

De bachelorproef informatica.

Een praktische gids.

Bert Van Vreckem, Jens Buysse.

Academiejaar: 2019-2020

Faculteit Bedrijf en Organisatie.

**HO
GENT**

Copyright © 2016-2020 Bert Van Vreckem & Jens Buysse

Dit werk valt onder de [Creative Commons Naamsvermelding-GelijkDelen 4.0 Internationale Publieke Licentie](#). De vormgeving is gebaseerd op Pieter van der Kloet's [thesis-sjabloon voor HOGENT](#).

14 april 2020

Inhoudsopgave

1	Inleiding	7
1.1	Soorten onderzoek	8
1.2	Structuur van deze gids	8
2	Een werkomgeving opzetten	9
2.1	Gebruik van \LaTeX	9
2.2	Bibliografische databank	10
2.2.1	APA-stijl	11
2.3	Versiebeheersysteem	11
2.4	Samenvatting	12
3	Een onderwerp kiezen	13
3.1	Het onderzoeksdomein kiezen	13
3.2	Onderzoeksvraag en -doelstellingen formuleren	14
3.3	Onderwerp uitschrijven	15
3.4	Struikelblokken	16
3.5	Samenvatting	17
4	Literatuuronderzoek	19
4.1	Doel van het literatuuronderzoek	19
4.2	Soorten bronnen	20
4.2.1	Publicatievormen	20
4.2.2	De kwaliteit van bronnen beoordelen	23
4.3	Publicaties bijhouden in JabRef	24
4.3.1	JabRef instellingen	24
4.3.2	Algemene bibliografische gegevens	26
4.3.3	Specifieke bibliografische gegevens	27
4.4	Op zoek naar relevante informatie	33
4.5	Samenvatting	35
5	Onderzoeksmethoden	37
5.1	De vergelijkende studie	37
5.1.1	Requirements-analyse	38
5.1.2	Long list	38
5.1.3	Short list en proof-of-concept	39
5.1.4	Conclusie	39

5.2	Proof-of-concept of testopstelling	39
5.2.1	Reproduceerbaarheid.	39
5.2.2	Repliceerbaarheid	40
5.2.3	Herbruikbaarheid	41
5.3	Experimenten uitvoeren en verzamelen van kwantitatieve data	41
6	Resultaten rapporteren	43
6.1	Visualisatie van cijfermateriaal	43
7	De bachelorproef schrijven	45
7.1	Algemene richtlijnen.	45
7.2	Structuur & taal	46
7.3	Afbeeldingen.	47
	Bibliografie	49

1

Inleiding

Deze gids is tot stand gekomen vanuit de begeleiding van de bachelorproef van de opleiding toegepaste informatica aan Hogeschool Gent. De bedoeling is om onze studenten wat houvast te geven in hoe ze hieraan in de eerste plaats moeten beginnen, en bevat heel wat tips i.v.m. methodologie, werken met \LaTeX voor een professionele opmaak, enz.

Hoewel de inhoud gegroeid is vanuit onze specifieke aanpak naar de bachelorproef toe, hopen we dat die ook relevant is voor andere (informatica-)studenten die een bachelorproef, thesis, of onderzoeksrapport moeten schrijven.

De doelstelling van de bachelorproef binnen onze opleiding is dat de student kan aantonen in staat te zijn om een “klant” (in de eigen organisatie of extern) **advies te geven over het in een bedrijfscontext toepassen van een actuele ICT-technologie**. Dat betekent concreet dat je in staat bent om je in te werken in een actueel onderwerp in ons vakgebied en dat je relevante, objectieve en gezaghebbende informatie over dat onderwerp kan verzamelen en kritisch beschouwen. Enerzijds gaat het dan over het opzoeken van bestaande kennis die te vinden is in de vakliteratuur, via experts of belanghebbenden (zogenaamde secundaire bronnen). Anderzijds verwachten we dat tijdens het onderzoek ook nieuwe kennis wordt gecreëerd, bijvoorbeeld via zelf opgezette experimenten, een proof-of-concept of prototype, interviews, enz. Al die informatie moet vervolgens gestructureerd en geanalyseerd worden, in het geval van kwantitatieve gegevens op een statistisch verantwoorde manier. Tenslotte wordt over het resultaat gerapporteerd in een op zichzelf staande tekst, de bachelorproef zelf.

Waar het op neerkomt is dat we verwachten dat een bachelor toegepaste informatica in staat is zich in te werken in een domein waar zij/hij nog niet noodzakelijk al expert in is en met de verzamelde informatie een oplossing uit te werken voor een *concreet* probleem uit het werkveld. We beogen geen academisch wetenschappelijke studie, maar wel een toegepast onderzoek met voldoende technische diep-

gang dat een meerwaarde biedt in het gekozen vakgebied.

1.1. Soorten onderzoek

Een bachelorproef hoeft geen academische masterthesis proberen na te bootsen. De professionele bachelor heeft zijn eigen waarde het is dus perfect mogelijk om te excelleren binnen de specifieke eigenheid van dit profiel. Het grote verschil bestaat in de eerste plaats uit het soort onderzoek dat gevoerd wordt.

Bij *fundamenteel onderzoek*, dat typisch aan universiteiten uitgevoerd wordt, ligt de nadruk op het uitbreiden van onze kennis in het algemeen. Binnen computerwetenschappen kan het bijvoorbeeld gaan over het ontwikkelen van nieuwe en/of efficiëntere algoritmen voor probleemoplossing. Of de resultaten van fundamenteel onderzoek ook onmiddellijk toepasbaar zijn is van secundair belang.

Toegepast onderzoek, daarentegen, probeert een antwoord te formuleren op een concrete vraag uit het werkveld. De onderzoeker zal dan aan de hand van de kennis die er op dit moment beschikbaar is (en gepubliceerd door vakexperten), proberen die vraag te beantwoorden. Dat kan bijvoorbeeld gaan over hoe een nieuwe technologie concreet in een bedrijfscontext kan toegepast worden, een keuze maken tussen verschillende alternatieve producten of technologieën, een vooronderzoek voorafgaand aan het ontwikkelen van een applicatie, enz. Bij toegepast onderzoek is de doelgroep ook heel specifiek, bijvoorbeeld één enkel bedrijf. We merken dan ook dat de onderwerpen die aangebracht zijn door bedrijven ook het best uitgewerkt zijn en het vaakst leiden tot een goede bachelorproef.

1.2. Structuur van deze gids

De rest van deze gids is als volgt gestructureerd:

Hoofdstuk 2 helpt bij het opzetten van een werkomgeving, meer bepaald \LaTeX en een versiebeheersysteem.

Hoofdstuk 3 geeft tips bij het kiezen van een onderwerp en uitschrijven van een onderwerp.

Hoofdstuk 4 gaat gedetailleerd in op het voeren van een literatuuronderzoek, het bijhouden van referenties in een bibliografische databank en hoe dit correct in een \LaTeX -document te verwerken.

Hoofdstuk 5 bespreekt een aantal onderzoeksmethoden die typisch gebruikt worden in een bachelorproef, o.a. de vergelijkende studie, of het opzetten van experimenten.

Hoofdstuk 7, tenslotte bevat een aantal tips in verband met het schrijven van de tekst, en enkele \LaTeX -specifieke richtlijnen.

2

Een werkomgeving opzetten

In dit hoofdstuk behandelen we het opstarten van het werk aan een bachelorproef. Je vindt er enkele aanbevelingen over te gebruiken tools en het onderzoeksproces.

2.1. Gebruik van \LaTeX

De meeste studenten zijn gewend om opgemaakte tekst met een klassieke tekstverwerker (typisch MS Word) te schrijven. Voor je bachelorproef is het aangewezen om hier van af te stappen.

Word, zeker met het standaardsjabloon, geeft een layout die niet geschikt is voor publicatie. Eens de lengte en complexiteit van een Word-document toenemen (en bij een eindwerk is dat zeker het geval), krijg je te maken met inconsistenties in de layout van je tekst, paginanummering en slecht gepositioneerde afbeeldingen.

Wanneer je tekst kopieert vanuit een ander (voorbereidend) document of vanuit een website, wordt de oorspronkelijke layout overgenomen. Als die niet consistent is met deze van je hoofddocument, moet je alles gaan aanpassen. Dit is een bijzonder tijdrovend werk met een grote kans op fouten.

Een klassieke tekstverwerker die gebaseerd is op het WYSIWYG-principe¹, laat je toe om tot in de puntjes te bepalen waar tekst op het papier terecht komt, maar eigenlijk is deze grote vrijheid in dit geval een nadeel. Een strakke en professionele vormgeving is een specialiteit die een grote aandacht voor vaak pietluttige details vraagt. Als informaticus hebben wij niet de nodige kennis om dit te realiseren. Wanneer je een significant deel van de tijd bezig bent met het vormgeven van je document, word je bovendien afgeleid van de kern van de zaak: de inhoud van de tekst!

Een ander nadeel van de klassieke tekstverwerker is het binaire bestandsformaat. Dit maakt het onmogelijk om een document in een versiebeheersysteem te ste-

¹What You See Is What You Get, zoals je wellicht weet

ken (zie Sectie 2.3). Al gauw gaan er verschillende versies van het document naast elkaar leven: 'bachproef 3.docx', 'bachproef 5 30 maart.docx', 'final draft.docx', 'final draft na feedback.docx', 'final final draft.docx' Je hebt versies op je laptop, op dropbox, op je vaste pc en in je mailbox. Op de duur is het overzicht zoek, vergeet je stukken tekst over te nemen of maak je andere fouten.

Voor het opmaken van een lange tekst met een professionele en strakke vormgeving is \LaTeX een aanrader. Zoals je weet is dit een *tekstzetsysteem* met een markuptaal (zoals HTML) die gespecialiseerd is in het op papier zetten van tekst. Je schrijft broncode in \LaTeX markup, een 'compiler' genereert een PDF. \LaTeX is tekstgebaseerd, dus je kan dit in een versiebeheersysteem steken.

Toegegeven, \LaTeX heeft wel degelijk enkele nadelen. Er is een niet te onderschatten leercurve, en zolang je vasthoudt aan de gewoonten die je overgehouden hebt aan het werken met een tekstverwerker, doet \LaTeX niet altijd wat je verwacht. Maar je moet de meeste inspanning leveren in het begin, om \LaTeX onder de knie te krijgen. Bij het schrijven van een bachelorproef met een tekstverwerker heb je het meeste werk op het einde, om alle onvolkomenheden, inconsistenties en fouten in de vormgeving weg te werken. Op dat moment heb je daar meestal niet meer voldoende tijd voor, want de deadline nadert. Het gevolg is vrijwel altijd een document dat onvoldoende afgewerkt is en er heel onprofessioneel uitziet voor de lezer. Dit past niet bij een werkstuk dat dient als afsluiter van een *professionele* bacheloropleiding.

In de rest van deze gids gaan we er van uit dat je \LaTeX gebruikt. Het is niet de bedoeling dat dit een \LaTeX handleiding wordt, daarvoor zijn er voldoende andere bronnen beschikbaar (Oetiker, Partl, Hyna & Schlegl, 2015).

Je kan een \LaTeX -sjabloon voor het opmaken van de bachelorproef vinden op de Github repository <https://github.com/HoGentTIN/bachproef-latex-sjabloon>. Via de groene knop rechtsboven kan je het sjabloon downloaden. De repository klonen of een fork aanmaken is niet aan te raden. Daarmee neem je immers ook de hele historie van het sjabloon over en die is niet relevant voor jouw werk.

2.2. Bibliografische databank

Een vast onderdeel van een bachelorproef is het voeren van een literatuurstudie om je in te werken in het onderzoeksdomein (zie Hoofdstuk 4). Het is belangrijk om goed bij te houden wat je allemaal leest, zodat je bij het schrijven van de inleiding kan verwijzen naar je bronnen. Het verwijzen naar bronnen en opmaken van een bibliografie moet op een strakke, strikt vastgelegde manier gebeuren. Dit is iets dat je niet manueel hoeft te doen, er bestaan verschillende softwarepakketten die dit grotendeels automatiseren: bibliografische databanken.

