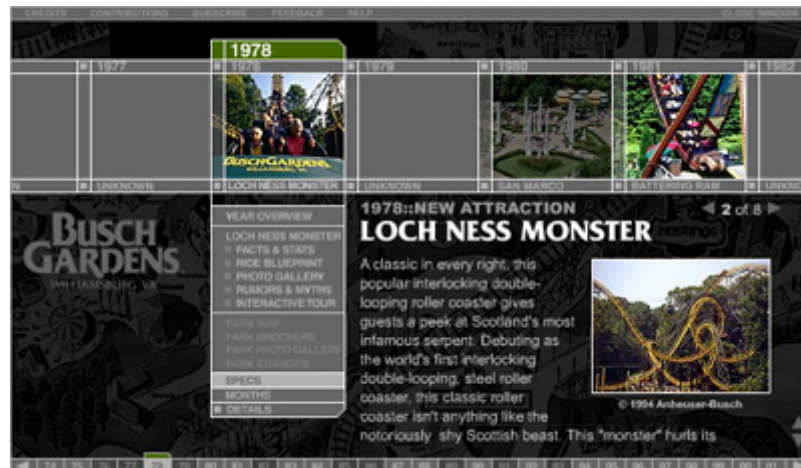
Description: Coming Soon

Live Site: [Launch Sample](#)

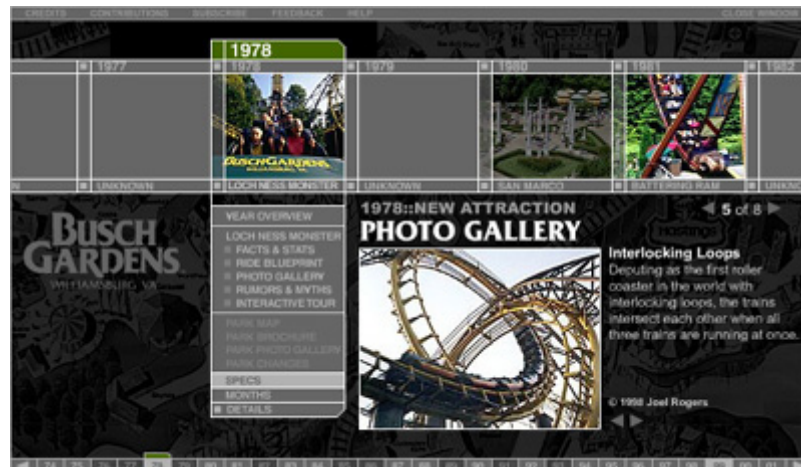
Awards: ThrillNetwork "Best of Coaster Web" Award
ScreamScape "Cool Site" Award
CoasterBuzz "Awesome" Award

File Size: 610k with preloader

Screen Shot #1:



Screen Shot #2:



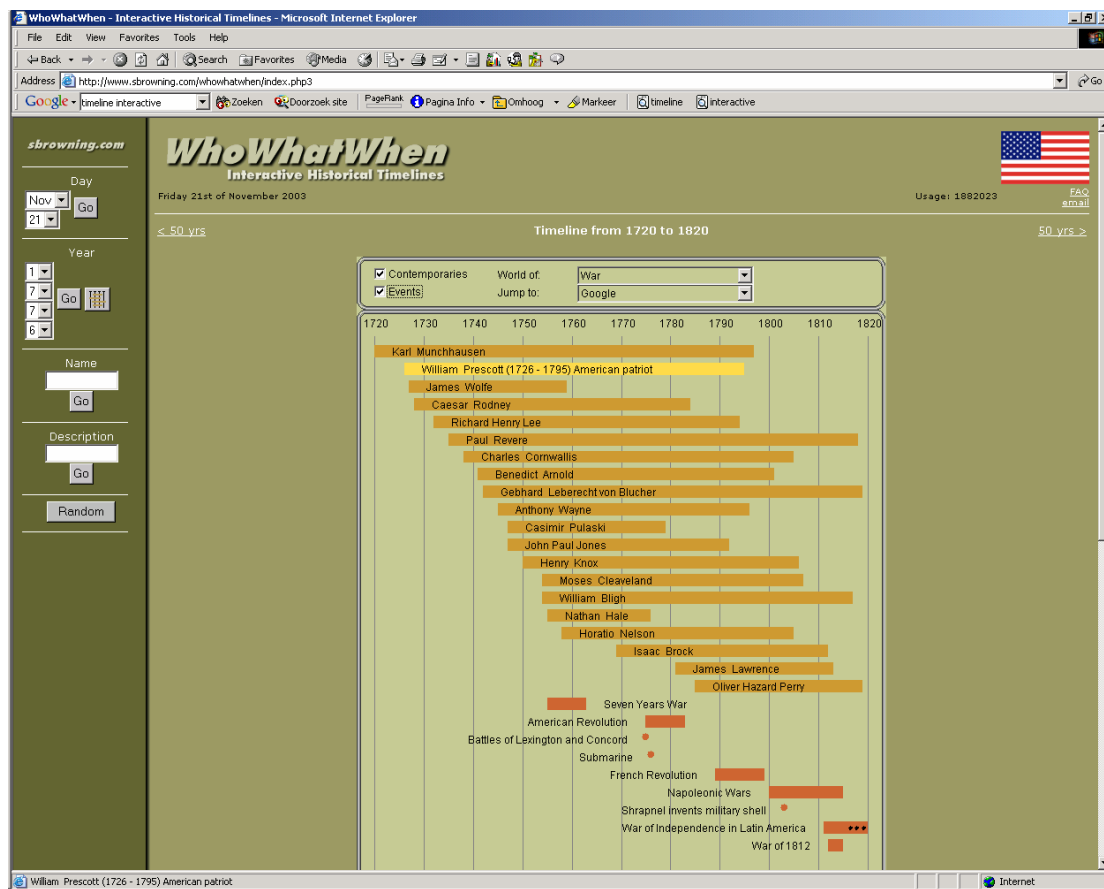
Screen Shot #3:



1.5 Who what when

- Java
- Events + contemporaries (personen/tijdgenoten)
- Overlappingsen zichtbaar
- Verbergen door vinkjes
- Filteren links met pull-downs
- Laat periodes van 50 jaar zien

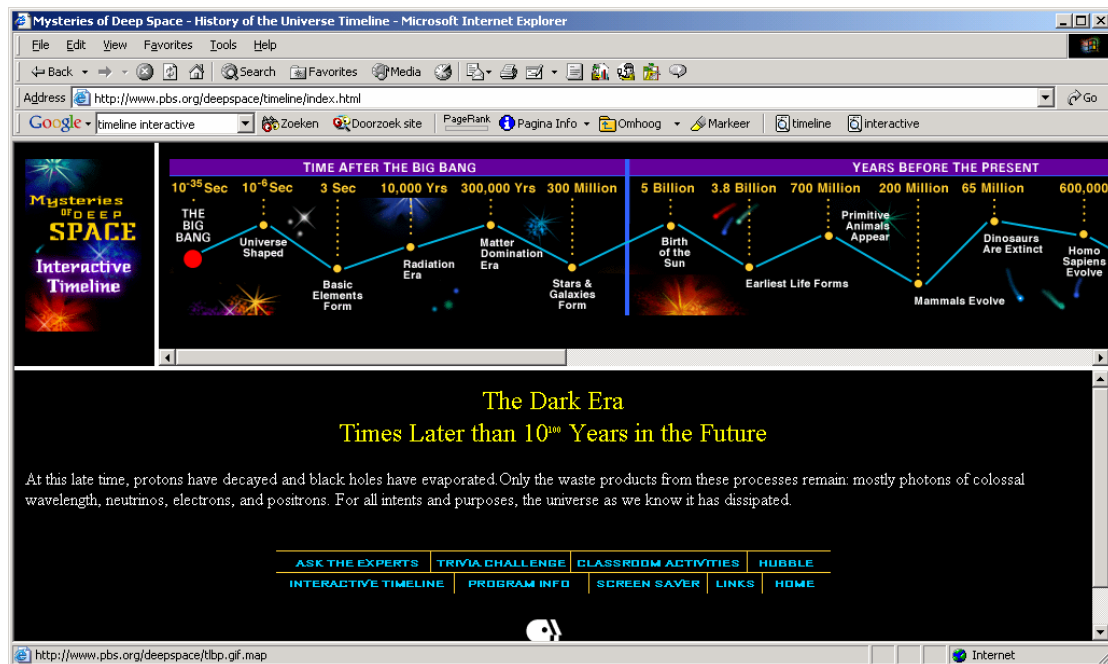
<http://www.sbrowning.com/whowhatwhen/index.php3>



1.6 Deep space timeline

- Statische html met plaatjes
- imagemap
- Frames

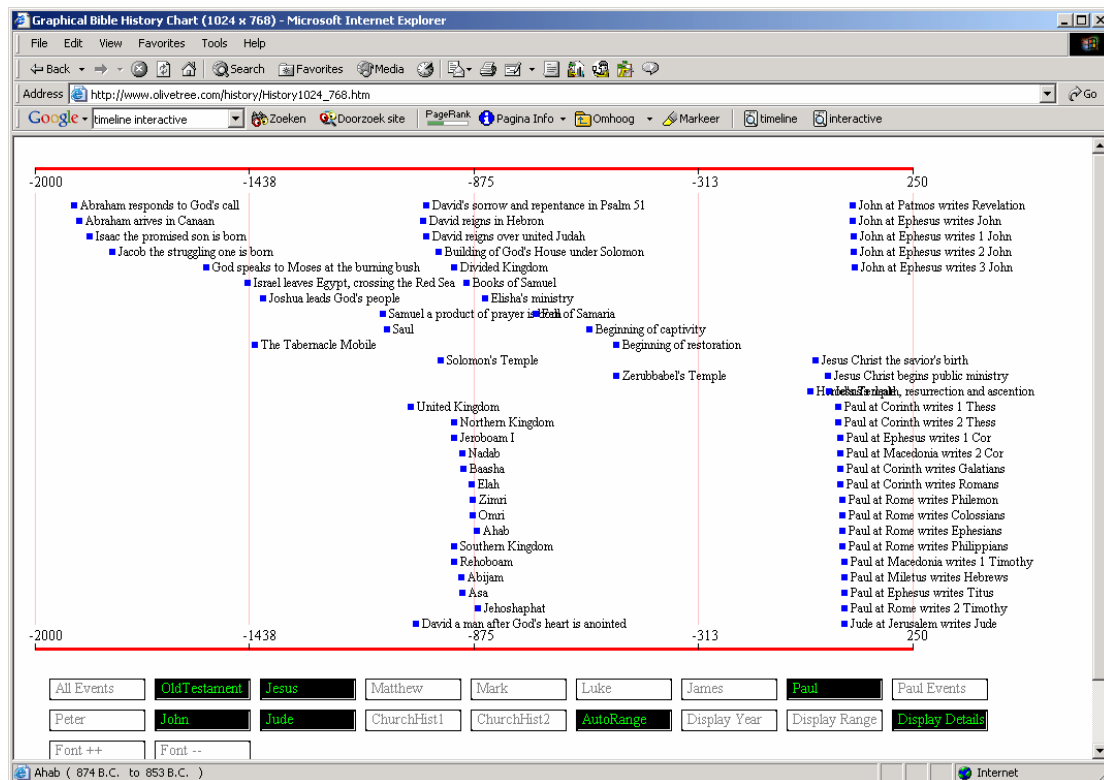
<http://www.pbs.org/deepspace/timeline/index.html>



1.7 Olive Tree Bible Software

- Java
- Statisch
- Slechte filter

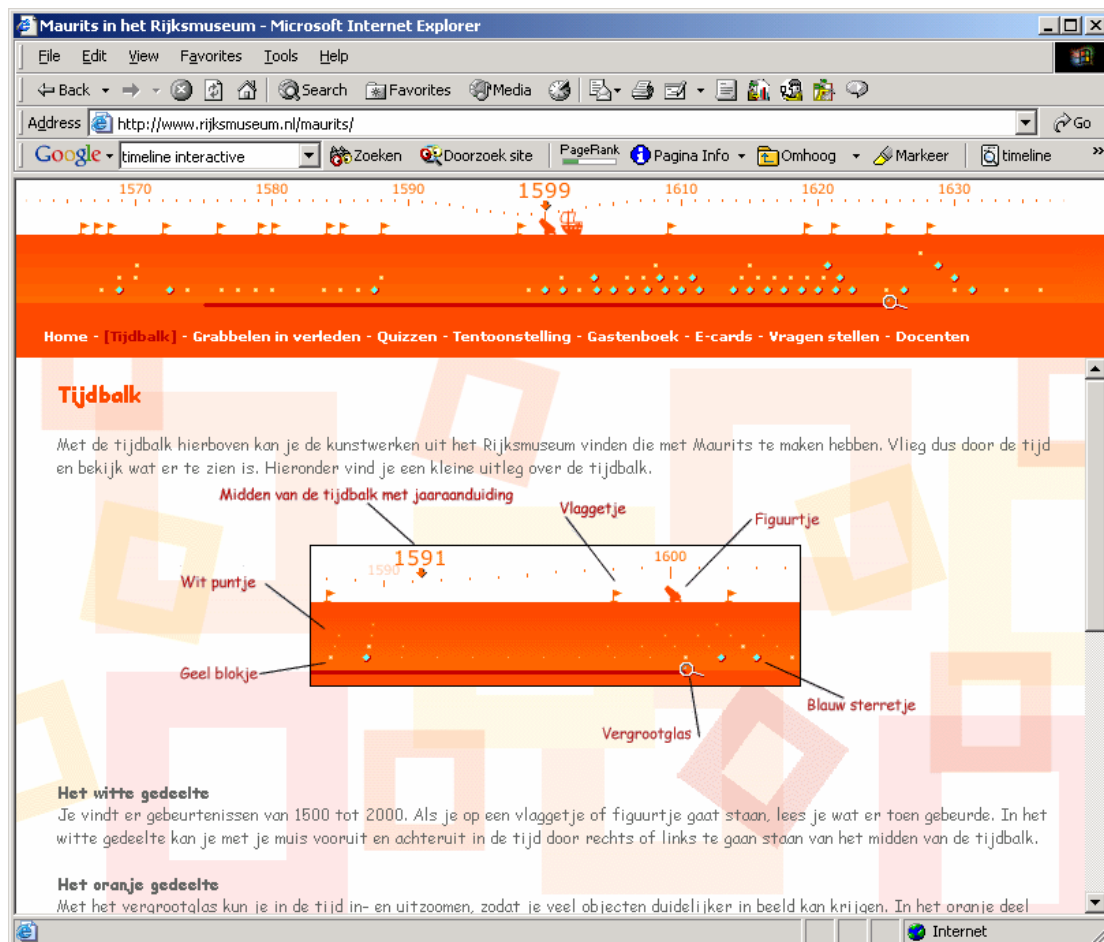
http://www.olivetree.com/history/History1024_768.htm



1.8 Rijksmuseum Maurits tentoonstelling

- Shockwave
- Statisch
- Goede geluidseffecten
- Uitleg
- 'Goed gevoel' dynamische zoom van 500 jaar tot 10 jaar
- Objectjes blijven stipjes in gedetailleerde zoom
- Animerende figuurtjes op tijdlijn bij gebeurtenissen
- Geen zwiep
- Korte beschrijving popt-up wanneer je op een stipje gaat staan
- Klikken linkt door naar andere plaats in de site
- duidelijk gericht op kinderen

