

Teacher

Citation for published version (APA):

Reek, van den, K. (1990). *Teacher: de leerkracht interface*. (IPO rapport; Vol. 749). Instituut voor Perceptie Onderzoek (IPO).

Document status and date:

Gepubliceerd: 30/05/1990

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Rapport no. 749

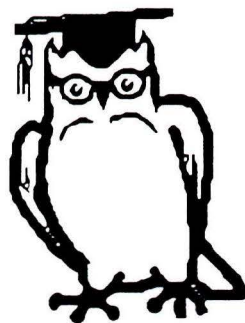
Teacher

De leerkracht interface

K. van den Reek

Hogeschool Eindhoven

Sector Technisch Onderwijs, afdeling Informatika



TEACHER

'De leerkracht interface'

Bedrijfsmentoren: Drs. H.H. Ellermann
Ing. C.D.J.M. v.d. Pol

Schoolmentor : Ir. H.Q.J. Meershoek

Auteur : K. van den Reek

IPO Eindhoven, juni 1990

Summary

The group Cognition and Communication of the Institute of Perception Research (IPO) has been working on the MIR (Monitor for Initial Reading) project for some time. The purpose of MIR is to investigate the utility and possible applicability of computers in the process of learning to read. The only point, that is concerned, is the technical reading skill, namely reading a text correctly without the necessity to understand it. The program is meant for group three of the primary school.

MIR is a program, that in the end has to function as an additional, but independent, reading aid for children. The development of MIR is based on a prototype, which is expanded on the basis of experiments.

At present MIR contains ten reading exercises. These exercises contain words and graphemes, which are presented to the children.

A teacher should be able to determine which exercises and words are to be presented to a specific child. Therefore a new module had to be developed, called the Teacher Interface. With the aid of this user interface, MIR can give the teacher an objective insight into the reading skill of the children, so that he is able to compare them with each other or with a standard. The teacher can adjust the correct combination of words and exercises on the basis of surveys of the results. By repeatedly evaluating these adjustments, which will be implemented in MIR, MIR will ultimately be able to work independently.

The most important tasks, which the module Teacher contains, are:

- * Start a training: The exercises are started.
- * Demonstration: It is possible to have a look at the exercises, which are used by the program. Also information is given here about the four feedback forms which are utilized by the program.
- * Class list: The list of students, which work with the program, is given here. Students can be added and removed from the list. Moreover the complete list can be removed.

- * Agenda: The function of Agenda is to look at and adapt the schedule for a particular child or a group of children. Agenda contains the sequence of exercises and the words, which are used in the exercises.

- * Progress: Summaries in tabular and graphical form can be shown. The teacher is able to look at the results of the exercises and the separate graphemes.

A prototype of a program with these tasks has been developed. In the future this prototype will be implemented and with the aid of experiments it will have be developed into a useful and user-friendly interface.

Samenvatting

Binnen de groep Cognitie en Communicatie van het Instituut voor Perceptie Onderzoek (IPO) wordt al enige tijd gewerkt aan MIR (Monitor for Initial Reading). Het doel van MIR is om de bruikbaarheid en mogelijke toepassing van computers bij het leren lezen te onderzoeken. Het gaat hierbij steeds om technische leesvaardigheid, het korrekt lezen van een tekst, zonder het noodzakelijkerwijs te begrijpen. De doelgroep is groep drie van de basisschool.

MIR is een programma, dat uiteindelijk moet fungeren als een additioneel, maar wel zelfstandig, leeshulpmiddel voor kinderen.

Bij de ontwikkeling van MIR is uitgegaan van een prototype, dat aan de hand van experimenten wordt uitgebouwd.

MIR bevat op dit moment een tiental oefeningen. Deze oefeningen bevatten woordjes en grafemen die aan de kinderen worden aangeboden.

Een leerkracht moet kunnen bepalen welke oefeningen en woorden een bepaald kind moeten worden aangeboden. Hiervoor moest een nieuwe module ontwikkeld worden, de leerkracht interface. Met behulp van deze user interface kan MIR een leerkracht een objectief beeld geven van de lees kennis van de leerlingen en deze vergelijken met de groep of een norm. De leerkracht kan vervolgens de juiste combinatie van woorden en oefeningen, aan de hand van overzichten, zelf instellen. Door het telkens evalueren van deze instellingen, die in MIR zullen worden geïmplementeerd, zal MIR uiteindelijk zelfstandig kunnen werken.

De hoofdtaken die de module Teacher op dit moment bevat zijn:

- * Training starten: de trainingen van de leerlingen worden gestart.
- * Demonstratie: de oefeningen waar het programma gebruik van maakt kunnen bekeken worden. Tevens wordt hier informatie gegeven over de vier feedbackvormen die het programma gebruikt.
- * Klasselijst: Deze bevat de lijst van leerlingen die met het programma werken. Er kunnen leerlingen toegevoegd en verwijderd worden, tevens kan de hele lijst verwijderd worden.

- * Agenda: Agenda heeft als functie het bekijken en aanpassen van de agenda van een leerling of een groep leerlingen. In agenda staat onder andere de volgorde van de oefeningen en de woordjes die de oefeningen gebruiken.
- * Vorderingen: Samenvattende overzichten in tabel en grafiekvorm kunnen bekeken worden. Resultaten van oefeningen zijn te bekijken en resultaten van afzonderlijke grafemen.

Voor deze hoofdtaken is een prototype gemaakt. Dit prototype zal in de toekomst geïmplementeerd worden waarna het door middel van experimenten verder uitgewerkt zal worden tot een bruikbaar en gebruikersvriendelijk interface.

Voorwoord

Mijn afstudeeropdracht heb ik volbracht op het Instituut voor Perceptie Onderzoek (IPO). Dit rapport bevat een beschrijving van mijn afstudeeropdracht, het ontwerpen van een user interface voor het MIR (Monitor for Initial Reading) project, waar ik in de periode van 15 januari 1990 tot 1 juni 1990 aan gewerkt heb.

Het Instituut voor Perceptie Onderzoek is een samenwerkingsvorm van de Nederlandse Philips Bedrijven B.V. en de Technische Universiteit Eindhoven. Het resorteert aan de kant van Philips onder het Natuurkundig Laboratorium en aan de kant van de Technische Universiteit onder de faculteit Wijsbegeerte en Maatschappijwetenschappen. Het IPO is ondergebracht in een stichting met als doel:

'de bevordering van het zuiver en toegepast wetenschappelijk onderzoek op het gebied van de perceptie en daarmee verwante gebieden, zulks in het belang van het Hoger Onderwijs en van die overheids-, semi-overheids- en particuliere instellingen, welke bij een dergelijk onderzoek zijn gebaat'.

Op het IPO onderzoekt men hoe mensen informatie in zich opnemen en verwerken, in wisselwerking met apparatuur en programmatuur. Het IPO onderzoek in z'n geheel wil leiden tot inzicht en tracht een bijdrage te leveren aan het doelgericht aansluiten van de grote informatiestroom uit onze samenleving op de informatiebehoeften van de gebruiker met zijn mogelijkheden en beperkingen.

Ten slotte wil ik iedereen bedanken die mij heeft geholpen bij het volbrengen van de afstudeeropdracht. Speciaal wil ik bedanken mijn begeleiders drs. H.H. Ellermann en Ing. C. van der Pol, schoolbegeleider Ir. H. Meershoek en de afstudeerder Frenk Mulder die mij heeft geholpen bij de implementatie van een deel van de Teacher interface. Verder wil ik de proefpersonen en collega afstudeerders bedanken voor de nodige kritische opmerkingen.

Inhoudsopgave

<i>samenvatting</i>	<i>i</i>
<i>voorwoord</i>	<i>iii</i>
<i>inhoudsopgave</i>	<i>iv</i>
<i>lijst met figuren</i>	<i>vi</i>
<i>informatie over de hoofdtaken van Teacher</i>	<i>vii</i>
1. Inleiding	1
2. Monitor for Initial Reading	3
2.1 De leesmethode Veilig Leren Lezen	4
2.2 De ontwikkelstrategie	5
2.2.1 De ontwikkelmethode	5
2.2.2 De structuur van MIR	7
2.3 De ontwikkelomgeving	9
3. De gebruikte methode voor het ontwikkelen van een User Interface	10
3.1 Globale functionaliteit	11
3.2 User Mapping	12
3.3 Gebruikers en taak eisen	12
3.4 Funktionaliteits matrix	13
3.5 Dialoog ontwerp	14
3.6 Prototyping	17
3.6.1 Ontwerp	18
3.6.2 Realisatie	18
3.6.3 Gebruik	18
3.6.4 Evaluatie	19
4. Teacher	20
4.1 Globale functionaliteit	20
4.2 User Mapping	21
4.3 Leerkracht en taak eisen	22
4.4 Funktionaliteits matrix	23
4.5 Dialoog ontwerp	23

4.5.1	Kies dialoog stijl	24
4.5.2	Kies de user interface stijl	25
4.5.3	Ontwerp dialoog structuur en navigatie faciliteiten	26
4.5.4	Verfijn het ontwerp en de stijl	30
4.5.4.1	Training starten	30
4.5.4.2	Demonstratie	30
4.5.4.3	Klasselijst	32
4.5.4.4	Agenda	33
4.5.4.5	Vorderingen	37
4.5.5	Gebruikers ondersteuning	41
4.5.5.1	Help	41
4.5.5.2	Feedback	41
4.5.5.3	Status informatie	42
4.5.5.4	Error management	42
4.5.5.5	Control akties	43
4.6	Prototyping	43
5	<i>Implementatie</i>	44
5.1	De file structuur van MIR	44
5.1.1	De file Vocabulary	45
5.1.2	De file Agenda	46
5.1.3	De file Users	46
5.1.4	De file Groupresult	47
5.1.5	De file Summary	47
5.1.6	De file Result	47
5.2	Werkwijze	48
5.3	Problemen	48
6	<i>Eindrapportage</i>	52
6.1	De verrichte werkzaamheden	52
6.2	Problemen	53
	<i>woordenlijst</i>	55
	<i>literatuur</i>	56
	<i>bijlagen</i>	58
Appendix A:	Indeling van Veilig Leren Lezen	58
Appendix B:	De enquête	59
Appendix C:	Listing Teacher	62

Lijst met figuren

2.1	Layout van een oefening voor het doelwoord 'boom'	5
2.2	Informatiestroom tussen de modules uit MIR	7
3.1	Voorbeelden van een scherm layout	16
4.1	Dialogbox	25
4.2	Schermlayout van Teacher	25
4.3	Demonstratie	26
4.4	Klasselijst	27
4.5	Agenda	27
4.6	Vorderingen	28
4.7	Startscherm van Teacher	30
4.8	Demonstratie	31
4.9	Klasselijst	32
4.10	Agenda	34
4.11	Aanpassen agenda	35
4.12	Vocabulaire	36
4.13	Vorderingen	37
4.14	De tabel van woorden en grafemen	38
4.15	Grafiek	39
4.16	De groep geselecteerde leerlingen voor grafiek 1	40
4.17	Een geselecteerde naam en een ongeselecteerde naam	41
4.18	Een routescherm	41
5.1	Overzicht files van MIR	45
5.2	Startscherm van Teacher	49
5.3	De tabel van woorden en grafemen	50
5.4	Het scherm voor het selecteren van oefeningen en woorden	51

Informatie over de hoofdtaken van Teacher

4.5.3

Demonstratie	26
Klasselijst	26
Agenda	27
Vorderingen	28

4.5.4

Training starten	30
Demonstratie	30
Klasselijst	32
Agenda	33
Vorderingen	37

1

Inleiding

Op het Instituut voor Perceptie Onderzoek (IPO) wordt sinds enige tijd gewerkt aan het MIR (Monitor for Initial Reading) systeem. MIR is een programma dat als doel heeft de bruikbaarheid en mogelijke toepassing in het onderwijs bij het leren lezen te onderzoeken.

MIR bevat oefeningen voor kinderen om te leren lezen. Resultaten van het doorlopen van deze oefeningen worden opgeslagen. De leerkracht moet de loop van het programma in sterke mate kunnen bepalen. Dat moet gebeuren door overzichten die de voortgang van de leerlingen weergeven. Met behulp van deze overzichten kan de leerkracht het systeem beheersen. Dit beheersen gaat met behulp van de 'Teacher' interface, Teacher genaamd.

Bij de ontwikkeling van Teacher ben ik allereerst de uitvoer van het MIR systeem gaan bestuderen, omdat dit tevens de invoer is van Teacher. Vervolgens ben ik de methode gaan bestuderen die ik zou gaan gebruiken bij het ontwikkelen van de interface. Bij Teacher is uitgegaan van de HUFIT methode (zie HUSAT 1989).

In de loop van de ontwikkeling van Teacher bleek deze methode toch niet helemaal aan onze verwachtingen te voldoen. Voordat er uiteindelijk over een prototype wordt gesproken is er al heel wat tijd gebruikt met het verzamelen van informatie over de taak van het systeem en de gebruiker. Er wordt verwacht dat deze informatie behoorlijk gedetailleerd verzameld wordt. Bij onze gebruiker, de leerkracht, bleek het verzamelen van gedetailleerde informatie over de systeemeisen een illusie. Al snel is toen begonnen met het ontwikkelen van een prototype. De gebruiker ziet nu met eigen ogen hoe het systeem eruit ziet en kan hierbij commentaar leveren en vragen stellen. Dit leverde uiteindelijk wel nuttige informatie op.

In hoofdstuk twee wordt allereerst het projekt MIR besproken. De funktionele eisen en de struktuur van MIR zullen worden besproken. Tevens bevat het hoofdstuk informatie over de ontwikkelingsmethode van MIR.

Hoofdstuk drie bespreekt de methode die gebruikt is bij de ontwikkeling van de user interface. Het volgende hoofdstuk, Teacher, bevat de invulling van de methode voor de module Teacher. Er wordt snel toegewerkt naar het prototype wat uitvoerig zal worden besproken. Dit is het laatste prototype dat is ontworpen.

Het laatste hoofdstuk bevat een beschrijving van de uitvoer van MIR dat door Teacher gebruikt wordt. Tevens is er een eerste aanzet gegeven voor de implementatie van het prototype van Teacher, zoals het op dit moment (mei 1990) bestaat. Een korte eindrapportage sluit het rapport af.

2

Monitor for Initial Reading

Binnen de Cognitie en Communicatie groep van het Instituut voor Perceptie Onderzoek is men geïnteresseerd in instructietechnologie en met name in de bruikbaarheid en mogelijke toepassing van computers bij het aanleren van taken en/of handelingen. Vanaf 1984 tot en met 1988 is er binnen deze groep een project geweest met de naam: "Ontwikkeling van een elektronisch leeshulpmiddel met spraak naar keuze, in het aanvankelijk lezen", ook wel Leesbord-project genoemd.

Het vervolgonderzoek in het Leesbord-project is voornamelijk gericht op het minimaliseren van de leertijd en het adaptief maken van de lees oefeningen aan de vorderingen van een leerling. Het programma dat daarvoor ontwikkeld gaat worden heet : 'Monitor for Initial Reading'.

Het doel van MIR (Monitor for Initial Reading) is om de bruikbaarheid en mogelijke toepassing van computers bij het leren lezen te onderzoeken. Het gaat hierbij steeds om technische leesvaardigheid, het korrekt lezen van een tekst, zonder het noodzakelijkerwijs te begrijpen. MIR is een programma dat afhankelijk van de individuele leesvaardigheid van leerlingen, van de effectiviteit van de diverse lees oefeningen en de mate van gekendheid van de woorden een zo optimaal mogelijke combinatie van lees oefeningen en woorden presenteert. De doelgroep is groep 3 van de basisschool. De leeftijd van de kinderen is ongeveer 5 à 7 jaar. MIR moet een programma zijn waar de kinderen mee kunnen leren lezen, met zo weinig mogelijk interventie van ouders of leerkrachten.

2.1 De leesmethode Veilig Leren Lezen

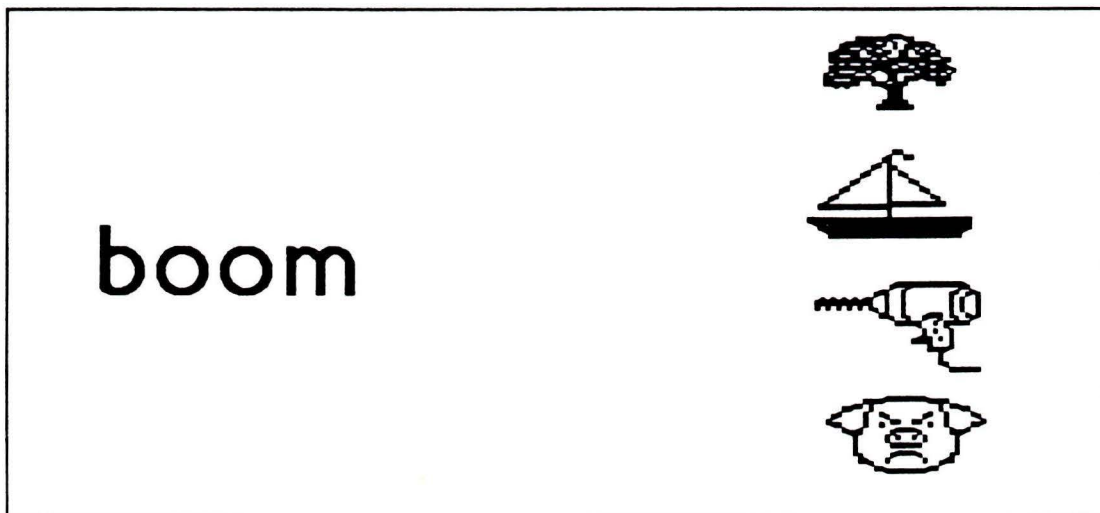
MIR is nu nog gebaseerd op een tiental oefeningen, genomen uit de leesmethode Veilig Leren Lezen, (Caesar 1988). Deze methode wordt op de meeste basisscholen toegepast.

Veilig Leren Lezen onderscheidt een drietal fases in het leesonderwijs. In de eerste fase worden woorden gelezen zonder de afzonderlijke letters (globaalwoorden) te kennen. Ook worden in deze fases de letters geleerd waaruit deze woorden zijn opgebouwd. In de tweede fase wordt geleerd om een onbekend woord te splitsen in zijn afzonderlijke letters (het dekoderen), deze een voor een uit te spreken en daarna achtereenvolgens vloeiend uit te spreken. De derde fase bestaat uit het veelvuldig oefenen van de laatste vaardigheid, het opvoeren van de leessnelheid en de leesvaardigheid.

Veilig Leren Lezen verdeelt de te behandelen letters en woorden in kleinere gedeeltes. Deze hoofdstukjes worden kernen genoemd. Iedere kern behandelt een aantal letters en past het gebruik van deze letters toe in de zogeheten globaalwoorden. In appendix A is een lijst gegeven van welke letters in welke kern voorkomen. Het gaat hier overigens niet over letters uit het alfabet, maar over grafemen. Een grafeem is een geschreven representatie van een spraakklank (foneem). Een grafeem kan uit een of meerdere letters bestaan. Een spraakklank die niet meer op te splitsen valt in andere klanken, wordt een foneem genoemd.

Een leerling begint met het oefeningen van globaalwoorden (immers dat heeft betekenis) en daarna de letters waaruit dat woordje bestaat. Bovendien geldt dat als een leerling de letters leert van een bepaalde kern, de woordjes uit voorgaande kernen geleerd moeten zijn. Voor de eerste kernen zijn er een voor een aantal globaalwoorden tekeningen, die gebruikt kunnen worden als extra hulpmiddel.

Een voorbeeld van een oefening die in MIR geïmplementeerd is, is weergegeven in figuur 2.1. Deze oefening presenteert midden op het scherm een doelwoord. Het doelwoord verschijnt met een drietal antwoordalternatieven (afleiders) in een willekeurige volgorde onder elkaar, rechts op het scherm. In deze oefening zijn de antwoordalternatieven en het doelwoord plaatjes. Hierna wordt door middel van spraak aangegeven dat de leerling de tekening van het doelwoord in het rijtje kan aanwijzen.



figuur 2.1: Layout van een oefening voor het doelwoord 'boom'

2.2 De ontwikkelstrategie

Momenteel is er nog weinig inzicht in de mogelijkheden en moeilijkheden, die kunnen optreden door het gebruik van een computer in een lessituatie. Daarom is MIR opgezet als een prototype. Bij MIR wordt er gebruik gemaakt van evolutionaire (referentie) prototyping. Bij deze vorm van prototyping evolueert het prototype tot een (onderdeel van een) definitief systeem.

2.2.1 De ontwikkelmethode

Omdat er in de loop van de tijd verschillende versies van MIR zullen ontstaan moet een degelijk raamwerk gemaakt worden dat een maximale herbruikbaarheid heeft en eenvoudig te wijzigen is. Er is daarom voor gekozen om MIR op een object georiënteerde manier te ontwerpen en te ontwikkelen (zie vd. Pol, 1989).

Object georiënteerd programmeren is een techniek, waarmee een complex systeem gezien wordt als een set samenwerkende objecten. Elk object heeft een eigen datastructuur en algoritmes. Op deze manier ontstaat een grote modulariteit en data-abstraktie.

Object georiënteerd ontwerpen modelleert het probleemgebied in concepten die zeer nauw aansluiten bij de werkelijkheid. Een van de gevolgen is dat met het analyseren

van een probleemgebied een deel van het ontwerp impliciet wordt uitgevoerd. Het toepassingsgebied en niet het probleemgebied wordt gemodelleerd, in het geval van MIR is het toepassingsgebied *het leren lezen in het algemeen* en niet *het leren lezen van Nederlands volgens een bepaalde leesmethode voor kinderen van de basisschool*, dit is het probleemgebied.

Het informatie model is een statische beschrijving van objecten in het probleemgebied op een bepaald moment. Het beschrijft welke feiten, situaties en gebeurtenissen mogelijk zijn in het complete toepassingsgebied. Hierdoor kunnen delen hergebruikt worden voor een soortgelijk probleem en is gemakkelijk uitbreidbaar en wijzigbaar. Het object georiënteerd werken heeft het voordeel dat er een model van de werkelijkheid gemaakt wordt in termen waar de informatiekundige mee vertrouwd is. Deze kan veel langer mee denken in de analyse fase. Analyse loopt dus parallel aan onderzoek, waardoor na herhaaldelijk overleg tussen ontwerper en informatiekundige een bruikbaar model van het toepassingsgebied ontstaat. Een voordeel van deze werkwijze is dat de abstrakties gebruikt in het toepassingsgebied automatisch opgenomen worden in het model, hierdoor wordt overspecificatie voorkomen.

