

Het methodologische belang van het ontwerp van het code-managementsysteem bij kwalitatieve data-analysesoftware (KDAS)

Reinoud Bosch en Agnes Mühlmeier-Mentzel

Recente inleidingen in het gebruik van kwalitatieve data-analysesoftware (KDAS, of QDAS in Engelstalige literatuur) bevatten een opvallende paradox. Aan de ene kant wordt gesteld dat de drie toonaangevende KDAS-pakketten (MAXQDA, NVivo, en Atlas.ti) het kwalitatieve onderzoekers mogelijk maken precies dat te doen wat zij willen. Aan de andere kant wordt aan de verschillen die tussen de drie programma's bestaan een groot belang toegekend. Ook wordt met regelmaat gewezen op het belang van de theoretische en methodologische uitgangspositie van de onderzoeker bij de keuze van de software en de elementen daarvan (Lewins & Silver, 2007; Di Gregorio & Davidson, 2008).

In dit artikel zullen wij betogen dat deze paradox voor een groot deel het gevolg is van de verschillende ontwerpen van het code-managementsysteem bij de KDAS-pakketten. Hoewel de pakketten een nagenoeg vergelijkbare functionaliteit bieden, worden door de verschillende ontwerpen van het code-managementsysteem verschillende manieren van werken gestimuleerd. Het hiërarchische code-managementontwerp dat dominant is bij MAXQDA en NVivo (tot en met NVivo8)¹ stimuleert een hiërarchisch georganiseerde en gefaseerde analyse, terwijl de netwerkstructuur van Atlas.ti bevorderlijk is voor een pluriform georganiseerde en iteratieve (herhalende) analyse.

Een onderkenning van dit verschil maakt het voor kwalitatieve onderzoekers eenvoudiger te kiezen tussen de verschillende pakketten. Ook zijn de pakketten nog in een dusdanige fase van ontwikkeling dat het zinvol is de discussie tussen kwalitatieve methodologen en de ontwikkelaars van de software op dit punt aan te scherpen. Zo heeft NVivo op het moment van schrijven van dit artikel een ingrijpende verandering van het code-managementsysteem aangekondigd, waarbij tot op zekere hoogte afstand wordt genomen van het hiërarchische code-managementontwerp.² Wat zijn de methodologische gevolgen hiervan?

1 Op het moment van schrijven (juli 2010) waren de meest recente versies van de KDAS-pakketten MAXQDA10, NVivo8 en Atlas.ti6.

2 Deze ingrijpende verandering zou inmiddels geïmplementeerd zijn in NVivo9, gelanceerd in oktober 2010.

Het belang van het ontwerp van KDAS-pakketten

De manier waarop een softwarepakket is ontworpen, kan grote gevolgen hebben voor de gebruikers ervan. Ontwikkelaars van software zijn zich hier over het algemeen van bewust. Binnen de informatica bestaan namelijk specifieke specialisaties, zoals *software engineering* en mens-computerinteractie, waarbinnen details van ontwerpen en de gevolgen daarvan voor gebruikers zorgvuldig tegen het licht worden gehouden (Sharp, Rogers & Preece, 2007; Van Vliet, 2008). Ook ontwikkelaars van KDAS-pakketten zijn zich vaak terdege bewust van het belang van hun ontwerpen voor de gebruikers van de pakketten (Muhr & Friese, 2003-4; Kuckartz, 2007; QSR, 2008).

Gebruikers van KDAS-pakketten hebben in het verleden van tijd tot tijd over de voor- en nadelen van specifieke KDAS-ontwerpen gepubliceerd. Aan de ene kant wordt hierbij gewezen op de manier waarop een ontwerp het werk van een onderzoeker kan ondersteunen, aan de andere kant wordt gewaarschuwd voor het risico dat het softwareontwerp de methode van de analyse bepaalt (Conrad & Reinhartz, 1984; Hesse-Biber, 1995; Coffey, Holbrook, & Atkinson, 1996). In recente publicaties wordt dat laatste risico niet zo hoog ingeschat. Volgens Lewins en Silver (2007) bieden KDAS-pakketten gewoon een aantal instrumenten die kunnen worden gebruikt om een verscheidenheid aan analytische processen te faciliteren. Het is aan de onderzoeker om een geïnformeerde keuze te maken uit die instrumenten. Toch geven Lewins en Silver tegelijkertijd aan dat de software wel degelijk invloed kan uitoefenen op de manier waarop de onderzoeker tewerkgaat. Bij de recente publicatie van DiGregorio en Davidson (2008) vinden we een soortgelijke paradox. Aan de ene kant wordt gesteld dat de onderzoeker ervoor kan zorgen dat de KDAS-instrumenten passen bij de gebruikte methode. Aan de andere kant wordt gewaarschuwd dat KDAS-pakketten onderzoekers kunnen aanzetten tot excessief coderen.