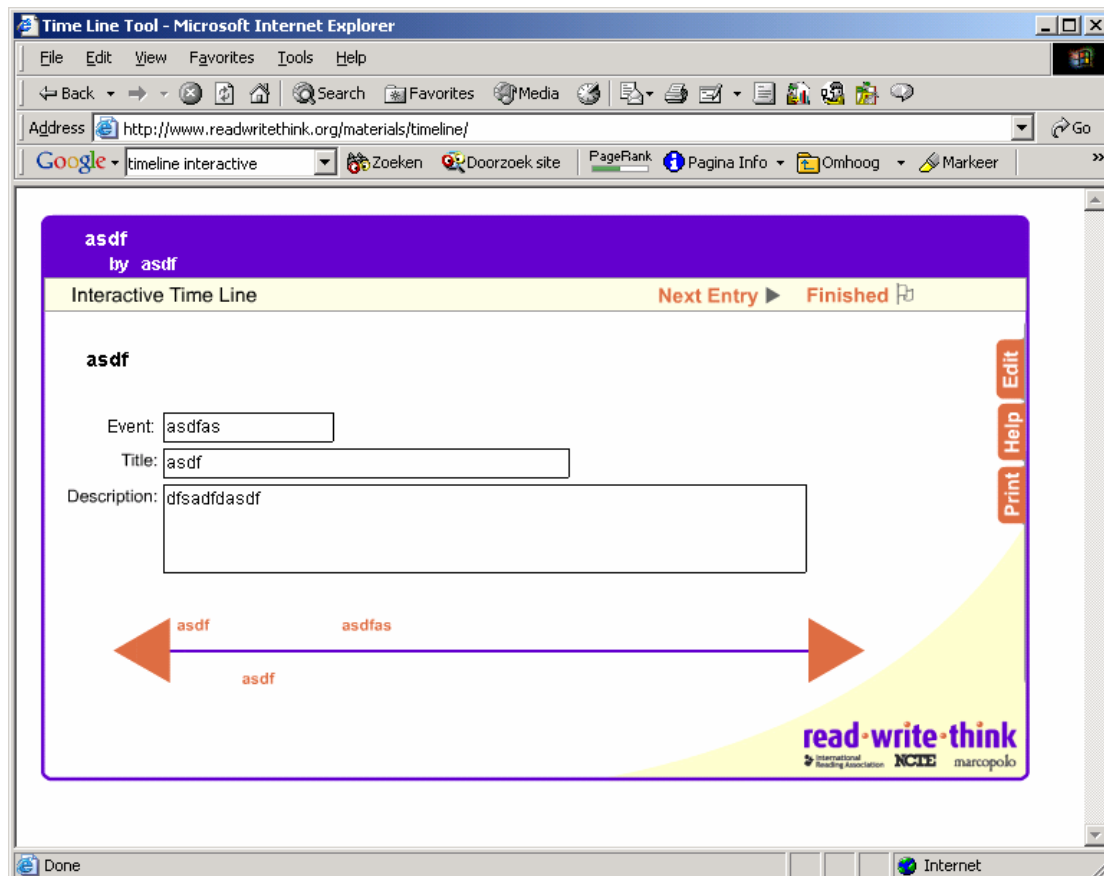
<http://www.rijksmuseum.nl/maurits/>



1.9 Time line tool

- flash
- interactief: zelf datums en gebeurtenissen aanmaken
- geen plaatjes

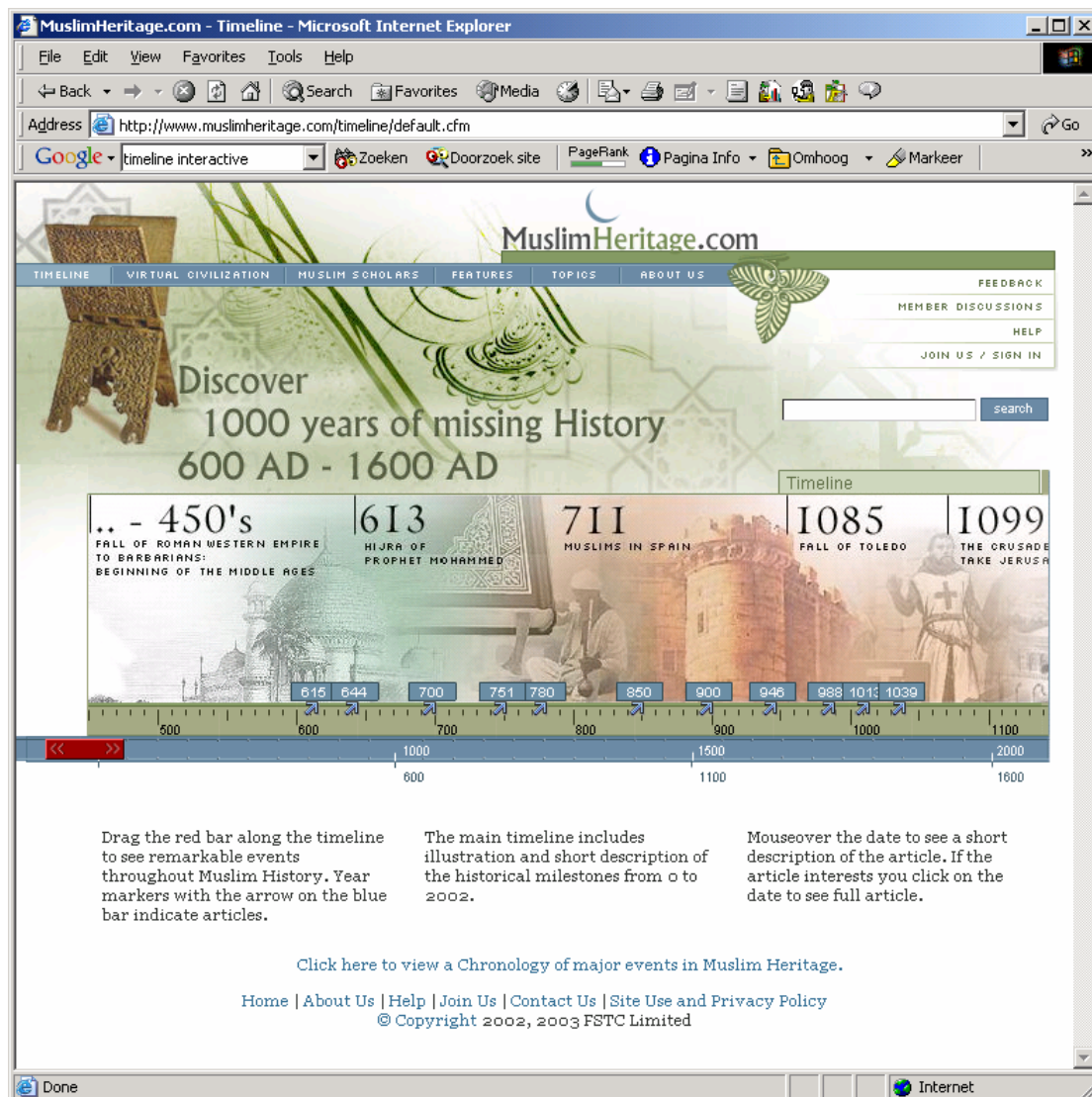
<http://www.readwritethink.org/materials/timeline/>



1.10 MuslimHeritage.com

- Flash
- Grafisch goed
- Weinig gebeurtenissen
- Goede scroll
- Achtergrond van de tijdlijn is in elkaar overlopende plaatjes van gebeurtenissen in de geschiedenis
- Mouseover op jaartallen zorgt voor popup met ongeveer 8 regels info
- **Heel goed zwiep gevoel**

<http://www.muslimheritage.com/timeline/default.cfm>



1.11 Business Law Timeline

- Flash
- Periodes horizontaal, jaartallen verticaal per periode
- simpel

Business Law Timeline

The slider at the bottom of the timeline can be moved to view legal happenings in different periods in time. Information appears in the lower window.

This interactive feature requires the free Macromedia Flash 5 plug-in. If you need to install this, [download Flash here](#).

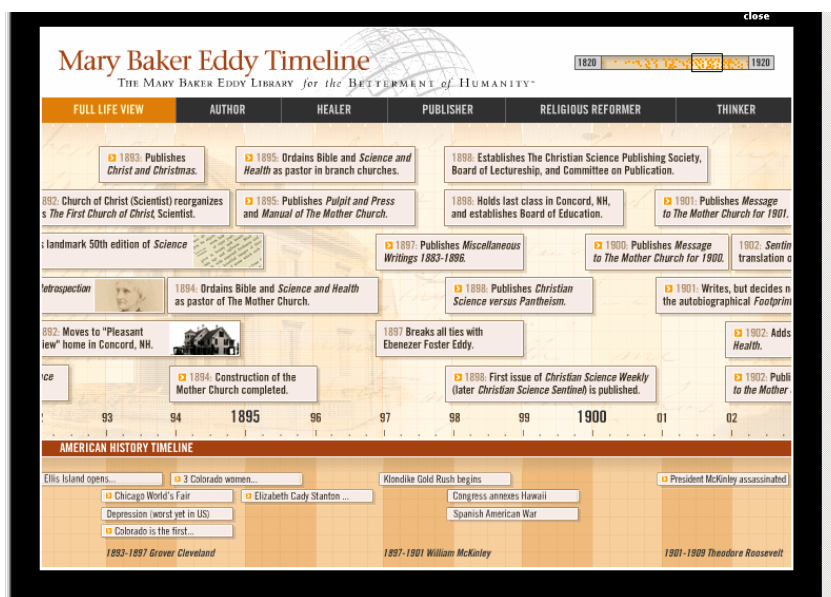
1700's	Late 1700s to 1870s	1800's	1900-1900
Blackstone's Commentaries. Boston Tea Party. s develop Corporation ration of Independence. : Common Law.	1700s-1800s : The Industrial Revolution. 1778 : Articles of Confederation is adopted. 1788 : State ratification of Constitution. 1788 : Supreme Court. 1791 : Bill of Rights.	1861-1865 : The Civil War. 1865 : Slavery is abolished. 1866 : Civil Rights Act. 1878 : Bankruptcy Act established. 1890 : Sherman Antitrust Act.	Early 1900's : English Courts. 1906 : Uniform Sales A 1906 : The Food, Drug Act is enacted.

Late 1700's: English common law is adopted in America. In the United States, law, equity, and merchant courts merged. All states except Louisiana base their legal systems primarily on the English common law. Because of its French heritage, Louisiana bases its law on civil law.

1.12 Mary baker eddy

- Erg goed!
- Flash 6
- Timeline in popup. Wordt geopend door klikken op een knop.
- Uitleg vind plaats voordat de timeline wordt geopend.
- Rechtsboven in beeld continu een kleine viewport van totale tijdlijn met stipjes
- Hoeft niet te klikken om te scrollen
- Beslaat maar 1 eeuw 1820-1920
- Mouseover gebeurtenis geen leuke effectje (maar alleen wanneer er een doorklik is)
- Bij klik op gebeurtenis klein popupje met een paar regels info
- Onder in beeld relevante gebeurtenissen uit de geschiedenis
- Begin een klein preloader procesbar
- Grafisch goed
- Geluidseffecten bij mouseovers en kliks, niet bij scrollen
- Plaatjes in tijdlijn mogelijk
- Sommige gebeurtenissen hebben veel achtergrond info op de popup (1887)
- Sneltoetsen via toetsenbord zouden goed zijn
- Simpele filter door te klikken op een 6 tal buttons. Bij filter blijven gebeurtenissen staan op de tijdlijn, maar worden verkleint tot een blokje en zijn niet klikbaar. Op die manier zie je dat er een filter is toegepast.
- Verdeling jaartallen:
 - o per vijf jaar groot getal, volledig bijv: 1915.
 - o jaren daartussen als korte notatie bijv: 16...17...18....19.
 - o daartussen stipjes die **kwartalen** voorstellen. Kwartaal is het kleinste element
- gebeurtenissen worden zoveel mogelijk achterelkaar geplaatst. Wanneer er overlappingen plaatsvinden wordt degene die als laatste begint op de volgende regel gezet. Er wordt dus verticaal ge-aligned.

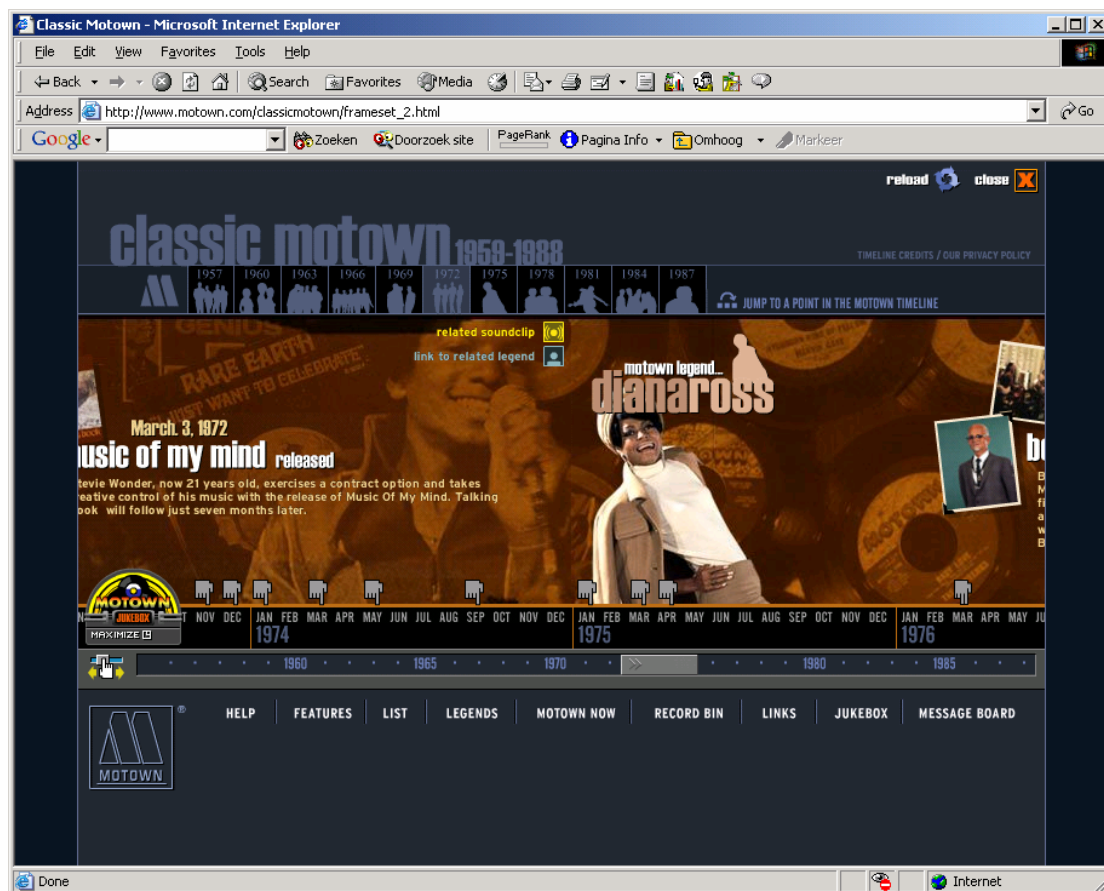
<http://www.marybakereddylibrary.org/marybakereddy/timeline.jhtml>