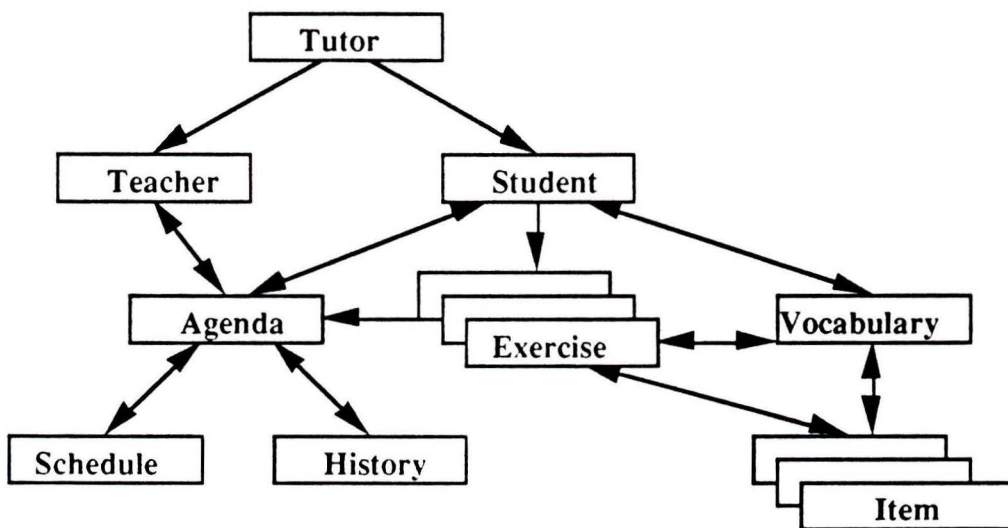
Een object georiënteerde variant van de informatie modellerings methode INFOMOD (Griethuysen, 1986) wordt gebruikt om MIR te ontwerpen. Deze methode kent een vijftal stappen, te weten:

1. Het *Informatie Structuur Model* maken: Er wordt eerst een model gemaakt van het applicatie gebied. Zo'n model is een hulpmiddel in de dialoog tussen de ontwerper en de informatiekundige. Een eerste aanzet voor het model is de terminologie, de wetmatigheden en de gebruiken die binnen het toepassingsgebied gelden. Hierna wordt het eerste model besproken met de informatiekundige, bijgesteld en getoetst. Dit wordt herhaald totdat er een bevredigend model is ontstaan.
2. *Informatie Vektoren en Informatie Regels* formuleren: Informatie vektoren zijn de formele weergave van het Informatie Structuur Model en bevatten informatie over een entiteit en de relaties met de andere entiteiten. De Informatie Regels specificeren de regels, constraints en kondities van de situaties, waarin de entiteiten betrokken kunnen zijn. De regels worden meestal gevonden door verdergaande analyse van de entiteiten en attributen in het toepassingsgebied.
3. Het *user interface* vastleggen: Dit kan onafhankelijk van de voorgaande stappen gebeuren. Het ontwerpen van de leerkracht interface gebeurt inderdaad onafhankelijk van de voorgaande stappen.

4. *Konstraints bepalen en vastleggen*: Dit beschrijft de probleem specifieke zaken die vergaande gevolgen hebben, bijvoorbeeld timing, performance en geheugen gebruik.
5. *Objekt specificaties maken*: Dit zijn de objecten, delen van het probleemgebied, die dezelfde eigenschappen hebben. Deze objecten zijn gemakkelijk uitbreidbaar en makkelijk aan te passen.

2.2.2 De structuur van MIR

MIR is opgebouwd uit modules. De modules die reeds zijn ontwikkeld en geïmplementeerd staan in figuur 2.2. In de figuur geven de pijlen de richting van de informatiestroom weer tussen de diverse modules. Als MIR wordt opgestart, wordt de besturing van het programma gegeven aan de module Tutor.



figuur 2.2: Informatiestroom tussen de modules uit MIR

- * *Tutor*
Bepaalt of de gebruiker een leraar of een leerling is door middel van de diskette die in het diskette station is geplaatst. Als het de leraar is, dan krijgt de Teacher module de controle, als het een leerling is dan wordt de Student module actief.
- * *Teacher*
De eerste aanzet voor deze module is ontwikkeld in het voorjaar van 1989. Dit prototype zou in het najaar van 1989 op school getest worden. Helaas wordt dit prototype nu nog niet getest. Dit zou waarschijnlijk goede informatie gegeven hebben over de behoefte van de gebruiker waar het interface voor ontworpen

wordt. Er is nu gekozen voor een geheel nieuwe methode voor het ontwerpen van een user interface. Deze methode zal in het hoofdstuk 4 beschreven worden. Er zal ook een heel nieuw prototype gemaakt worden, dat getest gaat worden op school.

* *Student*

Student zendt een boodschap aan Agenda om te vragen welke oefeningen en woorden moeten worden geoefend. Vervolgens start het de oefening en selecteert een bepaalde bibliotheek met items.

* *Agenda*

Bevat een vooraf gedefinieerd rooster van oefeningen en woorden, die gebruikt moeten worden in het programma. Het lesrooster krijgt allereerst de waarden zoals die gelden voor de gehele klas. Agenda zorgt dat de informatie in het lesrooster wordt bijgehouden en consistent blijft. De beschrijving van een oefening bestaat onder andere uit: de oefeningnaam, het aantal te oefenen woorden, de gebruikte woordenset en de antwoordtijd waarbinnen een antwoord moet worden gegeven.

Het lesrooster kan op twee manieren worden doorlopen. Een manier is sequentieel van de eerste tot de laatste oefening. Een andere manier is het doorlopen van het lesrooster in een willekeurige volgorde, totdat alle oefeningen een keer zijn geoefend.

* *Schedule*

Bevat een aantal oefeningen. Deze module moet adaptief worden. Dit wil zeggen dat de module zelf de volgorde en het soort woorden en oefeningen bepaalt, afhankelijk van de vorderingen van een leerling. Bijvoorbeeld als een leerling problemen heeft met een bepaalde letter, dan zullen woorden geoefend worden waarin deze letter voorkomt.

* *History*

Bevat een beschrijving van de resultaten van de leerlingen. Dit object moet kunnen worden bekeken vanuit het object Teacher en Student en dient als basis voor Schedule om een diagnose van de vorderingen te maken.

* *Exercise*

Deze module draagt zorg voor de presentatie van de verschillende items, registreert de responses en schrijft ze weg in een file voor latere bewerking. Deze presentatie omvat de lay-out van het beeldscherm, de feedback die moet worden gegeven. Als Exercise eindigt, worden de gegevens gezonden naar het object Agenda voor verdere verwerking.

* *Vocabulary*

Bevat een complete, alfabetisch gesorteerde lijst met woorden die aanwezig zijn. Dit zijn de Veilig Leren Lezen woordgroepen. De woordenlijst is in een aantal stukken verdeeld. Elk stuk bevat woorden van ongeveer dezelfde moeilijkheidsgraad. De woorden kunnen momenteel sekwentieel of willekeurig worden geselecteerd, precies zoals bij de oefeningen.

* *Item*

Bevat het doelwoord en de alternatieve antwoorden van de verschillende oefeningen. Ieder Item object bevat informatie over de manier waarop het gepresenteerd kan worden: als plaatje, als spraak, als karakters of als grafemen. Item houdt ook de resultaten per woord bij, zoals de tijd die nodig was om een antwoord te geven op een vraag.

2.3 De ontwikkelomgeving

De *hardware* waarmee MIR uiteindelijk moet werken is een Apple Macintosh Plus met een intern geheugen van 1 Mb en een harddisk van 20 Mb. MIR moet ook op andere systemen uit de Macintosh lijn kunnen werken. MIR moet op verschillende beeldschermen qua grootte en kleur kunnen werken. Deze hardware heeft een gebruikers-interface waaruit op eenvoudige wijze een interface voor kinderen kan worden geconstrueerd.

De *software* ontwikkeling gebeurt met behulp van de Macintosh Programmers Workshop (MPW). Dat is een programmeeromgeving in Object Pascal.

Het user interface van computer naar kind is onder andere de gesproken taal, dit is een voortvloeisel uit de eisen die gesteld zijn aan MIR. De communicatie van het kind met de computer geschiedt met behulp van een muis of een 'touchscreen'. De beide interfaces volgen de richtlijnen, die zijn gesteld in de 'Human Interface Guidelines' [Apple, 1986].

3

De gebruikte methode voor het ontwikkelen van een User Interface

Voor het ontwikkelen van user interfaces zijn er momenteel geen algemeen geaccepteerde methodes en er bestaat zelfs geen algemeen geaccepteerd lifecycle model, zoals dat bijvoorbeeld voor software ontwikkeling wel het geval is. Vroeger ontwierp een systeemanalist een user interface waar de klant maar blij mee moest zijn. Dit leidde vaak tot frustraties en is een van de redenen waarom veel software uiteindelijk in de kast verdween of dat de functionaliteit maar voor een zeer klein gedeelte gebruikt werd. In latere fases hield de analist een interview of enquête en ontwierp daarna zelf het user interface waar de klant zich vaak nog niet in kon vinden. Tegenwoordig wordt geprobeerd om een gebruiker zo vroeg mogelijk in het ontwerp traject actief mee te laten denken.

In de software ontwikkeling wordt nog dikwijls te weinig tijd besteed aan de ontwikkeling van de user interface. Een sprekend voorbeeld is de software ontwikkel methode System Development Methodology (SDM II, Pandata, 1988), waarschijnlijk het meest gebruikte software ontwikkel methode in Nederland. In de SDM II methode wordt pas in het detail ontwerp gepraat over de keuze van het user interface. Van het 750 pagina's dikke boek zijn er slechts vier ingeruimd om te vertellen hoe een user interface ontworpen kan worden.

Voor het ontwerpen van de leerkracht interface van MIR is een methode gekozen die vanaf het begin de gebruiker betreft bij het ontwikkelproces om zo een interface te ontwikkelen die consistent is over alle applicaties heen en ook heel duidelijk de wensen van de gebruiker weergeeft. Het huidige rapport is ook te beschouwen als een experiment: in hoeverre is het zinvol de leerkracht er vroeg bij te betrekken.

Voor de ontwikkeling van de leerkracht interface wordt een methode gebruikt die user-centered design (Norman & Draper, 1986) genoemd wordt. De hiergebruikte methode is de konkretisering van de aanpak zoals gesuggereerd door Norman & Draper in combinatie met ervaringen die opgedaan zijn in een ESPRIT project 385 -'HUFIT' (HUman Factors in Information Technology), zie HUSAT (1989). Deze methode kent een aantal fases, die in de volgende paragrafen zullen worden uitgelegd, te weten:

1. Leg *globaal de functionaliteit* en de grenzen van de applicatie.
2. Bepaal de verschillende soorten gebruikers van de applicatie en hun doelen en de voor- en nadelen van de applicatie voor hen. Dit wordt *User Mapping* genoemd.
3. Bepaal *gebruiker* en *taak eisen*. Per soort gebruiker worden eisen, kenmerken van de soort gebruiker en specifieke dingen over de taak vastgelegd. Deze beschrijvingen resulteren voor elke soort gebruikers in een *User Requirement Summary* (URS).
4. Vanuit de URS wordt de produkt specificatie gemaakt. Dit wordt gedaan met behulp van een *functionaliteits matrix*. Dit is een matrix waarin gebruikers en taak kenmerken uitgezet worden tegen de aangegeven functionaliteit. Deze produkt specificaties worden gebruikt als input voor het software ontwikkel proces. Deze URS kunnen ook gebruikt worden om testspecificaties te maken en om een bruikbaarheids evaluatie op te stellen.
5. *Dialog ontwerp*, het user-computer interface ontwerp. Vanuit de produkt specificatie wordt het user-computer interface ontworpen, rekening houdende met de verschillende soorten gebruikers.
6. Bouw een prototype en doe een evaluatie bij een aantal representatieve gebruikers van elke soort. Pas het prototype aan, ga weer terug en zo gaat het door totdat voor alle partijen een bevredigend resultaat is ontstaan.
7. Start met de software ontwikkeling, gebruik makend van de reeds verkregen specificaties.

3.1 Globale functionaliteit

Het betrekken van de gebruiker in de analyse fase kan als resultaat hebben dat er na herhaaldelijk overleg tussen ontwerper en gebruiker een bruikbaar idee van de uiteindelijke software ontstaat.

Het is heel moeilijk om alle eisen en de functionaliteit op voorhand vast te leggen. Zoals zal blijken zullen de eisen en de wensen van de gebruiker pas echt naar voren komen bij het bekijken van een prototype.

3.2 User mapping

User mapping wordt gebruikt om alle gebruikers van het geplande produkt te identificeren en bekijkt de kosten en baten voor deze groepen, voor zover dit mogelijk is, wanneer ze het produkt zouden gebruiken. Dit gebeurt in een aantal stappen.

Allereerst worden de achtergrond details bekeken. Vervolgens worden de gebruikers die bij het systeem betrokken zijn in gebruikersgroepen gezet, zodat er later een groep van mensen kan worden gevormd waar op een concrete manier over gedacht kan worden. Een gebruikersgroep is bijvoorbeeld de eindgebruikers of het systeembeheer en onderhoud.

Van deze groepen gebruikers kunnen concrete eisen opgesteld worden. Daarvoor wordt gebruik gemaakt van de basis informatie van het produkt idee. Bij het missen van informatie moet niet worden geschroomd om het eigen voorstellingsvermogen te gebruiken.

In hoeverre de eisen die nu zijn opgesteld concreet zijn, zal duidelijk worden in de fase van prototyping.

3.3 Gebruikers en taak eisen

Voor het vaststellen van gebruikers en taak eisen en kenmerken is een intensief contact met de verschillende gebruikers noodzakelijk. Verzamelen van deze gegevens wordt meestal door interviews en enquêtes gedaan. Het is belangrijk dat de gebruiker zich betrokken voelt in het ontwerp, want daarmee wordt de acceptatie graad van het ontwikkelde systeem vergroot.

Het achterhalen van de gebruikers kenmerken is noodzakelijk om een beeld van de gebruiker en de omgeving te krijgen. We bekijken onder andere de werksituatie van de gebruiker, de opleiding en waar en hoe vaak hij het systeem gaat gebruiken. Verder is het van belang om een idee te krijgen over de geoefendheid van de gebruiker: expert, incidenteel gebruiker of een ongeoefend gebruiker.

De taak analyse probeert een idee te geven van de taakstructuur en de taakfrequentie. Hoe vaker iemand met een programma werkt, des te meer vaardigheden krijgt die persoon. We onderzoeken wat de gebruiker op het moment doet, welke aspecten hem de meeste voldoening geven en welke aspecten niet prettig zijn.

Een probleem dat bij de taak analyse kan optreden is dat de gebruiker en de analist elkaar niet begrijpen of dat de gebruiker de dingen vaag vertelt terwijl de analist precisie prefereert.

Voor de taak analyse kan de volgende benadering, beschreven in de HUFIT methode (zie HUSAT, 1989), gebruikt worden:

1. Breng de omgeving en achtergronden in kaart, denk vooral aan sociale, technische en psychische aspecten.
2. Probeer vast te stellen wat je wilt weten, welke informatie je daarvoor nodig hebt en hoe je eraan kunt komen.
3. Verzamel informatie, denk aan kontrasterende situaties en uitzonderings situaties, verzamel eventueel in gedeeltes. Voor het verzamelen van informatie over kontrasterende en uitzonderings situaties is de analist heel afhankelijk van de informatie die de gebruiker geeft. Over het algemeen heeft de gebruiker de neiging om vage informatie te geven. De kontrasterende en uitzonderings situaties zullen waarschijnlijk pas bij het prototypen naar voren komen.
4. Maak een begrijpbaar geheel van de verzamelde informatie.
5. Bespreek de verzamelde informatie met de betrokken mensen.
6. Herhaal dit proces totdat voldoende informatie verzameld is.

De taak structuur geeft een idee hoe een taak opgebouwd is. Een open structuur heeft een voorspelbaar invoer en uitvoer patroon en heeft vaste procedures. Een gesloten structuur heeft grote fluktuaties in het uitvoer patroon.

De taakanalyse, taak structuur en de gebruikers karakteristieken vormen samen de basis systeem eisen.

3.4 Funktionaliteits matrix

De funktionaliteits matrix ondersteunt het beslissingsproces bij het vaststellen van de produkt specificaties. Het is voor ontwerpers met weinig kennis van het probleem gebied moeilijk om goed de gebruikers en taak informatie weer te geven en om te begrijpen of de voorgestelde funktionaliteit voldoet aan de eisen.

Elke eis van de taak en gebruiker worden als eerste element van een rij weergegeven. Deze eisen worden opgesplitst in funktionaliteit, hardware eisen en software eisen. Deze funktionaliteiten worden als een kolom hoofd weergegeven. Dan wordt per vakje aangegeven of de voorgestelde funktionaliteit, indien van toepassing, voldoet aan een gebruikers of taak kenmerk met een score van zeer goed- goed- matig-slecht- zeer slecht. Door per rij van deze scores het gemiddelde te nemen, kan gekeken worden of aan een gebruikers of taak eis wordt voldaan. Zo nee, dan moet er extra funktionaliteit toegevoegd worden, dus een extra kolom. Door per kolom het gemiddelde te nemen, kan beoordeeld worden in hoeverre de funktie bijdraagt aan het produkt of niet. Verder kan per funktie nog bepaald worden hoeveel tijd, mensen, middelen en geld het kost om de funktie te realiseren of te kopen.

Met behulp van deze matrix kan vastgelegd worden welke functies wel en niet gemaakt worden en in welke volgorde en wat het aan mensen en middelen gaat kosten en hoe lang dit duurt. Voor onderzoek is het kosten en in mindere mate het tijdsaspect niet belangrijk en kan dus buiten beschouwing gelaten worden.

Opmerkingen:

De methode gaat ervan uit dat hier alle functies vastgelegd worden en dat de kosten, de mensen en de middelen al te berekenen zijn, terwijl bij het dialoog ontwerp in de volgende paragraaf pas een *eerste aanzet* tot een ontwerp gemaakt wordt. Dit *globale* ontwerp zal door prototyping aangepast worden tot een ontwerp dat aan alle eisen, die in de loop van het proces zijn ontstaan, voldoet.

Het maken van een functionaliteits matrix is naar mijn mening op dit moment in het ontwerp proces niet optimaal op te stellen omdat niet alle eisen waaraan het systeem moet voldoen bekend zijn. We zullen deze fase daarom overslaan.

3.5 Dialoog ontwerp

In deze fase gaan we stap voor stap de user interface ontwerpen. We beginnen allereerst met het kiezen van relevante stijlen en het ontwerp van een globaal overzicht en een globale structuur van de user interface. Dit ontwerp wordt steeds meer gedetailleerd, totdat er een bevredigend resultaat is ontstaan.

Een user interface zal eerder succesvol zijn als hij gemakkelijk te gebruiken is, flexibel is, duidelijk, en consistent en begeleidend. Dit dialoog ontwerp is een goed eerste ontwerp maar zal zeker uitgetoetst moeten worden door middel van prototyping.

Uiteindelijk ontstaat in deze fase een beschrijving van hoe een dialoog er uit kan zien en eventueel wordt een model gemaakt.

Het ontwerpen gaat in een aantal stappen zoals beschreven in de volgende paragrafen.

Kies dialoog stijl: De ontwikkeling van het user interface ontwerp begint bij het selecteren van relevante interface stijlen die gebaseerd zijn op gebruiker en taak informatie en, als voornaamste, op de verschillende taken die de gebruiker wil uitvoeren.

Globaal gezien zijn er acht dialoog stijlen (zie HUSAT, 1989) te onderscheiden, gebaseerd op gebruikers en taak karakteristieken, nl:

- * Question and answer (vraag en antwoord)
- * Form filling (formulier invullen)

- * Menu selection (keuze uit een menu)
- * Function keys (functie toetsen)
- * Command Language (kommando taal):
- * Query language ('vraag' taal):
- * Natural language (natuurlijke taal):
- * Interactive graphics (interactieve symbolen):

Het zal echter vaak voorkomen dat men twee of meer stijlen moet combineren om een optimale afstemming op de gebruikers en taak eisen te krijgen.

Het kiezen van de relevante stijlen voor de taak eisen doen we aan de hand van een aantal vragen. Allereerst kijken we of de komputer of de gebruiker de dialoog gaat leiden. Dit is heel belangrijk voor de dialoog stijlen die we gaan kiezen. Bij een onervaren gebruiker zal gebruik gemaakt worden van een dialoog die door de komputer geleid wordt.

Een van de vragen die gesteld worden is of de gebruiker informatie invoert. Bij een ervaren gebruiker zullen we gebruik maken van de stijl 'formulier invullen', terwijl we bij een onervaren gebruiker liever de methode 'vraag en antwoord' gebruiken om informatie in te voeren. Zo kan de dialoog beter onder controle gehouden worden.

Een andere vraag die we kunnen stellen is hoe we een gebruiker in het programma keuzes laten maken. Bij een ervaren gebruiker kunnen we gebruik maken van flexibele menu's. Voor een onervaren gebruiker kan dit wat moeilijk en zullen we eerder kiezen voor gestructureerde, enkelvoudige menu's of voor keuzes die grafisch in beeld worden gebracht waarna ze met de muis aangewezen kunnen worden.

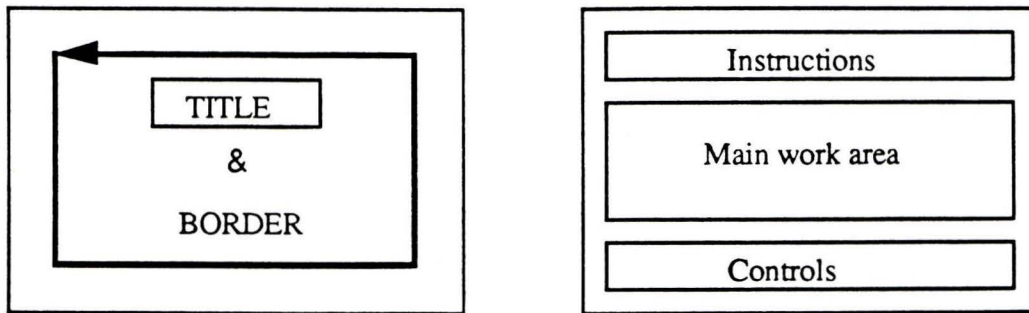
Zo proberen we alle dialoog taken te onderzoeken en daarbij een passende dialoogstijl te vinden.

Kies de user interface stijl: In deze fase van het dialoog ontwerp bekijken we wat er op het scherm zou moeten verschijnen in globale termen. Het is belangrijk dat wat de gebruiker ziet hetzelfde is als zijn of haar taak.

Als eerste kijken we of er grafische figuren gebruikt kunnen worden om gegevens in te representeren.

Een ander aspect van het dialoog uiterlijk is om in een vroeg stadium te overwegen of het gebruik van meerdere windows nodig is zodat de gebruiker meerdere sets van informatie op hetzelfde moment kan zien. Bij een onervaren gebruiker is dit niet mogelijk omdat het overzicht dan totaal verloren gaat en de dialoog ook niet meer goed en duidelijk te leiden is. Hier gebruiken we meestal een enkel scherm en geen windows. Als er wel meerdere windows gebruikt worden moeten we overwegen of er vaste windows gebruikt worden of bijvoorbeeld windows die de gebruiker zelf van grootte kan veranderen.

Een derde aspect dat we bekijken is hoe het scherm zal worden opgebouwd. Twee voorbeelden hoe een scherm opgebouwd zou kunnen worden zijn te zien in figuur 4.1.



figuur 3.1: Voorbeelden van een scherm layout

Het tweede scherm zal voor een onervaren gebruiker niet gauw worden gebruikt omdat er op een scherm drie velden staan met heel verschillende functies.

