# Project Information and Knowledge Engineering

# Fase 2

# 10-12-2011

## Groep 4

Ruben Visser 4081641

Olaf Maas 4086864 

Marieke van der Tuin 4079299

Inhoudsopgave

[Inleiding 3](#_Toc311235026)

[Doel 4](#_Toc311235027)

[Functionele omschrijving van het systeem 5](#_Toc311235028)

[Product fase 1 6](#_Toc311235029)

[Data 6](#_Toc311235030)

[Implementatie 6](#_Toc311235031)

[Product fase 2 8](#_Toc311235032)

[2e databron 8](#_Toc311235033)

[Input 8](#_Toc311235034)

[Output 8](#_Toc311235035)

[Komende fases 10](#_Toc311235036)

# Inleiding

In de periode van 16 t/m 25 november hebben Ruben, Olaf en Marieke samengewerkt om een eerste versie te maken van een muziek systeem, dat gebruikers muziek kan aanraden aan de hand van keuzes die vooraf worden ingevoerd.

Eerst is het globale doel voor dit systeem opgesteld. Dit doel beschrijft wat het systeem over 8 weken zou moeten kunnen. Dit zal gemaakt worden tijdens 4 periodes van elk 2 weken. Aan het einde van elke periode wordt een werkend product opgeleverd.

Vervolgens is er een lijst met functionele eisen van het programma opgesteld. Er is bepaald welke bronnen het systeem ongeveer zal gaan gebruiken, in welke programmeertalen het geschreven zal worden en hoe het systeem ongeveer zal werken.

In de eerste fase is een product gemaakt waarmee de genrevoorkeuren van gebruikers gebruikt worden om een lijst met hoog gewaardeerde artiesten te genereren welke aan de gebruiker worden getoond. Dit enkel met behulp van de MusicBrainz database.

In de tweede fase is dit product uitgebreid met een koppeling naar de Last.fm dataset. Naast de genrevoorkeuren kunnen gebruikers nu ook hun favoriete artiesten doorgeven. Daarnaast is er een log-in functie geïmplementeerd. De genre- en artiestvoorkeuren van de gebruiker worden daarmee ook opgeslagen, zodat dit een volgende keer niet opnieuw ingevoerd hoeft te worden. Ook is er een eigen waarderingsmethode geïmplementeerd: door middel van +1 (leuk) en -1 (niet leuk) kunnen gebruikers aangeven of zij een bepaalde suggestie wel of niet goed vonden.

In een later stadium zullen deze waarderingen van de gebruikers gekoppeld worden aan de gegeven suggesties om aan de hand van een genre-boom te bepalen welk (sub)genre gebruikers ook leuk zouden kunnen vinden.

# Doel

Er moet een systeem gebouwd worden dat mensen muziek kan aanraden die zij mogelijk leuk zullen vinden, aan de hand van de muziek die zij leuk vinden of andere opties die zij invoeren.

Hiervoor zal een gebruiker zijn profiel door kunnen geven. In dit profiel staat omschreven wat bijvoorbeeld de leeftijd en het geslacht is van de gebruiker, maar vooral wat zijn muziekvoorkeuren zijn.

Aan de hand van dit profiel en de waarderingen van andere gebruikers, zal het systeem een lijst genereren met daarin aanbevolen artiesten en nummers. Deze lijst bevat naast de naam of titel, ook links naar bijvoorbeeld YouTube of informatie over de artiest van Wikipedia.

Deze gegeneerde lijst kan vervolgens door de gebruiker beoordeeld worden. Deze beoordeling wordt gecombineerd met beoordelingen die reeds op het web te vinden zijn. Door de gebruikersinput zal het systeem steeds betere muziekkeuzes kunnen doorgeven naarmate het programma vaker wordt gebruikt.

Het systeem moet zijn informatie kunnen halen van verschillende bronnen. Deze informatie van bijvoorbeeld verschillende muziekdatabases dient door het systeem gecombineerd te worden.