1.13 Classic Motown

- Flash
- Traag, heel traag. Moet heel lang laden
- Grafisch goed
- Scroll/zwiep gevoel goed
- Beslaag een vaste periode van ongeveer 30 jaar
- Veel plaatjes en informatie op de tijdlijn zelf
- Schermruimte 800x600 wordt goed benut
- Via icons bovenaan direct springen naar jaartal/thema
- Alles in 1x ingeladen, geluid waarschijnlijk ook

http://www.motown.com/classicmotown/frameset_2.html



1.14 Nationaal archief

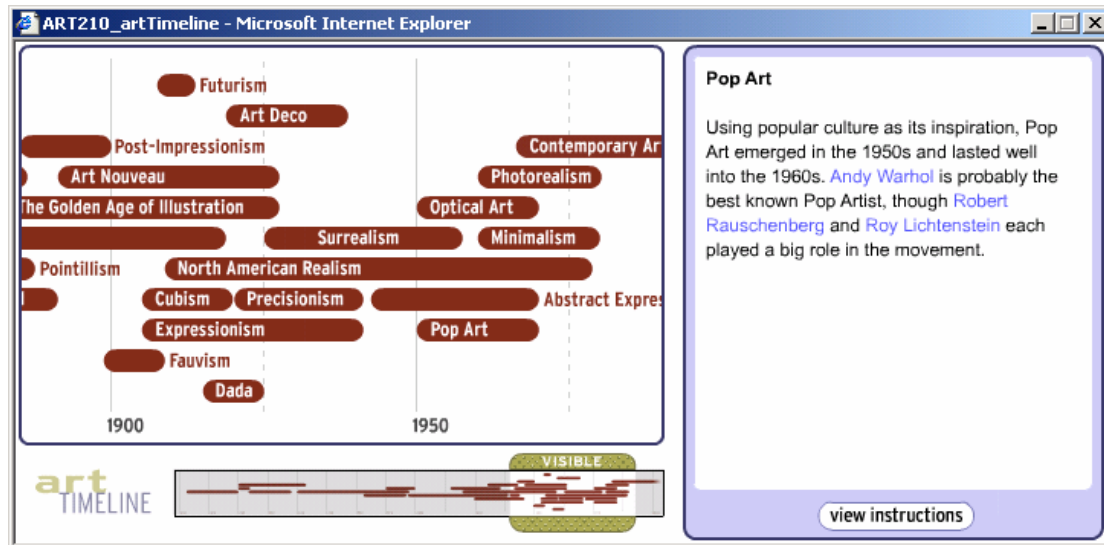
- Flash
- Grafisch goed
- Popup
- Redelijk vloeiend
- Grafisch ok
- Snel geladen
- Goeie doorklik naar meer info (maar wel in popup window)
- Introduceert 'tijdvakken'
- Belangrijke gebeurtenissen hebben een jaartal met groot font
- In eerste instantie zijn alle plaatjes cirkels, wanneer plaatjes geladen zijn verschijnt plaatje in cirkel
- (maar) 2 zoom levels
- loopt van 1200 tot 2000
- boven in beeld draggable schuifbalkje om door eeuwen te navigeren
- Drag en drop op kleine
- Kent geen balkjes om periodes weer te geven
- Drie lagen: curiosa, jaartallen en biografieën
- Tekst valt soms over elkaar heen wanneer er veel gebeurd is in een periode, ook bij meest ingezoomd (1946)
- Bepaalde periodes helemaal leeg bij geen info
- Goed: Meerdere thema's



<http://www.nationaalarchief.nl/>

1.15art timeline

- flash
- popup, alles gebeurt in deze popup
- overzichtsbalkje onderin
- balkjes voor periodes. Wanneer tekst niet in balkje past wordt ie er naast gezet.
- overlappende periodes worden onder elkaar geplaatst
- gaat over kunstperiodes
- soepel/snel

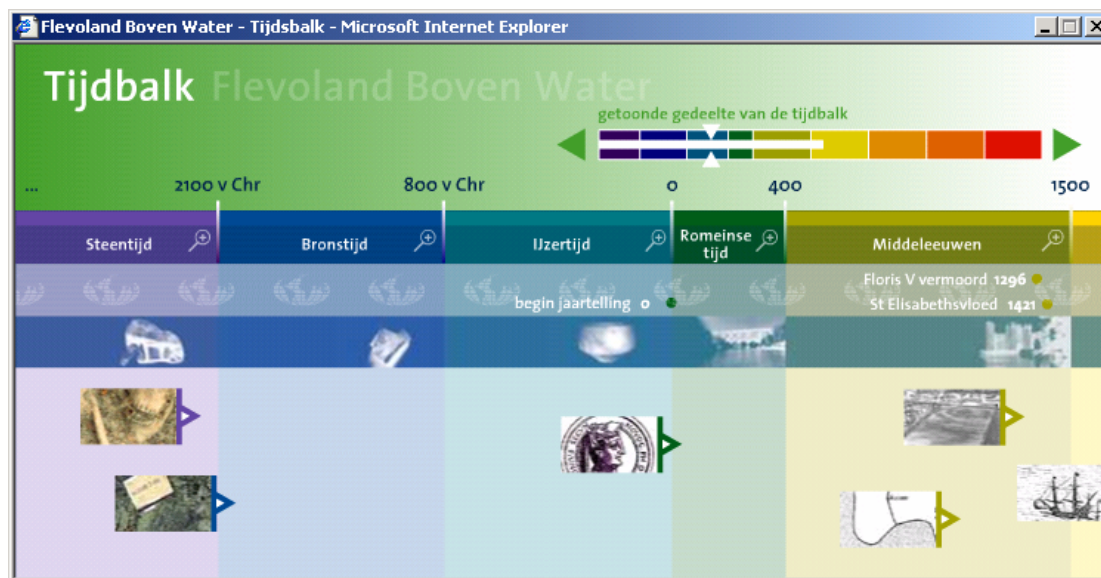


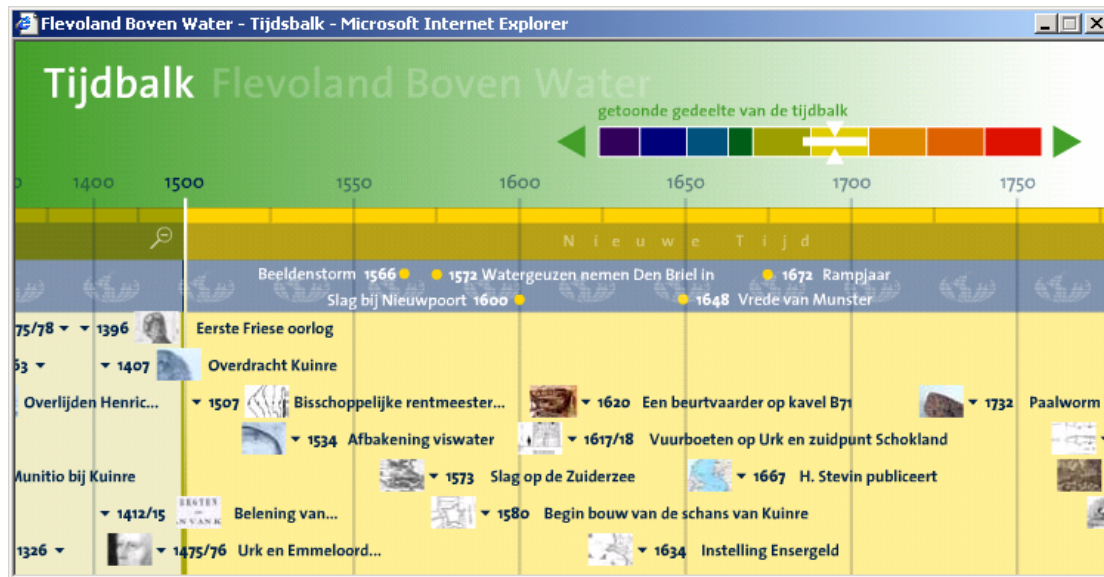
http://www.purple-gizmo.com/pFolio_Cont_interactive.html

1.16 Flevoland boven water

- flash
- loopt soepel
- snel
- super: periodes in geschiedenis kunnen ingeklapt en uitgeklaapt worden. (althans die suggestie wordt gewekt, in werkelijkheid is er sprake van twee zoomlevels. Bij het klikken op een plusje worden alle periodes uitgeklaapt)
- heel duidelijk kleurgebruik
- zowel AD als BC
- overzichtsbalkje geeft ook in kleuren de periodes weer
- overzichtsbalkje bevat een wit balkje die aangeeft welk gedeelte er in beeld is en geeft ook de exacte plek aan die bekeken wordt
- bepaalde titels van gebeurtenissen worden afgebroken met puntjes
- gebeurtenissen bevatten bijna allemaal een thumbnail
- overlappende gebeurtenissen worden onder elkaar geplaatst
- Gebeurtenissen in de rest van de wereld worden boven in de tijdslijn getoond zonder doorklik en zonder thumbnail maar met een stip/cirkeltje. Gebeurtenissen die wel een doorklik hebben staan in het grotere vlak eronder.
- **korte periodes met veel gebeurtenissen worden uitgerekt in de tijdslijn getoond!!!**
- Per periode een soort achtergrondplaatje gebruikt ter illustratie (bijv. een wazige steen)

