

Onderwerpen

[Logica simulator](#)
[Openstreetmap viewer](#)
[Redeneren met instroomgegevens](#)
[Automatische todo lijst](#)
[Visual Programming IDE](#)
[Klassediagram Editor](#)
[Flowchart Editor](#)
[Uurroosterproblemen](#)
[Codesubmissiesysteem](#)
[Robot Simulator](#)
[Routeplanner](#)
[Een Slim Toetsenbord](#)
[Analysetool voor Databases](#)
[Analysetool voor SCM repositories](#)
[Offline \(zaal\)quiz systeem](#)
[Offline quizsysteem voor kleine groepen](#)
[Opvolgingssysteem studentenvoortgang](#)
[Mobiele Feedly client](#)
[Facebook privacy tool](#)
[Step Sequencer](#)
[Simuleren van Verkeersnetwerken](#)
[File Watching System](#)
[OpenStreetMap: Data Cleaning](#)
[OpenStreetMap: Kortste pad tussen busstops](#)

Logica simulator

"Een educatieve tool om logische schakelingen te simuleren"

Uitgebreide beschrijving

In dit project is het de bedoeling om een simulatie programma te maken dat een logische schakeling simuleert op een visuele manier. In dit programma dient een gebruiker logische basisblokken (AND OR NOT, ..) te kunnen plaatsen en deze met elkaar te verbinden om een schakeling te bekomen. Verder moet er input (knoppen, oscillators, ...) en output (licht of geluid,...) voorzien worden om de schakeling te testen. De simulator moet voor didactische doeleinden (bijvoorbeeld bij het vak *Computersystemen*) ingezet kunnen worden.

Als extra functionaliteit kun je een manier voorzien om zelf componenten te bouwen die bestaan uit andere logische schakelingen om een hiërarchisch design te krijgen.

Probleem invoer

- user input om schema te tekenen
- schakeling in zelf te kiezen XML formaat

Probleem uitvoer

- simulatie van de schakeling m.b.v. input en output elementen
- automatische waarheidstabellen

Probleem requirements

De applicatie moet platformonafhankelijk (zowel op Mac OS, Linux als op Windows) kunnen werken. De programmeertaal mag vrij gekozen worden.

De tool dient gebruiksvriendelijk en stabiel te werken en moet een hoge tolerantie hebben voor verkeerde user input.

Bestaande software

KLogic, Logisim, ...

Openstreetmap viewer

"Visualisatie van een geografische kaart op basis van open vector data"

Uitgebreide beschrijving

Openstreetmap is een gratis open-source versie van Google Maps of Live Search maps. Net zoals bij Wikipedia wordt de kaart opgesteld door de community zelf zonder gebruik te maken van copyrighted materiaal. Iedereen kan de kaart aanpassen en gps logs uploaden om de kwaliteit van de wereldkaart te vergroten. Anderzijds kan ook iedereen de informatie gratis gebruiken onder een open-source licentie.

De straten worden intern in een xml formaat opgeslagen dat niet ideaal is om efficiënt te renderen. Deze opdracht bestaat er uit om (een deel van) de kaart om te vormen naar een efficiënter formaat en dit formaat te gebruiken om de kaart off-line te visualiseren. Een interactieve GUI is hierbij noodzakelijk.

Probleem invoer

- xml data van een bepaalde regio van openstreetmap

Probleem uitvoer

- visualisatie van de kaart
- gegenereerde tiles (zie vb. Google Maps)

Probleem requirements

De tool dient gebruiksvriendelijk en stabiel te werken. De kaartweergave dient duidelijk te zijn op elk zoom niveau. Navigatie dient interactief te zijn.

Er wordt een duidelijke omschrijving verwacht van het algoritme dat gebruikt wordt om de kaarten te tekenen.

Bestaande software

<http://wiki.openstreetmap.org/wiki/Software/Desktop>

Redeneren met instroomgegevens

“Het ontwikkelen van een tool waarmee instroomgegevens van de UHasselt eenvoudig onderzocht kunnen worden.”