# 

# Functionele omschrijving van het systeem

Via een formulier zal de gebruiker een bepaald profiel van zichzelf kunnen aanmaken, waarin bijvoorbeeld genres, favoriete nummers, leeftijd en geslacht kunnen worden aangegeven. Aan de hand van deze input zal het systeem bepaalde artiesten of nummers aanraden. Het systeem zal dit doen aan de hand van de voorkeuren van andere gebruikers, waarderingen van anderen over de muziek, genres en informatie over de luisteraars van deze muziek.

De aangeraden artiesten en nummers kunnen vervolgens door de gebruiker beoordeeld worden door middel van ‘vind ik leuk’ en ‘vind ik niet leuk’-knoppen.

Het systeem zal via het internet bereikbaar zijn. Hierdoor is het makkelijk voor iedereen te gebruiken; er hoeft niet een speciaal programma geïnstalleerd te worden. Een nadeel hiervan zou wel kunnen zijn dat het niet mogelijk is om alle muziekbestanden die op de computer van de gebruiker zijn, door te scannen.

Het systeem zal gebouwd worden in de talen HTML en PHP. Er is voor deze talen gekozen omdat deze uitermate geschikt zijn voor het gebruik via internet. Met de HTML zullen de pagina’s voor de gebruikers worden gemaakt. Met PHP wordt de achterliggende code geïmplementeerd waarmee goede suggesties kunnen worden gedaan qua muziekkeuze, welke vervolgens aan de gebruiker kunnen worden getoond. Daarnaast zal er nog gebruik worden gemaakt van een MySQL database om de gegevens over de gebruikers op te slaan.

Om de suggesties voor muziekkeuze te bepalen, zullen er diverse bronnen van het internet gebruikt worden, welke ook weer aan elkaar gekoppeld zullen worden. MusicBrainz is een van deze bronnen die gebruikt zal worden. In deze muziekdatabase is veel informatie beschikbaar over nummers, artiesten en uitgaven, met name feitelijke informatie. Verder zal LastFM gebruikt worden, een andere muzieksite waarin met name veel informatie te vinden is over welke nummers in combinatie met elkaar worden geluisterd. Naast deze muziekbronnen zal er ook gebruikt worden gemaakt van DbPedia, de voor het semantische web geschikt gemaakte versie van WikiPedia. Hieruit kan specifieke data worden verkregen over bijvoorbeeld een artiest. Met name geboortedatum, plaats, en een korte samenvatting over de artiest kunnen hieruit worden gehaald. Verder zal de YouTube API gebruikt worden voor een directe koppeling naar het aangeraden nummer.

# Product fase 1

Voor de eerste fase is er een systeem gemaakt waarmee de gebruikersinput beperkt blijft tot het kiezen van favoriete genres, en de output van het systeem slechts een rijtje met artiesten teruggeeft. Deze artiesten maken muziek binnen de gekozen genres en krijgen daarnaast een hoge waardering. Hiervoor maakt het systeem gebruik van één data bron. In latere fases zal het systeem worden uitgebreid met meer input van de gebruiker, met naast artiesten ook specifieke nummers als output. Het systeem zal dan verder gebruikmaken van meerdere bronnen en deze informatie gaan combineren.