<http://www.flevolandbovenwater.nl>

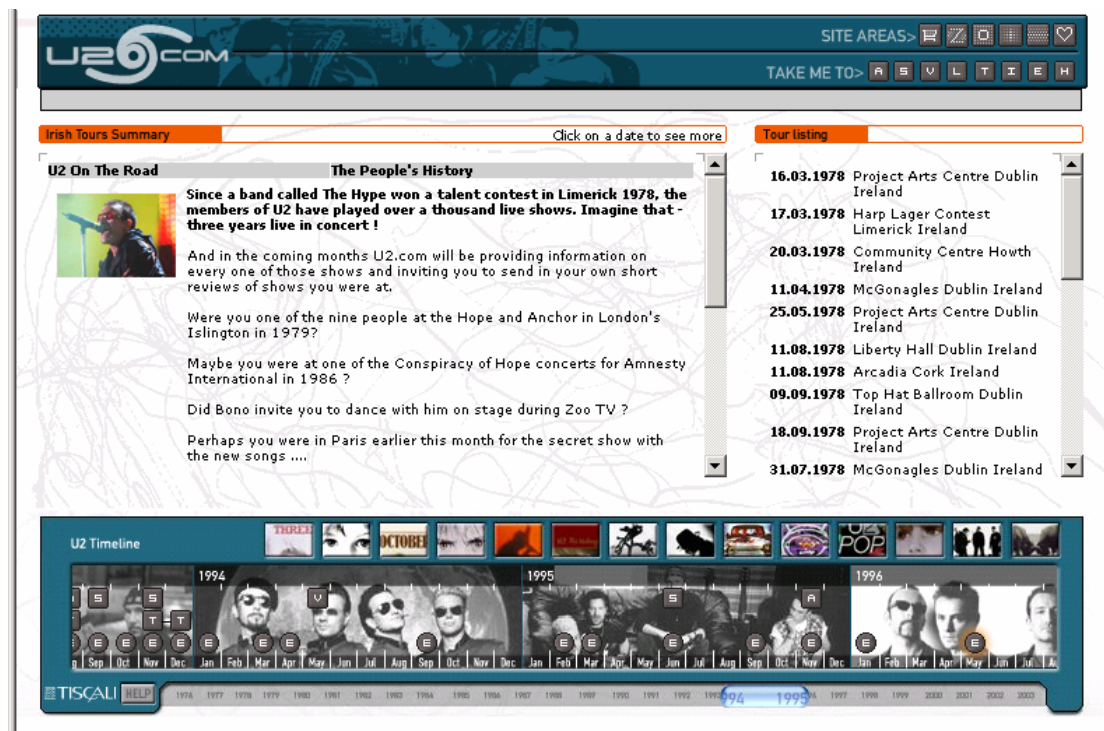




1.17 U2.com

- flash
- statisch
- buggy

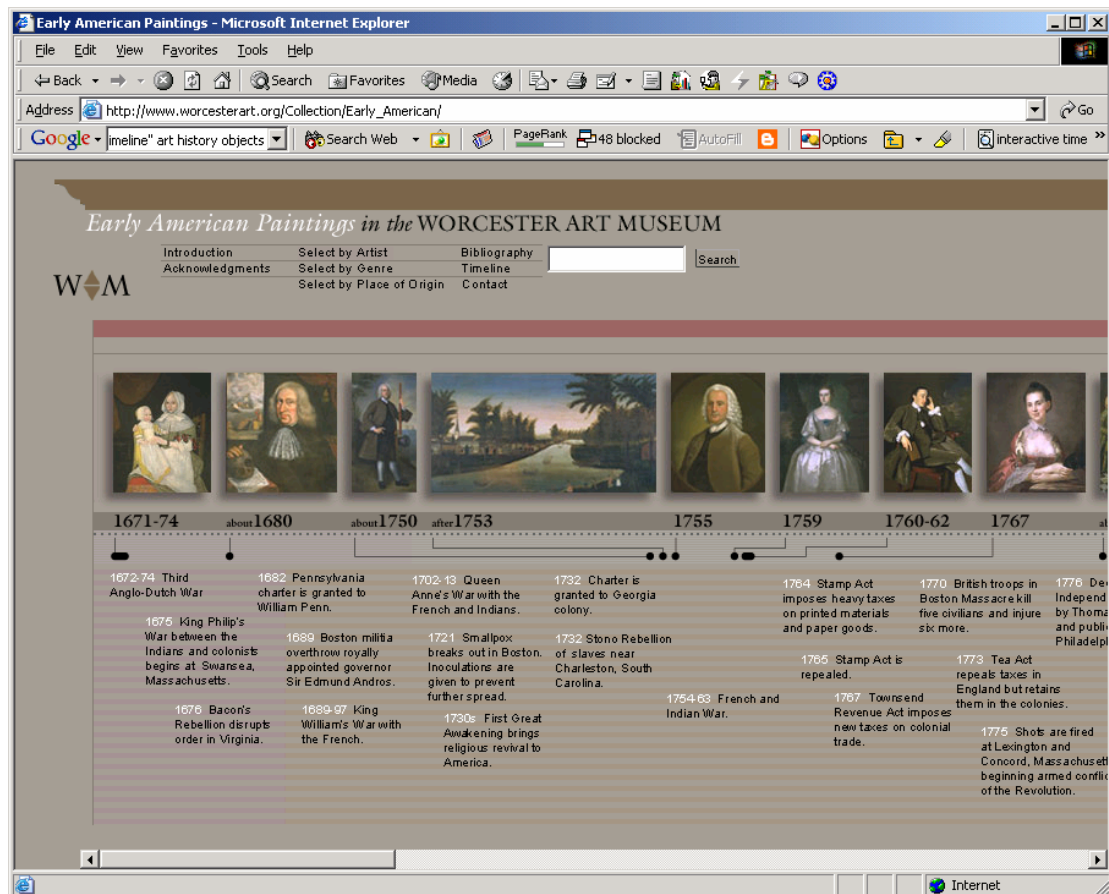
www.u2.com



1.18 Worcester art museum

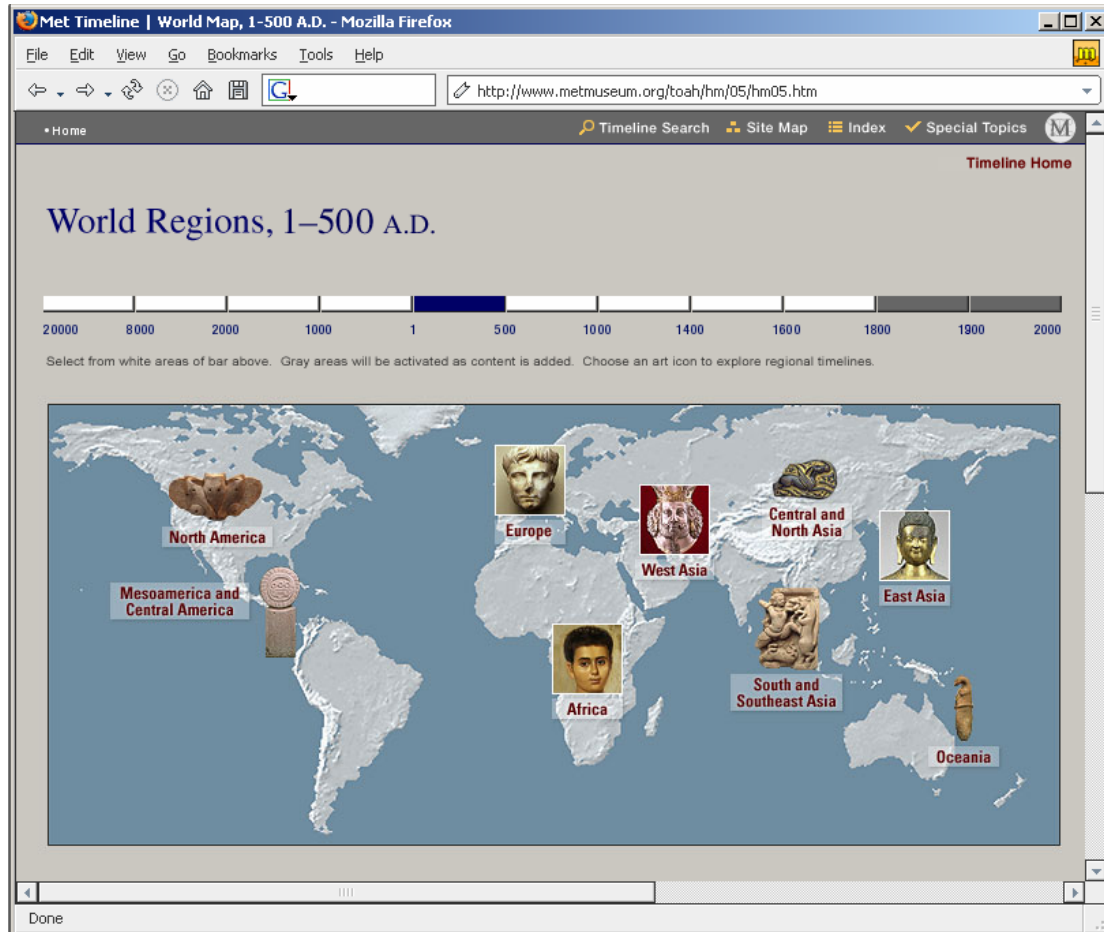
- Geen Flash maar DHTML!!! Het kan dus wel
- Traag
- Schaduw achter de plaatjes

http://www.worcesterart.org/Collection/Early_American/



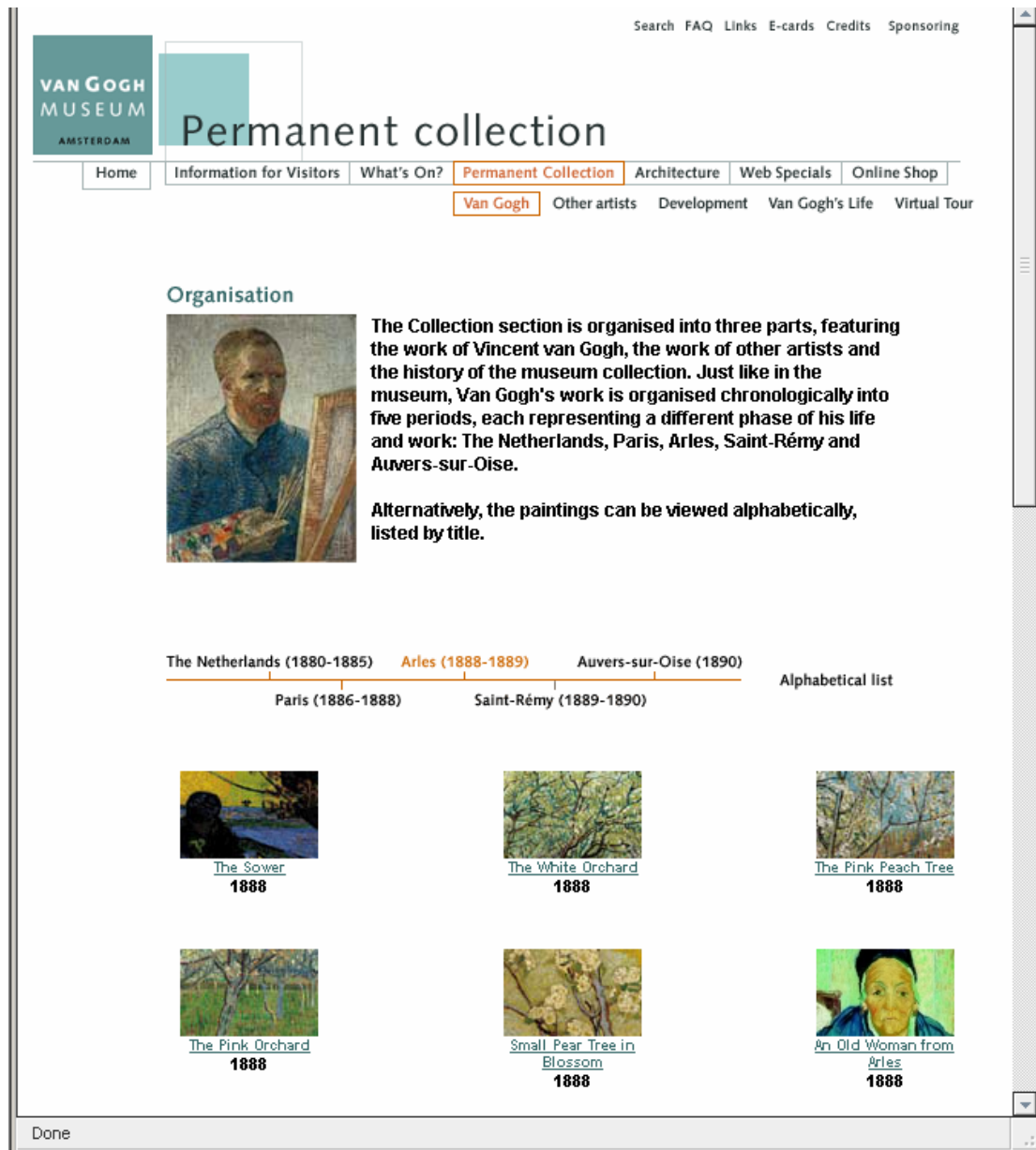
1.19 The Metropolitan Museum of Art

- geen flash maar gegeneerde html pagina's
- ontzettend veel data
- geografisch door de tijd navigeren
- groot tijdsbestek (vanaf 20000 BC)



1.20 Van Gogh museum

- geen flash maar gegeneerde html pagina's
- geen beweging
- korte periode
- goede doorklik
- alfabetisch zoeken mogelijk



2. CONCLUSIE

Het analyseren van bestaande tijdslijnen is erg nuttig geweest voor het bepalen van de functionaliteiten voor de tijdslijn die ik ontwikkelde. Iedere zojuist besproken tijdslijn heeft wel iets bijgedragen.

De Metropolitan tijdslijn vormde bijvoorbeeld een goede eenheid met de website waarin hij gebruikt werd. De tijdslijn van het Nationaal Archief introduceerde thema's. Flevoland boven water bracht mij op het idee tijdsvakken te creëren met variabele lengtes waarmee drukke periodes op de tijdslijn meer ruimte kregen. De tijdslijn van de Maurits tentoonstelling was voorzien van subtiele geluidseffecten. Toch was de pitch-demo die Fabrique voor het Rijksmuseum gemaakt had nog het meest bruikbaar. Een groot aantal interne doelen waren hierin al verwerkt, weliswaar niet uitgewerkt, maar er was wel goed over nagedacht.

‘De tijdslijn voor de nieuwe website van het Rijksmuseum’

Bijlage III

“Functioneel Ontwerp”

laatste wijziging:
24 maart 2004
Rick Companje

INHOUDSOPGAVE

1.	INLEIDING	3
2.	PLAATS VAN DE TIJDSLIJN IN DE WEBSITE	3
3.	FUNCTIONELE LAYOUT VAN DE TIJDSLIJN	3
4.	VOORLOPIG KLASSENDIAGRAM.....	4
5.	VISUALISATIE VAN ELEMENTEN MET ONZEKERE DATUM.....	5
6.	BIJGESTELD KLASSENDIAGRAM	6
6.1	Content package	6
6.2	GUI package.....	7
6.3	Scrollfunctionaliteit	7
6.4	Control package.....	8

1. INLEIDING

In de fase functioneel ontwerp heb ik alle functionaliteiten van het te ontwikkelen systeem bepaald en vastgelegd.

Dit document is een bijeenraping van de resultaten van deze fase.

Een aantal beschrijvingen zijn uit deze publieke versie van het functioneel ontwerp weggelaten.

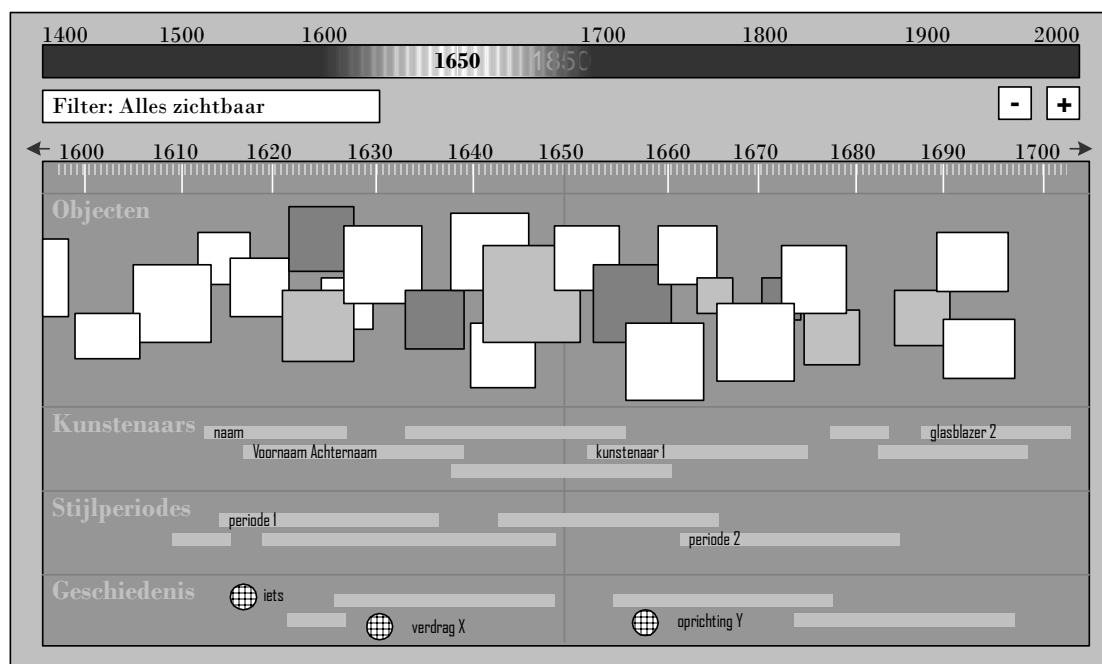
2. PLAATS VAN DE TIJDSLIJN IN DE WEBSITE

Het is nog niet met zekerheid te zeggen wat de rol van de tijdslijn wordt zolang het ontwerp van de overkoepelende website nog niet definitief was.

De tijdslijn moet daarom zo flexibel mogelijk opgebouwd worden zodat deze op iedere plaats op de website ingezet kon worden. Het formaat (de afmetingen) van de tijdslijn moeten configureerbaar zijn en de vormgeving moet later toegepast moest kunnen worden.

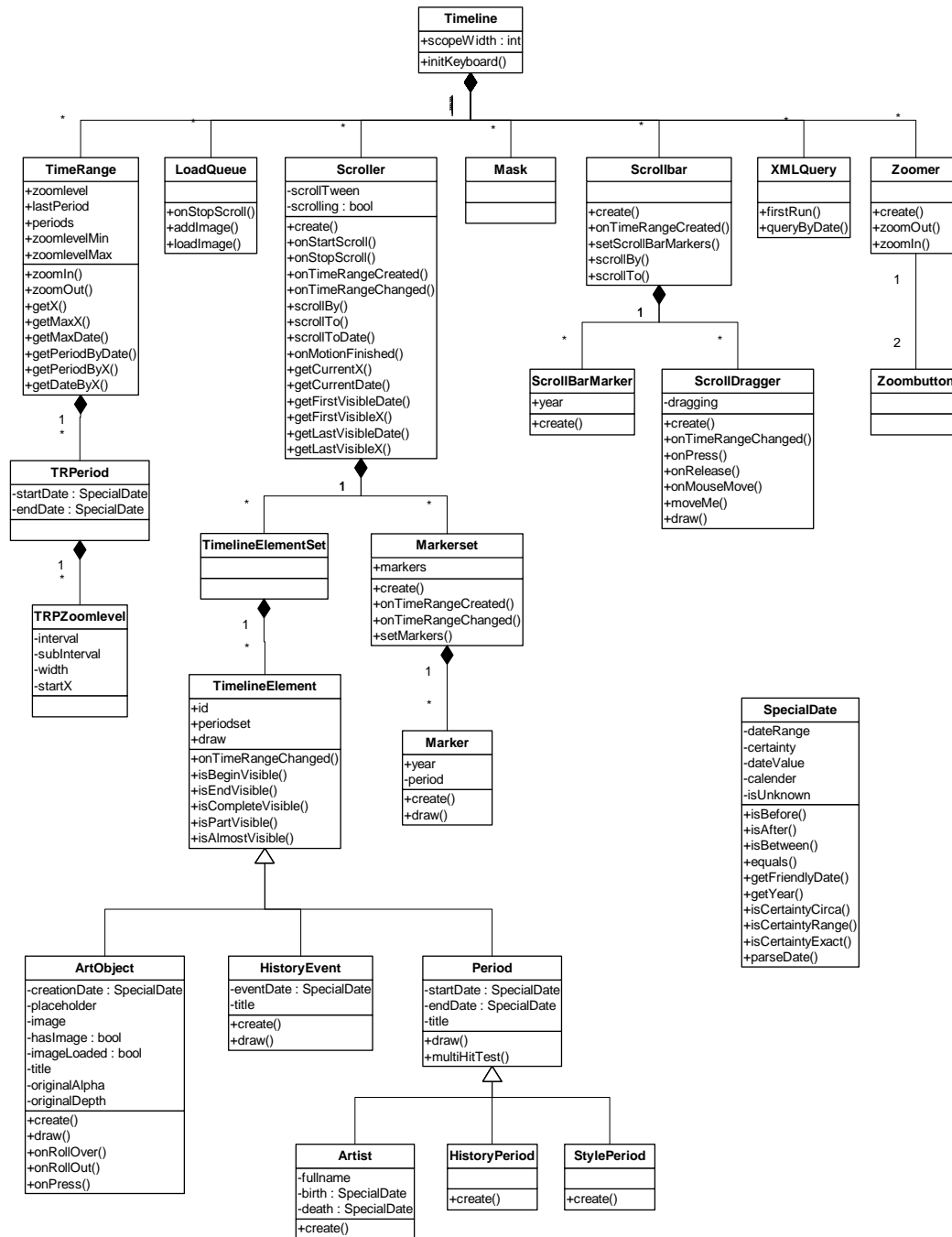
3. FUNCTIONELE LAYOUT VAN DE TIJDSLIJN

De afbeelding toont de functionele lay-out van de tijdslijn.