Uitgebreide beschrijving

De universiteit heeft een berg aan gegevens van de studenten. Denk aan instroomgegevens en gegevens omtrent het studieverloop. Het is dan ook nuttig als deze gegevens inzichtelijk gemaakt kunnen worden. Daarvoor is een tool nodig waarmee de gebruiker verschillende types vragen kan stellen (en deze zelf kan samenstellen) en de antwoorden op deze vragen op verschillende manieren kan visualiseren. Voor het beantwoorden van deze vragen dient de tool zelfstandig data te kiezen, combineren, filteren en visualiseren.

Probleem invoer

- Excel datasheets met gedetailleerde instroomgegevens
- Schema van de dataset met promotie-activiteiten:
(ACJAAR,ACTSOORTID,SOORT_ACTIVITEIT,ACTID,ACTIVITEIT,KALENDER_ACTIVITEIT,IDENTNR,GEBOORTEJAAR,GESLACHT,POSTNR,GEMEENTE,LAND,SCHOOLNR,SCHOOL,SCHOOLPOSTNR,GEMEENTE_SCHOOL,LAND_SCHOOL,RICHTING_SO)
- Schema van de dataset met instroomgegevens
(ACJAAR,STAMNUMMER,OPLEIDING,GESLACHT,GEBOORTEJAAR,POSTNR_WOONST,GEMEENTE_WOONST,LAND_WOONST,SCHOOLNR,SCHOOL,POSTNR_SCHOOL,GEMEENTE_SCHOOL,LAND_SCHOOL,RICHTING_SO,DATUIT,STPOPEGE NOMEN,STPBEHAALD,PTN_GEDELIBEREERD)

Probleem uitvoer

- Interactieve visualisatie van de datasets
- Documenten met verbanden en statistieken

Probleem requirements

De tool moet gebruiksvriendelijk en stabiel zijn. De tool moet intuïtief te gebruiken zijn, ook voor personen die niet vertrouwd zijn met de achterliggende data.

Bestaande software

Prefuse: <http://prefuse.org/>

Google Fusion Tables: <https://sites.google.com/site/fusiontablestalks/stories>

IBM Many Eyes: <http://www-958.ibm.com/software/data/cognos/manyeyes/>

Bronnen

http://en.wikipedia.org/wiki/Data_visualization

Automatische todo lijst

"Een intelligente takenplanner voor een takenlijst"

Uitgebreide beschrijving

Een lijst van taken dient ingeroosterd te worden in een aantal vrije tijdslots. De taken zijn allen voorzien van een bepaalde duur, prioriteit, deadline en afhankelijkheid. De vrije tijdslots zijn door de gebruiker te specificeren d.m.v. een bestaande kalender. De software plant via een intelligent algoritme automatisch de taken in zodat alle afhankelijkheden en deadlines gerespecteerd worden. De gebruiker krijgt ten slotte een overzicht van de ingeplande taken te zien en kan dit naar een kalender exporteren.

Indien het niet mogelijk is om de taken allemaal ingepland te krijgen, dient het systeem een aantal tussenoplossingen aan te bieden aan de gebruiker.

Probleem invoer

- Lijst van taken en kalender
- Kalender in iCal formaat / eigen formaat met converter voor iCal
- Taken in zelf te kiezen XML formaat

Probleem uitvoer

- Grafische weergave van ingeplande taken
- Kalender in iCal formaat / eigen formaat met converter voor iCal

Probleem requirements

De applicatie moet platformonafhankelijk (zowel op Mac OS, Linux als op Windows) kunnen werken. De programmeertaal mag vrij gekozen worden.

Het deel om de taken in te plannen moet volledig onafhankelijk van de rest gebruikt kunnen worden.

In het eindverslag wordt ook een motivatie verwacht voor de keuze van het algoritme, onderbouwd met testresultaten waarbij o.a. alternatieve aanpakken beschouwd worden.

Bestaande software

Bestaande software zoals Things (<http://www.culturedcode.com>) laat vaak niet toe om taken automatisch in te plannen, of zijn niet platformonafhankelijk. Wel laten ze al heel goed toe om taken te beheren met GTD principes en voorzien ze synchronisatie.