Een bibliografische databank laat je toe metadata over de gelezen werken gestructureerd bij te houden: titel, auteur, jaartal, en dergelijke, maar ook (aanklikbare) URLs, PDFs van artikels, nota's, enz.

Er zijn verschillende mogelijkheden, maar JabRef² is voor onze doeleinden wellicht de interessantste. JabRef is een open source bibliografische databank geschreven in Java en bij uitstek geschikt voor het werken met \LaTeX . Het gebruikt als bestandsformaat hetzelfde als Bib \LaTeX , het in \LaTeX ingebouwde systeem voor bibliografieën.

2.2.1. APA-stijl

De manier waarop een literatuurlijst en verwijzingen in de tekst moeten opge maakt worden ligt vast. Er zijn echter wel verschillende opmaakstijlen. Eén daarvan is de zogenaamde APA-stijl die afkomstig is van de American Psychological Association (APA). Het is de belangrijkste standaard voor publicaties in het domein van de sociale wetenschappen. De officiële handleiding van de APA-stijl is erg uitgebreid en kan daardoor onhandig in het gebruik zijn. In het \LaTeX -sjabloon voor de bachelorproef zit de APA-stijl al ingebakken en je hoeft je dus helemaal geen zorgen te maken over de correcte opmaak. Dit gebeurt automatisch, mits je de bibliografische gegevens (bv. titel, auteur, jaar van publicatie) correct bijhoudt in Jabref. Meer daarover in Sectie 4.3.

Let op!! Het gebruik van de APA-stijl is verplicht over heel HOGENT. Dat geldt zowel voor de opmaak van de literatuurlijst als verwijzingen naar bronnen in de tekst.

2.3. Versiebeheersysteem

Aan een informaticus hoeven hopelijk de voordelen van een versiebeheersysteem niet uitgelegd te worden? Gebruik altijd een versiebeheersysteem zoals Git om je werk bij te houden. Creëer ook een Github repository. Dit is enerzijds een goed backupsysteem (mits je regelmatig synchroniseert met Github), en anderzijds laat het je toe om je werk te delen met je promotor. Eén van de eigenschappen van een versiebeheer is dat het bij uitstek ontworpen is om wijzigingen in *tekstbestanden* op te volgen. Binaire bestandsformaten zoals documenten van een klassieke tekstverwerker zijn hiervoor niet geschikt, wat een extra motivatie is voor het gebruik van \LaTeX .

Volgende zaken horen zeker thuis in je repository:

- \LaTeX -broncode van de bachelorproef.
- in te voegen afbeeldingen.
- broncode van zelf geschreven scripts, benchmarks, experimenten, proof-of-concepts, enz. Dit maakt dat je experimenten makkelijker te reproduceren en te valideren zijn door derden.
- ruwe resultaten experimenten (in tekstformaat, bv CSV), transcripties van interviews, enz.

²<http://www.jabref.org/>

- losse nota's, ideeën, enz. Gebruik hiervoor Markdown en vermijd Word-documenten.

Kortom, *alle* artefacten die resulteren uit je onderzoek horen thuis in de repository. Voor werkdocumenten waar je opgemaakte tekst wenst, maar waarvoor \LaTeX overkill is gebruik je Markdown³.

Wat hoort *niet* thuis in je repository:

- Hulpbestanden aangemaakt bij het compileren van \LaTeX . Je kan ervoor zorgen dat deze niet in een repository opgenomen worden door een `.gitignore`-bestand aan te maken⁴. Het \LaTeX -sjabloon voor de bachelorproef is al goed ingesteld.
- Grote (binaire) bestanden zoals ISO's, virtuele machines (bv. `.ova`), enz.
- PDFs van de artikels/ebooks die je gelezen hebt (dit wordt beschouwd als “herdistribueren” en mag niet onder de auteurswetgeving).
- Binaire bestanden die vaak veranderen, bv. Word-documenten.
- Bestanden die automatisch gegenereerd worden uit code in Git, bv. gecompileerde code.

Een versiebeheersysteem wordt pas echt nuttig als je het goed gebruikt. Commit dus zo vaak mogelijk, schrijf duidelijke commit-boodschappen en synchroniseer regelmatig met Github!

2.4. Samenvatting

De kernpunten van dit hoofdstuk zijn:

- Schrijf je tekst in \LaTeX in plaats van een klassieke tekstverwerker voor een strakke, professionele opmaak;
- Gebruik een *reference manager* voor het bijhouden van een bibliografische databank (JabRef is aanbevolen).
- Gebruik een versiebeheersysteem om al je werk in op te slaan (Git is aanbevolen);

³<https://guides.github.com/features/mastering-markdown/>

⁴Bijvoorbeeld <https://github.com/github/gitignore/blob/master/TeX.gitignore>

3

Een onderwerp kiezen

In dit hoofdstuk worden enkele suggesties gegeven voor het zoeken naar een onderwerp. De website van de HOGENT bib heeft daar ook enkele algemene raadgevingen over¹, deze gids is specifiek gericht op de bachelor toegepaste informatica. Vanuit de opleiding krijgen we van externen regelmatig aanbiedingen van onderwerpen die geschikt zijn voor een bachelorproef, maar dat aanbod is niet groot genoeg om alle studenten van een onderwerp te voorzien. Langs de andere kant is het zelf uitwerken van een onderwerp een interessante kans om je te verdiepen in een onderwerp waar je na je afstuderen graag mee zou verder gaan.

3.1. Het onderzoeksdomein kiezen

Een eerste stap is het kiezen van je onderzoeksdomein. Dit is iets waar eigenlijk niemand je mee kan helpen. Kies een domein waar je zelf in geïnteresseerd bent, zodat je voldoende motivatie kan opbrengen om je hier in te verdiepen. Je gekozen specialisatierichting in het laatste jaar is wellicht een goed startpunt. In welk soort job zou je na je afstuderen graag starten? Met welke technologieën/platformen/...zou je graag werken?

Ga op zoek te naar de actualiteit binnen je gekozen vakdomein. Belangrijk is je tijd te nemen om je “onder te dompelen” in de actualiteit. Dit lukt niet op een avond. Het is efficiënter om hier gedurende een aantal weken regelmatig wat tijd in te steken (bv. elke dag een uur). Op de duur zou je in principe de belangrijkste thema’s moeten herkennen waar men op dit moment vooral mee bezig is en dat zou inspiratie kunnen geven voor je onderwerp.

In Sectie 4.4 vind je enkele concrete tips en startpunten voor het zoeken van relevante informatie.

¹<https://bib.hogent.be/how-to/onderwerp-formuleren/inleiding/>

3.2. Onderzoeksvraag en -doelstellingen formuleren

Eens je voeling krijgt met de actualiteit van een onderwerp, leer je typisch ook de belangrijkste problemen en discussiepunten kennen. Die kunnen aanleiding geven tot het formuleren van je hoofdonderzoeksvraag, die je verder kan opsplitsen in concretere deelonderzoeksvragen.

Een goede onderzoeksvraag voor een bachelorproef start vanuit een reële of in elk geval realistische *bedrijfscasus*, een concreet probleem waar een specifiek bedrijf mee worstelt. Dat zorgt meteen ook voor een duidelijke *afbakening* van het onderwerp. Een te vage of algemene onderzoeksvraag leidt nooit tot een goede bachelorproef.

Bijvoorbeeld, “Wat is de beste agile methodologie voor bedrijven?” is in dat opzicht *geen* goede onderzoeksvraag. Er zijn verschillende agile methodologieën, en daar is allicht een goede reden voor. Er is geen duidelijke winnaar die voor elk bedrijf het meest geschikt is. Als dat wel zo was, dan zou intussen, een twintigtal jaar na de publicatie van het Agile Manifesto (Beck e.a., 2001), al duidelijk zijn wat de beste methodologie is. Een geschikte methodologie hangt af van veel factoren: de bedrijfscultuur, het soort bedrijf, het soort it-oplossingen dat ontwikkeld wordt, de grootte van het bedrijf, enz. Als je je onderzoeksvraag te algemeen formuleert, zal je nooit tot een sluitende conclusie kunnen komen. Dit hebben we in het verleden al te vaak ondervonden. De student schrijft dan in de conclusie “de beste oplossing hangt af van je voorkeur en je specifieke situatie.” Met andere woorden, er *is* geen conclusie. Start dus vanuit een concrete casus, bv. bedrijf XYZ worstelt met het tijdig en binnen budget opleveren van software en wil een agile methodologie gaan toepassen. Wat is voor hun specifieke situatie de meest geschikte methodologie? Daarmee is het onderwerp duidelijk afgebakend en de meerwaarde ervan is meteen zichtbaar.

Op goede onderzoeksvraag bestaat er nu nog geen sluitend antwoord. Vragen als “Wat is data mining?”, “Welke PHP frameworks zijn er?”, “Wat is de geschiedenis van de Cloud?” zijn dus niet geschikt, want het antwoord is snel te vinden door even te zoeken op Wikipedia of Google, of door relevante vakliteratuur na te slaan. Het is ook belangrijk om meteen na te denken over het verwachte resultaat, de *onderzoeksdoelstelling*. Onder welke omstandigheden kan je spreken van een succesvolle bachelorproef? Wat is het concrete eindresultaat dat je wil bereiken?

Voor het formuleren van je onderzoeksdoelstelling(en) kan je gebruik maken van het S.M.A.R.T.-principe (van Uchelen & Jungjohann, 2003):

Specifiek Is de doelstelling eenduidig? Is het duidelijk wie er baat bij heeft?

Meetbaar Onder welke (meetbare/observeerbare) voorwaarden of vorm is het doel bereikt? Is het doel om een proof-of-concept of prototype af te leveren? Een analyse voor een te ontwikkelen applicatie? Een vergelijking tussen mogelijke alternatieve oplossingen met een duidelijk advies? Enz.

Acceptabel Zijn deze doelen acceptabel voor de doelgroep? Zal de doelgroep de voorgestelde oplossing ook daadwerkelijk kunnen gebruiken?

Realistisch Is het doel haalbaar? Is het onderwerp voldoende afgebakend? Is de moeilijkheidsgraad van het probleem op het niveau van de professionele bachelor?

Tijdsgebonden Wanneer (in de tijd) moet het doel bereikt zijn? Voor een bachelorproef is dit eenduidig vastgelegd: de deadline voor indienen.

3.3. Onderwerp uitschrijven

Eens je een onderzoeksvraag hebt, kan je je onderwerp uitschrijven om het in te dienen ter goedkeuring. Dat betekent dat je ook al wat literatuuronderzoek gaat uitvoeren. Voor aanwijzingen over de aanpak hiervan, zie Hoofdstuk 4.

Wat je leest over het onderwerp, moet je dan structureren en formuleren in je eigen woorden in een doorlopende tekst. Een goed hulpmiddel om je gedachten over een onderwerp te ordenen om er later een gestructureerde tekst rond te schrijven is het opzetten van een mindmap. Er bestaan hiervoor verschillende tools die je kosteloos kan gebruiken, zoals bijvoorbeeld XMind² of FreeMind³.

In dit stadium is het nog niet de bedoeling een volledig uitgewerkte literatuurstudie uit te schrijven, maar een aantal referenties worden wel verwacht. Je promotor moet na het lezen van je voorstel begrijpen wat de context van je onderzoek is en waarom er een probleem is dat om een oplossing vraagt. Je moet dit kunnen aantonen aan de hand van gezaghebbende vakliteratuur.

Denk ook na over een titel voor je bachelorproef. Die hoeft nog niet definitief te zijn, maar een goede titel maakt duidelijk welke richting je precies wil uitgaan. Een concrete titel geeft je begeleider(s) het vertrouwen dat je weet wat je precies wil gaan doen en dat dit een realistische doelstelling is. Probeer te letten op het volgende bij het formuleren van een titel:

- Formuleer de titel niet als een vraag.
- Gebruik geen vakjargon en zeker geen afkortingen. De titel moet ook begrijpbaar zijn voor iemand buiten jouw specifieke vakgebied.
- Enkel je vakdomein benoemen is onvoldoende want veel te vaag. Je titel moet concreet zijn en duidelijk maken wat je precies wil onderzoeken. Dat betekent dus ook dat een titel gerust lang mag zijn. bv. “Cloud computing” is een algemene term die veel ladingen dekt, en die dus niets zegt. “De selectie van een open source Infrastructure as a Service platform voor het opzetten van een testomgeving voor webontwikkeling” is een stuk concreter.