4. VOORLOPIG KLASSENDIAGRAM

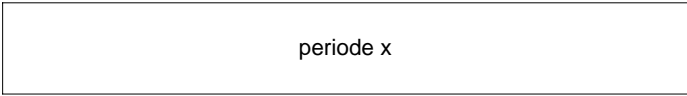
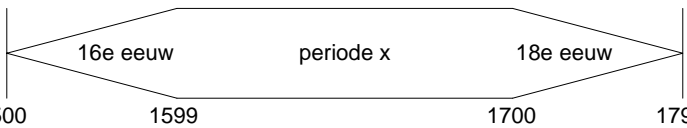
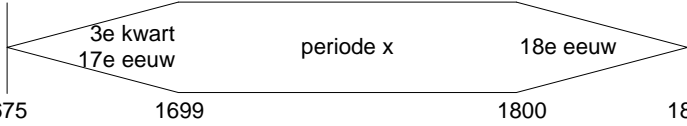
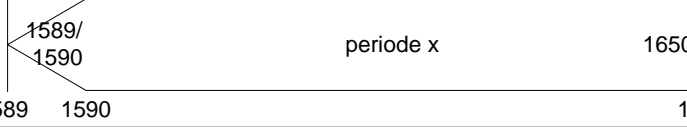
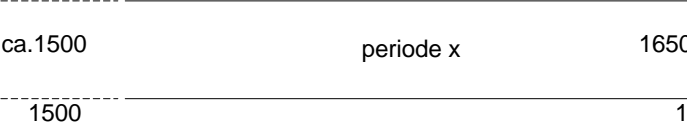
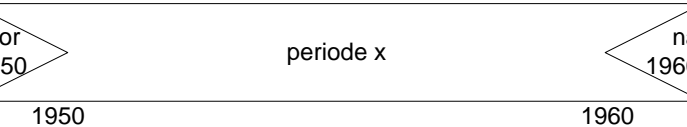
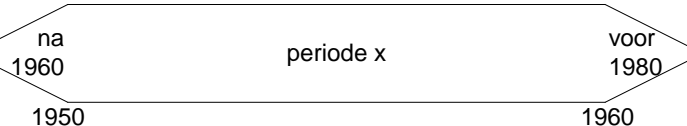
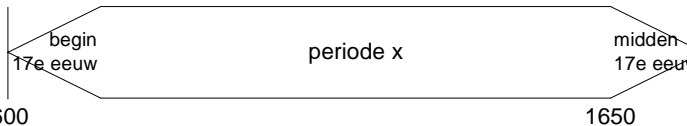
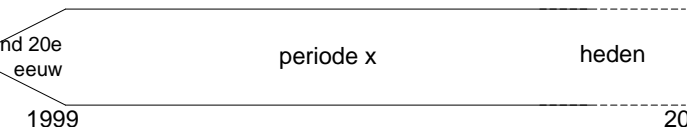
Een van de voorlopers van het uiteindelijke klassendiagram



5. VISUALISATIE VAN ELEMENTEN MET ONZEKERE DATUM

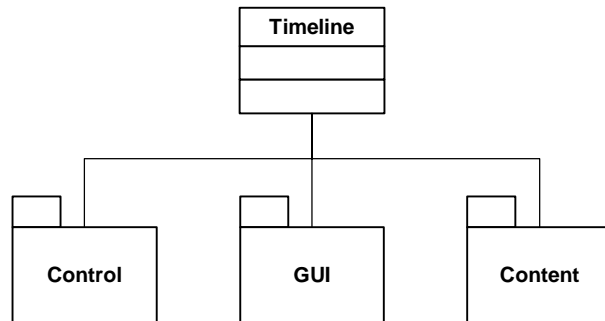
Periodes worden op de tijdslijn op verschillende manieren gepresenteerd.

Bepaalde periodes zullen een exacte begin en/of einddatum, van andere zal het begin of eind niet vast staan. Mogelijke combinaties en verschijningsvormen zijn in het volgende schema weergegeven:

Start	End	Weergave van een periode
1857	1950	
16 ^e eeuw	17 ^e eeuw	
3 ^e kwart 17 ^e eeuw	18 ^e eeuw	
1589/1590	1650	
ca. 1500	1650	
voor 1950	na 1960	
na 1960	voor 1980	
begin 17 ^e eeuw	midden 17 ^e eeuw	
eind 20 ^e eeuw	heden	

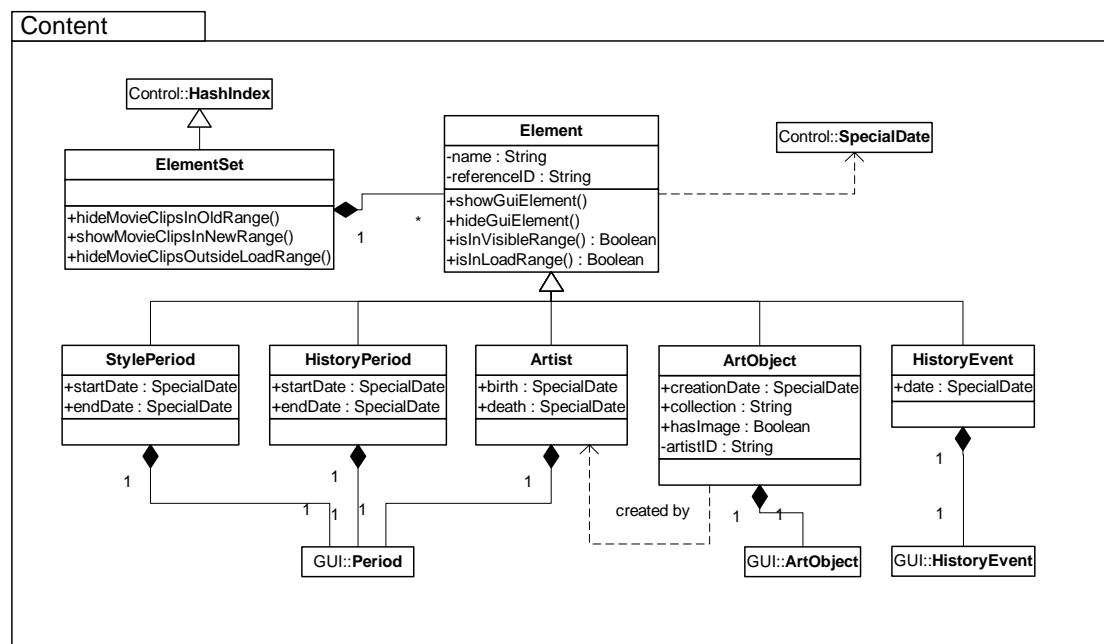
6. BIJGESTELD KLASSENDIAGRAM

De tijdslijn is onderverdeeld in packages. Het opdelen in packages maakt de tijdslijn (en ook het model daarvan) stukken overzichtelijker.



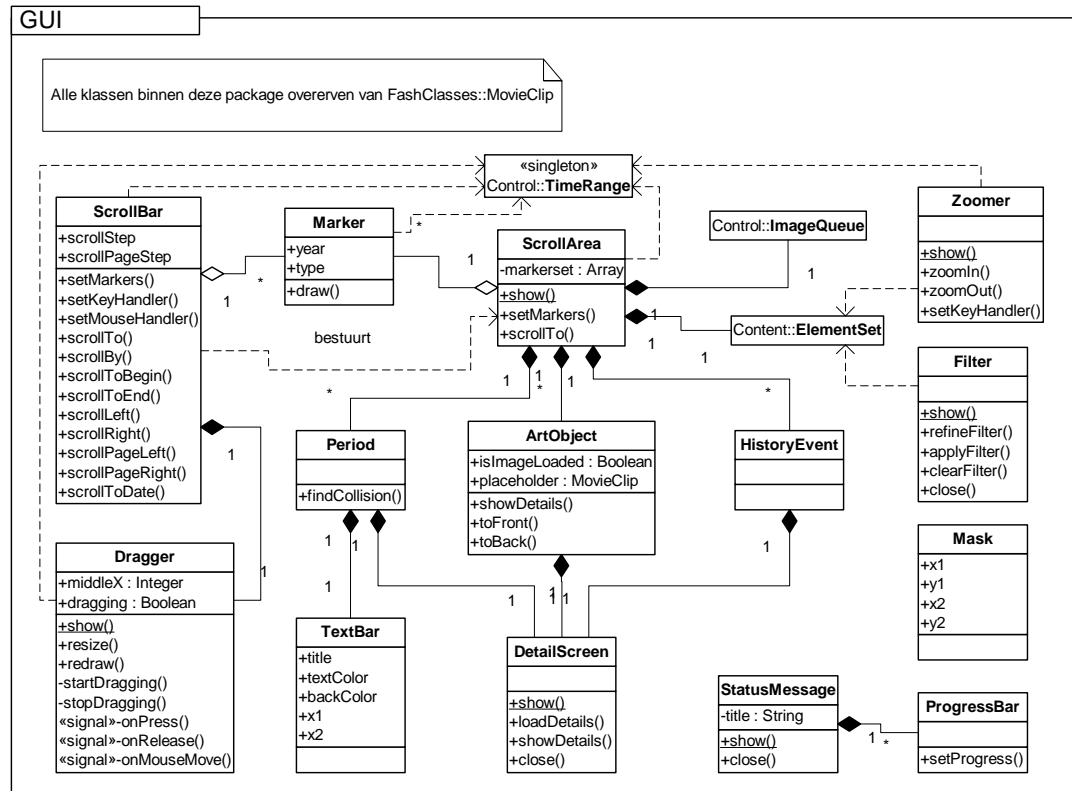
6.1 Content package

Onderstaande afbeelding toont het *content* package. Dit package bevat de content, oftewel de inhoud van de tijdslijn.



6.2 GUI package

De onderstaande afbeelding toont de inhoud van het package *GUI*. Deze package bevat alle interface elementen waarmee de tijdslijn bestuurd kan worden.



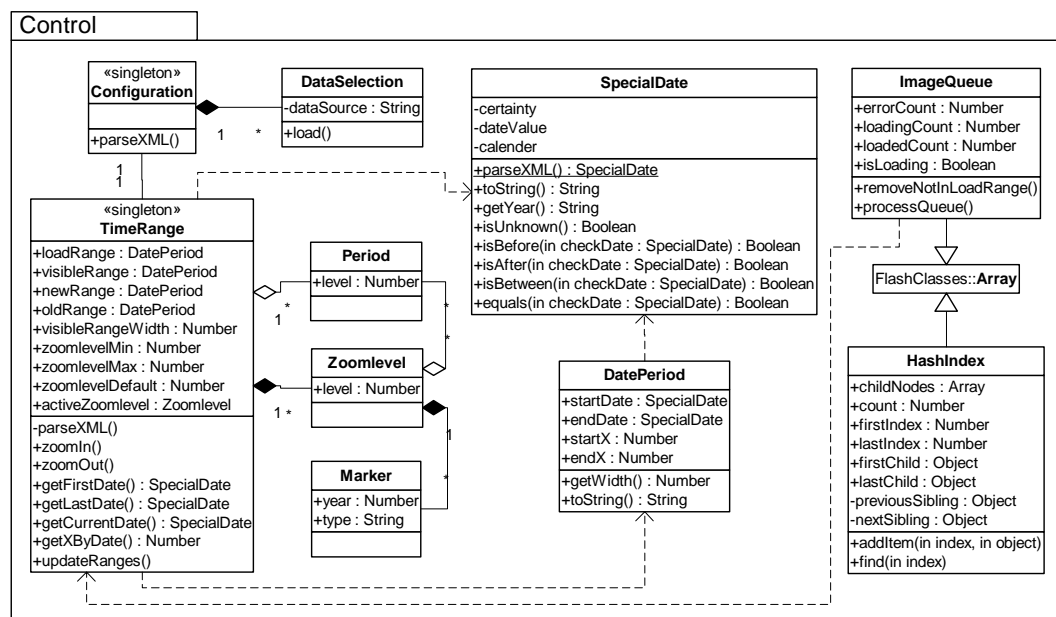
Zoals in het schema te zien is, wordt er door een groot aantal classes verwezen naar TimeRange. **Deze class is inmiddels hernoemt naar ElasticRange**

6.3 Scrollfunctionaliteit

De ScrollBar kan op de volgende manieren bestuurd worden:

commando	acties ScrollArea en ScrollDragger
slepen van de dragger met linker muisknop	ScrollArea volgt de bewegingen die de gebruiker met de ScrollDragger maakt.
neerwaarts scrollen met het scrollwiel van de muis	naar rechts. Stap is afhankelijk van de afstand die het scrollwiel heeft afgelegd.
opwaarts scrollen met het scrollwiel van de muis	naar links. Stap is afhankelijk van de afstand die het scrollwiel heeft afgelegd.
muisklik ScrollButtonLeft	klein stapje naar links
muisklik ScrollButtonRight	klein stapje naar rechts
muisklik op ScrollBar	(grote) sprong naar de nieuwe plaats
muisklik op Marker	sprong naar dat jaartal
toets Links (←)	klein stapje naar links
toets Rechts (→)	klein stapje naar rechts
toets PageUp (PgUp)	grote stap naar links
toets PageDown (PgDn)	grote stap naar rechts
toets Home	naar het begin van de tijdslijn
toets End	naar het eind van de tijdslijn

6.4 Control package



De classes binnen de control-package worden veelvuldig gebruikt door de classes in de andere packages.

De genoemde TimeRange heb ik geïntroduceerd als laag tussen de jaartallen en x-coördinaten op de tijdslijn. **Deze class is inmiddels hernoemt naar ElasticRange.**

De HashIndex wordt gebruikt door ElementSet om content geïndexeerd op jaartal te kunnen bewaren.

De ImageQueue handelt het laden van afbeeldingen af. Deze class wordt gebruikt om op een efficiënte manier de grote hoeveelheid afbeeldingen voor de ArtObjects te laden.