Ontwerp de dialoog structuur en de navigatie faciliteiten: Het ontwerp van de dialoog structuur zal gebaseerd worden op de lijst van functies die het systeem zal moeten vervullen. De dialoog structuur stelt de ontwerper in staat om te bedenken hoe deze functies met elkaar in verband staan en hoe de gebruiker zou willen navigeren tussen deze functies.

Dit ontwerpen gebeurt door de functies in vierkantjes te plaatsen en pijlen te gebruiken om te laten zien hoe de gebruiker kan navigeren. Hier kijken we ook naar bijvoorbeeld 'escape' of 'undo' mogelijkheden.

Verfijn het ontwerp en de stijl: Nu er een globaal overzicht en een globale structuur is ontworpen is het noodzakelijk dat de gedetailleerde aspecten van elke basis dialoog stijl worden ontwikkeld.

Gebruikers ondersteuning: Deze fase somt de verschillende soorten van ondersteuning op. Deze verschillende soorten van ondersteuning worden onderzocht aan de hand van vragen die een gebruiker bedenkt.

Er zijn verschillende soorten van ondersteuning zoals:

- * **Help:** Iedere gebruiker in een interactief systeem zal in een bepaalde fase van de dialoog in verwarring raken en zal ondersteuning vragen over hoe hij door moet gaan. Hier kan gebruik gemaakt worden van een 'help' faciliteit. Deze help faciliteit kan aangeroepen worden door bijvoorbeeld het klikken op een duidelijke herkenbare 'help' button of knop. Het help systeem moet gemakkelijk te gebruiken zijn, op ieder moment tijdens de dialoog aanwezig zijn.

- * **Feedback:** In het uitvoeren van een actie is het belangrijk dat de gebruiker meegedeeld wordt dat de actie naar tevredenheid is afgerond.
- * **Status informatie:** Lange dialoog onderbrekingen kan de gebruiker laten vergeten op welke plaats in de dialoog hij zich bevindt. In dit geval zal de gebruiker graag zijn huidige status of lokatie willen controleren.
- * **Error management:** Invoer errors kunnen in interactieve systemen op verschillende manieren verschijnen. De gebruiker kan de korrekte formulering van de invoer zijn vergeten, hij kan de invoer verkeert intypen of hij kan zelfs iets doen wat redelijkerwijs niet in de gedachten van de systeem is opgekomen. Het systeem zal zo efficiënt mogelijk tegen deze fouten afgeschermd moeten zijn.
Een aspekt van fouten afhandeling is bijvoorbeeld het nemen van voorzorgsmaatregelen. Hierbij wordt er gekeken of de invoer goed is voordat er iets mee gedaan wordt.
- * **Controle acties:** Er zijn veel dialoog situaties waarbij de gebruiker een of meerdere invoer mogelijkheden heeft die hem het verkeerde dialoog pad laten inslaan. Er wordt bijvoorbeeld het verkeerde pad in de menu hiërarchie gekozen.
Een aantal controle acties moeten dan aanwezig zijn, bijvoorbeeld in de vorm van functie toetsen. Een voorbeeld van zo'n controle actie is 'escape', deze gaat uit de huidige dialoog verschijning.

3.6 Prototyping

Een ontwerp kan nog zo mooi opgezet zijn, het blijft een papieren model met al zijn beperkingen. Een gebruiker kan een systeem eenvoudiger beoordelen als hij het gezicht van het systeem werkelijk te zien, te horen en te voelen krijgt. Dit kan gerealiseerd worden door het maken van een prototype van de dialoog. Het is belangrijk om dit zo snel mogelijk te doen, want op deze manier ontstaat bij de gebruiker een realistisch beeld van de definitieve dialogen in het systeem. Een eerste prototype kan een schets of een korte demonstratie, waarmee met de gebruiker over iets konkreets gepraat kan worden. Deze fase is een iteratie fase, die verschillende prototypes kent die langzaam aan naar het definitieve use interface groeien. Er moet goed voor ogen gehouden worden dat alleen over de user interface aspecten gepraat wordt en niet over de specificaties van het systeem. Deze fase kan resulteren in het besluit dat de informatie, die gebruikt is in eerdere stappen onvolledig of niet korrekt was en dat er dus fases over gedaan zullen moeten worden.

Een prototype komt via de volgende stappen tot stand: ontwerp, realisatie, gebruik en evaluatie. Voor prototyping is het noodzakelijk dat het ontwerp snel omgezet kan worden in een dialoog. Bovendien moeten wijzigingen snel doorgevoerd kunnen worden. Goede hulpmiddelen zijn daarbij onmisbaar, omdat een voorwaarde is dat snel een proefmodel gerealiseerd kan worden wat nauw aansluit bij de werkelijkheid. Prototyping is een iteratief proces. Het proces wordt beëindigd als de gebruiker en ontwerper tevreden zijn of als de kosten niet meer tegen de baten opwegen.

Bij het prototypen van systemen worden twee vormen onderscheiden, die ook van toepassing zijn op prototypen van dialogen. De eerste vorm, *specificatie prototyping*, of ook wel weggooi prototyping, is zuiver gericht op het opleveren van functionele specificaties op basis waarvan het definitieve systeem gebouwd wordt. Bij de tweede vorm, *evolutionaire prototyping* evolueert het prototype tot een (onderdeel van een) definitief systeem. MIR is een voorbeeld van een evolutionair prototype, terwijl de te ontwikkelen leerkracht interface prototype een specificatie prototype is.

3.6.1 Ontwerp

Het ontwerpen van een prototype begint met een studie en het uiteindelijke systeem komt via een aantal iteratie slagen tot stand. Het te snel maken van een prototype leidt tot extra iteratieslagen, of tot een prototype dat niet aan zijn doel geantwoordt. Als over de onderwerpen uit de taakanalyse geen duidelijkheid bestaat, dan moeten veronderstellingen getoetst worden aan de praktijk. In tegenstelling tot de ontwerpmethode die eerst volledige specificaties maakt komen de accenten bij prototyping anders te liggen. Bijvoorbeeld een prototype dat de voornaamste functies van het systeem kent, of een prototype gebruiken voor een nadere analyse van de taak van de gebruiker.

3.6.2 Realisatie

Vanuit de specificatie moet snel een realisatie en eventuele wijzigingen gemaakt kunnen worden. Het is belangrijk dat hiervoor goede hulpmiddelen aanwezig zijn. Afhankelijk van de gewenste functionaliteit van het prototype, zal ook een gedeelte van de applicatie gerealiseerd moeten zijn, of als prototype aanwezig moeten zijn.

3.6.3 Gebruik

Het prototype moet beoordeeld worden door de gebruiker. Hiervoor bestaan verschillende mogelijkheden:

1. de gebruiker speelt met de dialoog en levert zijn commentaar.
2. de gebruiker krijgt opdrachten, die met het systeem moeten worden uitgevoerd.
3. het systeem wordt gebruikt in de schaduw van de operationele situatie.

4. het systeem wordt ingezet in de operationele situatie.

Als niets van de eigenlijke applicatie is gerealiseerd, kan de gebruiker alleen dummy-waarden invoeren, die dan geen zinvolle reactie van het systeem opleveren. Als een beperkte functionaliteit gerealiseerd kan worden, kunnen opdrachten voor gebruik geformuleerd worden. Het derde geval duidt op het verzamelen van situaties in de bestaande omgeving waar het nieuwe systeem moet functioneren en deze naspelen met het prototype. Voor het vierde geval zal een groot gedeelte van de functionaliteit gerealiseerd moeten worden. Toepassingen met een hoge graad van interactie tussen mens en komputer kunnen alleen getoetst worden als nagenoeg alle functionaliteit gerealiseerd is. Het is aan te raden om toekomstige gebruikers, die niet bij de ontwikkeling van het systeem betrokken zijn, naar de dialoog te laten kijken. Zij moeten het terrein van de taak goed kennen.

3.6.4 Evaluatie

Een eerste oordeel van de dialoog met de user interface zal door de gebruikers worden gegeven. Zij zullen met opmerkingen komen over de bruikbaarheid van het systeem en over de overeenkomsten tussen de dialoog en hun verwachtingen. Soms is het mogelijk om de gebruiker al tijdens het gebruik zijn ideeën te laten spuien, door hem te vragen bijvoorbeeld hardop te denken. Hoe meer functionaliteit van het systeem gerealiseerd is, des te beter is de dialoog door de gebruiker te beoordelen. Hiernaast zal de ontwerper kwantitatief de bruikbaarheid van de dialoog moeten beoordelen, bijvoorbeeld hoe vaak worden gedeelten uit de dialoog gebruikt, waar worden veel fouten gemaakt.

Uit de beoordeling van de dialoog komen voorstellen voor verbetering naar voren, die leiden tot het ontwerp van een nieuw prototype of het besluit dat het prototype voldoet. Als men naar de gebruiker stapt met een nieuw prototype moeten hierin wel een aantal duidelijke veranderingen zijn aangebracht, omdat anders de gebruiker het idee krijgt dat hij beter zelf had kunnen specificeren.

Dit verslag is opgebouwd aan de hand van de HUFIT methode. De stappen zijn gevolgd en beschreven zoals in dit hoofdstuk is behandeld. Bij het ontwerpen van de user interface is er echter niet veel gebruik gemaakt van deze methode. Tijdens het ontwerpen bleek de methode te vaag en te onvolledig. Daarom is er vrij snel begonnen met het ontwikkelen van een prototype en is door het evalueren van dit prototype een bruikbaar ontwerp ontstaan.

4

Teacher

Teacher is de leraar interface die nog ontwikkeld moet worden binnen MIR. Het moet een gebruikersvriendelijke interface worden die geheel de besturing van MIR op zich neemt. De leerkracht moet het programma dat de leerlingen gaan doorlopen helemaal kunnen vaststellen en beheersen met behulp van dit interface. De resultaten worden opgeslagen en moeten later in een overzichtelijke, samenvattende vorm te bekijken zijn. Zo kan de meest effectieve combinatie van oefeningen en woordjes worden bepaald.

De user interface moet ook door de ouders van de kinderen gebruikt kunnen worden. Er zijn dan alleen een aantal functies, zoals het vergelijken van de resultaten van een leerling met de klas, die ze niet zullen gebruiken.

In dit hoofdstuk wordt het ontwerp van de user interface gemaakt, door toch de methode van hoofdstuk 3 gedeeltelijk te volgen. De problemen die bij het ontwerpen van de user interface zijn opgetreden worden beschreven aan het einde van het hoofdstuk.

4.1 Globale functionaliteit

MIR heeft voor een leerkracht twee voordelen, namelijk het genereren van een diagnose en individuele training. Met MIR kan een leerkracht een objectief beeld krijgen van de lees kennis van de leerlingen en deze vergelijken met de groep of een norm. Bepaalde leesgebreken die anders onopgemerkt zouden blijven kunnen nu gesignaleerd worden. Doordat MIR adaptief is aan de vorderingen van de leerling en doordat de leerkracht in kan stellen met welk materiaal geoefend moet worden, kan op deze manier een leerling individueel oefenen, met specifiek lesmateriaal.

De leerkracht start het systeem op dus MIR moet altijd beginnen met de leerkracht module van MIR (dit is Teacher). Teacher moet minimaal de volgende mogelijkheden hebben:

- * leerling trainingen starten: dit is het opstarten van een bepaalde oefening voor een bepaalde leerling.
- * leerlingen toevoegen en/of verwijderen: Hiermee kan de leerkracht nieuwe leerlingen toevoegen aan het leerlingbestand of oude verwijderen. Er moet dan meteen een rooster gemaakt worden per leerling en een diskette met de naam van de leerling. Deze diskette moet geformatteerd worden. Als dit fout gaat of als er nog data op de diskette staat, dan moet dit gemeld worden.
- * vorderingen tonen: Dit houdt in dat er een overzicht gemaakt moet worden van welke oefeningen een leerling gedaan heeft en met wat voor resultaat. Het gemiddelde resultaat van de rest van de groep moet ook getoond kunnen worden. Resultaten moeten in grafiek en tabel vorm weergegeven kunnen worden. De tabel moet minimaal bestaan uit de volgende gegevens: naam van de oefening, datum en tijd van de oefening, prestaties van de leerling per oefening, prestaties van de groep leerlingen en de duur van de oefening.
- * planning: Deze moet aangeven hoever de leerling is met de oefeningen en welke oefeningen nog moeten worden gemaakt. Per oefening moet ingesteld kunnen worden welke vocabulaire gebruikt moet worden, welke aanbodingsstrategie, welke feedbackvorm, hoeveel items (een item is een woordje of een grafeem) geoefend moeten worden, wat de maximale wachttijd is voor er feedback te horen is en hoe lang het scherm blanco blijft tussen het aanbieden van twee woordjes of grafemen.
- * leerstof toevoegen: De gebruikte vocabulaire moet gewijzigd en uitgebreid kunnen worden. Een leerkracht moet zelf een vocabulaire kunnen maken, eventueel specifiek voor een leerling.

4.2 User mapping

Het user interface is bedoeld voor de leerkracht en de ouders. In tweede instantie worden er functies toegevoegd voor de systeembeheerder en de demonstrateur.

In de volgende stappen zal alleen gesproken worden over de zaken die van belang zijn voor de leerkracht. Met de ouder als gebruiker wordt wel rekening gehouden.

4.3 Leerkracht en taak eisen

Om enig inzicht te krijgen in de taak van de leerkracht binnen het leesonderwijs is er een enquête opgesteld (zie appendix B). Deze enquête is opgestuurd naar twaalf basisscholen die ooit meegedaan hebben aan experimenten van het leesbord project. De response hierop was zeer slecht. Hieruit maken we op dat de leerkrachten het formulier niet in konden vullen vanwege de weinige kennis over computers en systemen in het onderwijs. Omdat we de enquête formulieren niet terug hebben gezien, is er met twee leerkrachten gepraat over de enquête en de wensen voor het systeem.

Met de volgende mogelijke karaktertrekken en opvattingen van leerkrachten voor het gebruik van de computer in de klas moet rekening worden gehouden:

- * de leerkracht wordt regelmatig in deze klassituatie gestoord door kinderen en ouders.
- * er bestaat nog enige weerstand tegen computers in het onderwijs.
- * de leerkrachten willen hun eigen leesmethode blijven gebruiken en liever niet overschakelen op een andere methode.
- * er zijn nauwelijks leesprogramma's die op de komputer werken bekend en dan wordt het moeilijk om eisen te stellen aan een systeem waar nauwelijks een beeld van gevormd kan worden.
- * de leerkracht heeft nauwelijks of nooit met de komputer gewerkt en kan dus als een onge oefend gebruiker gekenmerkt worden.
- * Het leren lezen van de kinderen gebeurt aan de hand van de leesmethode Veilig Leren Lezen die uit een aantal fases, afgesloten door toetsen, bestaat.
In de klas wordt er veel gebruik gemaakt van het leren lezen in groepjes. Deze methode wordt veel gebruikt en moet ook behouden blijven.
- * In de huidige leeslessen wordt meestal als negatief ervaren dat de kinderen niet genoeg individuele aandacht kunnen krijgen. Er zijn grote klassen en er is weinig tijd. Gedetailleerde resultaten, over bijvoorbeeld welke letters en grafemen de leerlingen beheersen, ontbreken vaak.

Doordat de leerkracht vrij onbekend is met leesprogramma's kan hij/zij moeilijk eisen formuleren. De taakeisen zijn daarom de eisen zoals beschreven in de globale functionaliteit.

De taakstructuur van het systeem zal bestaan uit vaste procedures omdat er een voorspelbaar in en uitvoer patroon is.

4.4 Funktionaliteits matrix

In dit ontwerp is geen gebruik gemaakt van deze matrix, omdat de werking en het nut van de matrix ons niet relevant leken voor het ontwerpen van de user interface. Iedere taak en eis van het systeem, zoals nu bekend is, is belangrijk en functioneel en voldoet zeker aan de gebruikers en taak kenmerken.

Verwacht wordt dat bij het evalueren van het prototype meer eisen en wensen van de gebruiker bekend worden.

4.5 Dialoog ontwerp

Bij het ontwerp van de leerkracht interface worden een aantal algemene User Interface ontwerp regels voor op windows gebaseerde interfaces (zoals die op de Macintosh aanwezig zijn) gebruikt.

Een view is een representatie van een gedachten eenheid en kan bestaan uit tekst, lijsten, of bijvoorbeeld plaatjes. De eerste ontwerp regel eist dat er slechts één taak is per window. Dit betekent dat elk window alle benodigde views moet hebben om de taak uit te voeren. Is dit niet mogelijk dan moet de taak opgesplitst worden in deeltaken.

Er moeten zo min mogelijk (verschillende) views per window te zien zijn. Dit betekent dat de taak eenvoudig moet zijn, zodat die met een minimaal aantal views beschreven kan worden. Het houdt ook in dat het scherm niet te vol moet staan en er overzichtelijk uit ziet.

De derde ontwerp regel adviseert het gebruik van standaard views. Deze views kunnen hun diensten al bewezen hebben en uitgebreid getest zijn. Verder kunnen deze standaard views in meerdere applicaties gebruikt worden, zodat een programmeur zich niet met het interface ontwerp bezig hoeft te houden en het interface over applicaties heen consistent blijft.

In de vierde ontwerp regel wordt het gebruik van enkelvoudige menu's voorgeschreven. Er wordt dus geen gebruik gemaakt van submenu's. Submenu's zijn moeilijk te bereiken en uit het zicht van de gebruiker. Verder geldt dat een menu niet meer dan zeven elementen mag bevatten.

De akties uit de menu's moeten zo veel mogelijk overeenkomen met werkwoorden en objecten uit bijvoorbeeld lijsten moeten overeenkomen met zelfstandige naamwoorden.

Als laatste ontwerp regel is het van belang dat de meest gebruikte keuzes in het zicht geplaatst moeten worden, omdat onervaren gebruikers de mogelijkheden niet kennen.

Verder geldt dat de schermverdeling goed moet zijn, kleurgebruik en verschillende types van tekst (vet, kursief enz.) beperkt moeten blijven. Het moet duidelijk leesbaar zijn en er mogen slechts twee lettertypes gebruikt worden.

Bij het ontwerpen van de user interface Teacher is met deze ontwerpregels rekening gehouden. Bij Teacher is echter geen gebruik gemaakt van standaard views.

4.5.1 Kies dialoog stijl

Omdat wij te maken hebben met een onervaren gebruiker zal de computer de dialoog gaan leiden.

Bij de user interface voor de kinderen is gebruik gemaakt van natuurlijke taal. De gebruiker is hier de leerkracht, een persoon met leesvaardigheid. Het voordeel van geschreven tekst is dat het eenvoudiger te programmeren is dan gesproken tekst. Daarom is ervoor gekozen de communicatie met de leerkracht via geschreven (nederlandse) teksten te laten verlopen. Er zal geprobeerd worden zoveel mogelijk gebruik te maken van de muis zodat de communicatie met de gebruiker zo eenvoudig en snel mogelijk kan verlopen. In sommige gevallen zal toch het toetsenbord gebruikt moeten worden bijvoorbeeld bij het intypen van een leerlingnaam.

De leerkracht zal gegevens opvragen, keuzes maken en inlichtingen vragen met behulp van de dialoogstijl 'Interactive graphics'. Zo kunnen de meeste taken die het systeem uit moet voeren gemakkelijk en snel uitgevoerd worden.

Het doorzoeken van informatie gebeurt met de zogenaamde dialoogboxen, zie figuur 5.1. De leerkracht kan de dialoogboxen met de muis bedienen. Door op de pijlen te klikken zal het veld regel voor regel gaan scrollen en kan alle informatie bekeken worden.



figuur 4.1: Dialoogbox

De gebruiker zal ook gegevens in moeten voeren. Het instellen van de agenda voor de leerlingen, het invoeren van leerlingnamen en het eventueel uitbreiden of veranderen

van de vocabulaire, de woordjes en grafemen waar de leerlingen mee werken, zal met het toetstenbord gebeuren.

4.5.2 Kies de user interface stijl

Bij deze interface zal er gebruik gemaakt worden van grafische figuren. De vorderingen van de leerlingen kunnen bekeken worden in tabelvorm en in grafiekvorm.

Als user interface stijl is er gekozen voor een scherm dat een titel draagt en een vierkant bevat waar mee gewerkt wordt en waar alle informatie in te vinden is. De opzet van het scherm is afgebeeld in figuur 5.2.



figuur 4.2: scherm lay-out van Teacher

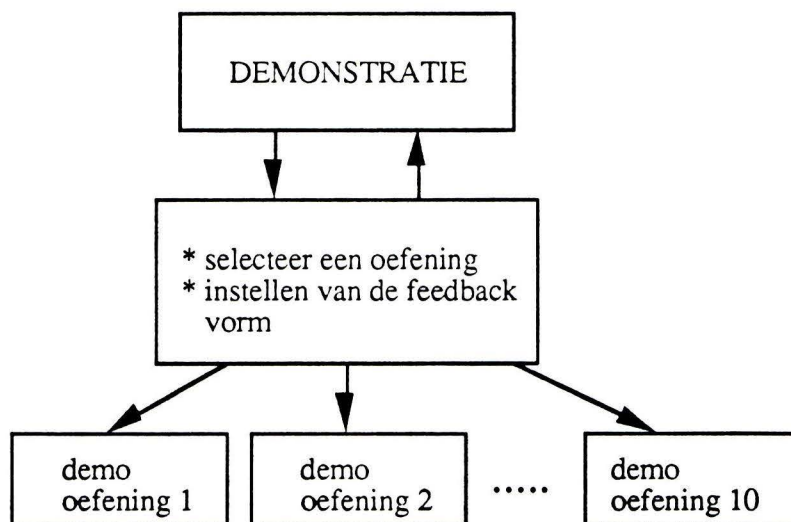
Er is gekozen om geen gebruik te maken van meerdere windows aangezien de leerkracht onervaren is, en dat er uit onderzoeken is gebleken dat dit verwarring met zich mee kan brengen.

4.5.3 Ontwerp dialoog structuur en navigatie faciliteiten

Teacher heeft een aantal hoofdtaken, namelijk het starten van trainingen, een demonstratie van de oefeningen, het bijhouden van de klasselijst, het bijhouden van een agenda en het bekijken van de vorderingen. De navigatie tussen deze taken en binnen de taken zullen in de volgende paragrafen beschreven worden. De vierkanten stellen de taken voor en de functies van de schermen binnen de hoofdtaken. De pijlen geven aan hoe tussen de schermen genavigeerd kan worden.

