



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2025, gehouden op donderdag 27 maart 2025 jl. en georganiseerd door Hogeschool Windesheim). Bij elkaar zo'n 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op 18 maart 2027 in Arnhem en wordt georganiseerd door HAN University of Applied Sciences.

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

Het gebruik van ALEXIS bij inleidend programmeeronderwijs, een experiment

T.C. Ruys
Universiteit Twente
faculteit Informatica
Postbus 217
7500 AE Enschede

Samenvatting

ALEXIS is een computerprogramma dat een hulp kan zijn voor de docent die vaardighedenonderwijs verzorgt. Dit artikel beschrijft een experiment met ALEXIS bij inleidend programmeeronderwijs, waaruit blijkt dat ALEXIS de kwaliteit van programmeeronderwijs kan verbeteren.

1 Inleiding

Vergissen is menselijk. En ook een docent is maar een mens. Wanneer je tijdens een college bij het oplossen van een vraagstuk onjuiste informatie verstrekt dan is het alsof de studenten de verkeerde oplossingsmethode veel beter onthouden. Dit fenomeen ontdekte ik enkele jaren geleden toen ik bij het corrigeren van huiswerkopgaven van inleidend programmeeronderwijs steeds dezelfde fout aantrof. De eerste keer dat ik de fout constateerde noteerde ik netjes waar de desbetreffende student de mist in was gegaan en wat hij kon doen om de fout in het vervolg te vermijden. Toen ik de fout voor de vijfde keer tegenkwam was mijn commentaar al veel beknopter en bij de negende student bekroop mij het vermoeden dat er tijdens het werkcollege iets was misgegaan. In de overige opgaven heb ik de fout niet meer verbeterd. Tijdens het volgende werkcollege heb ik de bewuste theorie nogmaals behandeld. Nu echter zonder fouten.

Dit soort ongelukjes komen gelukkig zelden voor. Wel valt op dat studenten over het algemeen vaak dezelfde fouten maken. En ook dan betrap ik mezelf er soms op dat ik bij het voor de vijfde keer constateren van een fout veel minder enthousiast de fout becommentarieer dan de eerste keer. Dezelfde fouten van commentaar voorzien is niet efficiënt en kan door irritatie

te beknopte feedback tot gevolg hebben. In zo'n geval heeft klassikale behandeling van de fout niet de gewenste uitwerking. 'Die fout zal mijn buurman wel gemaakt hebben', is veelal de reactie.

In dit artikel beschrijf ik een experiment met het computerprogramma ALEXIS bij inleidend programmeeronderwijs. ALEXIS heeft de pretentie een deel van de efficiëntie- en kwaliteitsproblemen bij vaardighedenonderwijs op te lossen. Allereerst zullen kort de problemen bij inleidend programmeeronderwijs worden geschetst. In paragraaf 3 worden enkele specifieke eigenschappen van ALEXIS besproken. Paragraaf 4 beschrijft het experiment met ALEXIS bij inleidend programmeeronderwijs. Vervolgens zullen enkele toekomstverwachtingen de revue passeren en wordt het artikel afgesloten met enkele conclusies.

2 Programmeeronderwijs als vaardighedenonderwijs

Programmeren is typisch een vaardigheid die men zich alleen eigen kan maken door veel te oefenen. De nadruk bij het inleidend programmeeronderwijs ligt dan ook doorgaans op het zelf schrijven van programma's. Maar alleen oefenen is niet voldoende: zonder goede feedback op het ontwerp en de implementatie van een algoritme komt de student er niet.

In de ideale onderwijsomgeving zou elke poging van de student uitvoerig van commentaar moeten worden voorzien. Niet alleen mondeling, maar ook schriftelijk zou de student over zijn sterke en zwakke punten moeten worden geïnformeerd en zonodig in een vroeg stadium worden bijgestuurd. In de praktijk echter wordt het commentaar vaak beperkt tot enkele losse krabbeltjes in de kantlijn.

De belangrijkste oorzaak van deze te bondige feedback is het feit dat de docent te weinig tijd heeft om elke poging uitvoerig van commentaar te voorzien. Ook het in de inleiding genoemde fenomeen dat studenten vaak dezelfde (standaard) fouten maken kan te beknopt commentaar tot gevolg hebben.