SpecialDate is de class die ingezet wordt voor het omgaan met datums volgens het ISO8601/Dublin Core formaat (zie analyse en definitie)

‘De tijdslijn voor de nieuwe website van het Rijksmuseum’

Bijlage IV

“Technisch Ontwerp”

laatste wijziging:
24 maart 2004
Rick Companje

INHOUDSOPGAVE

1.	INLEIDING	3
2.	PACKAGES	4
3.	SOFTWARE CLASSES IN PACKAGE 'CONTROL'	5
3.1	Configuration.....	5
3.2	DataSelection	6
3.3	SpecialDate.....	6
3.4	DatePeriod.....	7
3.5	SpecialDate.....	7
3.6	ImageQueue.....	8
3.7	HashIndex	9
3.8	ElasticRange (Elastische tijdvakken)	10
3.9	Beschrijving van de overige software classes	11
4.	SOFTWARE CLASSES IN PACKAGE 'GUI'	12
4.1	Zoomer.....	12
4.2	ScrollBar.....	13
4.3	Dragger.....	14
4.4	Detailscherm	15
4.5	Overige software classes	15
5.	SOFTWARE CLASSES IN PACKAGE 'CONTENT'	16
5.1	Beschrijving van de software classes	16
6.	GEBRUIK VAN BESTAANDE COMPONENTEN	17
MovieClip		17
Array		17
PopupManager		17
ProcessBar.....		17
7.	RAADPLEGEN DATARESOURCES	18
7.1	Spaces voor de tijdslijn	18
7.2	XML collecties.....	19
7.3	Schematische weergave van content-elements	20
7.4	XML communicatie in Flash	20
XML Connector.....		20
XML Socket Server		21
XML class in Flash.....		21
7.5	Meertalige ondersteuning.....	21
7.6	Datastructuren binnen de tijdslijn.....	21
8.	AANROEPEN EXTERNE RESOURCES.....	22

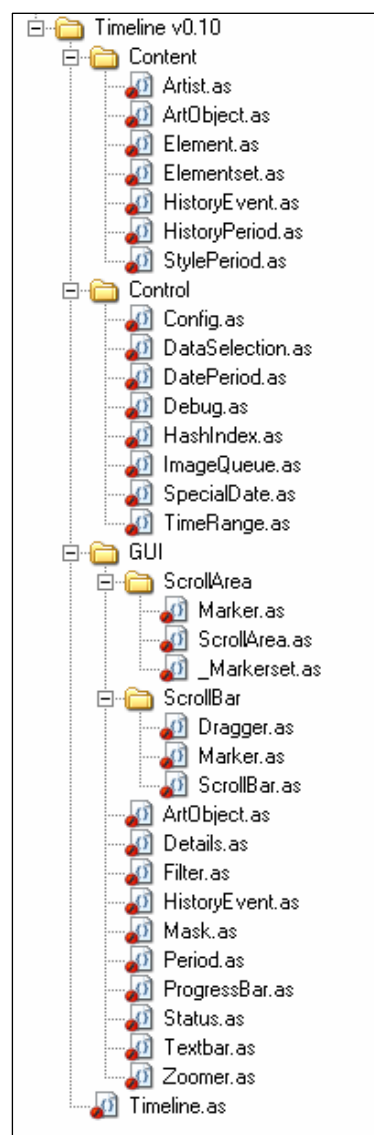
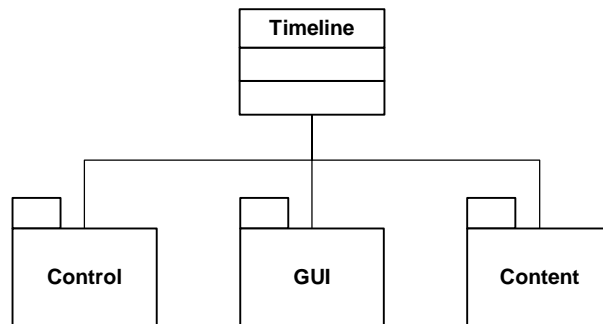
1. INLEIDING

In het technische ontwerp zijn alle objectclasses uit het klassendiagram technisch beschreven. Daarnaast ben ik dieper ingegaan op het benaderen van de XML database en heb ik vastgelegd op welke wijze externe resources (bijv. afbeeldingen) benaderd konden worden. Ook heb ik me tijdens deze fase verdiept in software-patterns.

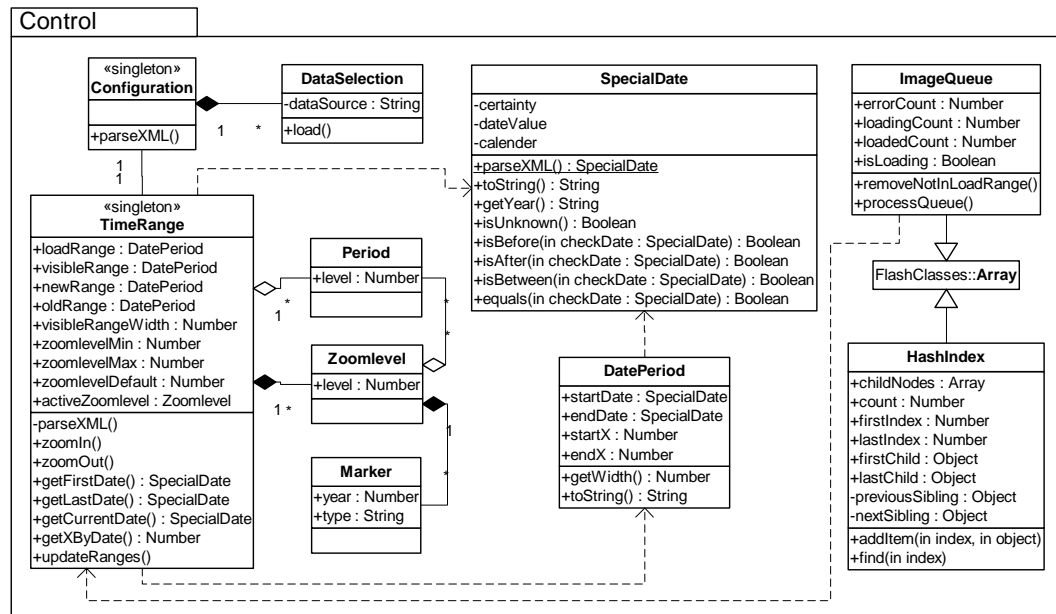
De technische beschrijving van een aantal classes zijn uit deze publieke versie van het technisch ontwerp weggelaten.

2. PACKAGES

De tijdslijn is opgedeeld in een drietal packages. De packages groeperen een aantal bij elkaar horende classes (klassen). Het opdelen in packages maakt de tijdslijn (en ook het model daarvan) stukken overzichtelijker.



3. SOFTWARE CLASSES IN PACKAGE 'CONTROL'



3.1 Configuration

Onderstaande programma code is een voorlopige versie van de configuration class. Deze code wordt nog herzien.

```

static class Configuration {
    private var xmlFormat:XML;
    private var dispatchEvent:Function;

    function parseXML(parent:MovieClip, filename:String) {
        xmlFormat = new XMLExt(this, "onLoadXML");
        XMLNode.prototype=new XMLNodeExt();
        xmlFormat.load(filename);
    }

    function onLoadXML(xmlData, success:Boolean) {
        if (success) {
            root.scopeWidth = Number(xmlData.firstChild.
                getChildNodeByName("scope").attributes.width);
            root.
                zoomlevelMin =
            Number(xmlData.firstChild.getChildNodeByName("zoomlevels").attributes.min);
            root.zoomlevelMax = Number(xmlData.firstChild.
                getChildNodeByName("zoomlevels").attributes.max);
            root.zoomlevelDefault = Number(xmlData.firstChild.
                getChildNodeByName("zoomlevels").attributes["default"]);
            root.timeRangeFilename = xmlData.firstChild.
                getChildNodeByName("timerange").attributes.src;
            root.dataselections = new Array();
            root.loadRangeWidth = xmlData.firstChild.
                getChildNodeByName("loadrange").attributes.width;
            root.viewDate = SpecialDate.parseDate(xmlData.firstChild.
                getChildNodeByName("viewdate"));
            var dataselectionNodes = xmlData.firstChild.
                getChildNodeByName("dataselection").childNodes;
            for (var i in dataselectionNodes) {
                root.dataselections.push(dataselectionNodes[i].attributes.src)
            }
            root.dispatchEvent({type:"onFormatLoaded"});
        }
        else {
            trace("format error");
        }
    }
}
  
```

Onderstaande XML is een voorlopige versie van een configuratie bestand voor de tijdslijn.

```
<?xml version="1.0"?>
<format title="meesterwerken">
  <scope width="600"/>

  <interfacesettings
    zoomer="true"
    userfilter="true"
    artobjects="true"
    artists="true"
    historyperiods="true"
    historyevents="true"
    styleperiods="true"
  />
  <zoomlevels min="1" max="4" default="2"/>
  <timerange src="xml/timeRange.xml"/>
  <loadrange width="800"/>
  <viewdate value="1650"/>
  <images location="http://devrijks.q42.net/aria/" />
  <dataselection>
    <data src="xml/list-rick.php.xml"/>
  </dataselection>
</format>
```

3.2 DataSelection

De dataselecties worden gemaakt met XOPUS, het content management systeem waarmee de totale website zal worden onderhouden. Dit valt buiten de grenzen van het afstudeerproject. De dataselectie wordt bewaard in een aparte space zodat deze gecached kon worden. Hierdoor zal de tijdslijn niet bij iedere request een complexe dataselectie query opleveren.

Het onderstaande schema toont de benodigde spaces voor de tijdslijn. Referenties naar elkaar zijn hier in opgenomen.

3.3 SpecialDate

Alle historische datums die door de tijdslijn gebruikt worden, worden opgeslagen als datatype SpecialDate.

SpecialDate is gebaseerd op de ISO 8601 standaard. Deze standaard beschrijft een internationale datumnotatie volgens het volgende formaat:

```
YYYY-MM-DDThh:mm:ss.sTZD (bijv. 2004-07-16T19:20:30.45+01:00)
```

Aangezien de meeste datums waarmee gewerkt onnauwkeurig zijn worden in de meeste gevallen alleen het jaar, en heel soms nog maand en dag gebruikt. De mogelijkheid tot het aanduiden van een exacte tijd wordt zeker niet gebruikt.

Voor het bijhouden van de zekerheidsgraad van een historische datum en het vastleggen van de bijbehorende kalendereenheid wordt gebruik gemaakt van de DublinCore standaard voor datumnotaties. Deze standaard beschrijft in XML formaat een aan te bevelen notatiewijze:

```
<date calender="AD" certainty="ca" value="1551"/>
```

De zekerheidsgraad van een datum wordt vermeld in het *certainty* attribuut. Een aantal mogelijke waarden voor certainty zijn: exact, ca (circa), approx, before, after, early, late en mid.

De tijdslijn zal in eerste instantie alleen ingezet voor content-elementen met een datumzekerheid van exact of circa.

De DublinCore standaard voorziet ook in vastleggen van datums uit andere kalenders. Hoewel de tijdslijn in eerste instantie alleen hoeft om te gaan met data uit onze Gregoriaanse kalender (AD Anno Domini of in het Engels CE Common Era) zijn toch voorbereidingen getroffen om ook museale objecten uit andere kalenders te kunnen opnemen. Dit kan zeker

nodig zijn wanneer het Rijksmuseum haar collectie uitbreidt met kunstobjecten van voor het jaar nul (BC - Before the Common Era / Before Christ).

De class SpecialDate bevat een statische functie parseXML die een instantie van SpecialDate teruggeeft, geparst uit een XML Node

```
static function parseDate(xmlNode:XMLNode):SpecialDate {  
    var tmpDate:SpecialDate = new SpecialDate();  
    tmpDate.dateValue = xmlNode.attributes.value;  
    tmpDate.certainty = xmlNode.attributes.certainty;  
    tmpDate.calender = xmlNode.attributes.calender;  
    return tmpDate;  
}
```

De constructor van SpecialDate heeft de volgende parameters:

```
function SpecialDate(inValue,inCertainty,inCalender) {  
    dateValue = inValue;  
    certainty = inCertainty;  
    calender = inCalender;  
}
```

3.4 DatePeriod

```
function DatePeriod(start,end) {  
    if ((typeof(start)=="number") && (typeof(end)=="number")) {  
        startX = (start>0)?start:0;  
        startDate = root.timeRange.getDateByX( startX);  
        endX = (end< root.timeRange.getMaxX())?end: root.timeRange.getMaxX();  
        endDate = root.timeRange.getDateByX( endX)  
    }  
    else if ((typeof(start)=="object") && (typeof(end)=="object")) {  
        startX = root.timeRange.getX(start);  
        startDate = root.timeRange.getDateByX( startX);  
        endX = root.timeRange.getX(end);  
        endDate = root.timeRange.getDateByX( endX);  
    }  
}
```

3.5 SpecialDate

SpecialDate is de class die ingezet wordt voor het omgaan met datums volgens het formaat dat ik bepaald heb in de gegevensanalyse

Zie ook functioneel ontwerp.

```
class SpecialDate {  
  
    //Private  
    private var    certainty:String;  
    private var    dateValue:String;  
    private var    calender:String;  
    private var    isUnknown:Boolean;  
    private var    dateRange:Array;  
  
    //Constructor  
    function SpecialDate(inValue,inCertainty,inCalender) {  
        dateValue = inValue;  
        certainty = inCertainty;  
        calender = inCalender;  
    }  
  
    //Getters & Setters  
    public function get certainty():String {  
        if ( dateValue.indexOf("/")>-1) certainty="range";  
        else if ( certainty==undefined) certainty="exact";  
        return ( certainty);  
    }  
    public function set certainty(tmpvalue:String) { _certainty = tmpvalue; }
```

```
public function get dateValue():String {
    return ( dateValue!=undefined? dateValue:"?");
}

public function set dateValue(value:String) {
    dateValue = value;
    if (certainty=="range") {
        dateRange = dateValue.split("/");
        dateValue = dateRange[0];
    }
    isUnknown = ( dateValue==undefined)
}

public function get calender():String { return
( calender!=undefined? calender:"AD"); }
public function set calender(tmpvalue:String) { calender = tmpvalue; }

public function get isUnknown():Boolean {
    return isUnknown;
}

//Methods
public function getFriendlyDate():String {
    return dateValue;
}
function isCertaintyExact():Boolean {
    return (certainty=="exact");
}
function isCertaintyCirca():Boolean {
    return (certainty=="circa");
}
function isCertaintyRange():Boolean {
    return (certainty=="range");
}
function getYear():Number {
    return ((dateValue.indexOf("-")>-1)?Number(dateValue.split("-"
)[0]):Number(dateValue));
}
function isBefore(checkDate:SpecialDate) {
    return (this.getYear()<checkDate.getYear())
}
function isAfter(checkDate:SpecialDate) {
    return (!isBefore(checkDate))
}
function isBetween(checkDate1:SpecialDate, checkDate2:SpecialDate) {
    return (isAfter(checkDate1) && isBefore(checkDate2));
}
function equals(checkDate:SpecialDate) {
    return (this.getYear()==checkDate.getYear());
}
function toString() {
    return this.getYear();
}

//Static functions
static function parseDate(xmlNode:XMLNode):SpecialDate {
    var tmpDate:SpecialDate = new SpecialDate();
    tmpDate.dateValue = xmlNode.attributes.value;
    tmpDate.certainty = xmlNode.attributes.certainty;
    tmpDate.calender = xmlNode.attributes.calender;
    return tmpDate;
}
}
```

3.6 ImageQueue

De ImageQueue handelt het laden van afbeeldingen af. Deze class wordt gebruikt om op een efficiënte manier de grote hoeveelheid afbeeldingen voor de ArtObjects te laden.

```
import mx.transitions.Tween;
import mx.transitions.easing.Strong;

class LoadQueue {
    public var imagebuffer:Array = new Array();
    private var dispatchEvent:Function;

    private var loadingCount:Number = 0;
    private var errorCount:Number = 0;
```

```

function LoadQueue(parent:MovieClip) {
    root.addEventListener("onStopScroll", this)
    root.addEventListener("onStartScroll", this)
    root.addEventListener("onUpdateScreen", this)
}
function onUpdateScreen() {
    if (imagebuffer.length>0) {
        // eerst alles verwijderen wat toch niet zichtbaar wordt maar nog wel in de
queue staat
        for (var i=0; i<imagebuffer.length; i++) {
            if (!imagebuffer[i].isInLoadRange()) {
                this["imagebuffer"].splice(i,1);
            }
        }
        if (loadingCount<25) processQueue()
        root.debug.imagebuffer.setProgress(imagebuffer.length, 200)
        root.debug.loadingCount.setProgress(loadingCount, 200)
        root.debug.errorCount.setProgress(errorCount, 200)
    }
}
function processQueue() {
    // daarna alles tonen wat al zichtbaar zou moeten zijn
    for (var i=0; i<imagebuffer.length; i++) {
        if (imagebuffer[i].isInVisibleRange()) {
            loadImage(imagebuffer[i]);
            this["imagebuffer"].splice(i,1);
            if (i>40) break; //max 40 images per frametijd laden
        }
    }
    // daarna pas alles wat bijna zichtbaar is
    for (var i=0; i<imagebuffer.length; i++) {
        if (imagebuffer[i].isInLoadRange() &&
!imagebuffer[i].isInVisibleRange()) {
            loadImage(imagebuffer[i]);
            this["imagebuffer"].splice(i,1);
            if (i>40) break; //max 40 images per frametijd laden
        }
    }
}
function addImage(target:MovieClip) {
    imagebuffer.push(target)
}
function onLoadError(target:MovieClip, errorCode) {
    //trace("load error: " + errorCode)
    target.parent.timelineElement.hasImage = false;
    loadingCount--;
    errorCount++;
}
function onLoadInit(target:MovieClip) {
    //new Tween(target, "_alpha", Strong.easeOut, 0, 100, Math.random()*50);
    target.parent.timelineElement.onImageLoaded();
    loadingCount--;
}
function loadImage(target) {
    if (!target.isImageLoaded) {
        loadingCount++;
        var mcLoader:MovieClipLoader = new MovieClipLoader();
        mcLoader.addListener(this);
        mcLoader.loadClip(target.image, target.placeholder);
    }
}
}