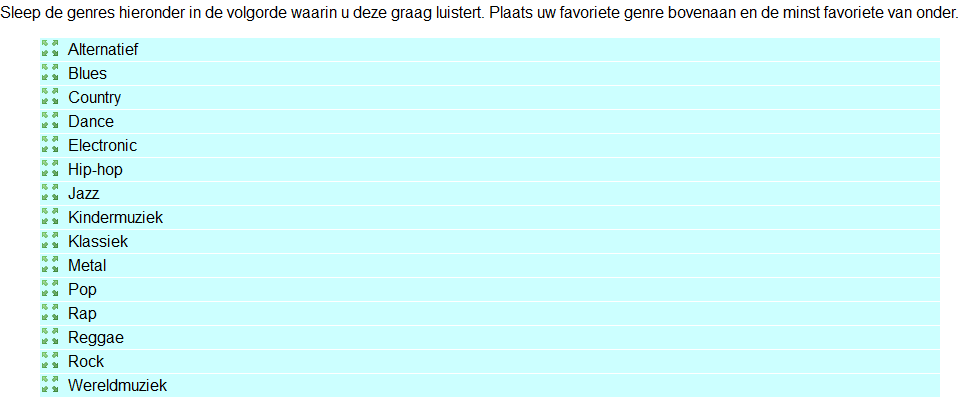
## Data

Voor de eerste fase is er gekozen om slechts van één data bron gebruik te maken. Dit is de muziekdatabase MusicBrainz. MusicBrainz is de open muziek encyclopedie waaraan gebruikers zelf data toevoegen en verbeteren. Er is allerlei feitelijke informatie in te vinden over artiesten, albums, nummers en diverse uitgaven hiervan.

Ook is de meeste muziek uit de database getagd. Deze tags zijn op de meeste punten vergelijkbaar met genres. Zo zijn er tags voor ‘classical’, ‘pop’, ‘jazz’, ‘electrical’, etc. Niet alle tags zijn even bruikbaar in het systeem. Sommige tags zijn maar een aantal keren gebruikt of redelijk nietszeggend (bijvoorbeeld de tag ‘production music’). Vandaar dat uit deze tags een bepaalde bruikbare selectie gekozen is die in het systeem worden gebruikt.

Daarnaast is er informatie beschikbaar over de waardering die gebruikers geven over bepaalde muzieknummers of artiesten. Deze informatie is verre van compleet: zo zijn er slechts 160 000 waarderingen gegeven, terwijl er zo’n 600 000 artiesten in de database aanwezig zijn. Deze waarderingen worden in de eerste fase echter wel gebruikt. In een latere fase zullen de waarderingen van de muziek verbeterd worden door waarderingen uit verschillende bronnen en de gebruiker zelf, te combineren tot een goede rating.

## Implementatie

Voor de daadwerkelijke implementatie van het product, is er eerst een keuze gemaakt over de beschikbare genres. Aan de hand van de tag-cloud van MusicBrainz (<http://musicbrainz.org/tags>), is een selectie gemaakt van tags welke vaak worden gebruikt en daarnaast ook nog een duidelijk genre aangeven. Uit deze selectie is een lijst met 10 genres gekomen waaruit de gebruikers kunnen kiezen. 

Nadat de gebruikers hun keuze hebben ingevoerd, wordt er voor de meest favoriete 3 genres bijbehorende artiesten opgezocht uit de MusicBrainz database.

De data uit de MusicBrainz database wordt verkregen door middel van directe zoekopdrachten via de XML web service welke zij zelf aanbieden. Dit geeft wel als beperking dat de berekeningen van het systeem zeer afhankelijk zijn van MusicBrainz: de muziekdatabase verwerkt slechts één zoekopdracht per seconde. Daarnaast slaat MusicBrainz zoekopdrachten over, indien binnen een seconde weer een nieuwe aanroep wordt gedaan. Hierdoor gaat er veel data verloren.

Van elk van deze verkregen artiesten wordt de waardering opgevraagd via een andere zoekopdracht. De waardering is een cijfer tussen de 0 en 5. Dit cijfer wordt gewogen met de positie van het genre uit de door de gebruiker gekozen lijst.

Door het wegen van de waardering zal het meestal het geval zijn dat een artiest welke getagd is met het eerste gekozen genre, hoger eindigt dan een artiest die nummers maakt uit het 3e gekozen genre. Dit hoeft echter niet zo te zijn. Zo krijgt bijvoorbeeld een artiest uit het 1e gekozen genre met waardering 2, een totale gewogen waardering van 10. Evenzo krijgt een artiest uit het 3e gekozen genre met waardering 4 een totale gewogen waardering van 12. Hierdoor zou de (beter gewaardeerde) artiest uit het 3e gekozen genre hoger uitkomen dan de artiest die nummers maakt in het meest favoriete genre van de gebruiker.