Bij het becommentariëren van tussenresultaten bij (inleidend) programmeeronderwijs zijn dus twee problemen waar te nemen. Enerzijds komt door te bondige feedback de kwaliteit van het onderwijs in gevaar en aan de andere kant is het werk van de docent niet al te efficiënt: hij moet vaak dezelfde (standaard) fouten van commentaar voorzien.

Een oplossing voor deze problemen lijkt gevonden in ALEXIS, een computerprogramma dat de docent die vaardighedenonderwijs verzorgt hulp biedt bij het becommentariëren van leerlingenwerk.

3 ALEXIS

ALEXIS is een programma, dat in samenwerkingsverband aan de Universiteit Nijenrode en de Universiteit Twente ontwikkeld is.

ALEXIS, ook wel ALECSYS (All-purpose Learner-oriented Efficiency-increasing Commentary SYStem) kan bij het corrigeren van (tussen-) resultaten van studenten het routineuze werk voor zijn rekening nemen, terwijl de taken die inzicht vragen in het leer- en onderwijsproces voor de docent blijven. De docent bekijkt het product van de leerling, gaat na wat de prestaties in het verleden zijn geweest en beslist welk commentaar er moet worden verstrekt.

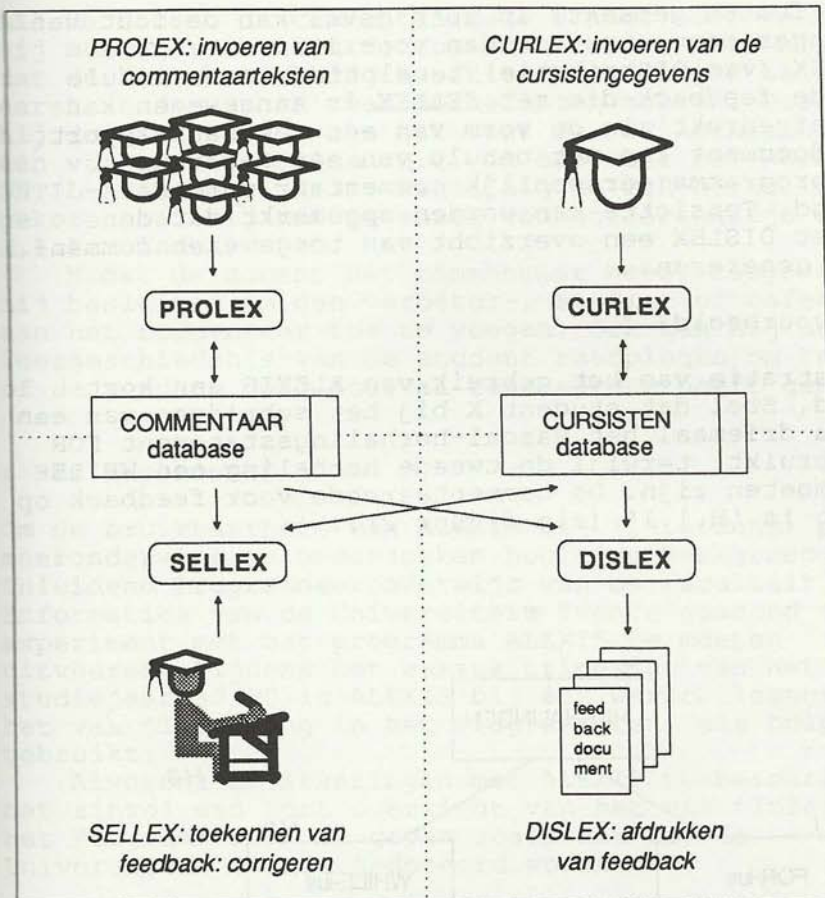
3.1 Het gebruik van ALEXIS

Een corrigeersessie met ALEXIS verloopt globaal als volgt. Allereerst zal men de gegevens van de cursisten (naam, studierichting en dergelijke) met ALEXIS moeten opslaan. Vervolgens kan men de opgaven van de studenten met ALEXIS corrigeren door het intypen van korte codes die corresponderen met commentaarteksten uit de feedbackverzameling van ALEXIS. Nadat men de opgaven heeft gecorrigeerd laat men ALEXIS een feedbackrapport generen waarin voor elke student de geconstateerde fouten en adviezen voor verbetering, oefening of studie zijn opgenomen. Dit feedbackrapport hoeft daarna alleen nog te worden afgedrukt.