²<https://www.xmind.net/>

³<http://freemind.sourceforge.net/>

Bij het beoordelen van een onderwerp, wordt rekening gehouden met volgende criteria:

- Er is een concrete, **duidelijk afgebakende onderzoeksvraag**, onderzoeksdoelstelling.
- Het voorstel is **vernieuwend** en heeft een duidelijke **meerwaarde** voor een specifieke doelgroep uit het ict-werkveld.
- De **methodologie** is duidelijk verantwoord, onderzoekstechnieken zijn geschikt voor beantwoorden onderzoeksvraag
- Er is een duidelijke **eigen bijdrage** en **technische diepgang**.

3.4. Struikelblokken

Volgende zaken geven ons het idee dat je onderwerp nog onvoldoende is uitgewerkt of dat er nog belangrijke struikelblokken zijn die een succesvolle bachelorproef in de weg staan:

- Er is geen concrete reële of realistische **bedrijfs casus** gekoppeld aan het onderwerp. Onderzoek op bachelorniveau is toegepast, m.a.w. we proberen concrete, reële problemen op te lossen. Dat moet ook in je onderwerp naar voor komen.
- (Eén van) de onderzoeksdoelstelling(en) is **speculatie over de toekomst** (bv. “wat zal de impact zijn van het Internet of Things op het dagelijks leven?”). De conclusie van een bachelorproef moet aantoonbaar zijn, toekomstvoorspellingen zijn dat nooit.
- Het resultaat van het onderzoek is in grote mate **afhankelijk van externe factoren**. Als je bijvoorbeeld een enquête zal voeren, is het belangrijk te beseffen dat het niet makkelijk is om voldoende respondenten te vinden. Je zal dus al meteen moeten aangeven op welke manier je van plan bent om een voldoende grote steekproef te nemen. Ook als je plan is om interviews af te nemen bij experten, bedrijven, enz. is het belangrijk om al voor de start van je onderzoek de nodige contacten te leggen. Als het immers niet lukt om tijdig de nodige personen te kunnen spreken, komt het resultaat van je bachelorproef in gevaar.
- Het onderwerp vertoont onvoldoende **technische diepgang**. De professionele bachelor toegepaste informatica is een technisch profiel en we verwachten dat je ook aantoonbaar technisch sterk te staan. Een onderwerp als “wat is de meest gebruiksvriendelijke mobiele applicatie voor thuisbankieren voor senioren?” voldoet hier bijvoorbeeld *niet* aan. Dit soort applicaties is wordt door

banken ontwikkeld en als student zal je nooit toegang krijgen tot de interne werking. Je zal je wellicht moeten beperken tot interviews en/of enquêtes bij de doelgroep. Op deze manier kan je niet aantonen wat je als informaticus in je mars hebt.

- De **referentielijst** is te kort of bestaat uit ongeschikte bronnen. Neem de nodige tijd om een initiële literatuurstudie te voeren en ga verder kijken dan het eerste het beste blogartikel. In Hoofdstuk 4 vind je aanwijzingen om dit aan te pakken.

3.5. Samenvatting

- Neem de tijd om een onderwerp te zoeken dat je ligt.
- Bij het uitschrijven van een onderwerp ben je best zo *concreet* mogelijk.
- Vermijd struikelblokken als het ontbreken van een concrete casus, speculatie, afhankelijkheid van externe factoren en een te oppervlakkige literatuurstudie.

4

Literatuuronderzoek

De eerste fase in elk onderzoek is typisch om een overzicht te schrijven van de huidige stand van zaken in het onderzoeksdomein. Daarvoor is het nodig om je te verdiepen in *alles* wat er over het onderwerp al geschreven is. Dit is het literatuuronderzoek. In elk verslag over onderzoek is het essentieel dat je elke bewering die je doet ook kan aantonen. Dat kan ofwel op basis van data die je zelf op een methodologisch correcte manier verzameld en geanalyseerd hebt (maar dat komt verder in deze gids aan bod), ofwel aan de hand van referenties naar *gezaghebbende* publicaties.

Dit hoofdstuk gaat dieper in op dit onderwerp: wat bereik je precies met een literatuurstudie, hoe kan je er aan beginnen, hoe kan je de bronnen die je vindt gestructureerd bijhouden en op welke manier gebruik je die dan in je tekst?

HOGENT heeft via Chamilo een cursus over informatievaardigheden gepubliceerd¹ die je zeker eens moet doornemen. Het is immers niet de bedoeling om in deze gids dezelfde inhoud te herhalen. Dit hoofdstuk spitst zich in de eerste plaats toe op literatuuronderzoek over een ict-gerelateerd onderwerp, en het correct opmaken van een referentielijst met \LaTeX en JabRef (zie Sectie 2.2).

Neem ook opnieuw de cursus Onderzoekstechnieken bij de hand, in het bijzonder het lesmateriaal ivm \LaTeX en rapporteren over onderzoek.

4.1. Doel van het literatuuronderzoek

Het belangrijkste doel van een literatuuronderzoek is om vertrouwd te worden met het onderzoeksdomein en die kennis ook door te geven aan de lezers van je bachelorproef. Je gaat dus zoveel mogelijk informatie verzamelen en lezen over het

¹https://chamilo.hogent.be/index.php?go=CourseViewer&application=Chamilo%5CApplication%5CWebIcms&course=22068&tool=LearningPath&browser=Table&tool_action=ComplexDisplay&publication=980981

onderwerp zodat je er eigenlijk alles over weet dat er op dit moment over te weten valt. Het is dan de bedoeling om alle voor je onderzoek relevante kennis die je op deze manier hebt opgedaan ook op een gestructureerde manier en in je eigen woorden samen te vatten in een doorlopende tekst. Dit vormt meestal het eerste hoofdstuk (of de eerste hoofdstukken) van je bachelorproef.

Aan de hand van de literatuurstudie geef je de lezer de nodige achtergrond om het onderwerp te begrijpen. Het is een inleiding op het onderwerp, en bespreekt de huidige stand van zaken. Je vermeldt wat de experts in het domein er over te zeggen hebben en welk onderzoek er in het verleden al over gedaan is (met uiteraard vermelding van de belangrijkste conclusies). Uit de literatuurstudie moet ook naar voren komen dat er nog hiaten in onze kennis zijn, dat er een probleem is dat om een oplossing vraagt. En dat is uiteraard precies het onderwerp van je bachelorproef.

4.2. Soorten bronnen

In het kader van een onderzoek kunnen we informatiebronnen ondeverdelen in deze drie categorieën:

Primaire kennis die je zelf vergaart tijdens je onderzoek. Bijvoorbeeld metingen uit experimenten, resultaten van enquêtes, transcripties van interviews, enz.

Secundaire publicatie van kennis, onderzoek, enz. door anderen. Bijvoorbeeld artikels in vaktijdschriften, boek, presentatie op een conferentie, enz.

Tertiaire zoekindexen en encyclopedieën. Bijvoorbeeld Google Scholar, Web of Science, Elsevier ScienceDirect, Arxiv.org, Wikipedia, about.com, Webopedia, enz.

Wanneer in een tekst verwezen wordt naar de literatuur, dan gaat het telkens over *secundaire* bronnen (ook *publicaties* genoemd). Dat betekent ook dat primaire of tertiaire bronnen *niet* kunnen. Er mag dus bijvoorbeeld nooit verwezen worden naar een Wikipedia-artikel, woordenboeken, enz. Tertiaire bronnen zijn wel goede *startpunten* van de zoektocht naar geschikte publicaties (Zie Sectie 4.4). Je kan ook niet verwijzen naar het verslag van een interview dat je gevoerd hebt, omdat dat ook niet gepubliceerd is en dus niet toegankelijk voor de lezer.

4.2.1. Publicatievormen

Kennis wordt doorgegeven en gepubliceerd in verschillende vormen. In deze sectie lijsten we de belangrijkste op en bespreken de betrouwbaarheid en objectiviteit van elk.

Artikel in wetenschappelijk tijdschrift

Van een artikel dat in een wetenschappelijk tijdschrift (of Eng. *journal*) gepubliceerd wordt, mag je uitgaan dat er eerst een rigoreus verificatieproces aan vooraf gegaan is. Ingezonden artikels worden door de redacteurs van het tijdschrift doorgegeven aan andere experts in het vakgebied die de verantwoordelijkheid hebben de inhoud ervan op een onafhankelijke manier te verifiëren. Dit noemt men in het Engels *peer review*. Dit proces duurt typisch verschillende maanden en kan zelfs uitlopen tot een paar jaar. Publicaties in wetenschappelijke tijdschriften worden algemeen beschouwd als de meest betrouwbare en als er over jouw bachelorproefonderwerp te vinden zijn, is het ten zeerste aan te raden die te lezen. Nadelen is dat het niveau (vooral op vlak van wiskunde) typisch vrij hoog ligt en dus niet altijd even toegankelijk voor de gemiddelde bachelorstudent.

Artikel in een vaktijdschrift

Vaktijdschriften zijn gericht op een professioneel publiek, dus dit zijn geen academische, wetenschappelijke teksten. Er gaat ook geen peer review-proces aan de publicatie van artikels vooraf. Typisch beslist een redacteur of een artikel al dan niet gepubliceerd wordt. Vaktijdschriften zijn ook meer en meer enkel online beschikbaar en we rekenen hieronder ook portaalsites rond een bepaald thema of vakgebied, zoals dZone², Informit³, enz.

Het is dan belangrijk te weten wie de auteur van het artikel is. Is dit een erkend vakexpert of een journalist? In het eerste geval is het artikel zeker bruikbaar als bron, maar in het andere moet je er toch eens kritisch naar kijken. Er is immers zelden garantie dat een journalist voldoende expertise in het onderwerp van zijn artikel heeft. Journalisten hebben ook andere drijfveren dan vakexperten en willen vooral dat hun artikel door zoveel mogelijk mensen gelezen wordt. Soms zullen ze de zaken dan wat sensationeler voorstellen dan ze eigenlijk zijn, of slaan ze de bal mis als het gaat over technische details.

Presentatie op een conferentie

Zowel de wetenschappelijke als de professionele gemeenschap organiseren wereldwijd conferenties om met elkaar te overleggen, om nieuwe resultaten te presenteren en kennis door te geven. Typisch wordt er enkele maanden voor de start een oproep gedaan om onderwerpen voor presentaties voor te stellen. Bij een wetenschappelijke conferentie wordt dan meestal gevraagd om een uitgeschreven artikel in te dienen dat wordt beoordeeld via een peer-reviewproces dat meestal wel een stuk minder zwaar is dan voor een journal. Voor vakconferenties en voor sommige wetenschappelijke conferenties is een paragraaf tekst met een samenvatting van de inhoud (abstract) voldoende. Bij vakconferenties zal meestal een

²<https://dzone.com/>

³<https://www.informit.com/>

panel dat door de organisator is samengesteld de inzendingen beoordelen en de presentaties selecteren.

Na afloop van een conferentie wordt de inhoud van de geselecteerde presentaties gebundeld en gepubliceerd. Bij een wetenschappelijke conferentie is dat een (e-)boek dat bestaat uit de ingezonden artikels, wat men in het Engels de *proceedings* noemt. Bij een vakconferentie gaat het meestal enkel over de gebundelde presentatieslides, of zetten de sprekers zelf hun slides op een website om presentaties te delen, zoals SlideShare of Speaker Deck. Meer en meer vakconferenties nemen sommige of alle presentaties op en maken die beschikbaar via bv. Youtube of Vimeo, of via een website in eigen beheer.