Bronnen

- [Allen, David](#) (2001). *Getting Things Done: The Art of Stress-Free Productivity*. Penguin Books. [ISBN 0-14-200028-0](#).
- Things for Mac <http://www.culturedcode.com>

Visual Programming IDE

“IDE voor een visuele programmeertaal”

Mensen die starten met programmeren staan voor de uitdaging om de syntax goed onder de knie te krijgen. Dit zal in het begin vaak voor problemen zorgen. Een oplossing voor dit probleem is om te werken met een visuele programmeertaal, waarbij de gebruiker niet moet typen, maar beschikt over een “blokkendoos” die de mogelijke opdrachten bevat. De gebruiker kan vervolgens deze blokken combineren/aan elkaar schakelen, om zo een programma te maken.

De blokken kunnen overigens nog meer visuele feedback geven: bijvoorbeeld de vorm en de kleur kunnen gelinkt zijn aan de betekenis ervan.

Probleem invoer

- Via IDE
- XML bestand inladen

Probleem uitvoer

- XML bestand wegschrijven

Probleem requirements

De applicatie moet platformonafhankelijk (zowel op Mac OS, Linux als op Windows) kunnen werken. De programmeertaal mag vrij gekozen worden.

De blokken moeten uitbreidbaar zijn.

In het eindverslag wordt een duidelijke beschrijving verwacht van de omzetting van de blokken naar een uitvoerbare structuur.

Bestaande software

Scratch is een voorbeeld van een Visuele programmeertaal. Het nadeel bij deze aanpak is dat de uitvoeromgeving volledig vast ligt, de grootte kan bvb. niet aangepast worden.

Bronnen

- Visual Programming Language, Wikipedia, http://en.wikipedia.org/wiki/Visual_programming_language
- Scratch, <http://scratch.mit.edu/>
- Open Blocks: <http://dspace.mit.edu/bitstream/handle/1721.1/41550/220927290.pdf>
- Blockly: <https://code.google.com/p/blockly/>
- RoboMind: <http://www.robomind.net/nl/>

Klassediagram Editor

"Editor voor klassediagrammen"

Uitgebreide beschrijving

Het klassediagram is een onderdeel van UML (Unified Modeling Language). Het bevat de statische structuur van een applicatie door de klassen, attributen en associaties in een diagram te tonen.

Probleem invoer

- Via GUI
- XML bestand

Probleem uitvoer

- XML bestand

Probleem requirements

De applicatie moet platformonafhankelijk (zowel op Mac OS, Linux als op Windows) kunnen werken. De programmeertaal mag vrij gekozen worden.

In het eindverslag wordt ook een beschrijving verwacht van het layouting algoritme.

Bestaande software

Als inspiratie kan je bestaande applicaties zoals Dia, Microsoft Visual Modeler, Rational Rose of OmniGraffle bekijken.

Bronnen

- Class Diagram, Wikipedia, http://en.wikipedia.org/wiki/Class_diagram

Flowchart Editor

"Editor en simulator voor flowcharts"

Uitgebreide beschrijving

Een flowchart geeft een grafische weergave van een algoritme of proces, waarbij de stappen uit het algoritme als geometrische figuren worden afgebeeld, verbonden door pijlen. Ze worden gebruikt bij het ontwerp van een proces of bij de documentatie ervan.

Probleem invoer

- via GUI
- XML bestand

Probleem uitvoer

- XML bestand

Probleem requirements

De applicatie moet platformonafhankelijk (zowel op Mac OS, Linux als op Windows) kunnen werken. De programmeertaal mag vrij gekozen worden.

Bestaande software

Als inspiratie kan je bestaande applicaties zoals Dia, Microsoft Visual Modeler of OmniGraffle bekijken.

Bronnen

- Flow Chart, Wikipedia, <http://en.wikipedia.org/wiki/Flowchart>

Uurroosterproblemen

“Interactieve hulptool om uurroosters op te stellen”

Uitgebreide beschrijving

Het maken van uurroosters is een taak die veel tijd en energie vergt: er moet een overeenkomst gevonden worden waarmee iedereen gelukkig is. Zo een overeenkomst vinden is geen eenvoudige taak: iedereen heeft beperkingen m.b.t. de tijdstippen, lokalen hebben een beperkte capaciteit, lokalen kunnen slechts voor 1 vak gebruikt worden op hetzelfde moment, er zijn afhankelijkheden tussen lessen, enz.