Demonstratie

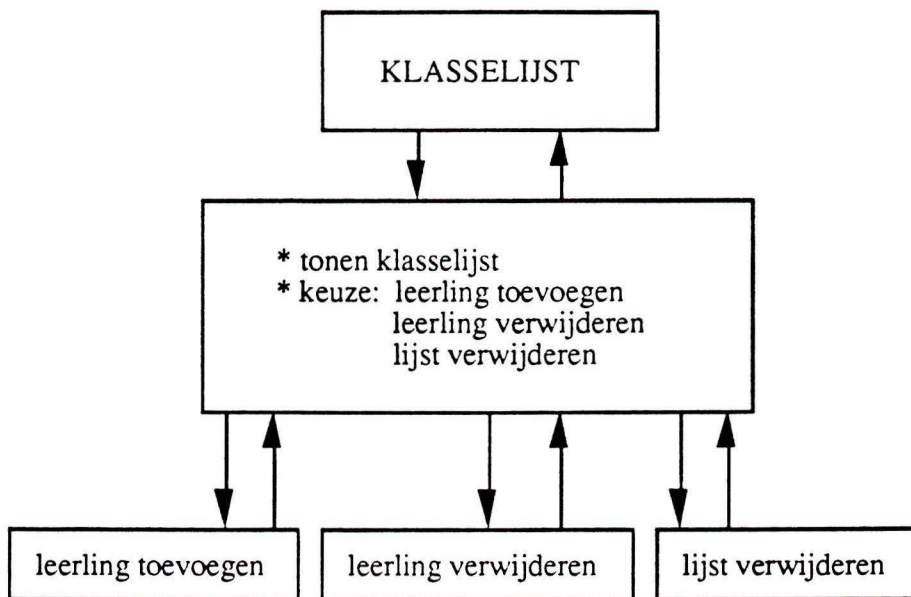
Binnen de hoofdtaak demonstratie moet de leerkracht de mogelijkheid krijgen om een keuze te maken tussen de oefeningen waar hij een korte demonstratie van wil bekijken. Deze functie is gewenst omdat het mogelijk is dat mensen die met het programma gaan werken de oefeningen die het programma gebruikt nog niet kennen. Bovendien dient het als extra faciliteit en als geheugensteun tijdens het werken met het programma. In figuur 5.3 zijn de functies en de navigatie faciliteiten van de hoofdtaak demonstratie in model gebracht.



figuur 4.3: Demonstratie

Klasselijst

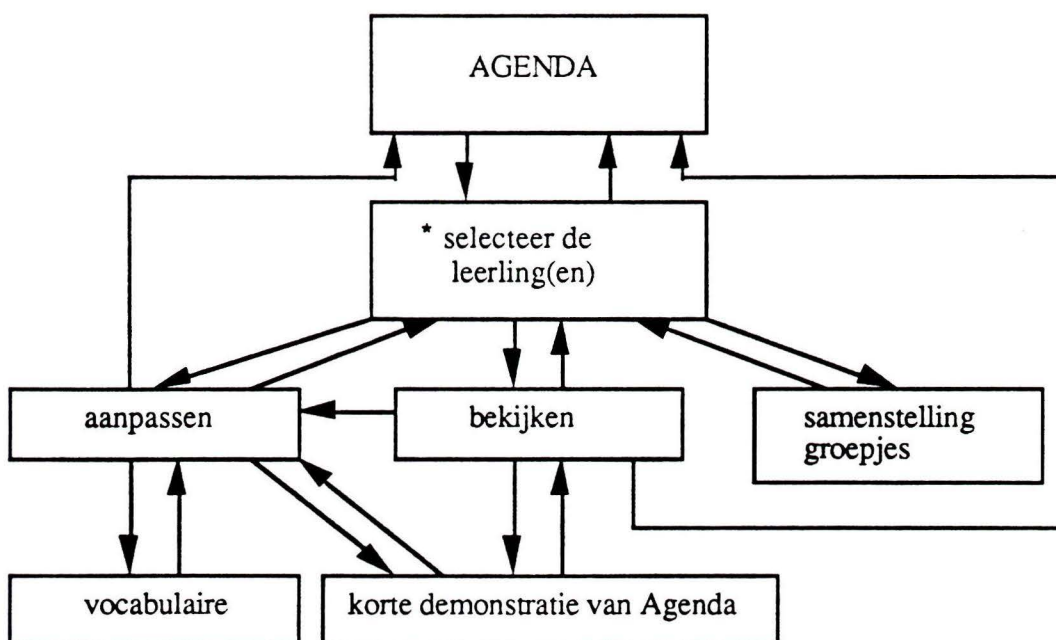
De hoofdtaak klasselijst heeft vier functies. Allereerst kan bekeken worden welke leerlingen met het programma werken. Dit zijn de leerlingen die in de klasselijst zijn opgenomen. Het verwijderen van een leerling is een functie en zo ook het toevoegen van een leerling. Het verwijderen van de gehele klasselijst is de laatste functie en zal waarschijnlijk aan het einde van een schooljaar gebruikt worden. In figuur 5.4 zijn de navigatie mogelijkheden getekend tussen de verschillende functies van de hoofdtaak klasselijst.



figuur 4.4: Klasselijst

Agenda

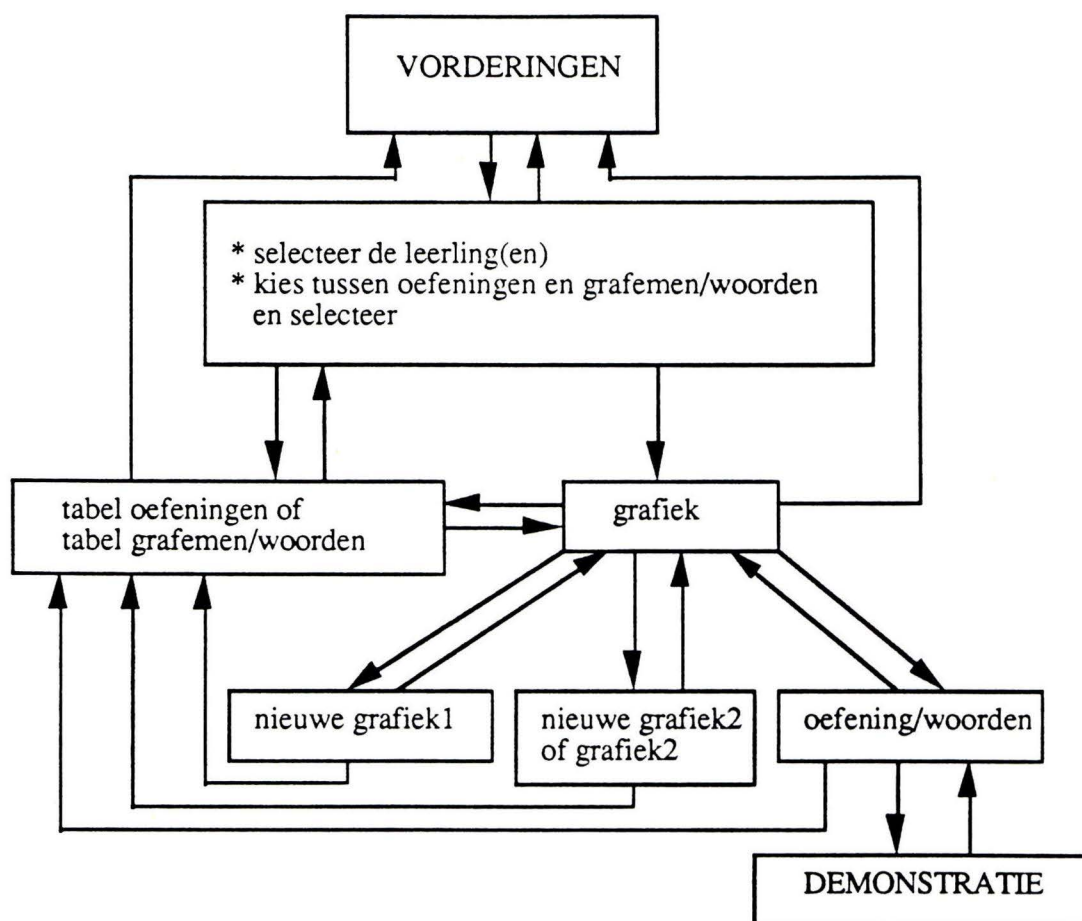
Agenda heeft als functie het bekijken en aanpassen van de agenda van een leerling of van een groep leerlingen. Tijdens het aanpassen van de agenda kan de leerkracht de vocabulaires bekijken. Dit zijn de kernen zoals die bekend zijn bij de methode Veilig Leren Lezen (zie appendix A). Deze vocabulaires kunnen aangepast en veranderd worden, tevens kan er een geheel nieuwe vocabulaire worden samengesteld.



figuur 4.5: Agenda

Vorderingen

De hoofdtak vorderingen heeft als taak het tonen van samenvattende resultaten van de leerlingen uit de klasselijst. De vorderingen van oefeningen kunnen bekeken worden en de vorderingen van grafemen en woorden. De leerkracht selecteert allereerst de leerling(en) waarvan hij/zij de vorderingen wil bekijken. Er moet een keuze gemaakt worden of er vorderingen bekeken worden van oefeningen of van woorden en grafemen. Vervolgens wordt er een selectie gemaakt uit bijvoorbeeld de oefeningen. Daarna kan de tabel of grafiek bekeken worden van deze selectie.



figuur 4.6: Vorderingen

Als voor tabel is gekozen heeft de leerkracht de mogelijkheid om weer terug te gaan naar vorderingen of de grafiek te bekijken. Nieuwe leerlingen en woorden/grafemen of oefeningen kunnen geselecteerd worden waarvan de tabel of de grafiek te bekijken is.

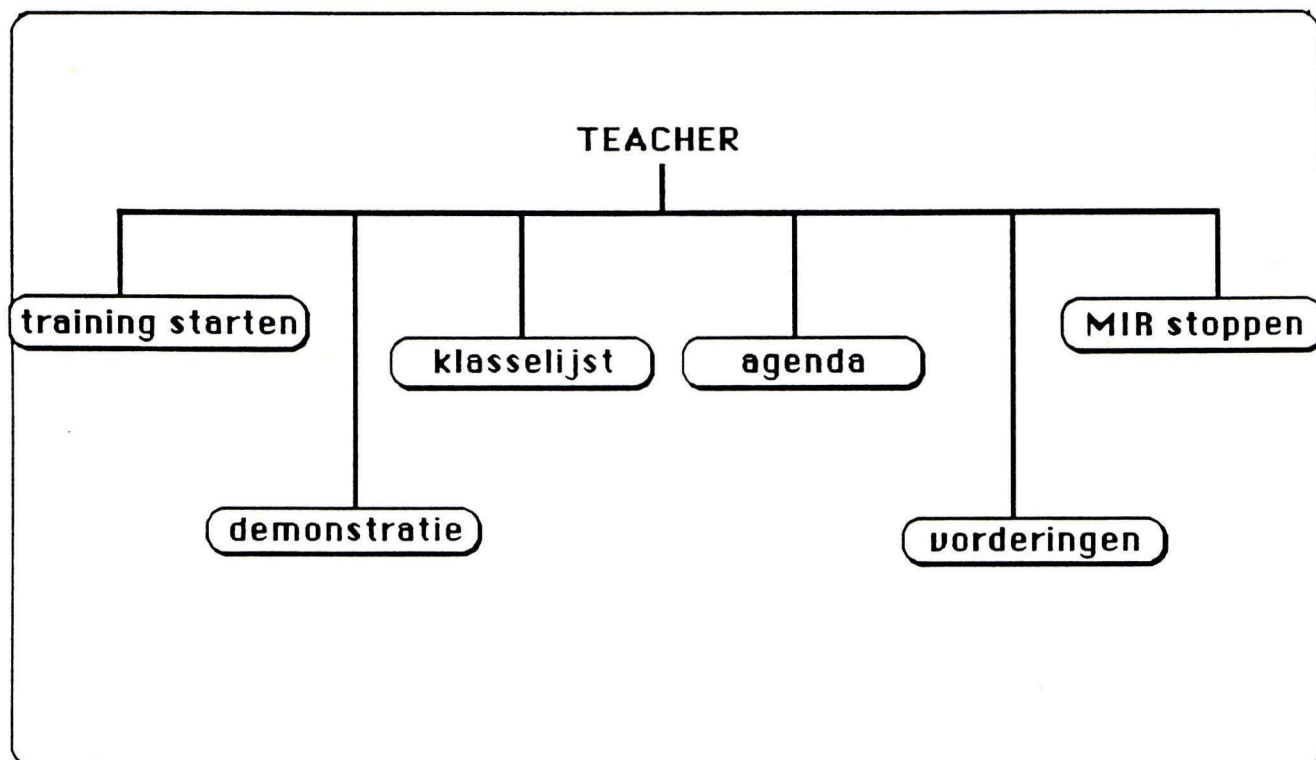
Als er voor de grafiek is gekozen heeft de leerkracht meerdere mogelijkheden. De tabel kan bekeken worden, of er kan een nieuwe grafiek gemaakt worden, dit betekent dat er een nieuwe groep leerlingen geselecteerd kan worden.

Een tweede grafiek kan erbij gemaakt worden zodat de resultaten van leerlingen met elkaar vergeleken kunnen worden. Gewenst is bijvoorbeeld om de vorderingen van een leerling ten opzichte van de klas te bekijken. Als er al een tweede grafiek bestaat kan er een nieuwe groep leerlingen geselecteerd worden voor deze tweede grafiek. Tevens kan de grafiek verwijderd worden.

Een laatste mogelijkheid is om de woordjes en de oefeningen te veranderen. Als de eerste keuze is geweest het bekijken van de resultaten van oefeningen, kan er later voor gekozen worden om de resultaten van woordjes en grafemen te bekijken. Dit kan altijd veranderd worden.

4.5.4 Verfijn het ontwerp en de stijl

Bij het opstarten van MIR wordt direkt het eerste scherm van Teacher getoond. Dit scherm toont de hoofdtaken van het programma zoals figuur 5.7 aangeeft.



figuur 4.7: Het startscherm van Teacher

De ovalen zijn de zogenaamde buttons. Als met de muis op deze button geklikt wordt gaat het programma naar het eerste scherm van deze hoofdtak. De buttons zijn de instrumenten waarmee genavigeerd wordt tussen de taken en de verschillende functies.

4.5.4.1 Training starten

Mir moet beginnen met de leerkracht module Teacher, als er een diskette met Teacher in zit. Deze start het systeem op. Dit opstarten van het systeem kan door op de button "training starten" te klikken. Het systeem zal dan met de oefeningen beginnen zoals beschreven in Agenda.

4.5.4.2 Demonstratie

Vanuit het hoofdscherm is er de mogelijkheid om de oefeningen die in het programma zitten te bekijken. Gebruikers die niet bekend zijn met het programma

kunnen zo snel zien hoe het programma eruit ziet. Ook als een gebruiker niet meer precies weet hoe een oefening eruit ziet kan hij/zij even kijken.

demonstratie

Selecteer een oefening voor een demonstratie:

wrdwrldmatch
wrdpicmatch
completion
picwrldmatch
audana
spel

De oefening maakt gebruik van de feedbackvorm:

None
Partial
Complete
Extensive

informatie over feedback

teacher

figuur 4.8: Demonstratie

Er kan een oefening gekozen worden door op de naam van de oefening te klikken. Een korte demonstratie van de oefening, zoals de leerlingen deze te zien krijgen, zal dan te zien en te horen zijn. Na de demonstratie komt de gebruiker automatisch terug bij het demonstratiescherm. Op dit moment is alleen de oefening Spel op de juiste manier te zien. Voor de andere oefeningen is er een voorbeeldscherm met schriftelijke informatie ontworpen. Dit zal niet geïmplementeerd worden.

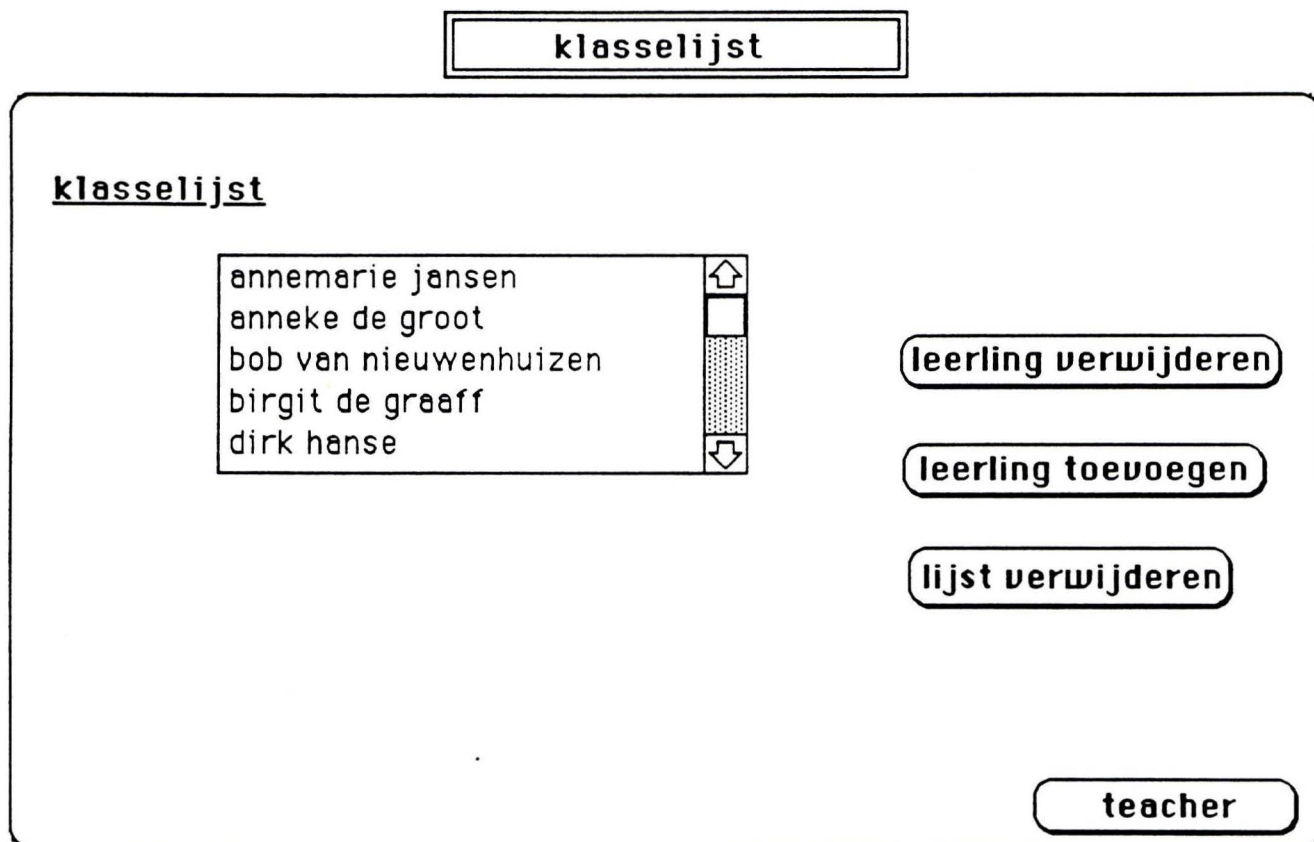
In het demonstratiescherm kan ook de feedbackvorm ingesteld worden. Voordat de oefening wordt gekozen moet deze feedbackvorm ingesteld worden. De demonstratie van de oefening zal dan gebruik maken van de ingestelde feedbackvorm. Als er geen feedbackvorm gekozen is wordt automatisch de feedbackvorm 'none' genomen. In figuur 5.8 is de feedbackvorm 'Extensive' anders van kleur en dat betekent dat deze feedbackvorm geselecteerd is.

De button 'informatie over feedback' geeft uitgebreide schriftelijke informatie over de vier feedbackvormen. De gebruiker die onbekend is met het programma kan eerst rustig bekijken wat een feedbackvorm inhoudt alvorens hem te selecteren en te beluisteren.

De button 'teacher' brengt de gebruiker weer terug naar het startscherm zoals beschreven in de vorige paragraaf.

4.5.4.3 Klasselijst

De hoofdtak klasselijst bevat mogelijkheden om een klasselijst bij te houden. Aan deze lijst kunnen nieuwe leerlingen toegevoegd worden en er kunnen ook leerlingen verwijderd worden. De lijst kan ook in z'n geheel verwijderd worden, wat aan het einde van een schooljaar heel praktisch kan zijn. Bij alle drie de mogelijkheden is een cancel mogelijkheid ingebouwd.



figuur 4.9: Klasselijst

Met de button 'leerling verwijderen' kan een leerling verwijderd worden. De leerling die verwijderd moet worden, wordt eerst geselecteerd door met de muis op de naam van de leerling te klikken. De naam van de geselecteerde leerling zal dan van kleur veranderen zodat de gebruiker kan zien wat er geselecteerd is. Daarna kan de leerling verwijderd worden.

Met de button 'leerling toevoegen' kan de naam van een leerling toegevoegd worden. Het toevoegen van de naam van de leerling gebeurt door de button 'leerling toevoegen' te gebruiken. De gebruiker krijgt de gelegenheid om de naam van de leerling in te typen. Het toevoegen gebeurt nu door op 'OK' te klikken. Als de gebruiker helemaal geen gebruik wil maken van toevoegen gebruikt hij 'cancel', er wordt nu geen naam toegevoegd.

Het verwijderen van de lijst kan met de button 'lijst verwijderen'. Ook hier is weer een 'cancel' mogelijkheid aanwezig zodat er terug gegaan kan worden zonder dat de lijst verwijderd wordt.

De button 'teacher' brengt de gebruiker weer terug naar het startscherm.

4.5.4.4 Agenda

De oefeningen die de leerlingen moeten doorlopen staan beschreven in de agenda. Er bestaat een agenda voor de hele groep leerlingen. Maar ieder kind kan ook individueel een agenda hebben.

De gegevens die in de agenda staan moeten allemaal aan te passen zijn door de leerkracht.

Allereerst moet de leerkracht de naam van de leerling of van de leerlingen selecteren (zie figuur 5.10). Als meerdere leerlingen geselecteerd worden moeten deze leerlingen wel dezelfde agenda hebben. Met de button 'samenstelling groepjes' kan bekeken worden welke groepjes van leerlingen met dezelfde agenda op dat moment aanwezig zijn. Als er leerlingen geselecteerd worden die niet dezelfde agenda hebben komt er een melding. Deze melding houdt in dat de groepjes leerlingen getoond worden en dat na klikken met de muis op een groepje, dit groepjes automatisch wordt geselecteerd. Nu kan er geen fout optreden bij het selecteren van groepjes van leerlingen. De leerkracht kan nieuwe groepjes van leerlingen samenstellen, aan de hand van de vorderingen van de leerlingen, door de leerlingen apart een nieuwe agenda te geven. De leerlingen met dezelfde agenda vormen dan een nieuw groepje.