Het loont de moeite om uit te zoeken welke conferenties er doorgaan rond het vakgebied waarbinnen jouw gekozen onderwerp past en zo mogelijk de presentaties te bekijken die er zijn doorgegaan.

Qua betrouwbaarheid benadert het niveau van artikels in *conference proceedings* die van wetenschappelijke artikels in journals. Dit geldt minder voor vakconferenties, omdat er geen peer-review proces aan vooraf gaat. Je kan er wel de belangrijkste experts over een bepaald vakdomein leren kennen en veel bijleren over recente ontwikkelingen binnen het vakgebied.

Thesis

Thesissen zijn ook vaak interessant als informatiebronnen. De diepgang hangt hier grotendeels af van de opleiding waarvoor de thesis geschreven is: doctoraatsthesis (PhD thesis), masterthesis of bachelorproef. Deze teksten zijn geschreven onder begeleiding van een promotor die borg staat voor de kwaliteit van de inhoud. Als je een gepubliceerde thesis vindt, dan is die dus in principe nagelezen door een expert. Doctoraatsthesisen staan wat dat betreft ongeveer op het niveau van wetenschappelijke artikels. Meestal is het ook zo dat een of meerdere onderdelen van een doctoraatsthesis ook als artikels in een wetenschappelijk tijdschrift zijn gepubliceerd.

Boek of handleiding

Ook bij boeken is het belangrijk om te weten wie de auteur is en in hoeverre die de autoriteit heeft om over een onderwerp te schrijven. In principe kan iedereen immers een boek uitgeven, en er is geen formele peer-review, dus geen onafhankelijke validatie van de inhoud.

White paper

Een *white paper* is een rapport over een bepaald onderwerp dat als bedoeling heeft de lezer voldoende achtergrondinformatie te geven over dat onderwerp om het te begrijpen, beslissingen te nemen of een probleem op te lossen. In ons vakdomein worden white papers typisch uitgegeven door bedrijven die een product verkopen

gerelateerd aan het behandelde onderwerp. Een bedrijf dat antivirussoftware verkoopt kan bijvoorbeeld white papers uitgeven over het beveiligen van computers, hoe wachtwoorden gekraakt worden, enz.

Het is belangrijk om te beseffen dat white papers meestal niet objectief zijn. De auteurs hebben iets te verkopen, dus het is voor hen belangrijk om het onderwerp op een zodanige manier te presenteren dat het aankopen van hun producten of diensten interessant lijkt. Een leverancier van beveiligingssoftware heeft er bijvoorbeeld belang bij om het probleem van cybercriminaliteit erger voor te stellen dan het in werkelijkheid is. De lezer die ongerust wordt over de toestand is meer geneigd om beveiligingssoftware te kopen.

Lees een white paper dan ook met een zeer kritische blik en tracht ook objectieve informatie uit andere bronnen te vinden.

Blogartikel

Een blog is meestal (een deel van) een persoonlijke website waar de auteur regelmatig artikels publiceert rond een bepaald onderwerp en haar/zijn kennis deelt met anderen in hetzelfde vakgebied. Over ict-gerelateerde onderwerpen zijn er duizenden blogs, en de kans is groot dat de belangrijkste experts binnen je gekozen onderwerp er één bijhouden.

Ook hier is het belangrijk om te achterhalen wie de auteur is en welke autoriteit die heeft over het onderwerp. Wanneer je een artikel vindt over bijvoorbeeld “continuous delivery” van Martin Fowler, een wereldwijd erkend expert en spreker rond software-ontwikkeling, dan is dit heel bruikbaar als bron. Een artikel over hetzelfde onderwerp door een andere auteur die bijvoorbeeld na wat zoeken op LinkedIn een marketeer blijkt te zijn, neem je best niet op in je literatuurlijst.

4.2.2. De kwaliteit van bronnen beoordelen

Uit de voorgaande sectie zou je al moeten opgevallen zijn dat de kwaliteit van bronnen niet altijd makkelijk te evalueren is. Veel hangt af van wie de auteur is en welke autoriteit die heeft binnen het vakgebied.

Een hulpmiddel bij het beoordelen van een bron is de *CRAP-test* (Gratz, 2015):

Currency of actualiteit: is de bron voldoende recent voor het onderwerp?

Reliability/Relevance of betrouwbaarheid/relevantie: is de inhoud goed onderbouwd? Wordt er naar bronnen verwezen? Is de inhoud relevant voor jouw onderzoek?

Authority of autoriteit: is de auteur een autoriteit over het onderwerp? Gaat het over een persoon of een organisatie?

Point of view of objectiviteit: wat is de intentie van de auteur? Wat wil die bereiken?

Bij het beoordelen van een bron is het dan uiteraard noodzakelijk om te weten te komen wie de auteur is en wanneer die geschreven en gepubliceerd is. Van vele websites is dit jammer genoeg niet mogelijk en wordt deze informatie niet gegeven. Dit soort bronnen hoort niet thuis in een literatuurlijst!

4.3. Publicaties bijhouden in JabRef

Voordat je op zoek gaat naar informatie is het belangrijk om je eerst voor te bereiden om alle bronnen gestructureerd bij te houden zodat je ze later kan terugvinden en een bibliografie kan opstellen. Bibliografieën moeten in een strak en vastgelegd formaat worden opgesteld. Er zijn veel verschillende bibliografiestijlen, maar op HOGENT is er gekozen voor één, het APA-systeem (van de American Psychological Association)⁴.

Het is ondoenbaar om manueel een bibliografie en vermeldingen in de tekst bij te houden. Gelukkig bestaan er verschillende applicaties die gespecialiseerd zijn in het bijhouden van een bibliografische databank, zgn. *reference managers*.

HOGENT stelt zelf Endnote voor, maar het nadeel is dat dit een commerciële applicatie is, waar je na je afstuderen geen toegang meer toe hebt. In deze sectie wordt Jabref voorgesteld, een open source applicatie voor het bijhouden van bibliografische gegevens, die bovendien ontwikkeld is specifiek voor \LaTeX . JabRef gebruikt het bestandsformaat van Bib \LaTeX , het ingebouwde systeem voor bibliografieën.

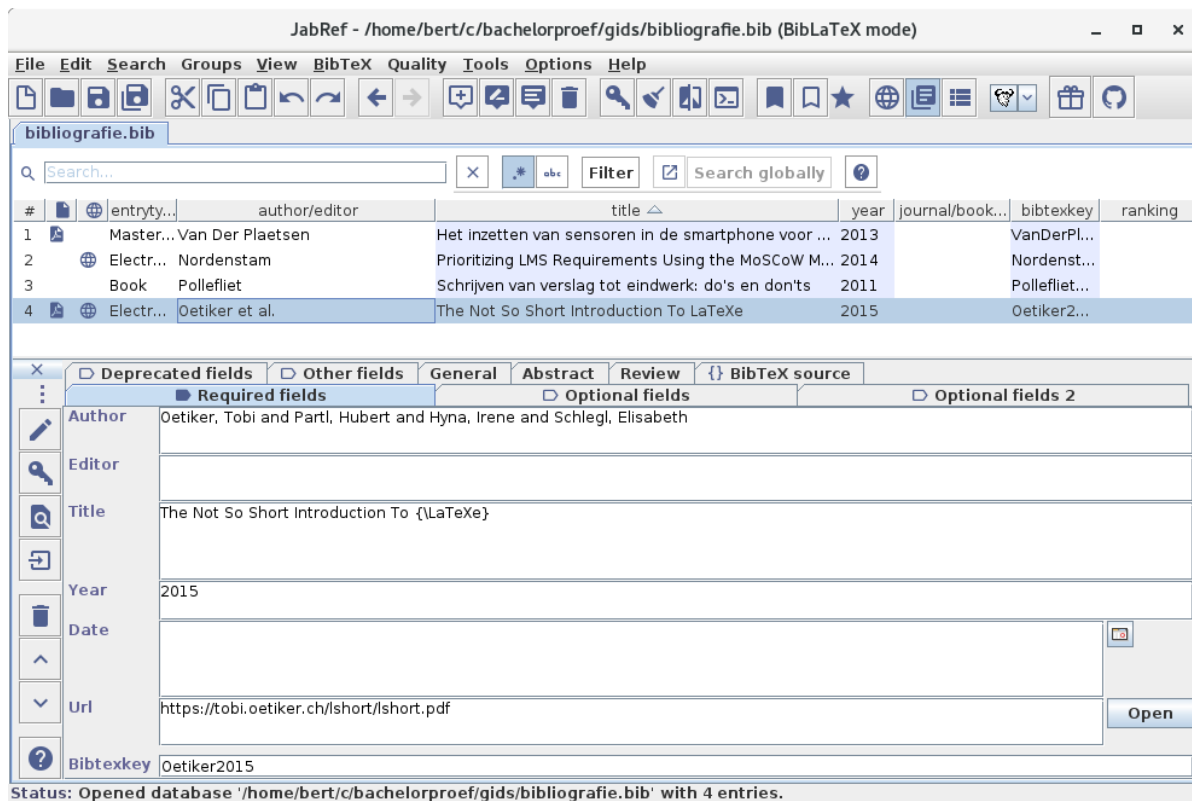
Als je meer gedetailleerde informatie over Bib \LaTeX nodig hebt die niet in deze gids te vinden is, raadpleeg dan de handleiding (Lehman, Kime, Boruvka & Wright, 2016).

4.3.1. JabRef instellingen

Je kan Jabref downloaden van <http://www.jabref.org/> en installeren op zowel Windows, MacOS als Linux. Bij het voor de eerste keer opstarten is het nuttig om volgende instellingen aan te passen:

- Kies in het menu voor File > Switch to BibLaTeX mode. Dit maakt de bestandsindeling van de bibliografische databank compatibel met het aangeboden \LaTeX -sjabloon voor de bachelorproef.
- Kies in het menu voor Options > Preferences en dan voor de categorie “BibTeX key generator”. Elke bron in de databank wordt geïdentificeerd door een unieke sleutel die je kan automatisch laten genereren. Je kan de vorm ervan zelf instellen. Het standaardformaat is de familienaam van de eerste auteur gevolgd door het publicatiejaar, bv. “Knuth1998”. Je kan dit naar eigen smaak aanpassen, maar kies dit vooraf en hou je er aan. Je zal deze sleutel gebruiken om vanuit de tekst te verwijzen naar je bronnen, bv. met het commando `\parencite{Knuth1998}`.

⁴<https://bib.hogent.be/how-to/bronnen-vermelden/refereren-volgens-apa6th/>



Figuur 4.1: Jabref. Centraal in de gebruikersinterface vind je een *overzicht* van de verschillende bronnen in deze bibliografische databank, in dit geval vier. De *pdf-icoontjes* links bij de eerste en vierde bron geven aan dat de bron lokaal opgeslagen is als pdf. Als je er op klikt, wordt de pdf geopend. De *wereldbol-icoontjes* bij de tweede en vierde bron duiden op een url die je in de webbrowser kan openen als je er op klikt. Onderaan bevindt zich een *detailvenster* met de bijgehouden gegevens voor de vierde bron. Deze worden opgedeeld in verschillende tabbladen, o.a. verplichte gegevens ("Required fields"), optionele ("Optional") en andere ("Other fields"), enz. In dit geval zijn de namen van de auteurs ingevuld (zie Sectie 4.3.2). Er zijn in dit geval geen redacteuren ("Editors"), en dat veld is dan ook leeg gebleven. Het veld "Bibtexkey" onderaan is automatisch gegenereerd (zie Sectie 4.3.1) door te klikken op het *sleutelsymbol* in de knoppenbalk links.

- Kies in het Preferences-venster voor de categorie File en geef een directory op voor het bijhouden van PDFs van de gevonden bronnen onder “Main file directory”. Het is heel interessant om de gevonden artikels te downloaden en onder die directory bij te houden. Nog beter is om als naam van het bestand de BibTeX key te nemen (bv. Knuth1998.pdf). Je kan het bestand dan makkelijk openen vanuit Jabref.