Het belangrijkste bestanddeel van ALEXIS wordt gevormd door de verzameling ('de database') commentaarteksten. Naast deze database heeft ALEXIS de beschikking over een database met cursisteninformatie en vier modulen, die de becommentariëring in goede banen leiden. De vier modulen hebben in ALEXIS allemaal een naam gekregen: PROLEX, CURLEX, SELLEX en DISLEX; kortom een voorvoegsel van drie letters en LEX, van aLEXis.

Met PROLEX (van PROductie) kunnen commentaarteksten aan de feedbackdatabase worden toegevoegd. Ook is het mogelijk om bestaande teksten te wijzigen of te verwijderen. Wanneer de verzameling feedbackteksten een min of meer compleet karakter heeft gekregen hoeft deze module eigenlijk niet meer gebruikt te worden.

Met CURLEX (van CURsistenadministratie) kunnen gegevens over de cursisten (naam, adres, en dergelijke) in de cursistendatabase worden ingevoerd. Ook kan men met CURLEX cursisteninformatie wijzigen of verwijderen. Het invoeren van de gegevens van een cursistengroep hoeft natuurlijk maar één maal te gebeuren.



Figuur 1
Globaal overzicht van ALEXIS

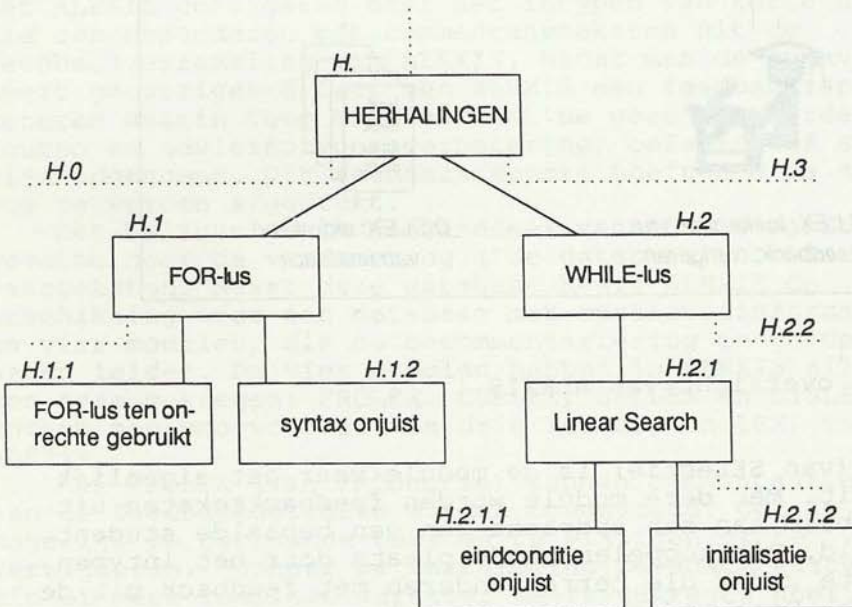
SELLEX (van SElectie) is de module waar het eigenlijk om draait. Met deze module worden feedbackteksten uit de database aan een opdracht van een bepaalde student gekoppeld. Dit koppelen vindt plaats door het intypen van korte codes die corresponderen met feedback uit de database. Deze codes worden vervolgens in de cursisten-database opgeslagen. Het zal duidelijk zijn dat de docent deze codes niet uit zijn hoofd hoeft te kennen: SELLEX biedt verschillende mogelijkheden om in de boom van commentaarteksten naar de gewenste feedback te zoeken (zie figuur 2 voor een voorbeeld van een gedeelte van een feedbackboom). Naast beschrijvingen van fouten kunnen ook verbeter-, oefen- en studieadviezen worden gegeven. Ook is het mogelijk om de leergeschiedenis van de student te raadplegen. Of met andere woorden: heeft de student bij vorige opdrachten

dezelfde fouten gemaakt? In zo'n geval kan de fout van een strenger commentaar worden voorzien.