De agenda van de geselecteerde leerling(en) kan nu worden bekeken of worden aangepast. Het aanpassen van de agenda gebeurt door het intypen van de nieuwe instelling. Deze instelling kan bepaalde waarden hebben. Als een verkeerde instelling wordt ingetypt zal dit gemeld worden.

agenda

selecteer de leerling(en) waarvan u de agenda wilt bekijken/aanpassen:

Annemieke Jansen

Anneke de Groot

Bob van Nieuwenhuizen

Birgit de Graaff

Dirk Hanse

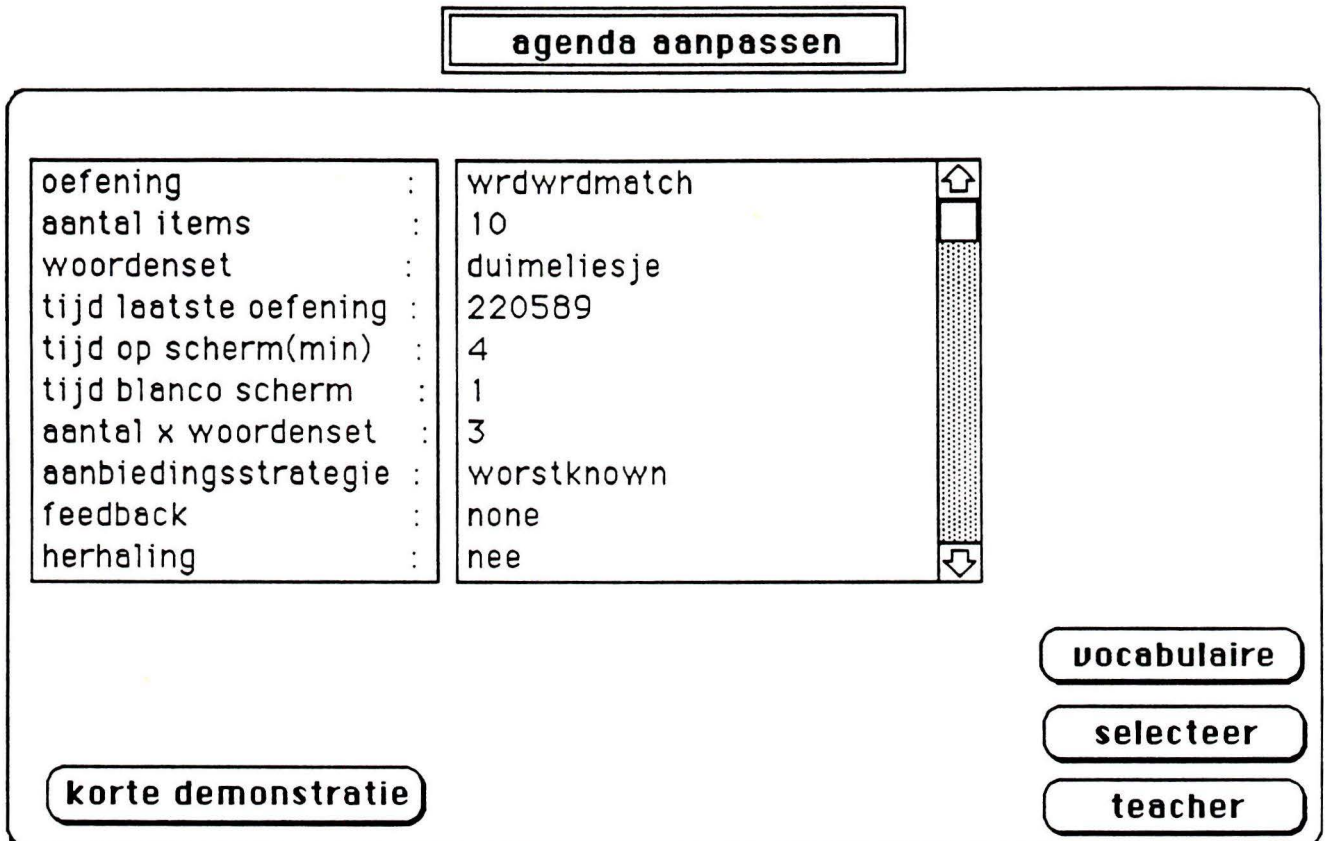
samenstelling groepjes

figuur 4.10: Agenda

De gegevens die in agenda staan en die aangepast kunnen worden zijn:

1. Oefening: hier kan de naam van de oefening worden ingevuld.
2. Aantal items: dit zijn het aantal items die in deze oefening geoefend moeten worden.
3. Woordenset: hier kan de naam van de woordenset ingevuld worden. De oefening maakt dan gebruik van de woordjes die in deze woordenset staan.
4. Tijd laatste oefening: hier komt de datum en de tijd dat de leerling(en) deze oefening voor het laatst hebben gedaan.
5. Tijd op scherm: de tijd dat woordjes of letters op het scherm blijven staan. Dit moet ingevuld worden in seconden.
6. Tijd blanco scherm: tijd dat het scherm blank blijft voordat een item wordt aangeboden (in seconden).
7. Aantal x woordenset: aantal keer dat de woordenset nog geoefend moet worden.
8. Herhaling: Na de tweede oefening kan bij herhaling 'ja' worden ingevuld. Als dit gebeurt dan komt er na de oefening een herhaling van alle vorige oefeningen. Deze oefeningen worden dan door elkaar aangeboden. Dit als extra oefening voor leerlingen die bijvoorbeeld moeite hebben met lezen.

In de toekomst zullen alle gegevens die in de file Agenda staan, zoals beschreven in [Pol,1989], aangepast moeten kunnen worden. Omdat deze gegevens niet allemaal op het scherm kunnen is het het makkelijkste om een veld te maken met deze gegevens erin die je kunt scrollen zoals te zien is in figuur 5.11 waar het rechtse veld met de instellingen gescrold kan worden.



figuur 4.11: Aanpassen agenda

Er moet een mogelijkheid gemaakt worden om een korte demonstratie te bekijken van de agenda. Dit is nu nog niet mogelijk, maar in de toekomst moet er een korte demonstratie komen van de oefeningen in de volgorde zoals ze in de agenda zijn vastgelegd. De oefeningen maken dan ook gebruik van de andere instellingen, zoals onder andere feedback en woorden set. Via de button 'selecteer' kan de gebruiker weer terug naar het vorige window om eventueel agenda's van andere leerlingen te bekijken.

Vanuit het scherm waar de agenda wordt aangepast kan de gebruiker de mogelijkheden voor een vocabulaire, dit is een woordenset, bekijken. Tevens kan de gebruiker zelf een vocabulaire opstellen. Het bekijken van de vocabulaire kan door op de naam van een vocabulaire te klikken (zie figuur 5.11). De vocabulaire komt dan in beeld tesamen met de buttons "verander" en "terug". Het veld waar de woordjes en letters in staan kan niet zomaar worden veranderd. Als de gebruiker de vocabulaire echt

wil gaan veranderen of woorden toevoegen dan klikt hij/zij op de button "verander". Het is nu mogelijk om de woorden van de vocabulaire te veranderen en woorden toe te voegen. Als de gebruiker nu weer terug gaat naar het vorige scherm wordt de nieuwe vocabulaire weggeschreven onder dezelfde naam. De gebruiker kan nu alleen de oorspronkelijke vocabulaire terug krijgen door de gewijzigde vocabulaire weer terug te veranderen. In de toekomst moet hier misschien een oplossing voor worden gevonden zodat de gebruiker in ieder geval altijd de beschikking heeft over de oorspronkelijke vocabulaires, die opgesteld zijn door de methode Veilig Leren Lezen.

Het maken van een eigen vocabulaire is ook mogelijk. Hier kan nu alles worden ingevuld. Later moet het mogelijk zijn om eventueel suggesties te geven hoe een nieuwe vocabulaire voor een bepaalde leerling eruit zou kunnen zien. Misschien moet het geven van een mogelijke woordenset waaruit dan een nieuwe vocabulaire kan worden samengesteld ook tot de mogelijkheden behoren. Hierover is nog niet nagedacht.

vocabulaire

Selecteer de vocabulaire die u voor het aanpassen van de agenda wil bekijken en/of wijzigen:

Duimeliesje
De School
De Dierentuin
Spel en Speelgoed
Sneeuwitje
Naar het bos
In de keuken
Klein Duimpje

Zelf vocabulaire samenstellen:

Vocabulaire opstellen

Agenda

figuur 4.12: Vocabulaire

4.5.4.5 Vorderingen

Deze hoofdtaak geeft de gebruiker de gelegenheid om naar de resultaten van een leerling of een groep leerlingen te kijken.

Allereerst wordt de leerling of de groep leerlingen geselecteerd waarvan de resultaten getoond moeten worden. Dit gebeurt weer door het klikken met de muis op de naam van de leerling. Deze naam zal van kleur veranderen zodat duidelijk zichtbaar is dat die betreffende leerling geselecteerd is. Er kunnen meerdere leerlingen geselecteerd worden door met de muis meerdere namen aan te klikken. Om een selectie ongedaan te maken klikt de gebruiker opnieuw op de naam van de leerling. De naam zal weer zijn oorspronkelijke kleur krijgen en de selectie is ongedaan gemaakt.

Vervolgens wordt er een keuze gemaakt of de resultaten van oefeningen of van woordjes en grafemen getoond moeten worden. Deze keuze kan gemaakt worden door op het rondje of de naam te klikken van een van deze mogelijkheden. Daarna kunnen de oefeningen of woordjes en grafemen geselecteerd worden waarvan de vorderingen in tabel of grafiek getoond moeten worden. In figuur 5.12 is te zien dat er gekozen is voor vorderingen van de oefeningen. Er kunnen nu oefeningen geselecteerd worden. Als er gekozen is voor woorden en grafemen kunnen geen oefeningen geselecteerd worden.

vorderingen

selecteer de leerling(en):

	annemarie jansen anneke de groot bob van nieuwenhuizen birgit de graaff dirk hanse	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
--	--	---

maak uw keuze:

oefeningen
 grafemen/woorden

en selecteer.

	wrdwrddmatch picwrddmatch wrdpicmatch audwrddmatch completion	<input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
--	---	---

tabel
grafiek
teacher

figuur 4.13: Vorderingen

Een keuze kan gemaakt worden uit het bekijken van de grafiek en het bekijken van de tabel. Als voor de tabel wordt gekozen komt de tabel in beeld. Er is dan de mogelijkheid om weer terug te keren naar het startscherm van vorderingen zoals getoond in figuur 5.12. Een nieuwe groep leerlingen en/of oefeningen en woorden kan geselecteerd worden en de tabel kan weer bekeken worden. Een voorbeeld van de tabel van woorden en grafemen is te zien in figuur 5.13.

De tabel bevat de woordjes en grafemen die door de gebruiker geselecteerd zijn. Van twee leerlingen of twee groepen leerlingen kunnen de resultaten in tabelvorm en grafisch getoond worden. Zo kunnen de resultaten van leerlingen met elkaar vergeleken worden. De tabel met de resultaten van de woordjes bevat andere gegevens. Allereerst worden de oefeningen die geselecteerd zijn vermeld. Per oefening en per grafiek worden de datum dat de oefening voor het laatst geoefend is, de effectiviteit en de duur van de oefening vermeld.

Resultaten van woorden en grafemen						
woorden/grafemen	% goed beantwoord		% fout beantwoord		% niet beantwoord	
	grafiek1	grafiek2	grafiek1	grafiek2	grafiek1	grafiek2
boom	87	86	12	14	1	3
roos	67	65	21	19	12	15
aan	34	66	0
aap	89	7	4
boor	65	18	17
beuk	49	51	0
buik	45	53	2
..	

grafiek
vorderingen
teacher

figuur 4.14: De tabel van woorden en grafemen

Als er in het startscherm van vorderingen gekozen wordt voor de grafiek dan kan de gebruiker niet meer terug naar het startscherm. Er is voor gekozen om verder alle keuzes vanuit de grafiek te laten plaatsvinden. In figuur 5.13 is de button 'vorderingen' aanwezig. Deze is aanwezig als vanuit vorderingen de tabel wordt gekozen. Als vanuit

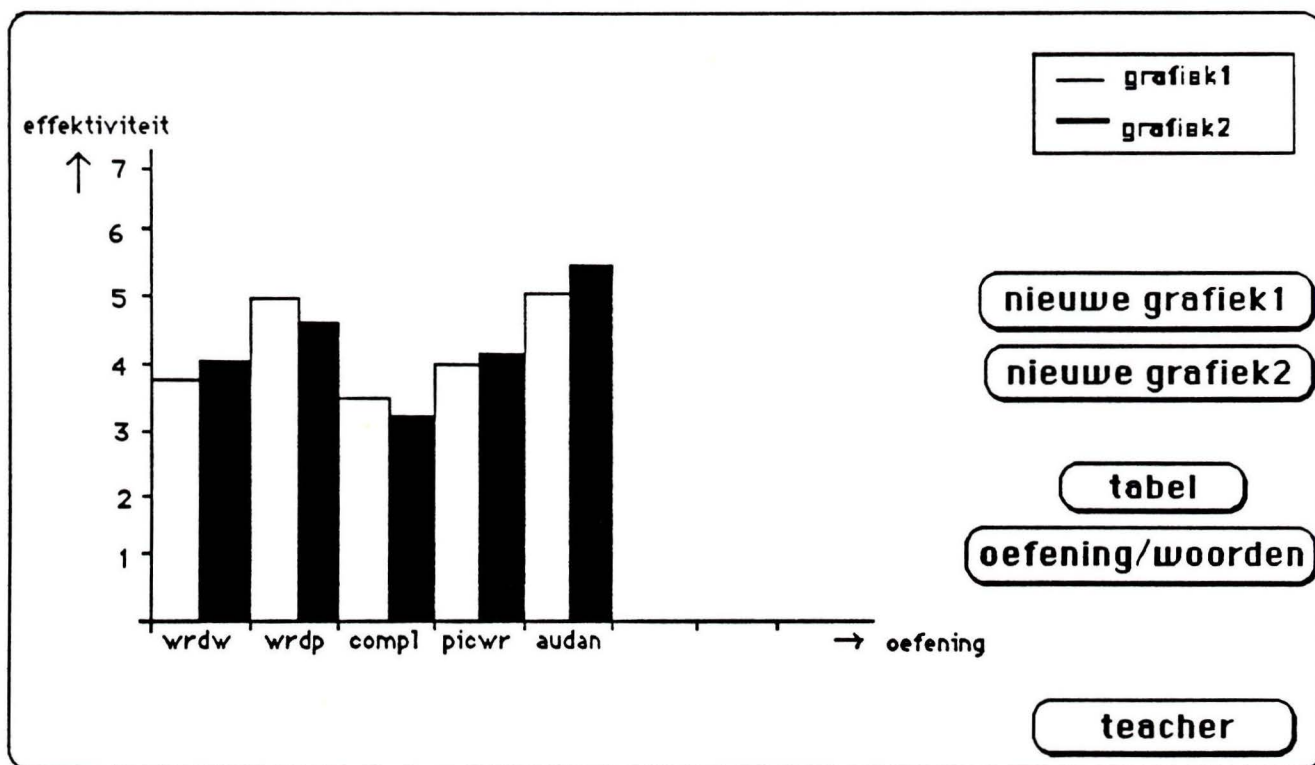
de grafiek de tabel wordt gekozen is deze button niet meer zichtbaar omdat er dan niet meer teruggegaan kan worden naar het startscherm.

Als de keuze op grafiek valt krijgt de gebruiker de resultaten te zien van de leerling(en) die geselecteerd zijn in grafische vorm. Vanuit het scherm waar de grafiek getoond wordt heeft de gebruiker verschillende mogelijkheden.

Met de button 'nieuwe grafiek1' kan de gebruiker een nieuwe groep leerlingen of een nieuwe leerling selecteren om de resultaten van te bekijken.

De button 'grafiek2' geeft de gebruiker de mogelijkheid om een tweede groep leerlingen of een tweede leerling te selecteren. Zo kunnen twee grafieken met elkaar vergeleken worden. De vorderingen van een leerling ten opzichte van de klas kan bijvoorbeeld bekeken worden. Als er een tweede grafiek is getekend verandert de button 'grafiek2' in 'nieuwe grafiek2', zoals in figuur 5.14 getoond wordt. De selectie van een bepaalde groep leerlingen kan ongedaan gemaakt worden en een nieuwe groep kan geselecteerd worden. Zo kunnen alle combinaties van vorderingen van leerlingen en groepen leerlingen in tabelvorm of grafisch getoond worden. Tevens kan deze tweede grafiek verwijderd worden zodat alleen de eerste grafiek zichtbaar is.

Er is gekozen voor een staafdiagram omdat gebleken is, door experimenten met proefpersonen van het IPO, dat dit het duidelijkste overkomt bij de gebruiker.

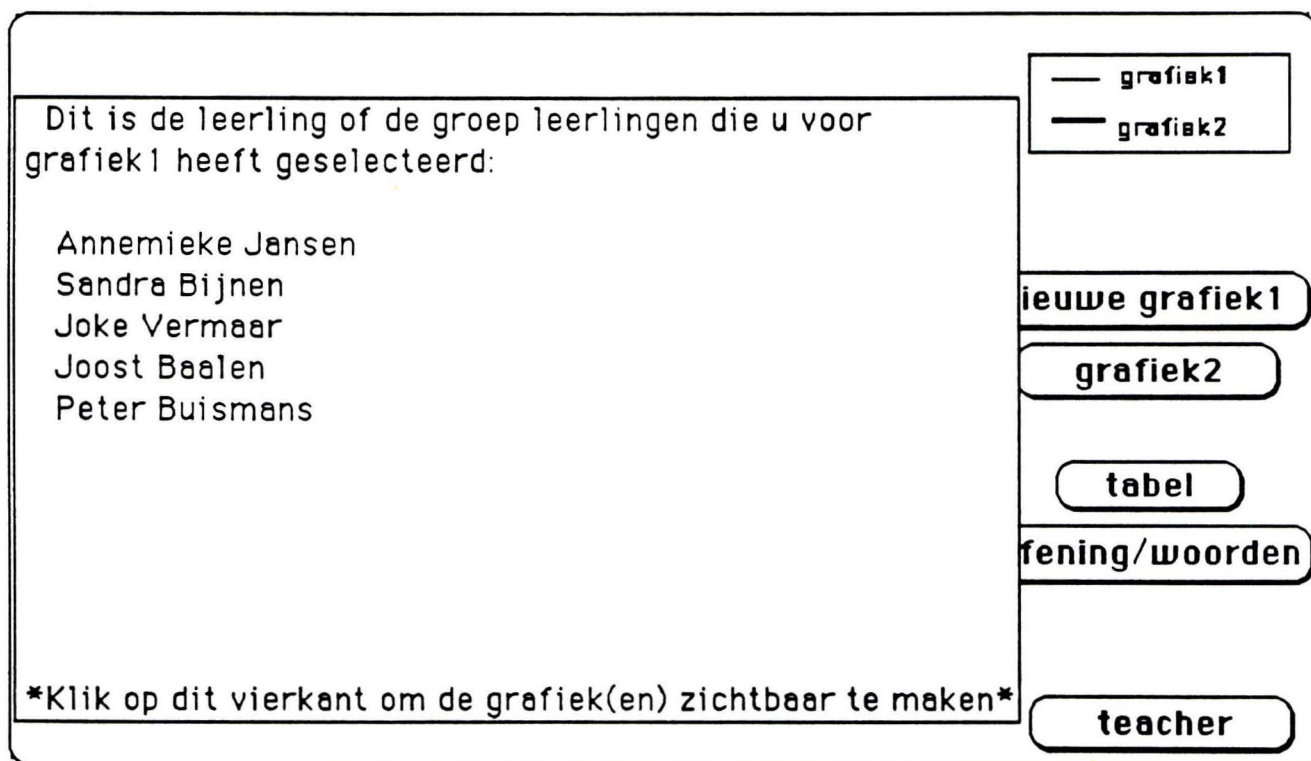


figuur 4.15: Grafiek

Via de button 'tabel' kan de tabel bekeken worden van de twee grafieken.

De button 'oefening/woorden' geeft de gebruiker de mogelijkheid om te wisselen van oefening naar woorden en grafemen of een nieuwe combinatie van oefeningen of van woorden en grafemen te maken. De oude keuzes zijn nog zichtbaar. Het selecteren gaat hetzelfde als bij het startscherm van vorderingen.

Rechtsboven in het scherm van grafiek staat een vierkantje met 'grafiek1' en 'grafiek2' erin. Als er met de muis op 'grafiek1' wordt geklikt zal er een veld verschijnen waarop de groep leerlingen staat beschreven die voor deze grafiek zijn geselecteerd, zoals te zien is in figuur 5.15. Door op het veld te klikken waar de namen in staan verdwijnt het veld en zijn de grafieken weer zichtbaar. Voor grafiek 2 bestaat dezelfde mogelijkheid, nu door op 'grafiek2' in het vierkantje te klikken.



figuur 4.16: De groep geselecteerde leerlingen voor grafiek 1

4.5.5 Gebruikers ondersteuning

Omdat we met een onervaren gebruiker te maken hebben is het heel belangrijk dat er een goede ondersteuning is. De verschillende soorten ondersteuning zullen in de volgende paragrafen worden besproken.

4.5.5.1 Help

Als help faciliteit is er gekozen voor een duidelijke en herkenbare button die op ieder moment van de dialoog aanwezig is.

De helpbutton is altijd rechts boven in het scherm zichtbaar. Als er met de muis op de button wordt geklikt zal er een helpscherm zichtbaar worden. Dit helpscherm geeft informatie over het informatiescherm waar het bijhoort. De mogelijkheden die de gebruiker heeft in dit informatiescherm worden besproken. Dit betekent dat de werking van de buttons die op dat scherm aanwezig zijn uitgelegd wordt.

Met de button 'terug' die op ieder helpscherm aanwezig is kan de gebruiker terug naar het informatiescherm waar hij/zij vandaan kwam.

Het startscherm van Teacher bevat nog een extra help faciliteit. Het scherm dat via de button 'instructie' te zien is, geeft globale informatie over de opbouw van het interface en legt het begrip 'button' uit aan de gebruiker.

4.5.5.2 Feedback

In dit programma moeten er onder andere namen van leerlingen geselecteerd worden. Als een naam geselecteerd is moet dit onmiddellijk te zien zijn. Dit gebeurt door het veranderen van de kleur van de naam die geselecteerd wordt, zie figuur 5.16. In figuur a is de naam niet geselecteerd, in figuur b is er wel een naam geselecteerd.



figuur 4.17: Een geselecteerde naam en een ongeselecteerde naam

Het selecteren van oefeningen en woorden en grafemen gebeurt op dezelfde manier als het selecteren van de namen van de leerlingen.

Als een button aangeklikt wordt zal deze oplichten waarna het nieuwe scherm zichtbaar wordt. Zo kan de gebruiker zien dat hij de button heeft aangeklikt en is het verschijnen van een ander scherm geen verrassing.

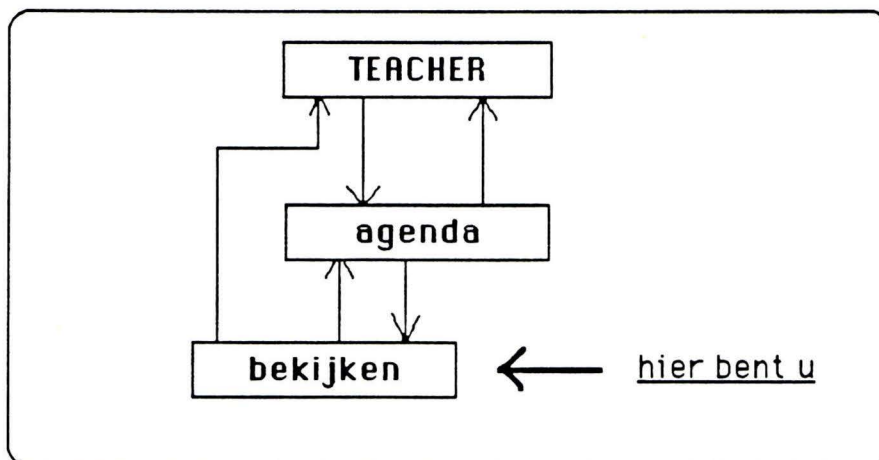
De feedback die gebruikt moet worden bij het veranderen van de agenda is de zogenaamde dynamische feedback. Dit betekent dat als er iets nieuws wordt ingetypt dit onmiddellijk zichtbaar is. Hetzelfde gebeurt bij het wijzigen en aanvullen van de vocabulaires.