```

3.7 HashIndex

```

class HashIndex extends Array {
    public var firstitem;
    public var count:Number = 0;

    function addItem(index, object):Object {
        var leftside, rightside;
        var newitem = {index:index, object:object}
        count++;

        if (!this[index]) this[index] = new Array();
        newitem.childindex = this[index].length;
    }
}

```

```

        this[index].push(newitem)

        if (firstitem==undefined) {
            firstitem = newitem;
            this.firstChild = this[index][0];
        }
        else {
            //this[index].firstChild and this[index].lastChild
            this[index].firstChild = this[index][0];
            this[index].lastChild = this[index][this[index].length-1];

            //if necessary update this.firstChild and this.lastChild
            if (index < this.firstChild.index) this.firstChild =
this[index].firstChild;
            if (index > this.lastChild.index) this.lastChild =
this[index].lastChild;

            //find position
            for (var cursor = firstitem; cursor.nextSibling; cursor =
cursor.nextSibling)
                if (cursor.nextSibling.index > newitem.index) break;

            //check for being last item and update siblings
            if (rightside = cursor.nextSibling) {
                rightside.previousSibling = newitem;
                newitem.nextSibling = rightside;
            }
            leftside = cursor;
            leftside.nextSibling = newitem;
            newitem.previousSibling = leftside;
            return newitem;
        }
    }
    function find(index) {
        //trace(this.firstChild.index)
        var tmp = index;
        var ttl:Number = 5000;
        if (!this[index]) {
            while(!this[index] && ttl-->1) index++;
            //trace(tmp + " " + index)
            //if (ttl==0) trace("TTL=0 in HashIndex.find(" + tmp + ")");
        }
        //trace(tmp + " " + index)
        return this[index];
    }
}

```

3.8 ElasticRange (Elastische tijdvakken)

Een grote hoeveelheid van de collectie van het Rijksmuseum bestaat uit kunstobjecten uit de Gouden Eeuw (17^e eeuw). Op de tijdslijn is dit een drukke periode. Er zijn ook periodes uit de geschiedenis waarvan het Rijksmuseum minder kunstwerken bezit.

Om te voorkomen dat sommige plekken op de tijdslijn overvol raken, en andere plekken te leeg lijken is in de fase functioneel ontwerp het principe elastische tijdvakken (ElasticRange) functioneel beschreven.

Bij dit principe wordt de tijdslijn onderverdeeld in een aantal periodes waarbij per periode een bepaalde hoeveelheid ruimte gereserveerd kan worden.

In dit technisch ontwerp worden een aantal extra beschrijvingen gegeven over de technische werking.

Onderstaande crosstable toont de drie dimensies van de ElasticRange.

zoomlevel \ tijdvak	level 1	level 2	level 3	level 4
1400 / 1600	150	1000	2000	4000
1600 / 1700	300	2000	4000	6000
1700 / 2000	150	1000	2000	4000
Totale breedte	600	4000	8000	14000

De tijdvakken worden handmatig geconfigureerd in een XML bestand:

```
<?xml version="1.0" encoding="UTF-8" ?>
<timerange>
  <period>
    <start value="1400"/>
    <end value="1600"/>
    <zoomlevel level="1" interval="100" subinterval="50" width="150"/>
    <zoomlevel level="2" interval="50" subinterval="5" width="1000"/>
    <zoomlevel level="3" interval="25" subinterval="5" width="2000"/>
    <zoomlevel level="4" interval="25" subinterval="5" width="4000"/>
  </period>
  <period>
    <start value="1600"/>
    <end value="1700"/>
    <zoomlevel level="1" interval="25" subinterval="5" width="300"/>
    <zoomlevel level="2" interval="5" subinterval="1" width="2000"/>
    <zoomlevel level="3" interval="5" subinterval="1" width="4000"/>
    <zoomlevel level="4" interval="5" subinterval="1" width="6000"/>
  </period>
  <period>
    <start value="1700"/>
    <end value="2000"/>
    <zoomlevel level="1" interval="100" subinterval="50" width="150"/>
    <zoomlevel level="2" interval="50" subinterval="5" width="1000"/>
    <zoomlevel level="3" interval="25" subinterval="5" width="2000"/>
    <zoomlevel level="4" interval="25" subinterval="5" width="4000"/>
  </period>
</timerange>
```

Met de volgende formule wordt de vertaalslag van datum naar x-coördinaat gemaakt:

```
function getX(datum:SpecialDate):Number {
  var p = getPeriodByDate(datum);
  if (p) {
    var x1 = p.startDate.getYear();
    var x2 = p.endDate.getYear();
    var y1 = p.zoomlevels[zoomlevel].startX;
    var y2 = Number(p.zoomlevels[zoomlevel].startX) +
              Number(p.zoomlevels[zoomlevel].width);
    return Math.round((datum.getYear() * ((y2-y1)/(x2-x1)) +
                      (y1-((y2-y1)/(x2-x1)) * x1));
  }
  else {
    trace("fout in getX. Periode niet gevonden");
    trace("zoomlevel: "+zoomlevel);
    trace("datum: "+datum.dateValue);
  }
}
```

uses:

```
function getPeriodByDate(datum:SpecialDate):Object {
  for (var i=0;i<periods.length;i++) {
    if (datum.isBetween(periods[i].startDate, periods[i].endDate) ||
        datum.equals(periods[i].endDate))
      return(periods[i]);
  }
  trace("fout in getPeriodByDate: "+datum.dateValue);
  return(null)
}
```

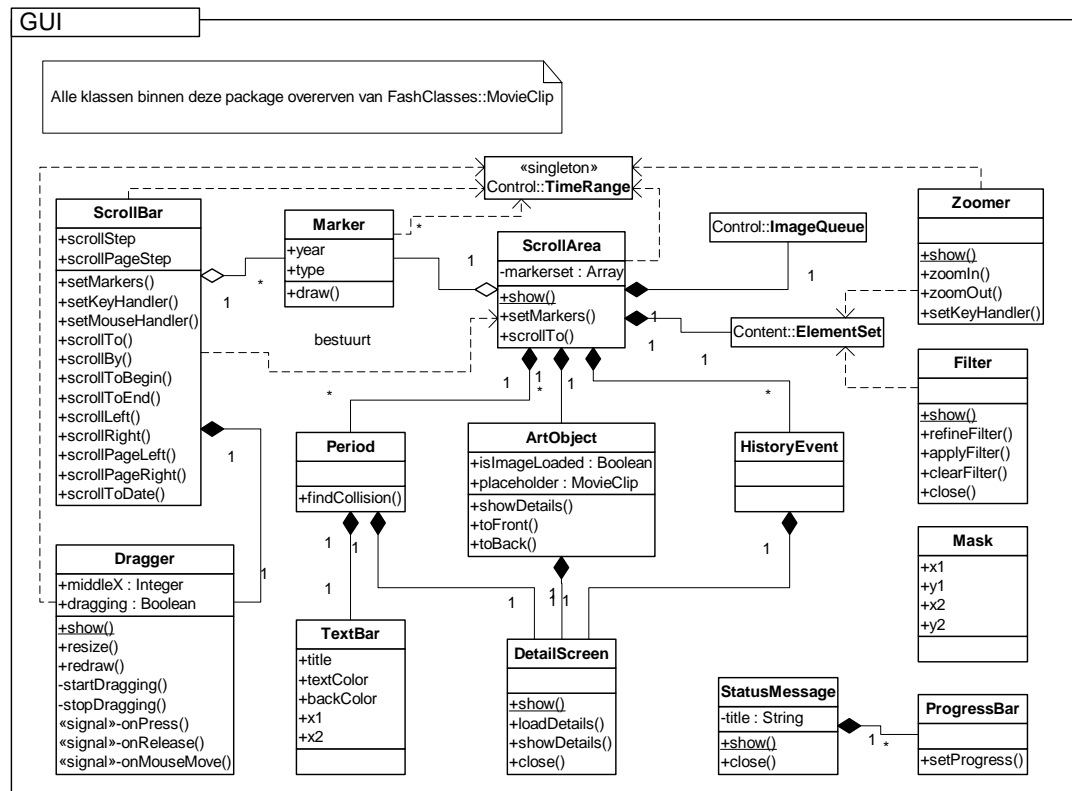
ElasticRange bevat nog de embedded classes:

- Period
- Zoomlevel
- Marker

3.9 Beschrijving van de overige software classes

De technische beschrijving van een aantal classes zijn uit deze publieke versie van het technisch ontwerp weggelaten.

4. SOFTWARE CLASSES IN PACKAGE 'GUI'



4.1 Zoomer

De configuratie van de zoomlevels vindt plaats in het XML bestand van ElasticRange.

```

class Zoomer extends MovieClip {
    static function create(parent:MovieClip) {
        return parent.attachMovie("Zoomer", "zoomer",
parent.getNextHighestDepth())
    }
    function Zoomer() {
        x = root.scopeWidth- width;
        y = 30;
        this["zoombuttonMin"].onPress = zoomOut;
        this["zoombuttonPlus"].onPress = zoomIn;
    }
    function zoomOut() {
        if (!root.timeline.scroller.scrolling) {
            var previousDate = root.timeline.scroller.getCurrentDate()
            root.timeRange.zoomOut();
            root.timeline.scrollBar.scrollToDate(previousDate);
        }
    }
    function zoomIn() {
        if (!root.timeline.scroller.scrolling) {
            var previousDate = root.timeline.scroller.getCurrentDate()
            root.timeRange.zoomIn();
            root.timeline.scrollBar.scrollToDate(previousDate);
        }
    }
}

```

4.2 ScrollBar

D tijdslijn is voorzien van een ScrollBar (schuifbalk) en een ScrollArea. Deze twee GUI elementen maken het mogelijk dat er naar links en naar rechts door de tijd 'gescrolled' kan worden.

commando	acties ScrollArea en ScrollDragger
slepen van de dragger met linker muisknop	ScrollArea volgt de bewegingen die de gebruiker met de ScrollDragger maakt.
neerwaarts scrollen met het scrollwiel van de muis	naar rechts. Stap is afhankelijk van de afstand die het scrollwiel heeft afgelegd.
opwaarts scrollen met het scrollwiel van de muis	naar links. Stap is afhankelijk van de afstand die het scrollwiel heeft afgelegd.
muisklik ScrollButtonLeft	klein stapje naar links
muisklik ScrollButtonRight	klein stapje naar rechts
muisklik op ScrollBar	(grote) sprong naar de nieuwe plaats
muisklik op Marker	sprong naar dat jaartal
toets Links (←)	klein stapje naar links
toets Rechts (→)	klein stapje naar rechts
toets PageUp (PgUp)	grote stap naar links
toets PageDown (PgDn)	grote stap naar rechts
toets Home	naar het begin van de tijdslijn
toets End	naar het eind van de tijdslijn

De ScrollBar is voorzien van een Dragger (meesleper). De dragger kan met de muis of het toetsenbord naar links en naar rechts worden verplaatst. De ScrollArea volgt de bewegingen van de ScrollBar.

De ScrollBar voor de tijdslijn imiteert zoveel mogelijk het gedrag van een windows ScrollBar.

```
class ScrollBar extends MovieClip {
    public var scrollDragger:ScrollDragger;
    private var scrollBarMarkers:Array = new Array();
    public var scrollStep:Number = 10;
    public var scrollPageStep:Number = 30;

    static function create(parent:MovieClip) {
        return parent.attachMovie("ScrollBar", "scrollBar",
parent.getNextHighestDepth());
    }
    function ScrollBar() {

        width = root.scopeWidth+3;
        setScrollBarMarkers();
        scrollDragger = ScrollDragger.create( parent);
        scrollTo( root.viewDate)

    }

    function scrollBy(value:Number) { scrollTo(scrollDragger.middleX+value); }
    function scrollToBegin() { scrollTo(0); }
    function scrollToEnd() { scrollTo( root.scopeWidth) }
    function scrollLeft() { scrollBy(-scrollStep) }
    function scrollRight() { scrollBy(scrollStep) }
    function scrollPageLeft() { scrollBy(-scrollPageStep) }
    function scrollPageRight() { scrollBy(scrollPageStep) }
    function onPress() { scrollTo( xmouse) }

    //function onEnterFrame() {
    //    if (_root.timeline.scroller.scrolling) sound.start();
    //}

    function scrollTo(value:Number) {
        scrollDragger.middleX = value;
        scrollDragger.update();
    }
}
```

```

    function scrollToDate(value:SpecialDate) {
        scrollTo( root.scopeWidth / root.timeRange.getMaxX() *
root.timeRange.getX(value))
    }

    function setScrollBarMarkers() {
        var startYear:Number =
Number( root.timeRange.periods[0].startDate.getYear());
        var endYear:Number =
Number( root.timeRange.lastPeriod.endDate.getYear());
        for (var year = startYear; year<=endYear; year+=100) {
            ScrollBarMarker.create(this,year)
        }
    }
}

```

4.3 Dragger

```

import mx.transitions.Tween;
import mx.transitions.easing.Strong;

class ScrollDragger extends MovieClip {
    public var dragging:Boolean;

    static function create(parent:MovieClip) {
        return parent.attachMovie("ScrollDragger","scrollDragger",
parent.getNextHighestDepth());
    }
    function ScrollDragger() {
        root.addEventListener("onTimeRangeChanged", this);
        y = 2;
        draw();
    }
    function get middleX() {
        return x+ width/2
    }
    function set middleX(value:Number) {
        value = value - width/2;
        if (value<0) value=0
        if (value+ width> root.scopeWidth) value = root.scopeWidth- width;
        x = value;
    }
    function onTimeRangeChanged() {
        var previousMiddleX = middleX;
        var newWidth = root.scopeWidth * root.timeRange.visibleRange.width /
root.timeRange.getMaxX();
        new Tween(this, " width", Strong.easeOut, width, newWidth, 10);
        new Tween(this, "middleX", Strong.easeOut, middleX, previousMiddleX,
10);
        update();
    }
    function update() {
        var scrollerX = middleX * root.timeRange.getMaxX() / root.scopeWidth;
        root.timeline.scroller.scrollTo(scrollerX);
        if ( width<80) {
            this["title txt"].text =
( root.timeRange.getDateByX(scrollerX).getYear());
        }
        else {
            this["title txt"].text =
( root.timeRange.visibleRange.startDate.getYear() + "-" +
root.timeRange.visibleRange.endDate.getYear());
        }
    }
    function onPress() {
        startDrag(this,false,0,this.y, root.scopeWidth- width, this.y);
        dragging = true;
    }
    function onRelease() {
        stopDrag();
        dragging = false;
    }
    function onMouseMove() {
        if (dragging) update();
    }
    function draw() {
    }
}