Je kan de andere instellingen nakijken en eventueel naar wens aanpassen, maar de hierboven opgesomde zijn de belangrijkste.

4.3.2. Algemene bibliografische gegevens

De bedoeling van een bibliografie is de lezer toelaten je bronnen zelf op te zoeken en te beoordelen op betrouwbaarheid. Dat betekent dat je voldoende informatie moet opgeven zodat de bron terug te vinden is. Afhankelijk van het soort bron (artikel in een journal, boek, website, enz.) moet je andere informatie opgeven. Dit wordt verderop uitgediept. Drie elementen zijn sowieso *altijd* essentieel: de **auteur**, de **titel** van de bron en het **jaar** (of datum) van publicatie. Als één van deze drie ontbreekt, wordt het bijzonder moeilijk om de oorsprong en de kwaliteit van de bron te evalueren. Dit soort bronnen kan je bijhouden ter info, maar zijn meestal niet geschikt om op te nemen in een bibliografie. Als de auteur onbekend is, is het niet immers mogelijk om te beoordelen of die wel de autoriteit heeft om op een objectieve en diepgaande manier over het onderwerp te schrijven. Als het jaartal niet opgegeven is, is het erg moeilijk om na te gaan in hoeverre deze bron nog niet achterhaald is door recentere ontwikkelingen in het vakgebied.

Enkele tips bij het invullen van auteursnamen:

- Noteer de naam van auteurs in de vorm “Familienaam, Voorna(a)m(en)”. Dus “Van Vreckem, Bert” en niet “Bert Van Vreckem.” In principe wordt de tweede notatie ook aanvaard, maar dit werkt enkel voor typische angelsaksische namen met een tweede voornaam (bv. “Donald Ervin Knuth”). De eerste twee woorden worden beschouwd als voornamen, het laatste woord als de familienaam. “Van” wordt in dat geval dus verkeerdelijk beschouwd als tweede voornaam.
- Als de auteur een bedrijf of organisatie is, met een naam bestaande uit verschillende woorden, zet die dan tussen accolades: bv. “[The Linux Foundation]”. Zoniet probeert L^AT_EX dit als een persoonsnaam te interpreteren. “Foundation” wordt dan de familienaam, “The” en “Linux” de twee voornamen.
- Als je meerdere auteurs hebt, scheid elke naam dan met and, bv. “Bernard, Anita and Buysse, Jens and Van Vreckem, Bert”.
- Na invullen van de auteursna(a)m(en) klik je op de knop met het sleutel-icoon (zie Figuur 4.1) om een unieke sleutel te genereren voor deze bron.

- Naast een auteurveld is er ook een veld voor eventuele redacteur(en) (*editor*) voorzien. Minstens één van de twee moet ingevuld zijn, soms allebei. Dit gebeurt bijvoorbeeld in een boek dat samengesteld is uit hoofdstukken die telkens door andere auteurs geschreven zijn en waar je naar één bepaald hoofdstuk wil verwijzen. Verderop vind je daar een voorbeeld van.

Merk trouwens op dat in een bibliografie **enkel werken mogen opgenomen zijn waarnaar verwezen wordt vanuit de tekst**. \LaTeX doet dit standaard automatisch, dus als je bronnen in de lijst mist, dan betekent dit dat je die niet in de tekst gebruikt hebt.

Probeer telkens zoveel mogelijk informatie bij te houden over je bronnen, zodat het later makkelijker wordt die terug te vinden. Dit is een tijdrovend proces, en niet al deze informatie wordt ook in de bibliografie opgenomen. Het raadplegen van de databank wordt wel een stuk makkelijker. Heb hier voldoende aandacht voor en controleer ook goed het resultaat in de bibliografie zelf. Zijn de auteurs correct weergegeven? Heb je een jaartal? Is er voor een online bron (zie verder) een URL en datum van raadplegen opgegeven? Enz.

Enkele velden die zinvol zijn om altijd trachten in te vullen:

Abstract Samenvatting van het artikel. Dit is meestal de eerste paragraaf van een artikel en wordt altijd duidelijk aangegeven.

DOI of “Digital Object Identifier”. Dit is een unieke code voor artikels in wetenschappelijke publicaties die het opzoeken makkelijker maakt (op voorwaarde dat de DOI gegeven is).

File Naam van het bestand met het gedownloade artikel. Je kan vanuit JabRef het artikel openen in een PDF-viewer of desgevallend tekstverwerker.

Keywords Kernwoorden die het onderwerp weergeven, gescheiden door komma's.

Review Je eigen opmerkingen over deze bron. Waarom heb je deze bijgehouden? Wat is het interessantste dat je er uit geleerd hebt?

URL De URL waar je het artikel gevonden hebt. Deze URL wordt niet altijd in de bibliografie opgenomen, maar is altijd nuttig om bij te houden. Je kan vanuit JabRef de website openen in een webbrowser.

Urldate de datum waarop je deze bron het laatst hebt geraadpleegd.

4.3.3. Specifieke bibliografische gegevens

In deze sectie wordt voor de meest relevante soorten bronnen uitgelegd hoe deze correct bij te houden en in de bibliografie op te nemen. JabRef geeft zelf al enige aanwijzingen over welke informatie minstens nodig is: in het detailvenster (zie Afbeelding 4.1) moet je minstens het tabblad “Required fields” invullen. Voor elke



Figuur 4.2: Soorten bronnen in Jabref. Bij het toevoegen van een nieuwe bron in JabRef (Ctrl+N) moet je eerst het soort publicatie kiezen. Afhankelijk van het soort moet in de literatuurlijst immers andere informatie gegeven worden.

soort publicatie (zie Sectie 4.2.1) vind je verderop een overzicht van de in te vullen velden en wat die precies betekenen, en hoe de referentie in de literatuurlijst er uit zal zien.

Bij het toevoegen van een nieuwe bron aan een bibliografische databank moet je eerst het type selecteren (zie Afbeelding 4.2). Hieronder bespreken we enkel de belangrijkste die relevant zijn voor een bachelorproef.

Article

Dit soort bron wordt enkel gebruikt voor artikels die verschenen zijn in een wetenschappelijke journal. Artikels in (vak)tijdschriften of kranten vallen hier *niet* onder. Verplichte velden:

Author De naam van de auteur;

Title De titel van het artikel;

Year Jaar waarin het artikel verschenen is;

Volume De jaargang van het tijdschrift waarin het artikel verschenen is;

Number Het nummer (binnen de jaargang) waarin het artikel verschenen is (soms niet gegeven);

Pages Paginanummers

Voorbeeld:

```
@Article{SabiEtAl2016,
  author      = {Sabi, Humphrey M. and Uzoka, Faith-Michael E. and
                 Langmia, Kehbuma and Njeh, Felix M.},
  title       = {Conceptualizing a model for adoption of cloud
                 computing in education},
  journaltitle = {International Journal of Information Management},
  year        = {2016},
  volume      = {36},
  number      = {2},
  pages       = {183--191},
  doi         = {10.1016/j.ijinfomgt.2015.11.010},
  url         = {http://www.sciencedirect.com/[...]8401215001115},
  abstract    = {Cloud computing is a pervasive computing [...]},
  keywords    = {Cloud computing, Educational technologies, [...]},
  owner       = {bert},
  timestamp   = {2016-09-01},
}
```

In de bibliografie ziet dit er zo uit:

Sabi, H. M., Uzoka, F.-M. E., Langmia, K. & Njeh, F. M. (2016). Conceptualizing a model for adoption of cloud computing in education. *International Journal of Information Management*, 36(2), 183–191. doi:[10.1016/j.ijinfomgt.2015.11.010](https://doi.org/10.1016/j.ijinfomgt.2015.11.010).

InProceedings

Dit soort bron wordt gebruikt voor artikels die gepubliceerd zijn in het verslag (proceedings) van een *wetenschappelijke* conferentie. Verplichte velden:

Author Naam van de auteur(s),

Title Titel van het artikel,

Booktitle De naam van de conferentie,

Year Het jaar waarin de conferentie doorging.

Daarnaast kan je optioneel ook volgende velden invullen:

URL naar de website van de conferentie waar het artikel kan gevonden (eventueel rechtstreeks naar de pdf);

Urldate datum waarop je deze bron het laatst geraadpleegd hebt.

DOI op voorwaarde dat er één toegewezen is aan dit artikel.

Voorbeeld:

```
@InProceedings{VanVreckemEtAl2013,
  author    = {Van Vreckem, Bert and Borodin, Dmitriy and De Bruyn,
              Wim and Now\'{e}, Ann},
  title     = {A Reinforcement Learning Approach to Solving Hybrid
              Flexible Flowline Scheduling Problems},
  booktitle = {Multidisciplinary International Scheduling Conference
              (MISTA) 2013},
  year      = {2013},
  url       = {https://expertise.hogent.be/files/10711623/hffsp_la.pdf},
  urldate   = {2016-09-01},
  abstract  = {In this paper, we present a method based on Learning
              Automata to solve Hybrid Flexible Flowline Scheduling
              Problems [...]}.},
  owner     = {bert},
  timestamp = {2016-09-01},
}
```

In de bibliografie:

Van Vreckem, B., Borodin, D., De Bruyn, W. & Nowé, A. (2013). A Reinforcement Learning Approach to Solving Hybrid Flexible Flowline Scheduling Problems. In *Multidisciplinary International Scheduling Conference (MISTA) 2013*. Verkregen 1 september 2016, van https://expertise.hogent.be/files/10711623/hffsp_la.pdf.

InBook

Dit type gebruik je als je wil verwijzen naar een specifiek hoofdstuk in een boek. Minstens volgende velden moeten dan ingevuld zijn:

Author De auteur(s) van het hoofdstuk,

Editor De redacteur(en) van het boek (indien van toepassing),

Year Jaartal waarin het boek werd uitgegeven,

Title Titel van het *hoofdstuk*,

Pages Begin- en eindpagina van het hoofdstuk,

Booktitle Titel van het *boek*,

Publisher Naam van de uitgeverij.

Optioneel kan je ook volgende informatie aanvullen:

Subtitle of Booksubtitle ondertitel van het hoofdstuk of boek, resp.,

Edition Nummer van de uitgave,

Location Stad waar de uitgeverij gevestigd is,

ISBN Het ISBN-nummer van het boek (ter info, wordt nooit getoond in de bibliografie).

Bij een boek is het ongebruikelijk om een URL op te geven. Als je bijvoorbeeld ter info voor jezelf de URL van het boek op de website van de uitgever wil bijhouden, doe je dit best in een ander veld, bv. Comment of Review.

Voorbeeld

```
@InBook{Meyr2008,
  author      = {Meyr, Herbert},
  title       = {Forecast Methods},
  booktitle   = {Supply Chain Management and Advanced Planning},
  year        = {2008},
  editor      = {Stadtler, Hartmut and Kilger, Christoph},
  booksubtitle = {Concepts, Models, Software, and Case Studies},
  edition     = {4e editie},
  publisher   = {Springer},
  location    = {Heidelberg},
  isbn        = {978-3-540-24814-9},
  pages       = {461--472},
  comment     = {https://www.springer.com/us/book/9783540248149},
  owner       = {bert},
  timestamp   = {2016-09-02},
}
```

In de bibliografie:

Meyr, H. (2008). Forecast Methods. In H. Stadtler & C. Kilger (Red.), *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies* (4e editie, pp. 461–472). Heidelberg: Springer.

Electronic of Online

Onder dit type publicatie vallen vrijwel alle online bronnen die niet onder een andere categorie te plaatsen zijn: blogartikels, artikels in online vaktijdschriften of portaalsites, Youtube-video's van presentaties op vakconferenties, online documentatie, enz.

Merk op dat je de algemene website van organisaties, softwarepakketten, enz. *niet* in je literatuurlijst mag opnemen. Deze kan je wel in een voetnoot zetten.

Deze velden moet je verplicht invullen:

Author Auteur(s) van de bron, spreker (in het geval van een video van een lezing op een conferentie), ...

Title Titel van de bron, lezing, ...