4.5.5.3 Status informatie

Omdat een leerkracht in de klas gestoord kan worden door ouders en leerlingen is het belangrijk dat op ieder moment in de dialoog de plaats in de dialoog waar de leerkracht zich bevindt bekend is.

Er gekozen voor een 'route' button die altijd links boven op het scherm te zien is. Via deze button komt de gebruiker bij een scherm zoals te zien is in figuur 5.16.

Het routescherm bevat vierkantjes die de schermen representeren die de gebruiker gepasseerd heeft. De pijlen geven aan hoe de navigatie is tussen de schermen.



figuur 4.18: Een routescherm

De leerkracht kan naar elk scherm terug door met de muis op de naam van het scherm te klikken waar hij/zij naar toe wil.

4.5.5.4 Error management

Bij het invoeren van een nieuwe agenda zal gebruik gemaakt worden van het nemen van voorzorgmaatregelen. Hierbij wordt gekeken of de invoer goed is ingevoerd voordat er iets mee gedaan wordt.

Bij het veranderen van de vocabulaire en het maken van een nieuwe vocabulaire zal voorlopig niet naar fouten worden gekeken.

4.5.5.5 Control akties

In dit programma zullen geen control akties aanwezig zijn. Als de gebruiker een verkeert pad is ingeslagen kan hij via het routescherm altijd weer terug naar de plaats waar hij vandaan kwam om zo een juiste keuze te maken. De dialoog is zo eenvoudig opgezet dat controle akties niet nodig zijn.

4.6 Prototyping

Er is besloten om al in een vroeg stadium een prototype te maken. Het was de bedoeling dit te doen aan de hand van de uitkomsten van de enquête. Zoals reeds vermeld is hier niet veel uitgekomen.

Twee leerkrachten zijn daarna bezocht om de enquête samen in te vullen. Er is geprobeerd om ideeën te krijgen over de wensen dit de leerkracht ten opzichte van het systeem zou hebben. De wensen en de ideeën over het systeem waren minimaal. Dit is naar mijn mening te wijten aan het feit dat de leerkrachten geen enkele voorstelling kunnen maken van een systeem dat kinderen leert lezen, omdat er in het onderwijs nog nauwelijks systemen en computers gebruikt worden. Zou dit wel het geval zijn dan zijn er vaak voordelen en nadelen aan het systeem dat ze gebruiken, tenzij dit het perfecte systeem is maar hier gaan we niet van uit.

Vrij vlug is begonnen met het ontwikkelen van het eerste prototype aan de hand van de gegevens die op dat moment bekend waren. Het prototype is gerealiseerd met Hypercard 1.2.2, een Apple Macintosh applicatie. Na het veelvuldig bespreken van het prototype en het invullen van de ideeën die aanwezig waren over taken en functies van het systeem ben ik met het prototype naar de twee leerkrachten gegaan waar we contact mee hebben.

Het prototype moest beoordeeld worden door de leerkracht te laten spelen met de dialoog en kommentaar te laten leveren over onduidelijkheden en functies die gemist werden of die juist overbodig waren.

De leerkrachten, overdonderd door het resultaat, vonden alles duidelijk en mooi. Er kwamen geen reacties over functies die overbodig waren of gemist werden. Dit is te verklaren door het feit dat er nog helemaal geen vergelijkende systemen op de markt zijn en zeker nog niet in het onderwijs gebruikt worden. Dit prototype is het eerste wat ze op dit gebied te zien krijgen. Wij kunnen dus niet van hen verwachten dat ze uitgesproken ideeën hebben over zo'n systeem. Over het algemeen vonden ze het heel interessant en wilden ze alles weten over het idee achter het systeem en of, en wanneer het op school gebruikt moest gaan worden.

Om toch de bruikbaarheid van het systeem te testen hebben een aantal medewerkers van het IPO als proefpersoon gefungeerd. Hier zijn een aantal nuttige ideeën uit voortgekomen. Tevens zijn een aantal onduidelijkheden naar voren gekomen die aangepast zijn. Uiteindelijk is het prototype ontstaan zoals beschreven in de vorige paragrafen.

5

Implementatie

Dit hoofdstuk bevat de beschrijving van een deel van de implementatie van de userinterface Teacher. Allereerst zal de file structuur van MIR worden besproken.

Er is gekozen om de hoofdtak 'vorderingen' van Teacher te implementeren. De hoofdtaken 'training starten', 'demonstratie' en 'MIR stoppen' zullen niet zoveel problemen opleveren. De hoofdtaken 'agenda' en 'vorderingen' zitten wat ingewikkelder in elkaar en deze taken gebruiken ook de uitvoer van MIR. Om een volgende afstudeerster een idee te geven hoe de interface moet worden geïmplementeerd is er gekozen voor het implementeren van 'vorderingen' in pascal. Later zal de userinterface Teacher in object georiënteerd pascal worden geïmplementeerd, waarin MIR ook is geschreven.

5.1 File structuur van MIR

Omdat Teacher de uitvoer van MIR gebruikt is het heel belangrijk te weten hoe de file structuur van MIR er uit ziet. Binnen MIR zijn er twee soorten files te onderscheiden, namelijk op klas nivo en op leerling niveau. De files die per klas aanwezig zijn bevatten alle data van de gehele klas, zoals een overzicht van wat de hele klas heeft gedaan. De data per leerling is specifiek op de leerling gericht, zoals woordmateriaal dat specifiek voor een leerling is samengesteld of een overzicht van de resultaten van een sessie van die leerling.

Elke leerling heeft een diskette met daarop zijn of haar specifieke files. De resultaten en Summary files worden op harddisk en op de diskette opgeslagen. De Users en Groupresult files staan op harddisk en een backup om de leerkracht diskette. In de volgende tabel wordt een overzicht gegeven van welke files op harddisk en welke op diskette staan.

	Vocabulary	Agenda	Summary	Result	Users	Groupresult
Klas	Hd I/O	Hd I/O	-	-	Hd I	Hd O
Leerling	d I/O	d I/O	Hd+d O	Hd+d O	-	-

figuur 5.1: Overzicht files van MIR

'Hd' betekent dat de file op harddisk staat en 'd' dat de file op diskette staat. De 'I' en de 'O' geven aan of het een in- of uitvoer file is. De files zijn opgeslagen als 'text' files zodat er nog in veranderd kan worden.

In de volgende paragrafen worden de files van MIR globaal besproken. Tevens worden er aanbevelingen gedaan hoe de files het beste door de user interface Teacher gebruikt kunnen worden. Voor een gedetailleerde beschrijving van de verschillende files verwijs ik naar [Pol, 1989].

5.1.1 De file Vocabulary

Per klas is er een file waarin een vocabulaire staat. Eventueel kan er voor een leerling ook een specifieke file worden gemaakt. Uit deze file worden de items voor de oefening gekozen. Per item zijn er een aantal regels en de items worden van elkaar gescheiden door een lege regel. Items moeten op een regel gescheiden worden door een spatie en/of komma en/of tabs.

Deze file zal door Teacher gebruikt gaan worden binnen de hoofdfunctie Agenda. Hier worden bij de oefeningen de namen van de kernen, zoals die bekend zijn in de methode Veilig Leren Lezen, geselecteerd die door de oefeningen worden gebruikt.

Aanbevelingen

Voor elke kern uit Veilig Leren Lezen zal er bij voorkeur een aparte file moeten worden gemaakt met de items, de spraak en de andere gegevens. Deze files worden dan in Agenda geselecteerd. Via de userinterface Teacher kunnen deze files aangepast worden. Als alle kernen in één file staan is het moeilijk om vanuit Teacher een kern van globaalwoorden te selecteren en te veranderen.

Deze file staat op harddisk en op het schijfje van de leerling. Als we de vocabulaire van de klas willen veranderen hebben we de file op de harddisk nodig. Willen we de vocabulaire van een leerling veranderen hebben we de diskette van de leerling nodig. Het is dus niet mogelijk om de vocabulaire van een geselecteerd groepje leerlingen te veranderen. Tot nu toe is dit niet gewenst. Als dit in de toekomst wel nodig is zullen de specifieke files van de leerlingen ook op harddisk opgeslagen moeten worden, zodat de files van een groepje leerlingen kunnen worden geselecteerd en veranderd.

5.1.2 De file Agenda

Per klas wordt een default agenda bijgehouden. In deze agenda staan de volgorde van de oefeningen, welke woordensets er worden gebruikt en de andere instellingen die mogelijk zijn. Deze file wordt ook door Teacher gebruikt en aangepast en wel in de hoofdtak Agenda. Het is de bedoeling dat *alles* wat in deze file staat aangepast kan worden. Op dit moment is er in de user interface Teacher de mogelijkheid om een aantal instellingen te doen.

Aanbevelingen

De agenda van de klas staat op de harddisk en kan daarop aangepast worden. De specifieke files van de leerlingen staan op de diskettes van de leerlingen. Het is ook hier niet mogelijk om de agenda van een groepje leerlingen aan te passen. Op dit moment is dit wel gewenst. De leerkracht wil groepjes kunnen vormen aan de hand van de agenda's.

Het is aan te bevelen om deze specifieke files van de leerlingen ook op harddisk te zetten. Dit kan gebeuren door de agenda's allemaal in één file te zetten. De agenda's vormen dan records in deze file met als sleutel het leerlingnummer. Dit levert wel wat zoekwerk op. De files kunnen ook allemaal apart opgeslagen worden. Het nadeel hiervan is dat bij het veranderen van de agenda's de files steeds geopend en gesloten moeten worden wat ook wel wat tijd kost. Wat de snelste en beste oplossing is zal de toekomst uit moeten wijzen. Verder kan de inhoud van de file zonder problemen worden gebruikt door Teacher.

5.1.3 De file Users

Deze file is een invoerfile van MIR en wordt gebruikt voor het koppelen van leerlingnamen en nummers. In deze file staan per leerling 3 regels: de eerste regel bevat de naam, de tweede regel het unieke leerlingnummer en de derde regel de konditie. Deze laatste regel wordt alleen gebruikt voor de experimenten en zal in de toekomst verdwijnen. De nummering van de leerlingen is doorlopend. Dus als er iemand uitvalt, wordt niet de opgevallen plaats opgevuld, maar achteraan toegevoegd.

Deze file kan gebruikt worden als klasselijst. Deze file zal dus beschikbaar moeten zijn voor Teacher. Bij het tonen van de klasselijst worden de namen van de leerlingen uit deze file gelezen en in alfabetische volgorde op het scherm gezet. Als er in de user interface een naam wordt geselecteerd kan deze opgezocht worden in deze file en kan het leerlingnummer erbij worden gezocht. Met behulp van dit leerlingnummer kunnen bijvoorbeeld de files met de resultaten van de leerling worden opgezocht. Een andere mogelijkheid is om de file Users in het begin van het programma in te lezen in een record met de opvolgende leerlingennamen. De unieke leerlingnummers moeten dan tevens worden opgeslagen in een record, zodat op ieder moment bekend is welke

leerlingen welk nummer hebben. In de file staan de leerlingen niet op alfabetische volgorde omdat als er een leerling wordt toegevoegd deze aan het einde van de file wordt toegevoegd.

5.1.4 De file Groupresult

Per sessie worden er een aantal gegevens bijgehouden van de hele klas. Deze file is zonder problemen te gebruiken door Teacher. De file zal gebruikt worden voor het bekijken van de resultaten van de oefeningen van de hele groep die met het programma werkt. Deze file is dus niet te gebruiken als er maar één leerling of een aantal leerlingen is geselecteerd.

In deze file staan de resultaten van de oefeningen per sessie. Dit betekent dat een oefening meerdere keren in deze file kan voorkomen. Om nu de resultaten van de oefeningen te laten zien bij de hoofdtak vorderingen, kan gekozen worden voor de gemiddelde effectiviteit van de gekozen oefening(en) of voor de effectiviteit van de laatste sessie, of nog een andere combinatie. Hierover is nog niets besloten maar het meest voor de hand liggende op dit moment vind ik het nemen van de gemiddelde effectiviteit van alle sessies van de geselecteerde oefeningen. Bij het ontwerpen van Teacher is er vanuit gegaan dat de gemiddelde effectiviteit van de sessies wordt genomen.

De file is op dit moment geschikt om te gebruiken voor Teacher.

5.1.5 De file Summary

Deze file bevat de resultaten per leerling per sessie. Deze file is op harddisk en op diskette te vinden. De inhoud van de file is hetzelfde als de file Groupresult.

Bij de hoofdtak 'vorderingen' kunnen deze files gebruikt worden om de resultaten van de oefeningen te bekijken. De files van de leerlingen die geselecteerd worden, worden geopend en gelezen. Per oefening kan nu de effectiviteit worden berekend voor de geselecteerde leerlingen. Zijn er meerdere leerlingen geselecteerd dan kan de effectiviteit van een oefening worden onthouden en gemiddeld met de effectiviteit van de oefening van de andere leerlingen. Deze files moeten dan op harddisk worden opgezocht.

5.1.6 De file Result

De resultaten van woorden en grafemen zijn te vinden in deze file. De file staat op diskette en op harddisk en heeft de naam van de leerling aangevuld met een sessienummer.

In de hoofdtaak 'vorderingen' zal er van deze files gebruikt worden gemaakt. De resultaten van de woorden en grafemen kunnen dan in deze files per leerling worden opgezocht. Dit levert wel heel wat zoekwerk op. Als bijvoorbeeld de resultaten van een bepaalde leerling van het grafeem 'ch' moeten worden gezocht, moeten alle files van deze bewuste leerling worden geopend en worden doorzocht.

De file is zonder problemen door de user interface Teacher te gebruiken.

Aanbevelingen:

In de toekomst is het misschien makkelijker om een file per leerling aan te leggen waar de resultaten per letter of grafeem in staan. Dit betekent dat na elke sessie de resultaten worden verwerkt in deze aparte file, zodat op ieder moment de resultaten per letter/grafeem van de gedane sessies bekend is.

5.2 Werkwijze

Bij het implementeren van een gedeelte van de user interface hebben we gebruik gemaakt van het software pakket 'Extender' (zie Juni, Feldt & Surovell, *Programmer's Extender volume 1 (1985)*). Dit software pakket geeft de gebruiker de mogelijkheid om gemakkelijk en flexibel te werken met de Macintosh interface. Er bestaan standaard procedures voor het maken van windows, dialoogboxen en buttons, zoals die in Teacher worden gebruikt.

Allereerst zijn de windows gedefinieerd met de buttons die bij die windows behoren. Vervolgens hebben we de windows door middel van de buttons aan elkaar gekoppeld. De tekst is op het scherm geplaatst en de dialoogboxen zijn werkend gemaakt.

Voor een listing van het programma met het bijbehorende commentaar verwijs ik naar appendix C.

Voor het bekijken van de ontworpen user interface kan gekeken worden in de folder Teacher met daarin de stack Teacher. Deze stack bevat de user interface die ontworpen is met behulp van het software pakket Hypercard (Goodman (1987)).

De plaats van de buttons en de velden met de tekst kunnen met behulp van dit pakket opgevraagd worden. Zo kan de stack precies geïmplementeerd worden zoals hij in Hypercard is beschreven.

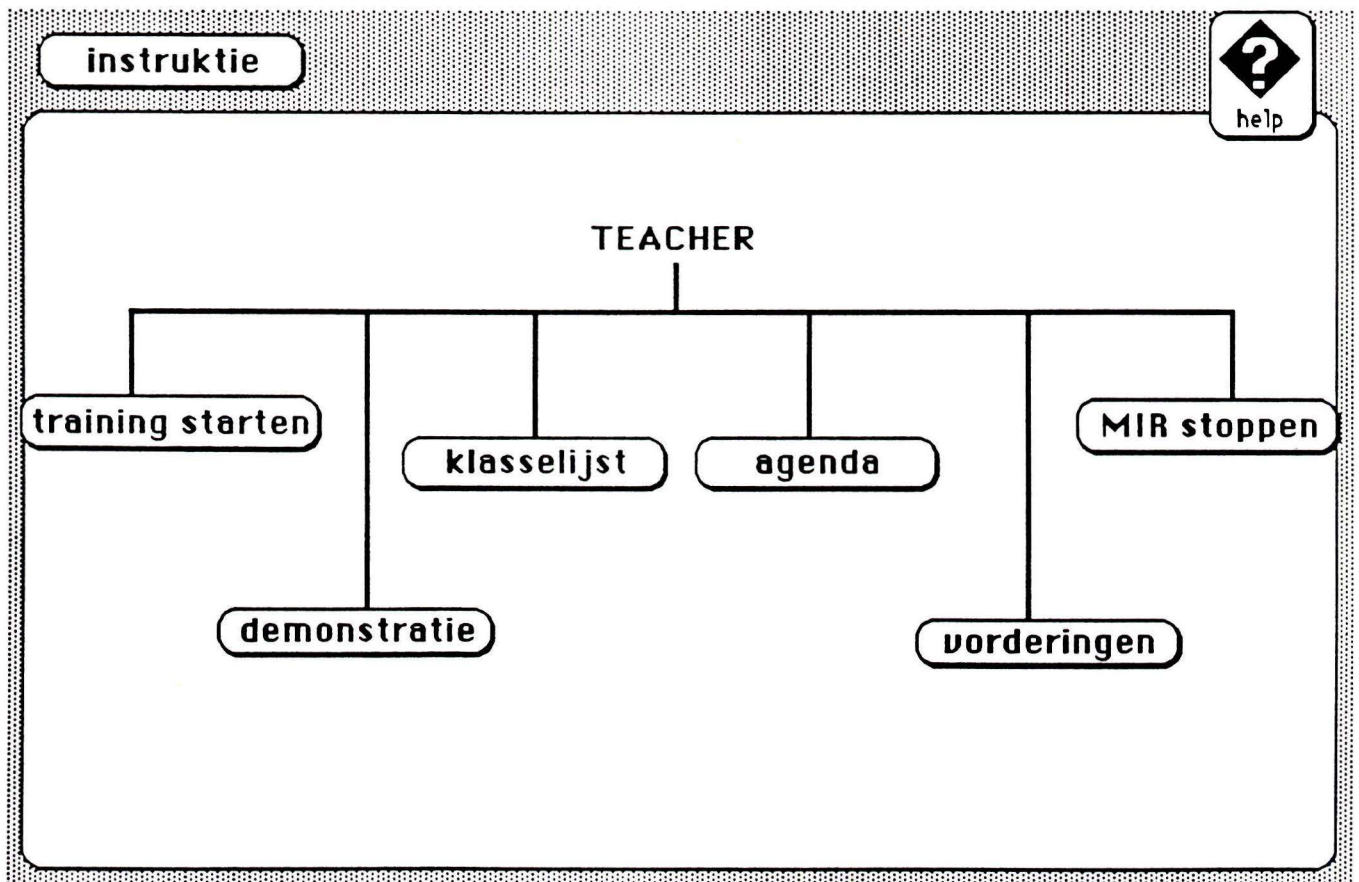
5.3 Wat is er geïmplementeerd

Op dit moment (mei 1990) is de hoofdtaak vorderingen gedeeltelijk geïmplementeerd. De schermen van vorderingen zijn gereed en tevens kunnen de buttons op deze schermen worden gebruikt. De dialoogboxen zijn gevuld met de namen van de leerlingen, de oefeningen en de woorden en grafemen.

Het selecteren van de leerlingen in het startscherm van vorderingen is op dit moment ook mogelijk. De files met de data worden echter nog niet ingelezen. In de listing van het programma is wel aangegeven waar deze files zullen moeten gelezen. Het lezen van de files zal niet veel problemen opleveren. Ook het berekenen van gemiddelden en het vullen van de tabel en de grafiek zullen niet veel problemen opleveren, dit zal slechts wat rekenwerk vereisen voor de plaats van de data op het scherm.

De helpbutton en de routebutton zijn op het scherm te zien maar er kan nog niet mee gewerkt worden. Het implementeren van de helpbutton zal geen problemen opleveren. De buttons in het routescherm kunnen nieuwe schermen oproepen. Dit werkt hetzelfde als bij de buttons die op dit moment geïmplementeerd zijn.

Op dit moment is het startscherm gereed zoals te zien is in figuur 5.1. De button vorderingen kan gebruikt worden om naar het eerste scherm van 'vorderingen' te gaan. Door de button 'MIR stoppen' te gebruiken stopt het programma en gaat terug naar waar het opgestart is.



figuur 5.2: Startscherm van Teacher

Het eerste scherm van vorderingen is te zien zoals beschreven in hoofdstuk 4, zie figuur 5.12. De buttons 'route' en 'help' zijn daarbij ook geïmplementeerd en hetzelfde stippeltjes patroon als in figuur 5.1 is daarbij aanwezig. Deze buttons en het patroon zullen standaard bij elk scherm te vinden zijn.

In het startscherm van vorderingen kunnen de velden gescrolld worden. Dit gaat op dit moment nog niet naam voor naam. Als er op een pijl geklikt wordt dan verplaatst het veld vaak twee of drie plaatsen tegelijk. Dit moet verbeterd worden zodat in de toekomst het veld per regel kan worden verschoven.

Als de button 'oefening' is geselecteerd dan is er een veld te zien met oefeningen en als de button 'grafemen/woorden' is geselecteerd dan bestaat het veld uit woorden en grafemen. De button 'tabel' gaat nu afhankelijk van deze buttons naar de tabel van de oefeningen of naar de tabel van de woorden en grafemen. De tabel van de woorden en grafemen zoals die nu geïmplementeerd is te zien in figuur 5.2.