De bedoeling van deze opdracht is een tool te ontwerpen die op een intuïtieve manier toelaat om een uurrooster samen te stellen. Ook constraints moeten op een snelle en eenvoudige manier ingevoerd kunnen worden. Wanneer er vervolgens conflicten plaatsvinden, dan moet dit zo snel en duidelijk mogelijk weergegeven worden.

De tool maakt het ten slotte ook mogelijk om bepaalde zaken snel in te voeren, denk hierbij bijvoorbeeld aan het dagsysteem dat er bijna altijd hetzelfde uitziet: HC,ZS,RC. Er zijn een heleboel van dit soort zaken die het opstellen van de uurroosters eenvoudiger en sneller kunnen maken.

Probleem invoer

- via GUI (o.a. kalender widget)
- XML bestand

Probleem uitvoer

- XML bestand

Probleem requirements

Experimenteren met haalbaarheid van berekeningen.

Duidelijke uitwerking van aanpak in eindverslag, tezamen met ondervonden moeilijkheden.

Codesubmissiesysteem

"Indien- en evaluatiesysteem voor programmacode"

Uitgebreide beschrijving

Initiatieven zoals de Vlaamse Programmeerwedstrijd en TopCoder hanteren een automatisch indiensysteem dat de ingediende programmacode compileert of interpreteert en uitvoert op verscheidene inputs. Op basis van het al dan niet correct uitvoeren van de opdracht, wordt een score toegekend aan de ingediende programma's. Doel van dit project is een dergelijk submissiesysteem op te zetten.

Probleem invoer

- Programmacode (nog te bepalen welke talen)

Probleem uitvoer

- Voor elk ingediend programma: score, gebaseerd op al dan niet correcte uitvoering van het programma op de gegeven set van inputs.

Probleem requirements

- Web-based interface.

Bestaande software

- Java-indiensysteem eerste bachelor
- Zie http://en.wikipedia.org/wiki/Online_judge voor een lijst van voorbeelden
- Sphere Online Judge <http://www.spoj.com/>

Robot Simulator

“Simulatieframework voor autonome robots”

Uitgebreide beschrijving

Robots beschikken over een heleboel sensoren en outputmogelijkheden. Soms is een probleem makkelijker op te lossen wanneer robots kunnen samenwerken. Tegenwoordig zijn er tal van manieren beschikbaar om robots met elkaar te laten communiceren. Een voorbeeld hiervan zijn LEGO Mindstorms robots die via een draadloze verbinding met elkaar kunnen communiceren.

Het is niet altijd eenvoudig om robot-software te testen in de praktijk: voldoende robots en een test-omgeving moeten ter beschikking staan van de ontwikkelaar. Daarom is het nuttig om een virtuele simulatieomgeving te hebben waarin robots “geprogrammeerd” en vervolgens getest kunnen worden.

Probleem invoer

- omgevingsbestand
- scripts/plugins voor robot gedrag

Probleem uitvoer

- simulatie

Probleem requirements

Uitbreidbaar: zowel omgevingsonderdelen, sensoren als scriptingtaal.

Routeplanner

“Basis routeplanner voor België”

Uitgebreide beschrijving

Routeplanners zijn alomtegenwoordig: iedere smartphone heeft al een GPS en een routeplanner applicatie. De routeplanner moet doorheen gigabytes aan geografische data gaan vooraleer de ideale route gevonden wordt. En toch slaagt bvb. Google Maps erin om binnen enkele seconden de ideale route tevoorschijn te toveren... Hoe kan dat?

Probleem invoer

- Autoroutenetwerk van België (OpenStreetMap)
- Punt A en Punt B

Probleem uitvoer

- Route van A naar B

Probleem requirements

Route moet binnen de 10 seconden gevonden zijn.

Bestaande software

- Google maps

Bronnen

- A*

Een Slim Toetsenbord

Uitgebreide beschrijving

Veel GSMs/smartphones hebben tegenwoordig ingebouwde software om een gebruiker sneller en gemakkelijker tekst te laten invoeren. Dit is vooral bijzonder nuttig bij het invoeren van tekst op kleine touch toetsenborden, zoals op de iPhone. Dit soort software gaat vaak fouten proberen te corrigeren door gebruik te maken van een woordenboek, en door rekening houden met aspecten zoals welke letters dicht bij elkaar staan op het toetsenbord, of welke woorden vaak na elkaar gebruikt worden.