Year Jaartal van publicatie (of eventueel **Date**, de dag van publicatie, als die bekend is),

URL naar de website waar de bron kan teruggevonden worden,

Urldate Datum van laatste raadplegen,

Bij dit soort bronnen worden veel fouten gemaakt bij het refereren. Het is essentieel dat de URL wordt meegegeven en ook de datum van raadplegen. Het web is voortdurend in beweging, en het is mogelijk dat de inhoud van een webpagina in de loop van de tijd verandert (bv. fouten die verbeterd worden) of zelfs dat een website herstructureert en de URL dus op een gegeven manier niet meer geldig is. Door de datum van raadplegen op te geven, bied je de lezer nog de kans om terug te vinden hoe die website er op dat moment in de tijd uitzag, via bijvoorbeeld de Wayback Machine van het Internet Archive⁵.

Voorbeeld van een blogartikel:

```
@Electronic{LewisFowler2014,
  author    = {Lewis, James and Fowler, Martin},
  title     = {Microservices: a definition of this new architectural
              term},
  date      = {2014-03-25},
  url       = {http://martinfowler.com/articles/microservices.html},
  urldate   = {2016-09-01},
  abstract  = {The term "Microservice Architecture" has [...]},
  keywords  = {application architecture, web services, microservices},
  owner     = {bert},
  timestamp = {2016-09-01},
}
```

In de bibliografie wordt dit:

Lewis, J. & Fowler, M. (2014, maart 25). Microservices: a definition of this new architectural term. Verkregen 1 september 2016, van <http://martinfowler.com/articles/microservices.html>.

⁵<https://archive.org/web/>

Een ander voorbeeld, deze keer van een presentatie op een vakconferentie die op Youtube is gepubliceerd. Omdat er niet meteen een apart veld voorzien is voor het vermelden van de naam van de conferentie, is die hier in het titelveld verwerkt.

```
@Online{Hykes2013,  
  author      = {Solomon Hykes},  
  title       = {The future of Linux Containers (PyCon 2013)},  
  date        = {2013-03-21},  
  url         = {https://www.youtube.com/watch?v=wW9CAH9nSLs},  
  urldate     = {2016-09-01},  
  abstract    = {At PyCon Solomon Hykes shows docker to the  
                public for the first time.},  
  owner       = {bert},  
  timestamp   = {2016-09-01},  
}
```

In de bibliografie:

Hykes, S. (2013, maart 21). The future of Linux Containers (PyCon 2013). Verkregen 1 september 2016, van <https://www.youtube.com/watch?v=wW9CAH9nSLs>.

4.4. Op zoek naar relevante informatie

Via de HOGENT bib krijg je toegang tot een grote hoeveelheid wetenschappelijke en vakliteratuur die niet publiek beschikbaar zijn. Dit gaat veel verder dan de boeken die in de bib beschikbaar zijn. Daarnaast zijn immers nog elektronische bronnen (ebooks, tijdschriften, enz.) te raadplegen. Ook (goede) bachelorproeven van vorige jaren kan je op deze manier vinden. De catalogus is te raadplegen via <https://bib.hogent.be/zoeken/>.

Databanken die niet publiek beschikbaar zijn kan je raadplegen als je op de campus bent, of thuis via Apollox: <https://apollox.hogent.be/>. Voor ons vakgebied zijn in het bijzonder volgende databanken interessant:

- Elsevier ScienceDirect
- Springer Online Journals
- Web of Science
- Google Scholar

Google Scholar is een zoekmachine voor wetenschappelijke literatuur die zowel in publiek toegankelijke ("open access") als betalende publicaties zoekt. Als je Scholar gebruikt vanop de campus of via Apollox, krijg je automatisch toegang tot de publicaties waar de HOGENT bib een abonnement op heeft.

Andere, publiek toegankelijke startpunten voor het zoeken:

- Arxiv.org is een database van Open Acces artikels in een hele reeks onderzoeksdomeinen, o.a. de Computing Research Repository⁶.
- Ook Wikipedia is een goed startpunt voor je onderzoek, maar vergeet niet dat Wikipedia-artikels op zich niet kunnen als referenties. Lees dus de oorspronkelijke bronnen van het artikel na en gebruik die als ze relevant zijn voor je bachelorproef.
- Er bestaan verschillende portaalsites voor actuele ict-gerelateerde onderwerpen waar technische artikels, presentaties, interviews, enz. op verschijnen, bv. dzone.com⁷, infoq.com⁸, enz.
- Ga op zoek naar voor je vakgebied relevante conferenties, workshops, symposia, enz. Je hoeft je daarbij niet te beperken tot conferenties in eigen land! Voorbeelden zijn Devoxx⁹ (Java), Google IO¹⁰ (Android), WWDC¹¹ (iOS), Configuration Management Camp¹² (Linux Systeembeheer), enz. Lanyrd¹³ is een website waar je vele conferenties kan terugvinden over allerlei onderwerpen en verspreid over heel de wereld.

Meer en meer worden de lezingen van conferenties gefilmd en gepubliceerd op Youtube of Vimeo¹⁴. Je kan ook de sprekers opzoeken en nagaan of ze hun slides gepubliceerd hebben op Slideshare¹⁵ of Speakerdeck¹⁶.

- Zijn er lokaal verenigingen die geïnteresseerd zijn in je vakgebied? Bijvoorbeeld OWASP Belgium¹⁷ (beveiliging van mobiele en webapplicaties). Je kan naar zulke groepen zoeken via Meetup¹⁸ of ook via LinkedIn¹⁹ (zoek specifiek naar groepen zoals het Belgian IT Infrastructure Network²⁰). Kijk eens na of er binnenkort evenementen in de buurt zijn en ga er naartoe.
- Wie zijn de belangrijkste namen in de “community”? Keynote-speakers op conferenties, auteurs van de belangrijkste boeken over het onderwerp, enz. Volg deze personen op Twitter, zoek uit of ze een blog hebben, actief zijn op LinkedIn, enz. Lees al wat je kan vinden dat ze geschreven hebben de laatste jaren.

⁶<http://arxiv.org/corr/home>

⁷<https://dzone.com/>

⁸<https://www.infoq.com/>

⁹<https://devox.be/>

¹⁰<https://events.google.com/io2016/>

¹¹<https://developer.apple.com/wwdc/>

¹²<http://cfmgmtcamp.eu/>

¹³<http://lanyrd.com/topics/>

¹⁴Voor Devoxx bijvoorbeeld op <https://www.youtube.com/user/parleysdotcom>

¹⁵<https://slideshare.net/>

¹⁶<https://speakerdeck.com/>

¹⁷<https://www.owasp.org/index.php/Belgium>

¹⁸<https://meetup.com/>

¹⁹<https://www.linkedin.com/>

²⁰<https://www.linkedin.com/groups/2092569>

- Zoek uit of er nieuwsbrieven bestaan over je onderwerp, die periodiek updates uit sturen over de actualiteit binnen dat vakgebied. Voorbeelden zijn Cron.Weekly²¹ (Linux systeembeheer) of DevOps Weekly²².

Het bijhouden van al je bronnen is tijdrovend, maar essentieel voor een goed onderbouwde bachelorproef! Besteed hier dus de nodige aandacht aan en controleer of de informatie over je bronnen correct is ingevuld in JabRef.

Alle relevante informatie die je in de loop van je onderzoek hebt verzameld en gelezen wordt dan gesynthetiseerd in een doorlopende tekst waarin je in je eigen woorden de situatie in het onderzoeksdomein schetst, met op gepaste plaatsen verwijzingen naar de literatuur. Hier gaan we dieper op in in Hoofdstuk 7.

4.5. Samenvatting

- Het doel van de literatuurstudie is om jezelf en de lezer van je bachelorproef voldoende context te geven om het onderwerp ten gronde te begrijpen.
- Er zijn drie soorten bronnen: primaire (resultaten van eigen onderzoek), secundaire (publicaties in wetenschappelijke of vakliteratuur) en tertiaire (encyclopedieën en zoekindexen).
- In een bibliografie horen enkel *secundaire* bronnen thuis.
- Zet je bibliografische databank op (bv. met Jabref) voordat je op zoek gaat naar informatie over je onderwerp en hou van wat je vindt nauwgezet zoveel mogelijk informatie bij.
- Zorg dat je altijd minstens de auteur, titel en jaartal hebt en daarnaast minstens alle andere verplichte informatie voor dat type publicatie correct noteert.
- Naast het raadplegen van publiek toegankelijke bronnen is het ook interessant om via de HOGENT bib informatie over je onderwerp op te zoeken.
- Gebruik de APA stijl.

²¹<https://www.cronweekly.com/>

²²<http://www.devopsweekly.com/>

5

Onderzoeksmethoden

In dit onderwerp vind je aanbevelingen in verband met een aantal vaak gebruikte onderzoeksmethoden. Meer bepaald bespreken we de aanpak van een vergelijkende studie, hoe je correct experimenten opzet en hoe je op een correcte manier rapporteert over bekomen resultaten (in het bijzonder cijfermateriaal).

5.1. De vergelijkende studie

Een type onderwerp dat vaak gekozen wordt voor een bachelorproef is een vergelijkende studie. Je bent op zoek naar een oplossing voor een probleem in de vorm van een software- of hardware-product, platform, dienst, enz. De bedoeling van de studie is om alle mogelijke alternatieven naast elkaar te zetten en een keuze te maken over de meest geschikte.

De ervaring leert dat een vergelijkende studie pas echt goed is als er een concreet doel is, een reële situatie waar de geselecteerde oplossing ook werkelijk zal toegepast worden. Het gevaar bestaat dat de studie zich beperkt tot het achter elkaar opsommen van enkele arbitrair gekozen mogelijkheden. Er volgt een paragraafje uitleg, soms gewoon van Wikipedia gehaald, met een opsomming van wat voor- en nadelen, maar niet gestructureerd en zonder rode draad. Een bepaald aspect als “gebruiksvriendelijkheid” wordt dan bijvoorbeeld in één product besproken, maar niet voor de andere, enz. De eigen inbreng is dan miniem: een dagje zoeken op Wikipedia, samenvatten of verder uitschrijven, klaar. Dit is op zich dus onvoldoende.

Maar hoe pak je het dan wel aan?

Laat ons veronderstellen dat je na je afstuderen aan de slag wil als webontwikkelaar, en je bent op zoek naar een geschikt PHP-framework om je websites mee te bouwen.

5.1.1. Requirements-analyse

Om een goede keuze te maken begin je met het verzamelen van *requirements*, zowel *functionele* als *niet-functionele*, bijvoorbeeld:

- Functionele requirements
 - Ondersteuning voor HTML5/CSS3
 - Responsive design
 - Er moet een authenticatiemodule in zitten dat OpenID, Facebook- en Google-authenticatie ondersteunt
 - ...
- Niet-functionele requirements
 - Moet bestand zijn tegen de top-10 beveiligingsproblemen van OWASP¹
 - Wachtwoorden worden opgeslagen volgens de state-of-the art (*salted* en *hashed*)
 - Moet open source zijn
 - Moet gratis zijn
 - ...

Als je het onderzoek doet voor een “klant” (i.e. je co-promotor), dan betrek je uiteraard alle belanghebbenden bij dit proces! Je lijst de requirements op en verdeelt ze onder naar belangrijkheid, bijvoorbeeld via de MoSCoW-techniek (Nordenstam, 2014). Je verdeelt de requirements dan in categorieën zoals “must-have”, “should-have” en “nice-to-have”.

5.1.2. Long list

Dan zoek je zoveel mogelijk alternatieven die in aanmerking komen om gebruikt te worden, m.a.w. al diegenen die je kan vinden. Je noemt ze in deze *long list* (die soms kan bestaan uit tientallen alternatieven) bij naam, met eventueel vermelding van een website en een beschrijving in één zin. Elk alternatief toets je af aan de requirements, voor zover dit al mogelijk is aan de hand van informatie die je op de website vindt of via andere bronnen. Zaken die je niet kan verifiëren laat je gewoon open om later na te kijken of misschien zelfs te negeren (als het bv. gaat om een onbelangrijke feature, of als verschillende andere must-haves niet voldaan zijn). Je sorteert de long list dan volgens het aantal voldane requirements, en maakt hier een overzichtelijke tabel van. Hopelijk heb je een aantal alternatieven overgehouden die voldoen aan alle must-haves en zoveel mogelijk should-haves en nice-to-haves.