DISLEX (van DIStributie) tenslotte, is de module waarmee de feedback die met SELLEX is aangewezen kan worden 'afgedrukt' in de vorm van een feedbackrapport. Aan dit document kan met behulp van een tekstverwerkingsprogramma persoonlijk commentaar worden toegevoegd. Tenslotte kan worden opgemerkt dat de docent met DISLEX een overzicht van toegewezen commentaar kan genereren.

3.2 Een voorbeeld

Ter illustratie van het gebruik van ALEXIS een kort voorbeeld. Stel dat student X bij het schrijven van een programma driemaal het Pascal-herhalingsstatement FOR heeft gebruikt, terwijl de tweede herhaling een WHILE-lus had moeten zijn. De commentaarcode voor feedback op deze fout is 'H.1.1' (zie figuur 2).



Figuur 2

Een gedeelte uit de commentaartekstenboom

Indien de docent deze code paraat heeft kan hij 'H11' intypen en op het beeldscherm verschijnt het toegewezen commentaar dat bij de ingetypte code hoort, zodat de docent kan controleren of de feedback juist is.

Als de docent de commentaarcode echter niet kent zal hij moeten zoeken in de commentaarboom. Daar hij weet dat de fout betrekking heeft op herhalingen zal hij als eerste onder de 'H' van herhalingen zoeken. Daar vindt hij drie 'takken', een tak voor de FOR-herhaling (H.1), één voor de WHILE-lus (H.2) en één voor het REPEAT-UNTIL-statement (H.3). Nadat hij de FOR-herhaling heeft gekozen zal hij het gewenste commentaartekstje onder H.1.1. vinden.

Nadat de docent het commentaar heeft toegewezen kan hij besluiten om een verbeter-, studie- of oefenadvies aan het commentaar toe te voegen. Ook kan hij de leergeschiedenis van de student raadplegen om te kijken of de student deze fout al eens eerder heeft gemaakt.

4 Het experiment

Om de bruikbaarheid van ALEXIS bij (inleidend) programmeeronderwijs te onderzoeken heeft de Werkgroep Inleidend Programmeeronderwijs van de faculteit Informatica van de Universiteit Twente gemeend een experiment met het programma ALEXIS te moeten uitvoeren. Tijdens het eerste trimester van het studiejaar 89/90 is ALEXIS bij één werkcollegegroep van het vak 'Inleiding in het Programmeren' als hulpmiddel gebruikt.

Alvorens de ervaringen met ALEXIS te beschrijven is het zinvol een kort overzicht van het vak 'Inleiding in het Programmeren' te geven zoals dat aan de Universiteit Twente gedoceerd wordt.

4.1 Het vak 'Inleiding in het Programmeren'

Het vak 'Inleiding in het Programmeren' is een propaedeutische vak (100 uur) voor de technische studierichtingen van de Universiteit Twente. Het doel van de cursus is de studenten te leren hoe ze problemen kunnen oplossen met een computer. Het gaat hierbij primair om het ontwerp van een algoritme dat het probleem oplost. De algoritmen worden uiteindelijk beschreven in de programmeertaal Pascal.

De organisatie van het vak kan ruwweg worden verdeeld in drie delen: een hoorcollege, een werkcollege en een practicum. In de hoorcolleges wordt de theorie aan de hand van een syllabus behandeld. Tijdens de werkcolleges worden klassikaal programmeeropgaven gemaakt. Na afloop van de cursus wordt de kennis van de student getoetst aan de hand van een tentamen.

Het belangrijkste bestanddeel van het vak wordt echter gevormd door het opgavengedeelte van het verplichte practicum. Hierbij moeten de studenten

individueel vier problemen uitwerken. De oplossingen worden mondeling met de werkcollegedocent of met een studentassistent besproken. Voorts zijn er werkcollegedocenten die de studenten de mogelijkheid bieden huiswerkopgaven te maken. Deze huiswerkopgaven worden gecorrigeerd en bij een volgend werkcollege geretourneerd.