Resultaten van woorden en grafemen						
woorden/grafemen	⌘ goed beantwoord		⌘ fout beantwoord		⌘ niet beantwoord	
	grafiek1	grafiek2	grafiek1	grafiek2	grafiek1	grafiek2

figuur 5.3: De tabel van woorden en grafemen

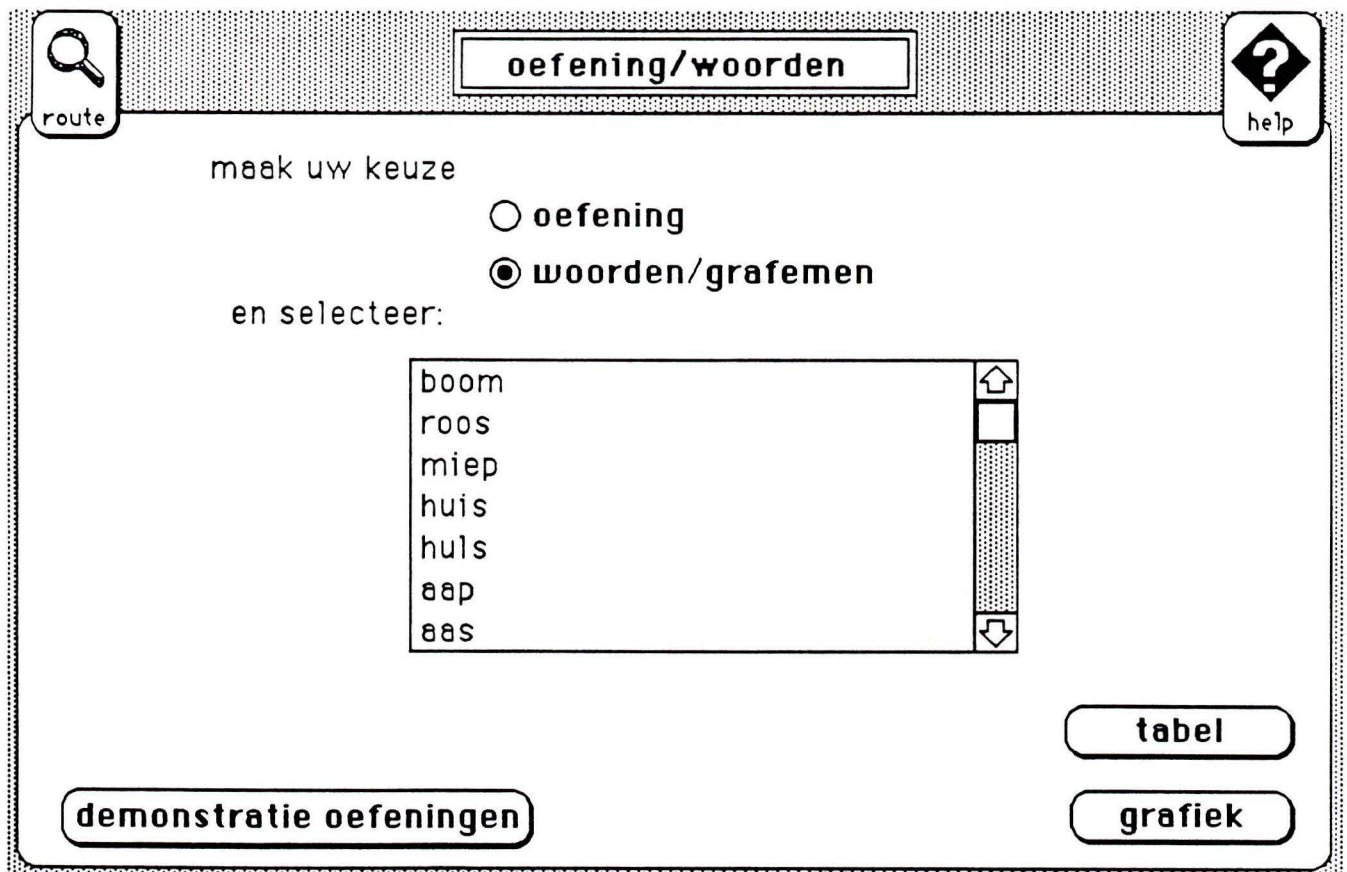
De tabel van oefeningen ziet eruit zoals beschreven in hoofdstuk 4. De gegevens kunnen nog niet worden ingevuld. Dit kan pas gebeuren als er via de dialogboxen

leerlingnamen en oefeningen en woorden kunnen worden geselecteerd.

Het window van de grafiek is geïmplementeerd zoals beschreven in hoofdstuk 4. Ook hier zijn de resultaten nog niet zichtbaar te maken. De buttons rechtsboven 'grafiek1' en 'grafiek2' zijn nog niet te gebruiken. Dit zal echter niet veel problemen opleveren. Als deze buttons worden aangeklikt moet er een window zichtbaar gemaakt worden met de namen van de leerlingen die geselecteerd zijn voor de betreffende grafiek. Door met de muis te detecteren waar er in dit nieuwe window geklikt is kan het window weer onzichtbaar gemaakt worden.

Het scherm waarin een nieuwe groep leerlingen kan worden geselecteerd voor een grafiek is helemaal geïmplementeerd.

Voor het laatste scherm van vorderingen geldt hetzelfde als voor de vorige schermen. De buttons werken en ook het selecteren via de dialogboxen is mogelijk, maar het inlezen van de files is nog niet geïmplementeerd (zie figuur 5.3).



figuur 5.4: Het scherm voor het selecteren van oefeningen en woorden

6

Eindrapportage

Op dit moment is er een prototype van de user interface Teacher gereed. Een gedeelte van dit prototype is in pascal geïmplementeerd. Deze user interface zal later in object pascal geschreven worden. Dit heeft als voordeel dat er bij veranderingen van de user interface niet zoveel aan het programma veranderd hoeft te worden, en het is een stuk overzichtelijker dan in pascal. Het nadeel van object geïmplementeerd pascal is dat het nog niet zo bekend is en dat er op een andere manier gedacht moet worden dan bij het programmeren in 'gewoon' pascal.

Als dit prototype geïmplementeerd is kan er geëxperimenteerd worden met de user interface Teacher, en kan de interface ontwikkeld worden tot een bruikbaar en gebruikersvriendelijk produkt.

Tot slot wil ik de verrichte werkzaamheden evalueren en de problemen die bij het ontwerpen van de user interface zijn ontstaan bespreken.

6.1 Verrichte werkzaamheden

De user interface Teacher die ontwikkeld is door [Drunen, 1985] zou in het najaar van 1989 op school gebruikt worden. Dit is helaas niet gebeurd, zodat ik van vooraf aan ben begonnen met het ontwikkelen van de interface.

Na het bestuderen van de omgeving waarin de user interface ontworpen moest worden, ben ik begonnen met het opstellen van de eisen die aan het programma worden gesteld. Dit kostte nogal wat tijd omdat er veel onduidelijkheid is en was over de concrete eisen. Op het IPO is het een en ander bekend over de functies die het systeem zou moeten bevatten en er werd verwacht dat door enquête's en interviews meer ideeën naar voren zouden komen. Helaas bleek dit niet het geval te zijn. Toen er een aantal weken waren verstreken zijn we zelf begonnen met het bedenken van functies voor het systeem met de eigen kennis die we hadden over de werkomgeving en de gebruiker.

Langzaam is de user interface ontwikkeld met behulp van het software pakket Hypercard, wat perfect het uiterlijk weer kan geven van de interface die de gebruiker te zien krijgt. Er kan de gebruiker een zeer realistisch beeld worden gegeven van de user interface. Dit is een enorm voordeel boven het praten over tekeningen en schetsen van de interface. Dit brengt toch vaak verwarring met zich mee omdat er veel mondeling zal moeten worden uitgelegd. Dit kan altijd verkeerd overkomen bij de gebruiker waardoor de kans bestaat dat er een verkeerd beeld wordt gevormd en dat er langs elkaar heen gepraat wordt.

Vervolgens is het prototype geëvalueerd met de gebruiker en proefpersonen van het IPO.

Een gedeelte van de user interface is geïmplementeerd. Dit als steun voor de stagiaire of afstudeerder die het prototype gaat implementeren.

Als laatste wil ik graag nog aanbevelen om de namen van de buttons en de oefeningen te bekijken en eventueel te veranderen. Naar mijn idee zijn de namen niet allemaal zo duidelijk, zeker de namen van de oefeningen niet.

6.2 Problemen

Het grootste probleem in het begin van de afstudeerperiode waren de gebruikers, de leerkrachten. Het was een groot probleem informatie te verzamelen over de functie van het systeem in het onderwijs. Dit is te verklaren dankzij het feit dat er nog nauwelijks computers gebruikt worden in het onderwijs. De leerkrachten hebben geen enkel idee wat de bedoeling is van het systeem en vragen zich voortdurend af wat de makers van het systeem nu precies willen bereiken. Verder bestaat er nog een angst voor computers. De leerkrachten zijn bang overbodig te worden of ze zijn gewoon 'tegen' computers. Er wordt veel gepraat over computers in het onderwijs in het algemeen. De leerkrachten onderling zijn het nog niet eens of er nu wel computers in het onderwijs gebruikt moeten worden of dat de oude vertrouwde methode toch het beste kan blijven. Zolang de gebruiker zelf nog in niet weet of hij de computer wil toelaten tot het onderwijs en er nog geen andere software op de markt is waar het systeem mee vergeleken kan worden zal het niet optimaal ontwikkeld kunnen worden.

Ik denk wel dat als het systeem straks in de experimentele fase is en op school gebruikt wordt dat dan de wensen van de leerkrachten naar voren komen. Het is belangrijk dat het systeem zo optimaal mogelijk aan de wensen voldoet, anders bestaat de kans dat het systeem wordt gekocht en in een stoffig hoekje wordt gezet zoals met de versie Veilig Leren Lezen op de MSX-computer op een aantal scholen is gebeurd.

Een tweede probleem was de ontwerpmethode HUFIT (zie HUSAT 1989). Deze methode is speciaal ontworpen voor het ontwikkelen van user interfaces. Naar mijn mening duurt het veel te lang voordat de methode begint met het globaal ontwikkelen van de interface. Er wordt heel veel aandacht geschonken aan het verzamelen van

informatie over de werkomgeving, de gebruiker en functionaliteit van het systeem.

Op dit moment begint het ontwikkelen van user interfaces wat meer in de belangstelling te staan. Er werd tot nu toe weinig aandacht besteed aan de user interfaces terwijl dit een complex en zeer belangrijk onderdeel is van het te ontwikkelen systeem. Helaas is er op dit moment nog weinig literatuur over dit onderwerp en zijn er geen standaard ontwerp methoden.

Woordenlijst

Question and answer:	Een dialoogstijl waarbij gebruikt gemaakt wordt van vraag en antwoord.
Form Filling:	Een dialoogstijl waarbij de vragen als een formulier worden gepresenteerd, waarbij de vragen stuk voor stuk beantwoord moeten/mogen worden, maar waarbij en een oogopslag alle vragen zichtbaar zijn.
Menu Selection:	Keuze uit een menu stelt de gebruiker in staat te kiezen uit een lijst van alternatieven. Deze methode wordt vooral gebruikt voor sturing van het systeem.
Function Keys:	Funktie toetsen , een enkele toets of een combinatie van toetsen activeert een kommando.
Command language:	Bij de kommando taal communiceert de gebruiker met het systeem door middel van kommando's.
Query language:	De vraagtaal is een dialoogstijl die veel gebruikt wordt bij databases. SQL (Structured Query Language) is de standaard. Gebruik vereist kennis van een groot gedeelte van de taal en de gebruikte database.
Natural language:	De natuurlijke taal sluit nauw aan bij de manier zoals mensen communiceren. Technisch gezien zijn er nogal wat problemen om lopende spraak te analyseren, om sprekers onafhankelijk te zijn en om achtergrond lawaai te elimineren.
Interactive graphics:	Interaktieve symbolen, de dialoogstijl waarbij een grafische voorstelling van objecten en/of taken in combinatie met een muis gebruikt worden voor de sturing en/of selectie van het systeem.

Literatuur

- APPLE (1985) *Inside Macintosh*, Volume 1-5, Addison - Westley publishing Company, Inc.
- Ceasar (1980) *Veilig Leren Lezen*, Structureermethode voor het aanvankelijk leesonderwijs, F.B. Caesar, Tilburg, Zwijssen.
- Chabay/Serwood (1989) *A practical Guide for the Creation of Educational Software*, Ruth W. Chabay, Bruce A. Sherwood, CDEC Technical Report No. 89-06, Carnegie Mellon University Pittsburgh.
- Drunen (1989) *Teacher, een leraar-vriendelijk interface*, G. van Drunen, Afstudeerverslag van de Hogeschool Eindhoven afdeling Informatika.
- Ellermann/Spaai (1990) *The Reading-Board Project*, H.H. Ellermann & G.W.G. Spaai, IPO manuscript no. 736.
- Goodman (1987) *The complete Hypercard handbook*, D. Goodman, Bantam Books, New York.
- Green (1986) *A Survey of Three Dialogue Models*, Mark Green, ACM Transaction on Graphics, Vol 5, No 3, juli 1986, p 244-275.
- Helander Martin (ed) (1988) *Handbook of human-computer interaction*, North-Holland, Amsterdam.
- HUSAT (1989) *The HUFIT Toolset*, proceedings of the seminar for Human Factors Practitioners, 1-3 november 1989, ESPRIT Project 385 - 'HUFIT'.
- Norman/Draper (1986) *User centered system design, new perspectives on human-computer interaction*, Donald A. Norman & Stephen W. Draper (ed), Hillsdale, Erlbaum.

- Pol (1989) *Specificatie en Ontwerp van MIR versie 3.0*, IPO rapport 719, C.J.D.M. van der Pol.
- Pandata (1988) *System Development Methodology*, W.S. Turner, R.P. Langerhorst, G.F. Hice, H.B. Eilers, A.A. Uijtenbroek, Pandata BV, Rijswijk.
- Willemse Henk & Lindijer Goof (1988) *Software ergonomie*, Schoonhoven, Academic Service.
- Juni, Feldt & Surovell *Programmer's Extender volume 1, Elegant & Flexibel Management of the Macintosh Interface* (1985), Invention Software Corporation, Ann Arbor, Michigan

Appendix A: Indeling van Veilig Leren Lezen

Kern	Titel		Letters	Globaalwoorden
1a	Duimeliesje			boom, roos, vis, vuur, mus, pim
1b	De school	1	r	kees, miep, bel, en , boek, raam, de
		2	s	school
		3	m	
1c	De dierentuin	4	l	oom, bus, sam, eet, een aap, pet
		5	p	
		6	t	
		7	b	
		8	v	
1d	Spel en Speelgoed	9	k	wip, kar, tol, het net, vaar, ik,
		10	aa	weeg, op, zit, loop
		11	00	
		12	ee	
2a	Sneeuwitje	13	i	moe, lief, is, wieg, pijl, boog, bos
		14	g	gaat, weg, huis, doos, dop, nies
		15	w	
2b	Naar het bos	16	u	papa, mama, met, haas, tak, eraf
		17	e	valt, zon, muur, voor, pijp, rook,
		18	d	noot, hij, aan, jas, riek
		19	z	
		20	o	
		21	a	
		22	ie	
2c	In de keuken	23	n	keuken, soep, joep, kaas, muis
		24	f	kook, pak, veeg, ui, zeef, uit,
		25	h	kookt, pakt, veegt, ei, zout, pan
		26	j	pop
2d	Klein Duimpje	27	ij	klein duimpje, hout, bij, reus, ziet
		28	oe	niet, ook, voorop, vooruit
		29	ui	
		30	uu	
		31	ou	
		32	sch	
2e	Afsluiting	33	eu	
		34	ei	
		35	au	
		36	ch	

Appendix B: De enquête

1. Algemeen:

- Uw naam en de school?.....
- Hoe lang geeft u al les?.....
- Hoelang voor deze groep?.....
- Heeft u ervaring met komputer gebruik, zo ja welk type en welk merk en in wat voor soort toepassing (tekstverwerking, gegevens opzoeken, rooster planning)?
.....
.....
- Heeft U in uw opleiding of daarna cursussen gevolgd over komputer gebruik of het toepassen op school, zo ja welke?
.....
.....
- Bereidt U uw lessen voornamelijk thuis of op school voor?.....
- Waar denkt u het systeem neer te zetten, er vanuit gaande dat U als enige het systeem zou gebruiken?
 - thuis
 - op school in een apart lokaal
 - in de klas

2. Taak

- Voor welke taken zou U het programma willen gebruiken?
 - Vergelijken met de groep
 - remedial teaching
 - ondersteuning bij beoordeling
 - individuele analyse
 - genereren van samenvattende overzichten
 - genereren van gedetailleerde overzichten
- Wat zou zo'n programma volgens U meer moeten kunnen?
.....
.....
- Hoe vaak denkt u zo'n programma gemiddeld per leerling te gebruiken (1 x per week, maand, dag)?
.....
- Hoeveel tijd denkt u aan het gebruik per leerling per keer gemiddeld te kunnen besteden?
.....
- Zou het handig zijn om een soortgelijk programma te hebben voor andere taken, eventueel binnen andere vakken, zo ja waar denkt u dan aan?

-
- Zou het handig zijn om het programma te gebruiken in samenwerking met een leerlingvolgsysteem?.....
 - Gebruikt U een leerling volgsysteem, zo ja welk?.....
 - Zou U de resultaten ook op papier willen zien om te bewaren en later weer te kunnen gebruiken, of is dit niet nodig?
.....
 - Het is mogelijk om de komputer per leerling te laten bepalen wat deze moet gaan leren. Zoudt U dergelijke adviezen op prijs stellen of wilt U liever zelf een rooster samenstellen?
.....
 - Welke aspecten van het formaat van de oefening zoudt U zelf willen controleren?
 - oefentijd
 - feedback vorm
 - woordmateriaal
 - lees oefening variant
 - zelf woord materiaal toevoegen
 - anders.....
-

3. Kondities

- Hoe vaak en hoe lang denkt U onderbroken te worden door collega's, ouders en kinderen?
.....
 - Vindt U het belangrijk dat er een schriftelijke dokumentatie is van hoe met het systeem omgegaan moet worden?.....
 - Zou het wenselijk zijn dat het programma zelf veel uitleg geeft over wat U kunt doen?.....
 - Hoe zou U dit dan willen zien?.....
 - Zou U het prettig vinden als de komputer U enigszins leidt in de taak die U wilt uitvoeren?.....
 - Heeft U ervaring met lees oefen programma's? Zo ja, welke zijn dat en wat vond U er de sterke en zwakke punten van?
.....
-

4. Opmerkingen

Hebt U nog opmerkingen of suggesties, die voor ons interessant kunnen zijn?

.....
.....

Appendix C: Listing Teacher

Implementatie van een gedeelte van de user interface, de hoofdtaak vorderingen

```
(*
COPYRIGHT: INSTITUUT VOOR PERCEPTIE ONDERZOEK 1990

Implementatie van TEACHER

*)
```

```
PROGRAM TEACHER;
```

```
USES MemTypes, Quickdraw, ToolIntf, PackIntf, Sound, TEACHERdefs, xtypedefs;
```

```
(*
derez RESOURCES Caesar:MPW:Interfaces:RIncludes:Types.r > test
*)
```

```
{SI XtendFast.p}
```

```
(*----- *)
```

```
(* TYPE DECLARATIE *)
```

```
TYPE
  arstring      = array[1..30] of str255;
  arbool        = array[1..30] of boolean;
```

```
(*----- *)
```

```
(* VARIABELEN DECLARATIE *)
```

```
VAR
  (* !!! SYSTEEM PARAMETERS !!! *)
  applemenu, filemenu, editmenu : MenuHandle; (* handle voor standaard apple-menu *)
  event                          : EventRecord; (* systeem event 'gebeurtenis' *)
  whatHappened                   : EventStuff;  (* extender uitbreiding op 'event' *)
  ibeam, cross, plus, watch      : cursor;     (* cursor vormen *)
```

```
(* DECLARATIE VAN DE WINDOWS *)
```

```
TEACHER      : TeacherWindowRecord;
VORDERINGEN  : vorderingenwindow;
GRAFIEK      : grafiekwindow;
NIEUWEGRAFIEK : nieuwegrafiekwindow;
OEFWOORD     : oefwoordwindow;
TABELOEF     : tabeloefwindow;
TABELGRAFEEM : tabelgrafeemwindow;

OefeningenGeselecteerd, IkWilStoppen      : Boolean;
helpselected, routeselected, grafiek2, grafiektwee : Boolean;
Klasselijstbool, oefeningbool, woordenbool,
Klasselijst2bool                          : arbool;

helprect, routerect                       : Rect;
klasselijst, woorden, oefeningen          : arstring;
vulklas, scroll, scroll1, vulwoord, vuloef,i : integer;
```

```
(* FORWARD DECLARATIONS *)
```

```
PROCEDURE MakeWindow;          FORWARD;
PROCEDURE InstallButtons;     FORWARD;
PROCEDURE teacher_graphics;   FORWARD;
PROCEDURE UpdateWindows;     FORWARD;
PROCEDURE handleTEACHER;     FORWARD;
PROCEDURE Init;              FORWARD;
```

```
(* END OF FORWARD DECLARATIONS *)
```

```
(* HET DEFINIEREN VAN EEN RECHTHOEK *)
```

```
FUNCTION Rectangle(x1,y1,x2,y2:Integer):rect;
VAR r:rect;
```

```

BEGIN
  SetRect (r,x1,y1,x2,y2);
  Rectangle:=r;
END; (* END OF RECTANGLE *)

(* HET DEFINIEREN VAN EEN GELUID *)
PROCEDURE PlayASound(st:Str255);
VAR resHndl:Handle;
    err:Oseerr;
BEGIN
  ResHndl:=GetNamedResource('snd ',st);
  err:=SndPlay (NIL,ResHndl,TRUE);
END; (* EINDE VAN PLAY A SOUND *)

(* HET PLAATSTEN VAN EEN STRING OP HET SCHERM *)
PROCEDURE PlaceString(x,y:Integer;s:str255);
BEGIN
  MoveTo(x,y); DrawString(s);
END;

(* HET PRINTEN VAN EEN STRING IN EEN DIALOOGBOX *)
PROCEDURE Printstring(s:string; x, maxy:integer; VAR y: integer);
BEGIN
  textfont(geneva);
  textsize(12);
  moveto (x,y+10);
  if ((y+10)<maxy) then drawstring (s); (* y< maxy: de string mag niet buiten de dialogbox
                                         geprint worden *)
  y:=y+12;
END;

(* HET VULLEN VAN DE DIALOOGBOX, SELECTIES IN DE DIALOOGBOX WORDEN ZICHTBAAR GEMAAKT *)
PROCEDURE schrijfdialoogbox (VAR lijst:arstring;VAR lijstbool:arbool; x1,x2,y1,y2:integer;
                             VAR vulklas, scroll:integer);
VAR r:rect;
BEGIN
  WHILE scroll< vulklas DO
    BEGIN
      IF (lijstbool[scroll]=true) AND (y1<y2)      (* deze zin is geselecteerd *)
      THEN BEGIN
        setrect (r,x1,y1,x2,12+y1);
        penpat (white);
        paintrect (r);
        printstring(lijst[scroll],x1+5,y2,y1);
        invertrect (r);                          (* het veranderen van de kleur van de zin *)
        END
      ELSE printstring(lijst[scroll],x1+5,y2,y1);
        scroll:=scroll+1;
      END;
    END;
  END;

(* HET DEFINIEREN VAN HET WINDOW *)
PROCEDURE MakeWindow;
BEGIN
  SetRect (TEACHER.wrect,59,69,59+512,69+342);      (* het definieren van de maten van het *)
  SetRect (VORDERINGEN.wrect,59,69,59+512,69+342);  (* vierkant in de windows *)
  SetRect (GRAFIEK.wrect,59,69,59+512,69+342);
  SetRect (NIEUWEGRAFIEK.wrect,59,69,59+512,69+342);
  SetRect (OEFWOORD.wrect,59,69,59+512,69+342);
  SetRect (TABELOEF.wrect,59,69,59+512,69+342);
  SetRect (TABELGRAFEEM.wrect,59,69,59+512,69+342);

  TEACHER.window:=CreateWindow(TEACHER.rec,TEACHER.wrect,' ',2,TRUE,FALSE,FALSE,FALSE);