Hoe zit het nou precies? In elk geval is het vanuit het oogpunt van de aandacht die binnen de informatica wordt besteed aan de gevolgen van de details van softwareontwerpen voor gebruikers opvallend dat er weinig analyses zijn gepubliceerd over de invloed van KDAS-ontwerpen op de manier van werken van kwalitatieve onderzoekers. Het empirische onderzoek van Lee en Fielding (1995:35) geeft daar wel degelijk aanleiding voor:

‘What happened when users became frustrated by their experiences in using a CAQ-DAS program? Broadly speaking, they either adapted their methods of working to the exigencies which faced them, or they ceased to use the program altogether.’

Het idee dat frustraties bij het gebruik van een software programma kunnen leiden tot aanpassingen in of ophouden met het gebruik van het programma is binnen de informatica algemeen geaccepteerd (Sharp et al., 2007). Net als bij andere software bestaat bij KDAS het risico dat onderzoekers hun manier van werken aanpassen of besluiten een pakket niet meer te gebruiken, omdat het gebruik ervan als frustrerend wordt ervaren.

In hun publicatie over KDAS kennen DiGregorio en Davidson (2008: 24) een groot belang toe aan de details van het ontwerp van een kwalitatief onderzoek:

‘the way researchers organize their research materials and their ideas has an impact on how they are able to “see” their data and on what they are able to pull out of their data’.

Dit geldt *mutatis mutandis* net zo goed voor het ontwerp van KDAS-pakketten. Hierbij speelt het ontwerp van het code-managementsysteem een centrale rol.

Het ontwerp van het code-managementsysteem in KDAS-pakketten

Het code-managementsysteem in KDAS bestaat uit een datastructuur waarin de door de onderzoeker gebruikte codes zijn gekoppeld aan datafragmenten, en waarin codes aan andere codes kunnen worden gekoppeld met behulp van zogenoemde *pointers*. Zulke *pointers* zijn niets anders dan bijvoegsels aan codes die de (database) adresgegevens bevatten van de door de onderzoeker aan die codes gekoppelde datafragmenten en andere codes. Wanneer een onderzoeker meerdere codes aan elkaar koppelt, ontstaan op deze wijze één of meerdere netwerken van codes. Dergelijke netwerken van codes vormen de basis voor de uit het onderzoek voortkomende theorie(ën) (Kelle, 1995).

MAXQDA, NVivo en Atlas.ti hebben elk hun eigen filosofie met betrekking tot de basisorganisatie van codes. Hoewel alle pakketten zodanig kunnen worden gemanipuleerd dat zij uiteindelijk een vergelijkbare functionaliteit bieden, ligt bij MAXQDA en NVivo (tot en met NVivo8) de nadruk op een hiërarchische organisatie. Bij Atlas.ti wordt daarentegen uitgegaan van een netwerk van codes waarvan de specifieke organisatie volledig wordt bepaald door de onderzoeker (Lewins & Silver, 2007).

Bij recente beschrijvingen van de effecten van ontwerpen van code-managementsystemen bij KDAS vinden we de al genoemde paradox. Codestructuren worden enerzijds als flexibel gezien; anderzijds worden onderzoekers geacht *workarounds* te ontwikkelen (manieren om met de software te werken wanneer deze geen optimale functionaliteit biedt) om aan de vereisten van hun methodologische tradities te voldoen (Lewins & Silver, 2007; DiGregorio & Davidson, 2008).

Hoewel aan het ontwikkelen van een aantal *workarounds* moeilijk te ontkomen is, is het toch belangrijk voor ogen te houden dat de reden voor een onderzoeker om een KDAS-pakket te gebruiken in de eerste plaats is gelegen in de *ondersteuning* die zo'n programma kan bieden bij het uitvoeren van een onderzoek. De software moet daarom zo goed mogelijk kunnen aansluiten bij de gebruikte methode. Het al geciteerde onderzoek van Lee en Fielding en inzichten uit de informatica geven aan wat onderzoekers doen als dat niet zo is: ze passen hun manier van werken aan, of ze stoppen gewoon helemaal met het gebruik van het programma.