Het moge duidelijk zijn dat slechts bij het opgavenpracticum en de huiswerkopgaven individuele feedback wordt gegeven. Het experiment met ALEXIS heeft zich geconcentreerd op het opgavenpracticum.

Voor zowel de student als de werkcollegedocent is het opgavenpracticum zeer tijdsintensief. Zoals vermeld krijgt de student bij dit opgavenpracticum vier opgaven voorgelegd, waarvoor hij een algoritme moet ontwerpen dat hij tenslotte naar een Pascalprogramma moet vertalen. Elke opgave wordt mondeling met de student besproken. In het geval de oplossing van het probleem niet juist is moet de student verbeteringen aanbrengen en de verbeterde opgave opnieuw ter correctie aanbieden. Dit proces gaat net zolang door totdat de docent respectievelijk studentassistent van mening is dat de student het probleem voldoende begrepen heeft en het eindproduct in Pascal zonder al te veel problemen op een computer zou kunnen worden getest. Voor de volledigheid kan nog worden vermeld dat de studenten één van de vier opgaven op de computer moet testen.

De auteur van dit artikel heeft ALEXIS bij de correctierondes van twee series opgaven van het opgavenpracticum gebruikt. Bij deze opgaven is ALEXIS met twee compleet verschillende doelstellingen ingezet. In de volgende twee subparagrafen zullen de twee gehanteerde methoden worden beschreven en geëvalueerd.

4.2 ALEXIS als vervanging van de mondelinge bespreking

Bij de eerste serie opgaven van het opgavenpracticum ligt de nadruk op het bewijzen van de correctheid van de gebruikte herhalingslus in het algoritme.

De doelstelling die aan de gebruikte methode bij opgave één ten grondslag lag was dat de feedback van ALEXIS de gebruikelijke mondelinge bespreking van 20 minuten zoveel mogelijk zou vervangen. De methode was vooral gericht op een efficiëntieverbetering van het onderwijs: de docent hoopte dat het corrigeren met ALEXIS minder tijd zou kosten dan de mondelinge bespreking.

De volgende methode is hierbij gehanteerd. De studenten leverden tijdens een werkcollege allen tegelijk de oplossingen van hun opgave bij de docent in. De docent zou de opgaven met ALEXIS van commentaar voorzien en bij het volgende werkcollege retourneren. Oplossingen die niet goedgekeurd waren moesten de

volgende week opnieuw worden ingeleverd en zouden wederom met ALEXIS worden gecorrigeerd.

Al gauw bleek deze *vervangings*-methode in de praktijk niet goed te werken. Er moest zeer veel opgavespecifiek commentaar met ALEXIS worden gegenereerd (en dus met PROLEX worden ingevoerd). De tijd, nodig voor het corrigeren van een opgave liep hierdoor enorm uit de hand. Het was onmogelijk de opgaven van 24 studenten in één week na te kijken. Dit was echter niet alleen het gevolg van het gebruik van ALEXIS, maar ook het gevolg van de bij aanvang bijna lege database met commentaarteksten: daar ALEXIS voor het eerst bij inleidend programmeeronderwijs werd ingezet moest tijdens het corrigeren de database voor een groot gedeelte worden gevuld.

Een veel groter nadeel dan de uit de hand gelopen tijdsbesteding van de docent was echter het niet functioneren van de feedback van ALEXIS. Een deel van de studenten begreep niet wat er met het commentaar van ALEXIS bedoeld werd. Vooral bij zeer slecht uitgewerkte problemen was het onmogelijk om de studenten op de goede weg te helpen. Het gevolg was dat deze studenten niet in staat waren hun opgave te verbeteren. Ook de periode tussen het inleveren van de opgave en het retourneren van de opgave met ALEXIS-feedback (in dit geval twaalf dagen) was veel te groot om de studenten snel en adequaat op hun fouten te wijzen.

Voor de studenten die een theoretische achterstand hadden was de in de feedback gebruikte terminologie onbegrijpelijk. Op deze studenten had het commentaar van ALEXIS wel een positieve uitwerking: ze werden op hun hiaten in theoretische kennis gewezen en gingen hard aan de slag om deze achterstand in te halen. Dit was en is echter geen doel van het gebruik van ALEXIS.