Deze gewogen waardering wordt opgeslagen in een database. Deze database zal elke keer voor gebruik worden geleegd. Er wordt in dit product nog geen informatie over de gebruiker opgeslagen, vandaar dat dit mogelijk is. De databasetabel van items die hieruit komt, wordt gesorteerd op aflopende grootte van waardering. Vervolgens worden hier de eerste 10 van getoond aan de gebruiker.

# Product fase 2

In fase 2 is het product uit fase 1 uitgebreid met extra functionaliteiten. Na de 1e fase was het product in staat om artiesten aan te raden aan de hand van een door de gebruiker ingevoerde genre top 10. De data werd hierbij uit MusicBrainz, een muziek encyclopedie, gehaald.

Zoals eerder vermeldt, leverde MuiscBrainz nogal wat beperkingen op. Zo werden de zoekopdrachten niet allemaal uitgevoerd in verband met een tijdslimiet. Aangezien de MusicBrainz database wel nuttige informatie bevat, is er gezocht naar een andere oplossing om MusicBrainz te doorzoeken. Hiervoor is de optie gevonden om zelf een lokale versie van MusicBrainz te laden, welke om de zoveel tijd wordt geüpdate. De lokale versie loste alle problemen met betrekking tot MusicBrainz op.

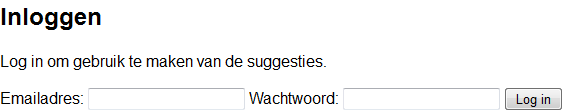
## 2e databron

Naast MusicBrainz is in deze versie ook een koppeling gemaakt met de dataset van Last.FM. De Last.FM dataset wordt op ongeveer dezelfde manier doorzocht als de MusicBrainz dataset: door middel van directe zoekopdrachten via de XML web service.

Last.FM biedt ten opzichte van MusicBrainz vele voordelen. Zo is er veel gebruikersdata in verwerkt, waardoor ook gemakkelijk top albums en populaire artiesten kunnen worden opgevraagd. Daarnaast is het handig dat Last.FM gebruikmaakt van de MBID’s, de MusicBrainz ID’s. Hierdoor zijn MusicBrainz en Last.FM gemakkelijk direct aan elkaar te koppelen.

## Input

De gebruikersinvoer is in deze versie vergroot. Naast de top 10 genres wordt er ook gevraagd om favoriete artiesten. Beide opties leveren op het moment aparte suggesties op.

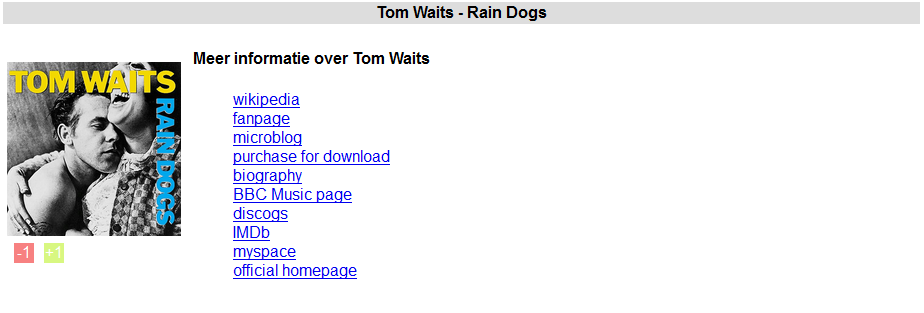
Daarnaast is er een log-in functie toegevoegd. Alle gebruikersinvoer wordt nu ook opgeslagen, zo hoeft een terugkomende gebruiker niet opnieuw zijn voorkeuren door te geven. Uiteraard kan hij deze wel bewerken als hij eenmaal weer is ingelogd.

Naast de invoer die de gebruiker in het begin geeft, heeft de gebruiker ook de optie om zelf waarderingen te geven aan de door het systeem gegeven suggesties. Dit gebeurt door middel van een relatief simpel systeem: een gebruik geeft de waardering +1 (leuk) of -1 (niet leuk) aan een suggestie.