¹https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

5.1.3. Short list en proof-of-concept

De meest veelbelovende alternatieven weerhoud je voor de volgende fase. De alternatieven in deze *short list* ga je in meer detail bespreken en verder tegenover elkaar afwegen. Zet eventueel een *proof-of-concept* op (zie Sectie 5.2), waarin je één of enkele van de meest veelbelovende alternatieven uitprobeert aan de hand van eenzelfde vastgelegd *scenario* waarin je de requirements die je nog niet hebt kunnen verifiëren aan bod laat komen.

5.1.4. Conclusie

Tenslotte geef je je *aanbeveling*, het alternatief dat het beste aansluit bij de requirements, en wat eventueel nog moet gedaan worden om het nog beter geschikt te maken.

5.2. Proof-of-concept of testopstelling

Met een proof-of-concept of testopstelling wil je aantonen dat een voorgestelde of veelbelovende oplossing voor de onderzoeksvraag ook realiseerbaar is in de praktijk. Vaak is het nodig verschillende varianten naast elkaar op te zetten, zodat je nog onderling kan vergelijken.

Een goede testopstelling voldoet aan drie eigenschappen (Lieberman, 2015):

Reproduceerbaarheid in staat zijn om de testopstelling opnieuw op te zetten (eventueel door een onafhankelijke partij) en de analyse opnieuw uit te voeren zodat je kan verifiëren dat je gelijkaardige resultaten krijgt.

Repliceerbaarheid lijkt op het voorgaande, maar is sterker uitgedrukt: bij een repliceerbare testopstelling is het mogelijk dat een onafhankelijke partij deze exact opnieuw kan opzetten en dezelfde resultaten zal bekomen.

Herbruikbaarheid varianten kunnen opzetten van de testomgeving zodat deze onderling kunnen vergeleken worden.

De beste manier om dit te bereiken is door het proces te automatiseren.

5.2.1. Reproduceerbaarheid

Een minimumverwachting van de tekst van een bachelorproef is dat de lezer in staat moet zijn om aan de hand van de beschrijving de testopstelling te reproduceren en je conclusies moet kunnen verifiëren. Dat betekent ook dat je beschrijving voldoende gedetailleerd moet zijn. Geef aan het begin van deze sectie in je tekst een duidelijke afbeelding met een schema van de opgezette testomgeving. Beschrijf vervolgens in detail:

- Welke hardware werd er gebruikt? Computers, netwerkkapparatuur, andere

ict-infrastructuur. Welke specificaties hebben deze apparaten (CPU, geheugen, type en grootte harde schijven, snelheid netwerkinterfaces, enz.)?

- Welke softwarepakketten zijn er geïnstalleerd die relevant zijn voor de testopstelling? Welke specifieke edities en/of versienummers?
- Hoe is de installatieprocedure precies verlopen? Welke instellingen zijn er aangepast, en hoe zijn alle componenten geconfigureerd?

5.2.2. Repliceerbaarheid

Het is zowel voor jezelf als voor de lezer van je bachelorproef belangrijk om het opzetten van de testomgeving zo veel mogelijk te automatiseren, bijvoorbeeld door het schrijven van een installatiescript. Op deze manier bereik je de *repliceerbaarheid* van je testopstelling.

Voor veel situaties kan je een proof-of-concept uitwerken met virtuele machines. Op die manier creëer je een afgeschermd omgeving zonder dat je extra software moet installeren op je fysieke systeem. Vagrant² is een interessante tool om reproduceerbare virtuele testomgevingen volledig geautomatiseerd op te zetten. Het is zelf geen virtualisatiesysteem, maar het spreekt VirtualBox, VMWare of Hyper-V aan vanop de command-line. Je beschrijft in een configuratiebestand (Vagrantfile) welke VMs je nodig hebt, met hoeveel processorkernen/geheugen/schijfruimte, met welk besturingssysteem, enz. Via een script of configuration management systeem (zoals Ansible) kan je de precieze configuratie van de VMs beschrijven (installatie software, gebruikers, netwerkservices, configuratiebestanden, enz.). De code/configuratie om een Vagrant-omgeving op te zetten is erg compact, volledig tekstgebaseerd en kan dus worden bijgehouden in een versiebeheersysteem.

Ook Docker³ is een interessante tool hiervoor. Docker is een vorm van zgn. containervirtualisatie. Virtuele machines (containers) hebben geen eigen OS, maar bevatten in principe enkel een applicatie. Het OS van het fysieke systeem wordt hergebruikt door de containers. Dat maakt dat containers een stuk efficiënter zijn qua schijf- en geheugengebruik voor het fysieke systeem. Er zijn echter een aantal beperkingen waardoor Docker niet altijd bruikbaar is als platform voor een testopstelling.

Als je merkt dat je een fout gemaakt hebt in de configuratie van je testopstelling, dan kan je het installatiescript aanpassen en de gehele testomgeving opnieuw opzetten. Als je dit manueel zou moeten doen, ben je wellicht minstens een dag kwijt. Bovendien heb je de kans dat je een kleinigheid over het hoofd ziet en dus niet in staat bent de testomgeving opnieuw exact te repliceren. Het initialiseren van een Linux-VM met Vagrant of Docker duurt typisch slechts enkele minuten. Windows-systemen nemen een stuk meer tijd in beslag. Het OS is op zich al een stuk groter⁴

²<https://www.vagrantup.com/>

³<https://docker.com/>

⁴Ca. 4GB voor Windows Server Core tegenover 256 à 512MB voor een minimale installatie van een

Een uitgebreide handleiding van Vagrant of Docker valt buiten het bereik van deze gids. Een voorbeeld ter illustratie is te vinden op <https://github.com/bertvw/lampstack>. Hier wordt een Apache webserver opgezet met een MariaDB databank, PHPMyAdmin, en Wordpress.

5.2.3. Herbruikbaarheid

Door het gehele proces te automatiseren creëer je een bijkomend voordeel: het is een stuk makkelijker om gelijkaardige varianten van een testopstelling te creëren. Stel dat je de performantie-impact van een cache op een databasesysteem wil meten. De installatie van beide varianten (met/zonder cache) zijn zeer gelijkaardig. In een Vagrant-omgeving kan je 2 VMs creëren, met hetzelfde installatiescript voor de basisopstelling, en een apart script om de verschillende configuratie-varianten te realiseren.

5.3. Experimenten uitvoeren en verzamelen van kwantitatieve data

Wanneer je een experiment opzet waarbij je kwantitatieve data gaat verzamelen, bijvoorbeeld voor een performantievergelijking, gelden dezelfde richtlijnen als bij een proof-of-concept: het is belangrijk dat die reproduceerbaar, repliceerbaar en herbruikbaar is. Wellicht start je vanaf een vooraf gedefinieerde testopstelling, en ga je herhaaldelijk eenzelfde handeling uitvoeren en metingen doen.

Ook hier is het belangrijk het uitvoeren van het experiment en het opslaan van testresultaten te automatiseren, bijvoorbeeld aan de hand van een shell-script. Als je een experiment manueel moet uitvoeren, wordt dit al snel een erg tijdrovende bezigheid. Stel je voor dat je telkens je het experiment wil uitvoeren op een knop moet drukken, ongeveer vijf à tien minuten moet wachten en dan de uitvoer naar Excel moet kopiëren en plakken en dan het rekenblad nog eens moet bewerken zodat je enkel de cijfers overhoudt om statistisch te verwerken. Deze werkwijze is onhoudbaar. Je moet immers voortdurend aanwezig en aandachtig blijven tijdens het uitvoeren. Bij elke manuele handeling is er bovendien een kans op het maken van fouten.

Een script dat het experiment in verschillende varianten kan uitvoeren zonder interactie door de gebruiker, en ook meteen de resultaten kan opslaan in een geschikt bestandsformaat bespaart je enorm veel tijd. Je kan het experiment 's nachts laten draaien en op die manier kan je het ook voldoende herhalen om statistisch bruikbare resultaten te bekomen. Als een experiment mislukt, volstaat het om het script te verbeteren en het opnieuw uit te voeren. Het is ook makkelijker om verschillende varianten te schrijven (zoals het eerdere voorbeeld van database-queries

Linux distributie. Installatie van applicaties hebben nog een bijkomende impact op de schijfgrootte van de VM.

met of zonder cache).

6

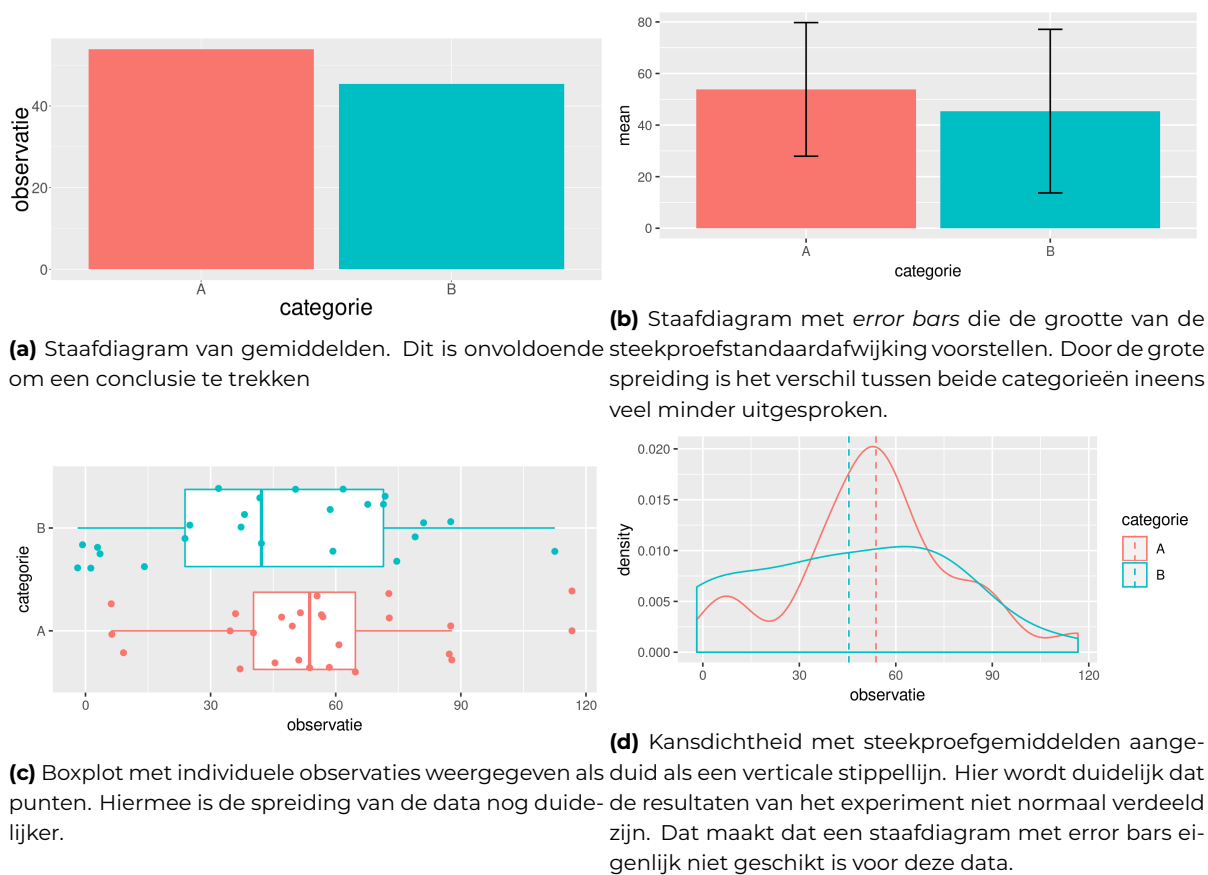
Resultaten rapporteren

In het vorige hoofdstuk hebben we onder andere besproken hoe je een experiment opzet om kwantitatieve data te verzamelen. In dit hoofdstuk gaan we in op hoe je deze resultaten correct rapporteert.