Een goed begrip van het ontwerp van het code-managementsysteem in KDAS en de effecten daarvan is dan ook van groot belang voor een onderzoeker om te kunnen bepalen of KDAS, en zo ja, welk KDAS-pakket geschikt is voor een onderzoek. Een essentiële keuze die hierbij bij de huidige KDAS-pakketten gemaakt moet worden, is die tussen een hiërarchische basis-organisatie van codes en een netwerkstructuur.

Hiërarchisch code-management

Zoals aangegeven is hiërarchisch code-management dominant bij MAXQDA en NVivo (tot en met NVivo8). De reden voor dit basisontwerp van het code-managementsysteem verschilt tussen de twee pakketten. Zoals Kuckartz (2007) aangeeft is MAXQDA specifiek ontwikkeld voor kwalitatieve inhoudsanalyse. Hierbij worden vaak hiërarchische organisatiestructuren gebruikt bij de bestudering van kenobjecten (Wester, 2006). Bij NVivo ligt daarentegen aan de keuze voor een hiërarchisch code-managementontwerp een levensbeschouwelijke filosofie ten grondslag, die stelt dat betekenis alleen bestaat bij de gratie van hiërarchie. Zo schrijven de ontwikkelaars van de software:

'hierarchical category systems provide fundamental classificatory features which weld topics into a significant totality without which there is no meaning' (Richards & Richards, 1995: 95).

Dit is een opmerkelijke uitspraak waar niet alle kwalitatieve onderzoekers zich in zullen kunnen vinden. Toch heeft het gebruik van hiërarchisch code-management ook voordelen voor onderzoekers die zich niet specifiek bezighouden met kwalitatieve inhoudsanalyse en die niet alleen betekenis vinden in hiërarchische structuren. Zo kan een codehiërarchie bijvoorbeeld helpen bij het overzichtelijk en georganiseerd houden van een onderzoek, *a fortiori* als meerdere onderzoekers tegelijkertijd aan een onderzoek werken. Bij teamprojecten kan een hiërarchische structuur namelijk helpen bij het afbakenen van interessegebieden en het bevorderen van gemeenschappelijkheid en consistentie (Lewins & Silver, 2007).

Omdat een hiërarchisch code-managementsysteem in de eerste plaats uitnodigt tot een hiërarchische uitwerking van categorieën, bevordert hiërarchisch code-management *logischerwijze* een hiërarchische analyse. Dit geldt in principe ook voor een eventuele causale en processuele analyse. In een hiërarchisch code-managementontwerp wordt de onderzoeker namelijk *gedwongen* om in hiërarchische termen te denken over causaliteit en processualiteit (Weitzman & Miles, 1995). Dit kan op zichzelf al vrij lastig zijn; daar komt nog bij dat niet iedere kwalitatieve onderzoeker geneigd zal zijn tot een hiërarchische analyse van causaliteit en processualiteit. Een hiërarchisch code-managementsysteem past daarom goed bij een gefaseerde analyse, waarbij eerst categorieën hiërarchisch worden uitgewerkt, en vervolgens pas een *niet-hiërarchische* causale/processuele analyse wordt uitgevoerd *buiten het kader van het hiërarchische code-managementsysteem* (Wester & Peters, 2004).

Een kanttekening is hierbij op zijn plaats. NVivo biedt verschillende *workarounds* voor de onderzoeker die wél hiërarchisch coderen hand-in-hand wil laten gaan met een *niet-hiërarchische* causale/processuele analyse. Dit geldt net zo goed voor MAXQDA (Lewins & Silver, 2007). Dat laat onverlet dat hiërarchisch code-management *op zich* logischerwijze een hiërarchisch georganiseerde en gefaseerde analyse stimuleert. Als dat is wat een onderzoeker wil, dan is een KDAS-pakket gebaseerd op hiërarchisch code-management een logische keuze.