## Output

De output in deze versie bestaat uit twee delen: suggesties die gebaseerd zijn op de genrevoorkeuren en suggesties die gebaseerd zijn op de artiestvoorkeuren. Beide voorkeuren worden via Last.FM opgehaald. Hiervoor worden de functies getSimilarArtist en getTopArtist gebruikt.

De suggesties bestaan uit artiesten. Bij elke artiest wordt het beste album gezocht via Last.FM (getTopAlbums). Van dit album worden diverse linkjes getoond naar andere bronnen, bijvoorbeeld naar Wikipedia en YouTube. Deze linkjes worden van MusicBrainz opgehaald. Daarnaast kan de gebuiker nu suggesties in en uitklappen om alles overzichtelijker te maken. Als de gebruiker een -1 geeft aan een suggestie wordt de suggestie standaard ingeklapt.

Daarnaast wordt ook de afbeelding van het album opgehaald. Het geheel ziet er per suggestie dan als volgt uit:

# Komende fases

Het product geeft met deze relatief kleine aanpassingen in fase 2 al aardig betere resultaten dan het product van de vorige fase. Het geeft echter nog niet de resultaten die wij wensen in het eindproduct. Om de resultaten in de volgende fase dusdanig te verbeteren, is er al divers voorwerk gedaan in fase 2.

Het is de bedoeling dat de suggesties die door het system gegeven gaan worden zo veel als mogelijk aan gaan sluiten bij de voorkeuren van de gebruiker. Op dit moment wordt dit met name gebaseerd op enkele vrij algemene muziekgenres die de gebruiker kan kiezen.

Om dit te verbeteren, hebben we besloten om een boom van muziekgenres op te bouwen. De meest algemene genres staan onder de wortel van de boom, en passende subgenres worden daar geautomatiseerd onder toegevoegd.

Het doel hiervan is om afstanden te gaan berekenen tussen verschillende genres, vanuit het perspectief van de gebruiker. Momenteel is al de waarderingsfunctie geïmplementeerd waarbij de gebruiker kan aangeven of hij een bepaalde suggestie leuk (+1) of niet leuk (-1) vind. Dit kan in de boom gebruikt worden door verbindingen te leggen tussen de genres van de suggestie (uiteraard ontstaat er dan een graaf). Op deze manier hopen we meer gerichte suggesties te kunnen doen dan in de vorige versies van ons systeem, maar daarbij ook bredere suggesties te kunnen geven doordat er minder krampachtig wordt vastgehouden aan de brede genres die de gebruiker in eerste instantie had opgegeven.

Voor dit gedeelte van het systeem is gebruikgemaakt van SPARQL queries naar DBPedia, de semantische versie van Wikipedia. Door middel van het attribuut ‘stylisticOrigin’ is de boom opgebouwd.

De boom zelf is nog niet gecombineerd met de waarderingen welke de gebruikers geven en de waarderingen van MusicBrainz en Last.FM. Het idee is dat dit voor de volgende fase wordt geïmplementeerd.

Het algoritme dat de muziek zal waarderen en daarmee de suggesties zal verbeteren, zal in grote lijnen als volgt werken:

* Er wordt een graaf gegenereerd van genres aan de hand van DBPedia.
* De afstanden in de graaf worden vervolgens gewogen aan de hand van de waarderingen van muziek op MusicBrainz en de playlists op Last.FM. Als bijvoorbeeld blijkt uit de playlists dat Metal vaak samen met Electrical wordt geluisterd, wordt de afstand tussen deze twee genres kleiner.
* De input bestaat uit favoriete genres en artiesten. De ingegeven artiesten worden aan een bepaald (sub)genre gekoppeld. De verbindingen tussen de ingegeven favoriete genres en de genres die bij de favoriete artiesten horen, krijgen een kortere afstand. De favoriete genres en artiesten van de gebruiker hebben een veel sterkere impact op de graaf dan de waarderingen van MusicBrainz en Last.FM.
* Aan de hand van de afstanden worden er artiesten gezocht, welke zoveel mogelijk aan de eigenschappen van de graaf voldoen.
* Als de gebruiker zelf waarderingen aan deze suggesties geeft, zal de graaf hierop worden aangepast.