De studenten waren in het algemeen negatief over de bij deze opgave gevolgde methode. Alleen de goede studenten, waarvan de opgave in de eerste correctieronde goed of vrijwel goed was bevonden, waren enigszins positief over deze vorm van begeleiding.

4.3 ALEXIS als aanvulling op de mondelinge bespreking

Naar aanleiding van opgave één is gebleken dat een mondelinge bespreking in veel gevallen onontbeerlijk is voor het op het juiste pad brengen van de student. Het algemene karakter van de commentaarteksten in ALEXIS bleek onvoldoende om de studenten verder te helpen.

Bij de tweede serie opgaven, met als thema 'files in Pascal', is getracht het algemene karakter van de commentaarteksten van ALEXIS zo goed mogelijk te benutten. Besloten werd alleen de echte 'standaard'-fouten in de opgaven van commentaar te voorzien. Hierbij valt bijvoorbeeld te denken aan: het

rechtstreeks inlezen van een recordvariabele van een tekstfile, géén test op het bereiken van het einde van een file, enz. De moeilijke problemen, zowel in het ontwerpgedeelte als op het Pascalvlak, zouden tijdens de mondelinge bespreking behandeld kunnen worden.

Met de studenten werd afgesproken dat ze het uitgewerkte probleem twee dagen voor de mondelinge bespreking bij de docent moesten inleveren. De docent keek de opgaven globaal na (alleen lettend op de eerder genoemde standaardfouten) en zorgde ervoor dat de gecorrigeerde opgave ruim voor de mondelinge bespreking beschikbaar was voor de student. Zo kon de student voor de bespreking het commentaar bekijken en bij eventuele onduidelijkheden tijdens de bespreking om opheldering vragen. De fouten die met ALEXIS van feedback voorzien waren kwamen tijdens de bespreking niet meer ter sprake. De bespreking van 20 minuten werd teruggebracht naar 15 minuten.

Deze *aanvullings*-methode bleek goed te functioneren. Het becomingentariëren van de opgave kon veel oppervlakkiger gebeuren dan bij opgave één: standaardfouten werden met ALEXIS van feedback voorzien, terwijl moeilijke fouten alleen werden aangestreept. Dit 'aanstrepen' van fouten kan ook met ALEXIS gerealiseerd worden. ALEXIS genereert dan geen foutbeschrijving maar een etiket (naam van de fout). Deze optie van ALEXIS is tijdens dit experiment niet gebruikt.

De studenten waren erg enthousiast over het op deze wijze gebruiken van ALEXIS. Als grootste voordeel zagen zij het feit dat ze nu over een schriftelijke beschrijving van hun standaardfouten beschikten. Doorgaans worden deze fouten tijdens een bespreking snel aangestipt en is de student na afloop de helft van de opmerkingen vergeten. Ook bij opgave één hadden de studenten de beschikking over feedback op hun standaardfouten, maar dat stond tussen het vaak onduidelijke opgavespecifiek commentaar. Hierdoor konden ze door de bomen het bos niet meer onderscheiden.

Ook ervoeren de studenten het als plezierig dat ze de opgave binnen een dag retour kregen waardoor de opgave nog vers in het geheugen lag en ze foutbeschrijvingen veel beter konden plaatsen dan bij de eerste opgave. Tevens konden de studenten die het commentaar van ALEXIS onverhoopt niet begrepen binnen een dag om opheldering vragen.

Voor de docent was deze manier van werken ook veel plezieriger. In plaats van 24 opgaven in één keer kreeg hij zo'n zes opgaven per dag ter correctie aangeboden. Het geven van feedback op de standaardfouten kostte minder dan tien minuten per opgave, zodat de correctie per dag goed te overzien was.

Het feit dat de docent de opgave van te voren met ALEXIS aandachtig bestudeerd had kwam de mondelinge bespreking ten goede. De docent wist precies waar de moeilijkheden zaten en kon de opgave veel efficiënter bespreken. Bij de traditionele werkwijze ging een deel van de mondelinge bespreking verloren doordat de docent de uitwerking van de student moest doorlezen.

Een nadeel van de gespreide methode was echter dat de arbeidsintensieve procedure van SELLEX naar eindproduct nu bijna dagelijks uitgevoerd moest worden.