Code-management op basis van netwerken

In tegenstelling tot het code-managementsysteem van MAXQDA en NVivo, is bij Atlas.ti het codemanagement niet hiërarchisch ontworpen. In plaats daarvan wordt de oorspronkelijke netwerkstructuur tussen codes behouden. Het is aan de onderzoeker om de aard van relaties tussen codes aan te geven. De reden die hiervoor wordt gegeven, is dat het de bedoeling is dat de software ondersteuning biedt voor de onderzoeker terwijl zo weinig mogelijk controle wordt uitgeoefend over het intellectuele proces (Muhr & Friese, 2003-4).

Omdat de onderzoeker zelf de aard van relaties tussen codes kan aangeven, stimuleert de netwerkstructuur van het codemanagement bij Atlas.ti een pluriform georganiseerde analyse. Allerlei soorten netwerken kunnen worden geconstrueerd, waarbij zowel hiërarchische als sequentiële en causale relaties vanaf het begin van het onderzoek kunnen worden geconcipieerd (Prein, Kelle, & Bird, 1995; Lewins & Silver, 2007). Een dergelijk proces van gelijktijdige ordening en analyse is karakteristiek voor de hermeneutische methodologische traditie die een belangrijke invloed heeft gehad op de ontwikkeling van Atlas.ti (Muhr, 1991).

Het op deze manier construeren van netwerken bevordert tevens een ‘iteratieve’ (herhaalde) analyse, waarbij het segmenteren en coderen van onderzoeksmateriaal continu wordt afgewisseld met het bouwen van conceptuele netwerken en *hypertext*-structuren (Muhr, 1991). Ook deze manier van werken is karakteristiek voor de hermeneutische methodologische traditie (Bosch & Boeije, 2010).

De pluriform georganiseerde en iteratieve analyse die wordt gestimuleerd door op netwerken gebaseerd code-management kan wel al snel leiden tot een onoverzichtelijk web van relaties tussen codes. Ook kan het gebruik van een op netwerken gebaseerd code-management leiden tot een gebrekkige analyse van structuur. Dit komt doordat een op netwerken gebaseerd code-managementontwerp suboptimale functionaliteit biedt voor een hiërarchische analyse van structuur.

Ook hier moet als kanttekening worden geplaatst dat zowel excessieve complexiteit als een gebrekkige analyse van structuur kunnen worden voorkomen. Dat laat onverlet dat het gebruik van op netwerken gebaseerd code-management *op zich* een pluriform georganiseerde en iteratieve analyse stimuleert. Een onderzoeker die een dergelijke methode

wil gebruiken, kan daarom het beste kiezen voor een softwarepakket met op netwerken gebaseerd codemanagement.

Conclusie

In dit artikel hebben wij betoogd dat verschillende ontwerpen van code-management bij KDAS-pakketten verschillende manieren van werken stimuleren. Hiërarchisch code-management bevordert een hiërarchisch georganiseerde en gefaseerde analyse, terwijl een op netwerken gebaseerd codemanagement een pluriform georganiseerde en iteratieve analyse stimuleert. Een onderkenning van dit essentiële verschil kan kwalitatieve onderzoekers helpen bij hun keuze van een KDAS-pakket.

Idealiter zou het theoretische en methodologische uitgangspunt van de onderzoeker bepalend moeten zijn voor het ontwerp van de software. In de praktijk is het natuurlijk onmogelijk voor ontwerpers om zulk maatwerk te leveren. Toch wordt nu wat te gemakkelijk gesteld dat onderzoekers maar *workarounds* moeten ontwikkelen zodat ze uiteindelijk tóch datgene met KDAS kunnen doen wat ze – vanuit hun methodologisch en theoretisch uitgangspunt – wensen. Als onderzoekers daar al toe bereid zouden zijn, dan is er nog steeds sprake van tijd- en efficiëntieverlies – en dat staat haaks op een van de belangrijkste redenen voor het gebruik van KDAS. Om de software beter te laten aansluiten op de wensen van onderzoekers, is het daarom belangrijk om de discussie tussen methodologen en de ontwikkelaars van KDAS aan te scherpen.