Het doel van dit project is om zelf een implementatie van zulk een slim toetsenbord te maken dat fouten van de gebruiker zo goed mogelijk probeert te corrigeren, en dit toetsenbord tevens te evalueren. Het is tevens belangrijk dat het systeem leert van vorige ingegeven teksten, en zich dus kan aanpassen aan de gebruiker.

Let op! Dit is een algoritmisch moeilijk onderwerp, waarbij vooral het bereiken van een goede performantie belangrijk zal zijn!

Probleem invoer

- Woordenboek van woorden voor een aantal talen (bijv. [Aspell](#))
- Lijsten van veelgebruikte woorden/zinnen (bijv. via crawling van kwaliteitsvolle teksten in combinatie met teksten die de gebruiker al heeft ingegeven)
- Door de gebruiker ingevoerde letters

Probleem uitvoer

- Een voorspelling van het getypte woord
- Eventuele volgende woorden of tekens die na dit woord kunnen volgen (bijv. op “ik” kan “ben”, “heb” of “wil” volgen)

Probleem requirements

De achterliggende voorspellingsfunctionaliteit dient op een platformonafhankelijke manier geïmplementeerd te zijn, en modulair opgebouwd te zijn zodat het herbruikt kan worden voor verschillende gebruikersinterfaces. De uiteindelijke user interface mag uiteraard specifiek voor een bepaald platform geïmplementeerd worden (bijv. de Windows Surface 2.0 SDK).

In het eindverslag wordt ook een motivatie verwacht voor de keuze van het algoritme, onderbouwd met testresultaten waarbij o.a. alternatieve aanpakken beschouwd worden.

Bestaande software

Een voorbeeld van zo'n keyboard is: [Swiftkey](#) voor Android smartphones.

Bronnen

- **Swiftkey:** <http://www.swiftkey.net/en/>

Analysetool voor Databases

Uitgebreide beschrijving

Veel databases worden niet netjes ontworpen maar groeien dynamisch over tijd: attributen, tabellen en relaties worden toegevoegd en verwijderd in de levensloop van de database. Dit gaat vaak niet gepaard met goed onderhouden documentatie en uitgedachte designs.

Voor onderhoudstaken, en voor het re-engineeren van applicaties is zo'n database niet optimaal. Om deze taken te ondersteunen maken we een analysetool die de structuur van een bestaande database kan (helpen) ontrafelen. Enkele mogelijke functies van zo'n tool zijn:

- Geef een visualisatie van de structuur van de database.
- Beantwoord queries over de structuur van de database.

Analysetool voor SCM repositories

Uitgebreide beschrijving

Grote softwareprojecten evolueren over tijd: de belangrijke onderdelen in de code veranderen, de rol van ontwikkelaars veranderen, de 'hot-spots' qua development veranderen. Inzicht in dit soort informatie kan het managen van softwareprojecten vereenvoudigen. Voor veel projecten worden deze gegevens indirect al bijgehouden via het versiebeheersysteem dat word gebruikt.

Daarom is het een goed idee om deze informatie inzichtelijk en toegankelijk te maken. Enkele mogelijke functies van zo'n tool zijn:

- Visualisatie van de evolutie van de rol van ontwikkelaars van het project.
- Visualisatie van de evolutie van de rol van onderdelen van het project.
- Zorg dat in je visualisatie gerelateerde bestanden ook visueel gerelateerd zijn.

Offline (zaal)quiz systeem

“Geïntegreerde omgeving om een (zaal)quiz te ontwerpen”

Uitgebreide beschrijving

Het organiseren van een quiz is veel meer dan enkel het stellen van vragen; zo moet er bijvoorbeeld een score bijgehouden worden van de verschillende teams en moeten video- en geluidsfragmenten afgespeeld kunnen worden. Hiertoe wordt vaak een hele resem aan softwaretoepassingen gebruikt: een presentatieprogramma (zoals Powerpoint), een videospeler, een spreadsheet applicatie, een set van vragen ...

Het doel van dit project is om het organiseren van een quiz te vereenvoudigen door één applicatie te maken waarin de (voor een quiz relevante) functionaliteit van de voorgaande toepassingen gecombineerd wordt. De applicatie bestaat uit twee verschillende vensters, waarvan er één enkel aan de quizmaster getoond wordt, terwijl het andere geprojecteerd wordt voor alle deelnemers.