Het lijkt alsof ALEXIS als 'aanvulling' op de mondelinge bespreking goed te gebruiken is om de kwaliteit van de feedback op standaardfouten te verbeteren. Of ALEXIS ook een directe tijdswinst kan opleveren is nog de vraag. De mondelinge bespreking kan bij een volgend gebruik van ALEXIS worden teruggebracht naar 10 minuten. De nakijkslag met ALEXIS lijkt echter niet onder 10 minuten per opgave gebracht te kunnen worden.

5 Toekomstig gebruik van ALEXIS

Het resultaat van het experiment met ALEXIS is positief genoeg om het gebruik van ALEXIS als hulp bij het opgavenpracticum van Inleiding in het Programmeren verder te onderzoeken. Allereerst zal echter de gegevensbank met commentaarteksten (momenteel circa 140) sterk moeten worden uitgebreid. Ook voor de arbeidsintensieve procedure van SELLEX naar eindproduct zal een oplossing moeten worden gevonden.

Daarna hoopt de Werkgroep Inleidend Programmeeronderwijs het experiment met ALEXIS uit te breiden door meerdere docenten met ALEXIS te laten werken. Hierbij zal onderzocht worden of docenten, die niet aan het totstandkomen van de database meegewerkt hebben, zodanig met ALEXIS overweg kunnen, dat hun onderwijs verbetert.

Ook kunnen in de toekomst de huiswerkgaven van Inleiding in het Programmeren met ALEXIS gecorrigeerd worden.

6 Afsluiting

Geconcludeerd kan worden dat ALEXIS als hulpmiddel bij een mondelinge bespreking van opgaven van inleidend programmeeronderwijs goed inzetbaar is. Vervanging van de mondelinge bespreking lijkt vooralsnog onmogelijk. Een kwaliteitsverbetering van onderwijs lijkt zeker haalbaar, een tijdswinst (nog) niet. Maar ook alleen een verbetering van de kwaliteit van het onderwijs moet een meer dan voldoende reden zijn om ALEXIS te

gebruiken. Tijdens het gebruik van ALEXIS bleek het programma een positief neveneffect te hebben: de docent werd bij het kiezen van het juiste commentaar gedwongen zijn leerdoelen helder te expliciteren in termen van eigenschappen die de oplossingen van de studenten moeten vertonen.

Toch zitten er nog wat haken en ogen aan het gebruik van ALEXIS. Op het moment kan er bij de beide opgavenseries van het experiment redelijk met ALEXIS worden gewerkt, daar de database op deze twee typen van opgaven afgestemd is. Voor een nieuwe toepassing (bijvoorbeeld: functioneel programmeeronderwijs, vertalerbouw, enz) begint men echter met een schone lei en zal men moeten beginnen met het schrijven van commentaarteksten. Wat dat betreft biedt ALEXIS nog geen gebruikersvriendelijke toevoegmogelijkheid. Maar misschien dat dit bij een volgende versie van het programma verbeterd wordt.

En mocht de docent in de toekomst tijdens een werkcollege onverhoopt een foute oplossingsmethode geven dan kunnen ook de uit deze fout voortvloeiende onjuiste oplossingen met ALEXIS van commentaar worden voorzien. Op deze wijze kan de docent met ALEXIS een persoonlijk overzicht van zijn eigen blunders opbouwen.

Gebruikte literatuur

P. Looijmans e.a. (1986) Een schrijfcursus via ALEXIS: teamwork van docent en computer (1). In: Tijdschrift voor Taalbeheersing 8 (1986), 24-41.

P. Looijmans e.a. (1987) Een schrijfcursus via ALEXIS: teamwork van docent en computer (2). In: Tijdschrift voor Taalbeheersing 9 (1987), 181-200.

P. Looijmans e.a. (1989) Handleiding ALEXIS, versie 4.00, Groningen: Wolters-Noordhoff.

T.C. Ruys (1990) Experiment: Gebruik van ALEXIS bij Inleiding in het Programmeren, Interne publicatie voor de werkgroep inleidend programmeeronderwijs, faculteit Informatica, Universiteit Twente, Enschede.