Het zou daarbij goed zijn als methodologen beter zouden aangeven wat zij precies van een KDAS-pakket verwachten. Dit zal afhangen van het type kwalitatief onderzoek: etnografen, fenomenologen, gefundeerde theoretici, hermeneuten, inhouds- en discoursanalysten, actie-onderzoekers, enzovoort: allen zullen zij naar iets anders op zoek zijn. Op hun beurt zouden de ontwikkelaars van de toonaangevende KDAS-pakketten zich kunnen afvragen of zij er wel verstandig aan doen een enkel pakket te ontwikkelen voor alle vormen van kwalitatief onderzoek. Zoals besproken is MAXQDA specifiek ontwikkeld voor kwalitatieve inhoudsanalyse terwijl Atlas.ti sterk is beïnvloed door de hermeneutiek. Die pakketten lijken dan ook zeer geschikt voor juist die respectievelijke soorten van kwalitatief onderzoek. NVivo lijkt zich op dit moment in een overgangsfase te bevinden. Zoals de ontwikkelingen bij NVivo aangeven, staat hierbij het herontwerp van het code-managementsysteem centraal. Een beter moment om na te denken en te discussiëren over het methodologische belang van het code-managementsysteem bij KDAS-pakketten is haast niet denkbaar.

Literatuur

Bosch, R. & Boeije, H. (2010). Wetenschapsfilosofische grondslagen bij *Analyseren in kwalitatief onderzoek*. Bijlage bij Boeije, H. (2005). *Analyseren in kwalitatief onderzoek*. Amsterdam: Boom Onderwijs. http://www.analysereninkwalitatiefonderzoek.nl/documenten/artikel_wetenschapsfilosofische_grondslagen.pdf.

- Coffey, A., Holbrook, B. & Atkinson, P. (1996). Qualitative data analysis: technologies and representations. *Sociological research online*, 1(1), <http://www.soc.surrey.ac.uk/socres-online/1/1/4.html>.
- Conrad, P. & Reinhartz, S. (1984). Computers and qualitative data: Editor's introductory essay. *Qualitative Sociology*, 7(1/2), 3-15.
- Di Gregorio, S. & Davidson, J. (2008). *Qualitative research design for software users*. Maidenhead, England: Open University Press.
- Hesse-Biber, S. (1995). Unleashing Frankenstein's monster? The use of computers in qualitative research. In R.G. Burgess (Ed.), *Computing and qualitative research. Studies in qualitative methodology*, vol. 5 (pp. 25-41). Greenwich, CT: JAI Press.
- Kelle, U. (1995). Introduction: An overview of computer-aided methods in qualitative research. In U. Kelle (Ed.), *Computer-aided qualitative data analysis: Theory, methods and practice* (pp. 1-17). Thousand Oaks: Sage.
- Kuckartz, U. (2007). *MAXQDA 2007 reference manual*. (J. Poppe, Transl.). Marburg: VERBI Software.
- Lee, R.M. & Fielding, N.G. (1995). User's experiences of qualitative data analysis software. In U. Kelle (Ed.), *Computer-aided qualitative data analysis: Theory, methods and practice* (pp. 29-40). Thousand Oaks: Sage.
- Lewins, A. & Silver, C. (2007). *Using software in qualitative research: A step-by-step guide*. Los Angeles: Sage.
- Muhr, T. (1991). ATLAS/ti – A prototype for the support of text interpretation. *Qualitative Sociology*, 14(4), 349-371.
- Muhr, T. & Friese, S. (2003-4). *User's manual for ATLAS.ti 5.0*, 2nd ed. Berlin: Scientific Software Development.
- Prein, G., Kelle, U. & Bird, K. (1995). An overview of software. In U. Kelle (Ed.), *Computer-aided qualitative data analysis: Theory, methods and practice* (pp. 190-210). Thousand Oaks: Sage.
- QSR (2008). *NVivo8-Getting-Started-Guide*. QSR International. www.qsrinternational.com.
- Richards, T. & Richards, L. (1995). Using hierarchical categories in qualitative data analysis. In U. Kelle (Ed.), *Computer-aided qualitative data analysis: Theory, methods and practice* (pp. 80-95). Thousand Oaks: Sage.
- Sharp, H., Rogers, Y. & Preece, J. (2007). *Interaction design: Beyond human-computer interaction*. 2nd ed. Chichester, England: John Wiley & Sons.
- Vliet, H. van (2008). *Software engineering: Principles and practice*. 3rd ed. Chichester, England: Wiley & Sons.
- Weitzman, E.A. & Miles, M.B. (1995). *Computer programs for qualitative data analysis*. Thousand Oaks, CA: Sage.
- Wester, F. (red.) (2006). *Inhoudsanalyse: Theorie en praktijk*. Deventer: Kluwer.
- Wester, F. & Peters, V. (2004). *Kwalitatieve analyse: Uitgangspunten en procedures*. Bussum: Coutinho.