Let op! Het is voor dit project niet de bedoeling om een Powerpoint-kloon te maken! De presentatie van de vragen gebeurt ahv door het programma automatisch gegenereerde slides (met daarop bijvoorbeeld de vraag en bijhorende mediabestanden), niet ahv manueel ontworpen slides.

Bronnen

- Een gemiddelde quiz van de jeugdbeweging / school
- De Slimste Mens
- De Pappenheimers
- Blokken
- ...

Offline quizsysteem voor kleine groepen

“Geïntegreerd systeem om een quiz voor kleine groepen spelers te ontwerpen”

Uitgebreide beschrijving

Geregeld worden in familie- of vriendenkring gelegenhedsquizen georganiseerd. Bijvoorbeeld als men in groep op reis gaat, is dit een leuke avondactiviteit.

Het doel van dit project is om het organiseren van dergelijke kleinschalige quiz te vereenvoudigen door één applicatie te maken die toelaat vragen te stellen, te laten antwoorden met drukknoppen (deze worden voorzien), scores bij te houden, etc. De gedetailleerde requirements worden in overleg met de opdrachtgever afgesproken.

Bronnen

- De Pappenheimers
- Blokken
- ...

Opvolgingssysteem studentenvoortgang

“Tool voor interne opvolging van studentenvoortgang”

Uitgebreide beschrijving

Studenten krijgen in hun opleiding vaak te maken met projecten. Het is echter moeilijk om deze voortgang goed op te volgen.

Een opvolgingssysteem stelt de onderwijsteams in staat om over verschillende vakken feedback met elkaar te delen. Het systeem biedt een eenvoudige interface aan waarin finale versies van projecten en scores bijgehouden kunnen worden, tezamen met verslagen van iedere samenkomst en een finaal verslag. Werkpunten en goede punten kunnen ook aangegeven worden. Alles wordt ingegeven via de applicatie zelf, via een intuïtieve interface.

Mobiele Feedly client

“Een nieuwsreader voor feedly.com”

Uitgebreide beschrijving

Feedly.com is een web-based news reader, met content gebaseerd op de RSS feeds van een gebruiker. Feedly biedt ook een API aan die gebruikt kan worden om de lijst met ongelezen artikels binnen een andere applicatie te openen.

Het doel van dit project is om een mobiele (Android) applicatie te ontwikkelen die gebruikt kan worden om Feedly nieuwsbronnen te raadplegen.

Bronnen:

Feedly API documentatie: <https://developer.feedly.com/>

Android Material design guidelines: <https://developer.android.com/design/material/index.html>

Press, een Android client voor Feedly: <http://twentyfivesquares.com/press/>

Facebook privacy tool

Uitgebreide beschrijving

Facebook biedt een aantal instellingen om de zichtbaarheid van je posts en gegevens te beperken tot bepaalde personen. Helaas zijn deze privacy-instellingen niet altijd even doorzichtig. Het doel van dit project is om een tool te ontwikkelen die de Facebook API gebruikt om een aantal privacy-processen te vereenvoudigen. De tool geeft de gebruiker in eerste instantie een overzicht van de gemaakte posts, samen met een aantal statistieken: welk percentage van de posts heeft welke privacy instellingen?

Verder laat de tool ook toe om bepaalde privacy voorkeuren te automatiseren, door de gebruiker in eerste instantie toe te laten om een bepaalde privacy voorkeur toe te passen op alle posts uit het verleden, maar ook door de gebruiker automatische privacy-suggesties te geven bij het aanmaken van nieuwe posts.

Bronnen:

Facebook API documentatie: <https://developers.facebook.com/>

Step Sequencer

“Maak een drum computer/step sequencer om liedjes te maken”

Uitgebreide beschrijving

In deze opdracht is het de bedoeling om een step sequencer te maken, waarmee de gebruiker gemakkelijk (eenvoudige) liedjes kan maken. Dit gaat als volgt: de gebruiker krijgt de mogelijkheid om samples/beats te programmeren in korte fragmenten (sequences) van een aantal maten. Deze fragmenten kunnen afzonderlijk afgespeeld worden aan een bepaalde snelheid. Vervolgens kan men deze fragmenten combineren door ze achter elkaar te plakken. Deze aaneenschakeling moet natuurlijk ook afgespeeld kunnen worden.