6.1. Visualisatie van cijfermateriaal

In de cursus Onderzoekstechnieken wordt het goed visualiseren van cijfermateriaal in detail besproken en geoefend. Toch zien we dat in de bachelorproef nog heel veel fouten gemaakt worden met foute conclusies als gevolg. Daarom komen we hier nog even terug op dit onderwerp.

Stel, je voert een performantievergelijking uit tussen twee systemen, A en B. Je hebt op elk systeem 50 keer een experiment uitgevoerd en de resultaten geregistreerd. Laat ons veronderstellen dat het resultaat van de meting een



Figuur 6.1: Verschillende manieren om dezelfde data te visualiseren.

7

De bachelorproef schrijven

Op een bepaald moment heb je een heleboel informatie verzameld en geanalyseerd, en wordt het tijd om het finale document op te maken. In dit hoofdstuk vind je enkele tips en richtlijnen om tot een goed resultaat te komen.

In deze gids beperken we ons tot enkele belangrijke punten en specifieke aanbevelingen voor de opmaak van de tekst in \LaTeX . Een meer omvattend overzicht kan je o.a. vinden in het boek van Pollefliet (2011).

7.1. Algemene richtlijnen

Wanneer je een tekst schrijft wil je uiteraard je best doen om deze zo aangenaam mogelijk te maken om te lezen. Er zijn echter een aantal dingen waar je moet op letten. In je bachelorproef toon je aan dat je een professionele instelling hebt en een gezonde dosis maturiteit op zak hebt. Dat moet ook tot uiting komen in je schrijfstijl. De tekst moet professioneel en objectief overkomen. Vermijd in het bijzonder volgende zaken:

- Schrijven vanuit je eigen standpunt is uit den boze. Gebruik dus nooit de **ik-vorm**. Die geeft immers de indruk dat je je *eigen mening* geeft, en als junior in je vakgebied heb je daarvoor onvoldoende autoriteit. De beweringen die je in je bachelorproef doet moeten objectieve aantoonbare feiten zijn, die ofwel gesteund worden door een verwijzing naar gezaghebbende vakliteratuur, hetzij volgen uit je eigen onderzoeksresultaten.
- Gebruik geen **spreektaal**. Blijf formeel en zakelijk.
- Gebruik geen **vage termen** om een hoeveelheid aan te duiden, bv. lang, groot, snel, populair, ...Quantificeer al deze uitspraken met cijfers en meeteenheden (uiteraard ondersteund door literatuurverwijzingen of resultaten van eigen onderzoek).

- Gebruik geen **toekomstige tijd**. Op het moment dat je afgewerkte bachelorproef gelezen wordt, is het een verslag van in het verleden uitgevoerd onderzoek. Dus niet “Eerst zal worden gekeken naar...” maar “Eerst werd onderzocht...”

Hou rekening met je *doelpubliek*. In principe kan je er van uitgaan dat de lezer een ict-achtergrond heeft (tenzij je je onderwerp uitwerkt in samenwerking met of in opdracht van iemand uit een andere discipline). Je kan dus meestal een zekere basiskennis veronderstellen, maar termen en afkortingen die specifiek zijn voor je eigen onderzoeksdomein moet je zeker uitleggen.

Doe een *grondige controle* op spellings- en grammaticale fouten. Deze *moeten* er uit zijn bij indienen. Voor een informaticus telt elke punt en komma. Een tekst vol fouten geeft een heel slechte indruk over je capaciteiten.

Begin elk hoofdstuk (en sectie) met een inleidende paragraaf die aangeeft wat de inhoud ervan is wat de link is met het vorige hoofdstuk. Iemand die meteen dat hoofdstuk begint te lezen krijgt op die manier wat context.

7.2. Structuur & taal

Een voorbeeld van een goede structuur kan je terugvinden in de template die beschikbaar is. Enkele losse raadgevingen (onder andere verbatim uit [Toolkit VTV](#)):

1. Probeer een aantrekkelijke titel te bedenken. Een goede titel is het halve werk.
2. Gebruik enkel afkortingen wanneer die relevant zijn en vermijd jargon. We verwijzen hiervoor naar de cursus onderzoekstechnieken. Enkele tips:
 - (a) Vervang waar mogelijk jargon door een algemene term. Vooral in teksten of rapportonderdelen die voor een bredere doelgroep bestemd zijn (bijv. de samenvatting, inleiding en conclusies) is dit van belang. Het wordt daardoor leesbaarder.
 - (b) Verhelder vaktermen door middel van voorbeelden.
 - (c) Als een vakterm niet te vermijden is, verklaar deze dan bij het eerste gebruik. Doe dit in elk langer tekstonderdeel (bijvoorbeeld: per hoofdstuk).
 - (d) Neem in het rapport een aparte lijst op met vaak voorkomende definities en hun beschrijving.
3. Vermijd subjectieve formuleringen: Vermijd daarom woorden die een persoonlijke betrokkenheid of mening suggereren: helaas, jammer, gelukkig. Wees ook voorzichtig met woorden die kunnen worden uitgelegd als een eigen interpretatie van de feiten: een sterke daling, een opvallende toename.
4. Wees spaarzaam met we/wij: veelvuldig gebruik van we/wij maakt de toon persoonlijk en daarmee minder objectief. Het risico bestaat dan dat de lezer

de boodschap afdoet als een visie van de schrijver(s) en niet als een objectief feit. Vermijd zeker ook het schoolmeesterachtige we-perspectief dat zowel de schrijver als de lezer insluit, of zelfs alle Nederlanders, of alle mensen.

5. Gebruik actieve werkwoordsvormen. Je maakt de tekst levendiger en beter leesbaar door zo veel mogelijk in de actieve vorm te schrijven, dus zo min mogelijk lijdende vormen te gebruiken. De lijdende vorm laat namelijk in het midden 'wie wat doet', wat de tekst onduidelijk, onpersoonlijk of saai maakt. Een lijdende vorm is te herkennen aan een hulpwerkwoord of door een bijzin die begint met 'door'.
6. In de regel verwijst het betrekkelijk voornaamwoord *die* naar de-woorden en *dat* naar het-woorden.
7. Kijk zeker nog eens de -dt regels na

7.3. Afbeeldingen

Het invoegen van afbeeldingen in een \LaTeX -document is één van de struikelblokken voor beginnende gebruikers van het tekstzetsysteem. In een klassieke WYSIWYG tekstverwerker ben je gewend om zelf te bepalen waar een afbeelding op het papier terecht zal komen. Meestal doe je dat dan meteen onder het deel van de tekst waar naar de afbeelding verwezen wordt. In een groter document wordt dit problematisch. Het is immers mogelijk dat door toevoegingen van tekst vóór de afbeelding, die de ondermarge gaat overschrijden. De tekstverwerker zal dan wellicht de afbeelding naar de volgende bladzijde verplaatsen en je krijgt onderaan extra witruimte. Dit ziet er niet goed uit, en je verliest dan kostbare tijd met het goed positioneren van de afbeeldingen ten opzichte van de tekst.

\LaTeX kan zelf bepalen waar een afbeelding het best gepositioneerd wordt. Het is in eerste instantie vervelend om deze controle te moeten afstaan, maar het document zal er wel een stuk beter door uitzien. Soms kiest \LaTeX er voor om de afbeelding een bladzijde verder of eerder te plaatsen dan de tekst die er op betrekking heeft. Dat betekent dus dat de context voor de afbeelding niet meer bij de afbeelding staat. Dit kan je echter oplossen door bij elke afbeelding een bijschrift (*caption*) te plaatsen dat volledig beschrijft wat er op de afbeelding te zien is. Beperk je niet tot enkele woorden. Gebruik volledige zinnen zodat de lezer de afbeelding kan begrijpen zonder naar de bijhorende tekst te moeten gaan zoeken. In de tekst zelf verwijst je dan uiteraard naar de afbeelding (met `\ref{fig:label}`).

Onder de auteurswetgeving is het toegelaten om binnen de context van onderwijs afbeeldingen uit andere publicaties over te nemen zonder voorafgaande toestemming van de auteur. Het is dan uiteraard essentieel dat je in het bijschrift een literatuurverwijzing toevoegt. Zoniet wordt dit beschouwd als plagiaat.

Logo's van producten of bedrijven toevoegen is *geen* goed idee. Dit valt immers

niet enkel onder de auteurswetgeving, maar onder de merkenwet. Een logo weerspiegelt de identiteit van een bedrijf en het gebruik van het logo wordt dan ook sterk gecontroleerd (grootte, correct kleurgebruik, enz.). Derden mogen enkel na expliciete toestemming en onder specifieke voorwaarden het logo gebruiken. In een thesis voegt een logo trouwens geen enkele inhoudelijke meerwaarde toe, wat op zich al een reden is om er geen in te voegen.

Wanneer je een afbeelding overneemt van een website, zorg er dan zeker voor dat de resolutie voldoende hoog is. Afbeeldingen die er op een beeldscherm goed uitzien, kunnen op papier duidelijk de individuele pixels zichtbaar worden, wat niet goed overkomt. Een beeldscherm heeft typisch een veel lagere resolutie (96 DPI of *dots per inch*, beeldpunten per duim) dan een printer (300 of 600 DPI). Voor een goed resultaat moet een afbeelding minstens evenveel pixels bevatten als de gewenste hoogte of breedte van de afbeelding op papier, vermenigvuldigd met de resolutie. Als je dus bijvoorbeeld een afbeelding op 5×5 cm (of ongeveer 2 inch) wil afdrukken op een 300 DPI printer, moet je afbeelding minstens 600×600 pixels groot zijn.

Een alternatief is werken met *vector graphics*, m.a.w. figuren die in het document getekend worden aan de hand van wiskundig gedefinieerde vormen. Er bestaat een zeer uitgebreide package voor \LaTeX hiervoor: *TikZ*. Dit heeft wel een steile leercurve, maar het resultaat is wel van hoge kwaliteit. Een inleiding op *TikZ* valt buiten de scope van deze gids, maar er zijn veel tutorials en voorbeelden te vinden op het Internet¹.

¹Bijvoorbeeld <http://www.texample.net/tikz/>

Bibliografie

- Beck, K., Beedle, M., van Bennekun, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Manifesto for Agile Software Development. Verkregen 13 juni 2019, van <https://agilemanifesto.org/>
- Gratz, A. E. (2015). Evaluating Sources - Use the C.R.A.P. test! Verkregen 1 september 2016, van <https://libraries.mercer.edu/research-tools-help/citation-tools-help/evaluating-sources>
- Lehman, P., Kime, P., Boruvka, A. & Wright, J. (2016, mei 14). *The biblatex package*. Versie 3.4. Verkregen 1 september 2016, van <http://tug.ctan.org/macros/latex/exptl/biblatex/doc/biblatex.pdf>
- Lieberman, M. (2015, oktober 31). Replicability vs. reproducibility — or is it the other way around? Verkregen 14 juni 2019, van <https://languagelog.ldc.upenn.edu/nll/?p=21956>
- Nordenstam, G. (2014). Prioritizing LMS Requirements Using the MoSCoW Method. Verkregen 18 februari 2016, van <http://www.learninglensinc.com/learninglens-offerings/prioritizing-lms-requirements-using-the-moscow-method/>
- Oetiker, T., Partl, H., Hyna, I. & Schlegl, E. (2015). The Not So Short Introduction To \LaTeX 2_ε. Verkregen 17 februari 2016, van <https://tobi.oetiker.ch/lshort/lshort.pdf>
- Pollefliet, L. (2011). *Schrijven van verslag tot eindwerk: do's en don'ts*. Gent: Academia Press.
- van Uchelen, I. S. & Jungjohann, S. M. (2003). *Performancemanagement en resultaatgerichte beloning: een winnend duo voor sturen op resultaten* [Praktijkboek belonen: Themakaternen]. Kluwer.