```

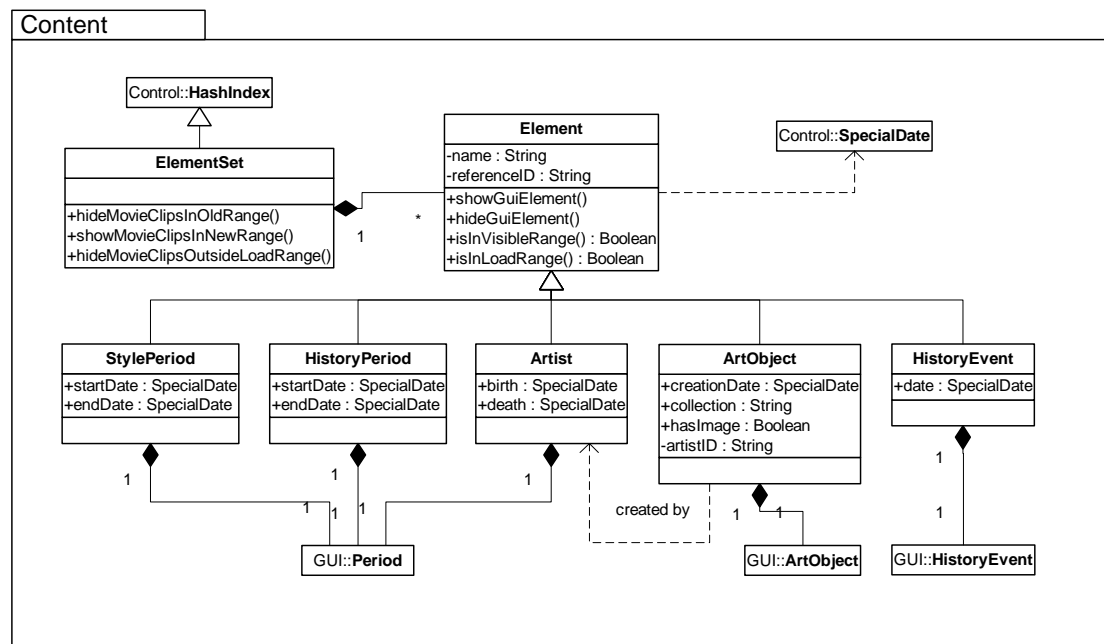
4.4 Detailscherm

Bij het opvragen van het detailscherm wordt a.d.h.v een nieuwe XML Query de benodigde detailgegevens gedownload. Parameter bij deze query is "ReferenceID" uit de space "Screen"

4.5 Overige software classes

De technische beschrijving van een aantal classes zijn uit deze publieke versie van het technisch ontwerp weggelaten.

5. SOFTWARE CLASSES IN PACKAGE 'CONTENT'



5.1 Beschrijving van de software classes

De technische beschrijving van een aantal classes zijn uit deze publieke versie van het technisch ontwerp weggelaten.

6. GEBRUIK VAN BESTAANDE COMPONENTEN

- van welke bestaande klassen/componenten ik gebruik maak
- of juist van welke niet en waarom.

MovieClip

- bestand wordt er niet groot door
- enige class die visuele/grafische dingen kan
- alle andere grafische dingen erven over van movieclip

Array

- wordt ge-extend door HashIndex en door ElementSet

PopupManager

- toch niet gebruikt,
- maakt swf te groot,
- zelf net zo makkelijk te maken en dan is ie wel klein.

ProcessBar

- toch niet gebruikt,
- maakt swf te groot,
- zelf net zo makkelijk te maken en dan is ie wel klein.

7. RAADPLEGEN DATARESOURCES

De informatie uit de Aria database wordt geconverteerd en wordt samen met de gegevens uit de ADLIB database gemigreerd in een nieuwe XML Database.

Het Aria-systeem is hierdoor na de implementatie van de nieuwe website overbodig geworden en wordt als zodanig opgeheven.

De ADLIB database blijft bestaan en wordt op gezette tijden gesynchroniseerd met de XML Database.

Q42 heeft voor de XML Database van het Rijksmuseum gebruik gemaakt van Oracle versie 9.

7.1 Spaces voor de tijdslijn

De XML Database is opgedeeld in een aantal spaces. Spaces zijn te vergelijken met tabellen uit een relationele database. Een aantal spaces in de nieuwe database zijn artists, artobjects, screens, catalogs, dictionary en encyclopedia. Een space bestaat uit een collectie van XML documenten met per document een eigen unieke ID, dat wil zeggen, alle ID's binnen een space moeten uniek zijn, daarbuiten mogen dubbelingen voorkomen.

De XML documenten binnen een collectie zouden als records beschouwd kunnen worden met een ID als primaire sleutel. Een groot verschil is echter tussen een tabel en space dat een tabel een vaste definitie heeft met een vooraf bepaald aantal datavelden terwijl binnen de spaces vrij genest kan worden. Ook kunnen er referenties opgenomen worden naar documenten in andere spaces.

Het grote voordeel van het kunnen nesten is dat de database niet zal gaan bestaan uit talloze koppeltabellen zoals dat het geval zou zijn bij een relationele database. Een ander groot voordeel is dat er binnen spaces referenties naar elkaar opgenomen kunnen worden waardoor het bevragen van een XML database minder complexe queries vereist. Een XML database kent net als een relationele database indexen waardoor snel in de database gezocht kan worden. Bij een relationele database worden indexen aangemaakt op tabelvelden. Bij een XML database worden de indexen aangemaakt op elementen en hun attributen.

Space	Element	Datatype	Opmerking
artists	<u>ID</u>	String	uniek binnen space
	name	String	naam van kunstenaar
	birth	SpecialDate	geboortedatum van kunstenaar
	death	SpecialDate	overlijdensdatum van kunstenaar
	screenReference	String	referentie naar gegevens voor detailscherm
artObjects	<u>ID</u>	String	uniek binnen space
	name	String	naam van kunstwerk
	creationDate	SpecialDate	creëerdatum van kunstwerk
	collection	String	in welke collectie kunstwerk thuis hoort. Hierop moet gefilterd kunnen worden
	hasImage	Boolean	afbeelding beschikbaar ja/nee, de eventuele naam van de afbeelding wordt afgeleid van ID
	artistReference	String	referentie naar de maker
	screenReference	String	referentie naar gegevens voor

Space	Element	Datatype	Opmerking
			detailscherm
stylePeriods	<u>ID</u>	String	uniek binnen space
	name	String	naam van de periode
	startDate	SpecialDate	begin van de periode
	endDate	SpecialDate	eind van de periode
	screenReference	String	referentie naar gegevens voor detailscherm
historyPeriods	<u>ID</u>	String	uniek binnen space
	name	SpecialDate	naam van de periode
	startDate	SpecialDate	begin van de periode
	endDate	SpecialDate	eind van de periode
	screenReference	String	referentie naar gegevens voor detailscherm
historyEvents	<u>ID</u>	String	uniek binnen space
	name	SpecialDate	naam van de gebeurtenis
	date	SpecialDate	datum van de gebeurtenis
	screenReference	String	referentie naar gegevens voor detailscherm

De space *screen* is nodig voor het verkrijgen van de data voor het opbouwen van de detailschermen. De uiteindelijke structuur van deze space is nog niet bekend.

De totale database bestaat uit 400.000 kunsthistorische entrees (in eerste instantie werd gesproken over 100.000)

Het is duidelijk dat een client-sided web-toepassing zoals de tijdslijn nooit in staat zal zijn al deze gegevens te verwerken, daarom is bepaald dat er van server-sided dataselecties gebruikt gemaakt wordt.

De dataselectie wordt bewaard in een aparte space zodat deze gecached kon worden. Hierdoor zal de tijdslijn niet bij iedere request een complexe dataselectie query veroorzaken.

7.2 XML collecties

Tijdens de ontwikkeling van de tijdslijn zullen de gegevens worden verkregen uit een statische XML collectie, indien dat goed functioneert wordt de werkelijke XML database benaderd.

De structuur van XML bestanden waar de tijdslijn haar data uit put is als volgt:

```
<?xml version="1.0"?>
<ricksmuseum xmlns:s="http://www.rijksmuseum.nl/xmlns/shared/1.0" size="1021">

  <artObject id="AK-MAK-1117" hasImage="true" title="Tsuba" artistID="AT-155"
  refID="SC-122">
    <creationDate value="1600" certainty="circa">
  </artObject>

  <artist id="AT-151" fullname="Vincent van Gogh">
    <birthDate value="1853"/>
    <deathDate value="1879"/>
  </artist>

  <styleperiod id="SP-122" title="Barok" refID="SC-24">
    <startDate value="1600" certainty="circa"/>
    <endDate value="1750" certainty="circa"/>
    <reference space="artist" id="AT-12123" role="?" />
    <reference space="artist" id="..." role="?" />
    <reference space="artobjects" id="SK-15-aa-222" role="?" />
  </styleperiod>
</ricksmuseum>
```

```

    <reference space="artobjects" id="..." role="?"/>
  </styleperiod>

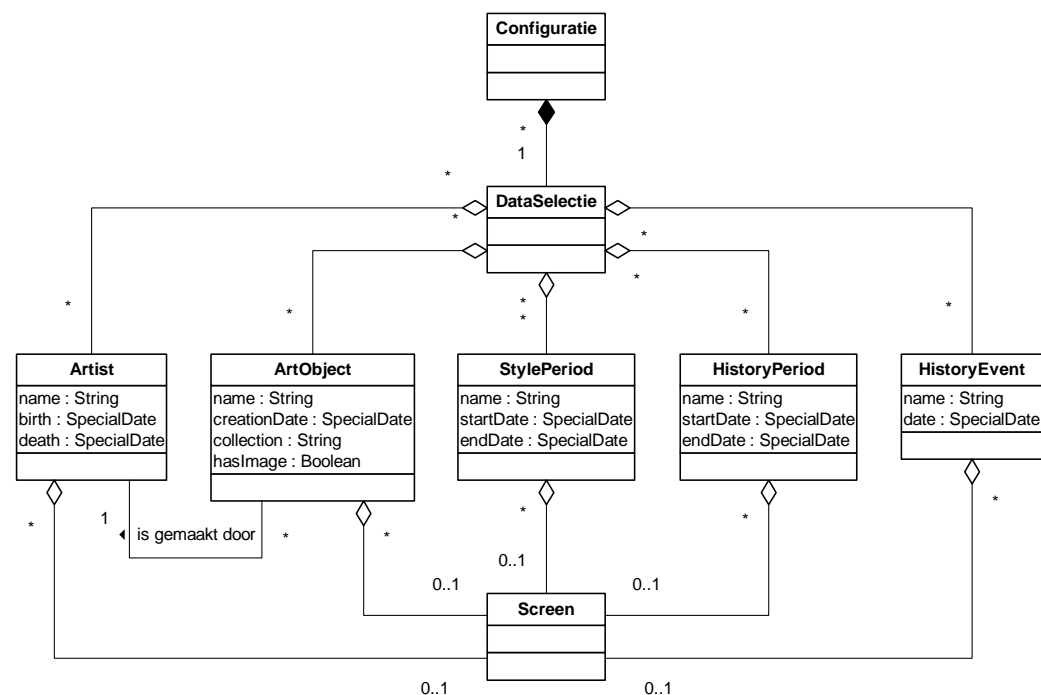
  <historyperiod id="HP-22" title="WOII" refID="SC-11">
    <startDate value="1939"/>
    <endDate value="1945"/>
  </historyperiod>

  <historyevent id="HE-12" title="De slag van november 1800">
    <date value="1800-11"/>
  </historyevent>

</ricksmuseum>

```

7.3 Schematische weergave van content-elements



7.4 XML communicatie in Flash

Vanuit Flash kan er op verschillende manieren gecommuniceerd worden met de XML Database.

- XML Connector
- XML Socket Server
- XML Class in Flash

XML Connector

nadelen:

- weinig documentatie

voordelen:

- nog niet bekend

```

var doc:mx.data.components.XMLConnector;
doc = new mx.data.components.XMLConnector();
doc.ignoreWhite = true;
doc.URL = "http://localhost/Timeline v0.01/periodes.xml";

```

```
doc.direction = "receive";
doc.addEventListener("send", sendFunction);
doc.addEventListener("status", statusFunction);
doc.params = new XML("<mydoc><name>Bob</name><city>Oakland</city></mydoc>");
doc.trigger();
```

XML Socket Server

nadelen:

- er moet een XML SocketServer draaien op de web- / dataserver

voordelen:

- waarschijnlijk sneller dan standaard XML object omdat de connectie 'open blijft'

voorbeeld:

```
var server:XMLSocket = new XMLSocket();
server.connect("localhost", 9875);
server.onConnect = serverConnected;
server.onXML = xmlReceived;
server.onClose = socketClosed;
server.send("<Text>Hi</Text>");
```

XML class in Flash

nadelen:

- je kunt niet rechtstreeks elementen benaderen. Je moet eerst door het hele bestand heen lopen. Je zou zoiets willen als `getElementByNodeName`.
- Je kunt de childs ook niet bij de naam aanspreken, maar allen bij het nummer in de array. Alleen dus `children[0].nodeName`
- waarschijnlijk langzamer dan een socket connectie

voordelen

- makkelijk te openen
- veel documentatie
- je kunt attributen (toch wel) bij de naam aanroepen:
`e.childNodes[0].attributes.calender`
- **later toegevoegd: XPath wordt ondersteund**

voorbeeld:

```
var doc:XML = new XML();
doc.ignoreWhite = true;
doc.onLoad = buildObjects;
doc.load("http://localhost/Timeline v0.01/periodes.xml");
```

7.5 Meertalige ondersteuning

De taal waarin de content verschijnt is afhankelijk van de dataselectie. De dataselectie wordt voor iedere taal apart gecached.

Tijdens het aanroepen van de tijdslijn wordt meegegeven in welke taal de website bekeken wordt. Aan de hand van dat gegeven wordt door de tijdslijn de juiste dataselectie uit de servercache opgevraagd. Alle interface-elementen zijn eveneens op de hoogte van de geselecteerde taal en zullen zichzelf in de juiste vorm presenteren.

7.6 Datastructuren binnen de tijdslijn

zie hashIndex

8. AANROEPEN EXTERNE RESOURCES

Afbeeldingen zullen on-the-fly worden gerenderd (en serversided worden gecached) door een component dat door Q42 wordt/is ontwikkeld.

Een image wordt via een URL als volgt opgevraagd:

```
/img/{imageID}?{templateName}
```

- imageID is de verwijzing naar de originele afbeelding
- templateName is de naam van een XML bestand waarin de beeldbewerkings opdrachten staan

voorbeeldtemplate 1:

```
<template output="png">
  <resize width="50" />
  <image src="mask.gif">
    <rotate angle="45" />
  </image>
  <resize />
</template>
```

voorbeeldtemplate 2:

```
<template output="png">
  <resize width="300" height="300" keepaspectratio="true" />
  <rotate angle="0" />
</template>
```

Een image die eenmaal gerenderd is wordt door de server gecached waardoor de beeldbewerkingen niet nogmaals uitgevoerd hoeven te worden.