Een intuïtieve interface is noodzakelijk. Ook moet de muziek handig opgeslagen kunnen worden zodat er in meerdere iteraties aan gewerkt kan worden, of stukjes herbruikbaar zijn in andere projecten.

Bronnen

JACK Audio Connection Kit (sound server daemon - <http://jackaudio.org/>)

Bestaande drum machines/beat sequencers:

- Hydrogen: <http://www.hydrogen-music.org/hcms/>
- Fruityloops: <http://www.image-line.com/flstudio/>
- SoundGrid: <http://www.mifki.com/>
- Tonematrix: <http://tonematrix>

Simuleren van Verkeersnetwerken

“Verkeerssimulaties & de Braess Paradox”

Uitgebreide beschrijving

Het doel van dit project is het simuleren van verkeersnetwerken door gebruik te maken van het zogenaamde Nagel-Schreckenberg model. Met deze simulaties willen we de Braess paradox illustreren. De Braess paradox beschrijft het volgende eigenaardige verschijnsel: voor sommige verkeersnetwerken kan het toevoegen van extra wegen zorgen voor een vermindering van de verkeersstroom. Deze paradox werd ontdekt door de Duitser Dietrich Braess in 1968. Dit verschijnsel is daarna ook beschreven in tal van andere netwerken, waaronder computernetwerken.

Het Nagel-Schreckenberg model is een microscopisch verkeersmodel, d.w.z. een model dat elk individueel voertuig in rekening brengt. De weg wordt voorgesteld als een opeenvolging van vakjes, met daarin maximaal 1 voertuig. Elk voertuig heeft een bepaalde snelheid die aangeeft aan hoeveel vakjes het voertuig opschuift per tijdseenheid.

Het verkeersnetwerk waarover de voertuigen rijden kan voorgesteld worden als een gerichte en samenhangende graaf. De knopen stellen bijvoorbeeld steden of kruispunten voor, de zijden verbinden de knopen en stellen de wegen voor. Elke zijde heeft een bepaalde lengte (het aantal vakjes) en een bepaalde rijrichting, aangegeven door een pijl. De voertuigen rijden van de startknoop tot aan de eindknoop. Bij vertrek krijgt elk voertuig een bepaalde route toegewezen.

Bronnen:

- Nagel, K.; Schreckenberg, M. (1992). "A cellular automaton model for freeway traffic". *Journal de Physique I* 2 (12): 2221
- D. Braess, Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung* 12, 258–268 (1969), Engelse vertaling: *Transportation Science* 39: (2005) pp 446–450

File Watching System

Uitgebreide beschrijving

Op IMOB worden er verschillende netwerkschijven (NAS'en) gebruikt. Deze netwerkschijven worden elke nacht gebackupt. Dergelijke backups worden x-aantal tijd bijgehouden.

Verschillende onderzoekers gebruiken deze netwerkschijven om grote hoeveelheden data bij te houden. Het grote probleem hierbij is dat deze files door iedereen kunnen worden aangepast. Uit schrik dat onderzoekers deze files per ongeluk aanpassen, dupliceren zij deze files naar een privé map op de netwerkschijf (uit veiligheid). Wanneer verschillende onderzoekers met dezelfde file werken, zijn er bijgevolg verscheidene kopies aanwezig. Een ander probleem is dat onderzoekers soms bepaalde tijd geen gebruik maken van een bepaalde file op de netwerkschijf. Als ze na x-aantal tijd deze file terug nodig hebben, kan het zijn dat deze file niet meer aanwezig is, omdat deze per ongeluk is verwijderd. Het kan zijn dat deze file dan niet meer in de backups aanwezig is waardoor de file niet meer te recoveren valt.

Omwille van deze reden willen we een file watch systeem creëren. Bij dit systeem heeft een folder/file (waarop een file watch staat) een eigenaar en verschillende volgers. De bedoeling van dit systeem is dat wanneer iemand een bepaalde file in een folder aanpast, de volgers en de eigenaar van deze map een melding krijgen over de verandering van de file. Wanneer de eigenaar een file aanpast, geeft hij hier ook de reden van. In het andere geval (als de file per ongeluk is aangepast), kunnen de veranderingen ongedaan gemaakt worden door een backup terug te zetten.

De bedoeling is dat er een programma gemaakt wordt dat toelaat om dit alles te managen. De verschillende gebruikers moeten folders/files kunnen volgen, er moet een history geraadpleegd kunnen worden (die mogelijk bijgehouden wordt via svn) en het moet mogelijk zijn om wijzigingen ongedaan te maken. Dit alles dient duidelijk gevisualiseerd te worden, zodat onervaren personen er snel mee aan de slag kunnen.

Probleem Invoer

Files/folders die gewacht willen worden

Bestaande Software

- Data Nerd
- Filesystemen met history (zoals HAMMER, Shadow Copy, ...)?

OpenStreetMap: Data Cleaning

Uitgebreide beschrijving

Het OSM netwerk is opgebouwd door middel van XML. Tussen deze XML elementen bevindt zich extra informatie in de vorm van het “tag” element. Dit element ziet er bijvoorbeeld als volgt uit:

```
<tag k="maxSpeed" v="50" />
```

Dit element beschrijft het key-value paar {"maxSpeed",50}. Doordat OSM wordt onderhouden door een community duiken er echter regelmatig fouten (typefouten, synoniemen, foute data, etc.) op in deze key-value paren. Een voorbeeld van een typefout van “bicycle” kan “bycycle” zijn, terwijl een synoniem voor “bicycle” “bike” kan zijn. Het probleem hierbij is dat consistentie van deze key-value paren zeer belangrijk is voor softwaredoeleinden. We willen bijvoorbeeld alle “bicycle” keys kunnen vinden door te zoeken op een enkele key.

Het doel van dit project is om de key-value paren in gegeven OSM data te onderzoeken en mogelijke fouten aan te geven. Enkele voorbeeldmethoden zijn string-distance, XML-structuur onderzoeken, data-detectors, synoniemen en typefout detectie. De gevonden anomalieën dienen overzichtelijk weergegeven te worden aan een gebruiker, die vervolgens kan beslissen om de data op te schonen. Ook is er de mogelijkheid dat de gebruiker op zoek gaat naar anomalieën door zelf filters te construeren.

Probleem Invoer

OSM netwerk

Probleem Uitvoer

Mapping van alle typefouten/synoniemen

Bestaande Software

- Spellcheckers
- Aspell
- Jazzy Woordenboek
- Wordnet

OpenStreetMap: Kortste pad tussen busstops

Uitgebreide beschrijving

Op IMOB wordt de GTFS database van “De Lijn” gebruikt om busstops aan het OSM netwerk te koppelen. Het resulterende netwerk wordt bijgehouden in een PostGis database. De bedoeling van dit project is om het *snelste* en het kortste pad te berekenen tussen twee opeenvolgende haltes van een buslijn.

Aan de hand van dit pad kunnen we bepalen aan welke zijde van de weg de bushaltes zich bevinden. Deze informatie is nuttig wanneer we verschillende rijstroken van een weg apart willen beschouwen: afhankelijk van de gebruikte rijstrook kan een bus je wel of niet afzetten aan de halte. Ten slotte laat deze informatie ons ook toe om te bepalen hoe snel de bus moet rijden op een bepaald traject, en of dit in strijd is met de toegelaten snelheid.

Definities:

Lijn: sequentie van bushaltes in volgorde van bediening

Dienst: lijn aangevuld met tijdstip voor bediening van elke halte

Netwerk: bestaat uit nodes en links. (Netwerk, Nodes, Links) stemt overeen met (Graph, Vertices, Edges)

Probleem Invoer

PostGis database met de nodige informatie over het OSM netwerk GTFS database: men mag er van uit gaan dat de in GTFS vermelde haltes aan links in het netwerk gekoppeld zijn.

Bestaande Software

- JGraphT - a free Java Graph Library voor graph-theory based tools (kan gebruikt worden voor opzetten van netwerk graph)
- Visualisatie Software: QGis (voor visuele verificatie van resultaten: dit vergt wel het wegschrijven van elk kortste pad tussen twee opeenvolgende haltes naar een postgis database)