```

```

VORDERINGEN.window:=CreateWindow(VORDERINGEN.rec,VORDERINGEN.wrect,'',2,TRUE,FALSE,FALSE,
                                FALSE,FALSE);
GRAFIEK.window:=CreateWindow(GRAFIEK.rec,GRAFIEK.wrect,'',2,TRUE,FALSE,FALSE,FALSE,FALSE);
NIEUWEGRAFIEK.window:=CreateWindow(NIEUWEGRAFIEK.rec,NIEUWEGRAFIEK.wrect,'',2,TRUE,FALSE,
                                FALSE,FALSE,FALSE);
OEFWOORD.window:=CreateWindow(OEFWOORD.rec,OEFWOORD.wrect,'',2,TRUE,FALSE,FALSE,FALSE,FALSE);
TABELOEF.window:=CreateWindow(TABELOEF.rec,TABELOEF.wrect,'',2,TRUE,FALSE,FALSE,FALSE,FALSE);
TABELGRAFEEM.window:=CreateWindow(TABELGRAFEEM.rec,TABELGRAFEEM.wrect,'',2,TRUE,FALSE,FALSE,
                                FALSE,FALSE);

SelectWindow(TEACHER.window);          (* het startwindow wordt geselecteerd *)
END;

(* HET INSTALLEREN VAN DE BUTTONS *)
PROCEDURE InstallButtons;
BEGIN
  SetRect(TEACHER.stopr,404,148,504,170);
  SetRect(TEACHER.vorderingenr,344,234,444,253);
  SetRect(TEACHER.agendar,261,164,353,185);
  SetRect(TEACHER.klasselijstr,151,164,251,185);
  SetRect(TEACHER.demonstratier,82,230,186,249);
  SetRect(TEACHER.training_startenr,8,148,121,170);
  SetRect(TEACHER.instructier,15,9,115,31);
  SETRECT(TEACHER.helpr,450,0,482,32);

  TEACHER.stop           :=CreatePushButton(TEACHER.window,TEACHER.stopr,'MIR stoppen');
  TEACHER.vorderingen    :=CreatePushButton(TEACHER.window,TEACHER.vorderingenr,'vorderingen');
  TEACHER.agenda         :=CreatePushButton(TEACHER.window,TEACHER.agendar,'agenda');
  TEACHER.klasselijst    :=CreatePushButton(TEACHER.window,TEACHER.klasselijstr,'klasselijst');
  TEACHER.demonstratie   :=CreatePushButton(TEACHER.window,TEACHER.demonstratier,'demonstratie');
  TEACHER.training_starten:=CreatePushButton(TEACHER.window,TEACHER.training_startenr,
                                             'training starten');
  TEACHER.instructie     :=CreatePushButton(TEACHER.window,TEACHER.instructier,'instructie');

  SetRect(VORDERINGEN.tabelr,162,283,262,305);
  SetRect(VORDERINGEN.grafiekr,272,284,372,306);
  SetRect(VORDERINGEN.teacherr,398,303,498,325);
  SetRect(VORDERINGEN.oefeningr,69,162,169,184);
  SetRect(VORDERINGEN.grafeemr,69,183,222,205);
  SetRect(VORDERINGEN.scrollnaamr,437,63,452,147);
  SetRect(VORDERINGEN.scrollgrafoefr,437,165,452,248);

  VORDERINGEN.TABEL      :=CreatePushButton(VORDERINGEN.window,VORDERINGEN.tabelr,'tabel');
  VORDERINGEN.GRAFIEK    :=CreatePushButton(VORDERINGEN.window,VORDERINGEN.grafiekr,'grafiek');
  VORDERINGEN.TEACHER    :=CreatePushButton(VORDERINGEN.window,VORDERINGEN.teacherr,'teacher');
  VORDERINGEN.OEFENING   :=CreateRadioButton(VORDERINGEN.window,VORDERINGEN.oefeningr,'oefening');
  VORDERINGEN.GRAFEEM    :=CreateRadioButton(VORDERINGEN.window,VORDERINGEN.grafeemr,
                                             'grafemen/woorden');
  VORDERINGEN.SCROLLNAAM :=CreateScrollBar(VORDERINGEN.window,VORDERINGEN.scrollnaamr,1,1,0);
  VORDERINGEN.SCROLLGRAFOEF:=CreateScrollBar(VORDERINGEN.window,VORDERINGEN.scrollgrafoefr,1,1,0);

  SetRect(GRAFIEK.nieuwgraf1r,375,139,502,162);
  SetRect(GRAFIEK.nieuwgraf2r,375,167,502,189);
  SetRect(GRAFIEK.grafiek2r,375,167,502,189);
  SetRect(GRAFIEK.tabelr,403,212,474,232);
  SetRect(GRAFIEK.oef_woordr,368,237,501,260);
  SetRect(GRAFIEK.teacherr,393,303,493,325);
  setRect(GRAFIEK.graf1r,400,60,475,75);
  setRect(GRAFIEK.graf2r,400,75,475,90);

  GRAFIEK.NIEUWGRAF1     :=CreatePushButton(GRAFIEK.window,GRAFIEK.nieuwgraf1r,'nieuwe grafiek 1');
  GRAFIEK.NIEUWGRAF2     :=CreatePushButton(GRAFIEK.window,GRAFIEK.nieuwgraf2r,'nieuwe grafiek 2');
  GRAFIEK.GRAFIEK2       :=CreatePushButton(GRAFIEK.window,GRAFIEK.grafiek2r,'grafiek 2');
  GRAFIEK.TABEL          :=CreatePushButton(GRAFIEK.window,GRAFIEK.tabelr,'tabel');
  GRAFIEK.OEF_WOORD      :=CreatePushButton(GRAFIEK.window,GRAFIEK.oef_woordr,'oefening/woorden');
  GRAFIEK.TEACHER        :=CreatePushButton(GRAFIEK.window,GRAFIEK.teacherr,'teacher');

  SetRect(NIEUWEGRAFIEK.tabelr,396,229,496,251);

```

```

SetRect (NIEUWEGRAFIEK.grafiekr, 396, 257, 496, 279);
SetRect (NIEUWEGRAFIEK.scrollnaamr, 328, 107, 343, 192);
SetRect (NIEUWEGRAFIEK.verwijderr, 396, 285, 496, 307);

NIEUWEGRAFIEK.TABEL      :=CreatePushButton(NIEUWEGRAFIEK.window, NIEUWEGRAFIEK.tabelr, 'tabel');
NIEUWEGRAFIEK.GRAFIEK    :=CreatePushButton(NIEUWEGRAFIEK.window, NIEUWEGRAFIEK.grafiekr,
                                             'grafiek');
NIEUWEGRAFIEK.SCROLLNAAM:=CreateScrollBar(NIEUWEGRAFIEK.window, NIEUWEGRAFIEK.scrollnaamr, 1, 1, 10);
NIEUWEGRAFIEK.VERWIJDER :=CreatePushButton(NIEUWEGRAFIEK.window, NIEUWEGRAFIEK.verwijderr,
                                             'verwijder');

```

```

Setrect (OEFWOORD.tabelr, 402, 267, 489, 288);
Setrect (OEFWOORD.grafiekr, 404, 300, 489, 321);
Setrect (OEFWOORD.demonstratier, 26, 300, 203, 323);
Setrect (OEFWOORD.oefeningr, 183, 68, 283, 90);
Setrect (OEFWOORD.oef_woodr, 183, 90, 335, 112);
Setrect (OEFWOORD.scrolloefr, 370, 134, 385, 217);

```

```

OEFWOORD.TABEL      :=CreatePushButton(OEFWOORD.window, OEFWOORD.tabelr, 'tabel');
OEFWOORD.GRAFIEK    :=CreatePushButton(OEFWOORD.window, OEFWOORD.grafiekr, 'grafiek');
OEFWOORD.DEMONSTRATIE :=CreatePushButton(OEFWOORD.window, OEFWOORD.demonstratier,
                                           'demonstratie oefeningen');
OEFWOORD.OEFENING   :=CreateRadioButton(OEFWOORD.window, OEFWOORD.oefeningr, 'oefening');
OEFWOORD.OEF_WOORD  :=CreateRadioButton(OEFWOORD.window, OEFWOORD.oef_woodr,
                                           'woorden/grafemen');
OEFWOORD.SCROLLOEF  :=CreateScrollBar(OEFWOORD.window, OEFWOORD.scrolloefr, 1, 1, 10);

```

```

Setrect (TABELOEF.grafiekr, 85, 300, 185, 322);
Setrect (TABELOEF.teacherr, 397, 300, 497, 322);
setrect (TABELOEF.vorderingenr, 190, 300, 290, 322);

```

```

TABELOEF.GRAFIEK:=CreatePushButton(TABELOEF.window, TABELOEF.grafiekr, 'grafiek');
TABELOEF.TEACHER:=CreatePushButton(TABELOEF.window, TABELOEF.teacherr, 'teacher');
TABELOEF.VORDERINGEN:=CreatePushButton(TABELOEF.window, TABELOEF.vorderingenr, 'vorderingen');

```

```

Setrect (TABELGRAFEEM.grafiekr, 85, 300, 185, 322);
Setrect (TABELGRAFEEM.teacherr, 397, 300, 497, 322);
Setrect (TABELGRAFEEM.vorderingenr, 190, 300, 290, 322);

```

```

TABELGRAFEEM.GRAFIEK:=CreatePushButton(TABELGRAFEEM.window, TABELGRAFEEM.grafiekr, 'grafiek');
TABELGRAFEEM.TEACHER:=CreatePushButton(TABELGRAFEEM.window, TABELGRAFEEM.teacherr, 'teacher');
TABELGRAFEEM.VORDERINGEN:=CreatePushButton(TABELGRAFEEM.window, TABELGRAFEEM.vorderingenr,
                                             'vorderingen');

```

```
END;
```

```
(* PROCEDURE VOOR HET VERANDEREN VAN WINDOW *)
```

```

PROCEDURE SwitchWindow(VAR w:WindowPtr);
BEGIN
  SelectWindow(w);
END; (* END OF SWITCHWINDOW *)

```

```
(* HET TEKENEN VAN DE WINDOWS MET HET STIPPELTJES PATROON, DE TITEL EN DE HELP EN ROUTE BUTTONS *)
```

```

PROCEDURE DrawWindows(wr:rect;titel:str255;help,route:Boolean);
  VAR r,r2:rect;
BEGIN
  PenSize(1,1); TextFace([]);
  ValidRect(wr);
  Fillrect(wr,ltgray);
  SetRect(r,10,42,502,332);
  PenPat(white); PaintRoundRect(r,10,10);
  PenPat(black); FrameRoundRect(r,10,10);
  (* titel *)
  IF titel<>' ' THEN
  BEGIN
    SetRect(r,173,8,347,32);
    SetRect(r2,176,11,344,29);

```

```

PenPat (white); PaintRect (r);
PenPat (black); FrameRect (r); frameRect (r2);
TextFace ([bold]);
MoveTo (260- (StringWidth (titel) DIV 2), 24);
DrawString (titel);
END;

(* SPECIALE HELP BUTTON *)
IF Help THEN
BEGIN
textfont (courier);
PenPat (white); PaintRoundRect (helprect, 10, 10);
PenPat (black); FrameRoundRect (helprect, 10, 10);
SetRect (r, 461, 3, 493, 35); PlotIcon (r, GetIcon (100));
TextSize (10); MoveTo (465, 47); DrawString ('help');
textfont (0);
END;

(* SPECIALE ROUTE BUTTON *)
IF route THEN
BEGIN
textfont (courier);
PenPat (white); PaintRoundRect (routerect, 10, 10);
PenPat (black); FrameRoundRect (routerect, 10, 10);
SetRect (r, 20, 3, 52, 35); PlotIcon (r, GetIcon (101));
TextSize (10); MoveTo (20, 47); DrawString ('route');
textfont (0);
END;

TextSize (12); PenPat (black); TEXTFONT (0); PenNormal; TextFace ([]);
END; (* END OF DRAWSPECIALBUTTONS *)

(* HET TEKENEN VAN HET TEACHER SCHERM *)
PROCEDURE teacher_graphics;
VAR r, r2:rect;
BEGIN
DrawWindows (teacher.window^.portrect, '', TRUE, FALSE);
PenSize (2, 2);
MoveTo (64, 110); LineTo (64, 148); (* de verticale lijnen *)
MoveTo (134, 110); LineTo (134, 230);
MoveTo (201, 110); LineTo (201, 164);
MoveTo (307, 110); LineTo (307, 164);
MoveTo (394, 110); LineTo (394, 234);
MoveTo (454, 110); LineTo (454, 148);
MoveTo (64, 110); LineTo (454, 110); (* horizontale lijn *)
MoveTo (252, 110); LineTo (252, 100);
TextFont (geneva); TextFace ([bold]); TextSize (12); MoveTo (225, 90); DrawString ('TEACHER');
PenNormal; textFont (0); Textface ([]);
DrawControls (TEACHER.window); (* de buttons en scrollbars *)
END;

(* HET TEKENEN VAN HET VORDERINGEN SCHERM *)
PROCEDURE vorderingen_graphics;
VAR r:rect;
BEGIN
DrawWindows (vorderingen.window^.portrect, 'vorderingen', TRUE, TRUE);
MoveTo (16, 82); DrawString ('selecteer de leerling(en):');
MoveTo (16, 160); DrawString ('maak uw keuze:');
MoveTo (16, 226); DrawString ('en selecteer. ');
SetRect (r, 230, 63, 452, 147); FrameRect (r);
SetRect (r, 230, 165, 452, 248); frameRect (r);
setctlmax (vorderingen.scrollnaam, vulklas-6); (* het zetten van de max waarde van de scrollbar *)
scroll:=getctlvalue (vorderingen.scrollnaam); (* het uitlezen van de huidige waarde vd scrollbar
als scroll bijvoorbeeld de waarde 3 heeft dan
betekent dat het derde element uit de lijst op
de eerste plaats in de dialoogbox staat *)

```

```

setrect(r,231,64,436,146); (* het tekenen van de dialoogbox *)
eraserect(r);
schrijfdialoogbox (klasselijst,klasselijstbool, 231,436,69,147, vulklas, scroll); (* het vullen
van de dialoogbox *)

IF OefeningenGeselecteerd THEN BEGIN
    SetCtlValue(vorderingen.oefening,1);
    SetCtlValue(vorderingen.grafeem,0);
    setctlmax(vorderingen.scrollgrafeef,5);
    scroll1:=getctlvalue(vorderingen.scrollgrafeef);
    setrect(r,231,166,436,247);
    eraserect(r);
    schrijfdialoogbox (oefeningen,oefeningbool, 231,436,171,248,
vuloeff, scroll1);
    END
ELSE BEGIN
    SetCtlValue(vorderingen.oefening,0);
    SetCtlValue(vorderingen.grafeem,1);
    setctlmax(vorderingen.scrollgrafeef,vulwoord-6);
    scroll1:=getctlvalue(vorderingen.scrollgrafeef);
    setrect(r,231,166,436,247);
    eraserect(r);
    schrijfdialoogbox (woorden,woordenbool, 231,436,171,248,
vulwoord, scroll1);
    END;
DrawControls(VORDERINGEN.window); (* het tekenen van de buttons en scrollbars *)
END;

(* HET TEKENEN VAN HET GRAFIEK SCHERM *)
PROCEDURE grafiek_graphics;
VAR r:rect;
BEGIN
    textsize(12);
    DrawWindows(grafiek.window^.portrect,'grafiek',TRUE,TRUE);
    DrawControls(GRAFIEK.window);
    pensize(1,1);
    moveto(405,68); lineto(425,68); (* het tekenen van de assen van de grafiek *)
    moveto(50,300); lineto(300,300);
    moveto(75,300); lineto(75,302);
    moveto(100,300); lineto(100,302);
    moveto(125,300); lineto(125,302);
    moveto(150,300); lineto(150,302);
    moveto(175,300); lineto(175,302);
    moveto(200,300); lineto(200,302);
    moveto(225,300); lineto(225,302);
    moveto(250,300); lineto(250,302);
    moveto(275,300); lineto(275,302);
    moveto(50,300); lineto(50, 80);
    moveto(48,275); lineto(50, 275);
    moveto(48,250); lineto(50, 250);
    moveto(48,225); lineto(50, 225);
    moveto(48,200); lineto(50, 200);
    moveto(48,175); lineto(50, 175);
    moveto(48,150); lineto(50, 150);
    moveto(48,125); lineto(50, 125);
    moveto(48,100); lineto(50, 100);
    pensize(2,2);
    moveto(405,80); lineto(425,80);
    textfont(geneva);
    textsize(9);
    moveto(25,75); drawstring('effektiviteit'); (* het schrijven van de tekst bij de grafiek *)
    moveto(290,310); drawstring('oefening/woorden');
    moveto(430,72); drawstring('grafiek1');
    moveto(430,85); drawstring('grafiek2');
    pensize(1,1);
    framereact(rectangle(400,60,475,90));
    PenNormal; textFont(0); Textface({});
END;

```



```
(* HET TEKENEN VAN HET SCHERM VOOR HET MAKEN VAN EEN NIEUWE GRAFIEK *)
```

```
PROCEDURE nieuwegrafiek_graphics;
```

```
VAR r:rect;
```

```
BEGIN
```

```
DrawWindows(nieuwegrafiek.window^.portrect,'(nieuwe) grafiek',TRUE,TRUE);
```

```
SetRect(r,143,107,343,192); FrameRect(r);
```

```
setctlmax(nieuwegrafiek.scrollnaam, vulklas-6);
```

```
scroll:=getctlvalue(nieuwegrafiek.scrollnaam); (* uitlezen van de waarde van de scrollbar,  
als scroll is bijvoorbeeld 3, dan betekent dat dat het derde element  
uit de lijst als eerste element in de dialogbox te zien is *)
```

```
setrect(r,144,108,327,191);
```

```
erasetrect(r);
```

```
IF grafiek2=false
```

```
THEN schrijfdialoogbox(klasselijst,klasselijstbool,144,327,113,192,vulklas,scroll)
```

```
ELSE schrijfdialoogbox(klasselijst,klasselijst2bool,144,327,113,192,vulklas,scroll);
```

```
MoveTo(19,94); drawString('selecteer de leerling(en):');
```

```
DrawControls(NIEUWEGRAFIEK.window); (* het tekenen van de buttons en scrollbars *)
```

```
END;
```

```
(* HET TEKENEN VAN HET SCHERM WAAR NIEUWE OEFENINGEN EN WOORDEN GESELECTEERD KUNNEN WORDEN *)
```

```
PROCEDURE oefwoord_graphics;
```

```
VAR r:rect;
```

```
BEGIN
```

```
DrawWindows(oefwoord.window^.portrect,'oefening/woorden',TRUE,TRUE);
```

```
SetRect(R,156,134,385,217); FrameRect(r);
```

```
MoveTo(77,71); DrawString('maak uw keuze');
```

```
MoveTo(77,126); DrawString('en selecteer:');
```

```
IF OefeningenGeselecteerd THEN BEGIN
```

```
SetCtlValue(OEFWOORD.oefening,1);
```

```
SetCtlValue(OEFWOORD.oef_woord,0);
```

```
setctlmax(OEFWOORD.scrolloef,5);
```

```
scroll1:=getctlvalue(OEFWOORD.scrolloef);
```

```
setrect(r,157,135,369,216);
```

```
erasetrect(r);
```

```
schrijfdialoogbox (oefeningen,oefeningbool, 157,369,140,217,  
vuloef, scroll1);
```

```
END
```

```
ELSE BEGIN
```

```
SetCtlValue(OEFWOORD.oefening,0);
```

```
SetCtlValue(OEFWOORD.oef_woord,1);
```

```
setctlmax(OEFWOORD.scrolloef,vulwoord-1);
```

```
scroll1:=getctlvalue(OEFWOORD.scrolloef);
```

```
setrect(r,157,135,369,216);
```

```
erasetrect(r);
```

```
schrijfdialoogbox (woorden,woordenbool, 157,369,140,217,  
vulwoord, scroll1);
```

```
END;
```

```
DrawControls(OEFWOORD.window);
```

```
END;
```

```
(* HET TEKENEN VAN HET SCHERM MET DE TABEL VAN DE OEFENINGEN *)
```

```
PROCEDURE tabeloef_graphics;
```

```
VAR r:rect;
```

```
BEGIN
```

```
DrawWindows(tabeloef.window^.portrect,'tabeloef',TRUE,TRUE);
```

```
TextSize(12);
```

```
PlaceString( 25,102,'oefening'); (* de tekst in de tabel *)
```

```
PlaceString( 132,102,'datum/tijd');
```

```
PlaceString( 236,102,'grafiek1');
```

```
PlaceString( 313,102,'duur');
```

```
PlaceString( 364,102,'grafiek2');
```

```
PlaceString( 441,102,'duur');
```

```
Textface([underline]); PlaceString( 170,80,' Resultaten van de oefening'); Textface([]);
```

```
FrameRect( Rectangle(17,86,480,280) ); (* de lijnen van de tabel *)
```

```

FrameRect( Rectangle(124,86,480,280) );
FrameRect( Rectangle(228,86,480,280) );
FrameRect( Rectangle(305,86,480,280) );
FrameRect( Rectangle(356,86,480,280) );
FrameRect( Rectangle(433,86,480,280) );
FrameRect( Rectangle(17,86,480,108) );
FrameRect( Rectangle(17,60,480,87) );
DrawControls(TABELOEF.window);          (* de buttons van het scherm *)
END;

(* HET TEKENEN VAN HET SCHERM MET DE TABEL VAN DE WOORDEN EN GRAFEMEN *)
PROCEDURE tabelgrafeem_graphics;
  VAR r:rect;
BEGIN
  DrawWindows(tabelgrafeem.window^.portrect, 'tabelgrafeem', TRUE, TRUE);
  DrawControls(TABELGRAFEEM.window);
  textsize(12);
  textfont(geneva);
  textface([underline]);placestring(150,80, 'Resultaten van woorden en grafemen');Textface({});
  placestring(19,100,'woorden/grafemen');          (* het schrijven van de tekst *)
  placestring(140,100,'% goed beantwoord');
  placestring(259,100,'% fout beantwoord');
  placestring(378,100,'% niet beantwoord');
  placestring(140,116,'grafiek1');
  placestring(200,116,'grafiek2');
  placestring(259,116,'grafiek1');
  placestring(319,116,'grafiek2');
  placestring(378,116,'grafiek1');
  placestring(438,116,'grafiek2');
  framerect(rectangle(17,86,495,280));          (* het tekenen van de lijnen *)
  framerect(rectangle(17,86,495,120));
  framerect(rectangle(17,86,495,103));
  framerect(rectangle(17,60,495,87));
  framerect(rectangle(138,86,495,280));
  framerect(rectangle(257,86,495,280));
  framerect(rectangle(376,86,495,280));
  framerect(rectangle(198,102,495,280));
  framerect(rectangle(317,102,495,280));
  framerect(rectangle(436,102,495,280));
END;

(* PROCEDURE VOOR HET UPDATEN VAN DE WINDOWS *)
PROCEDURE UpdateWindows;
  VAR TheWindow:WindowPtr;
      saveport:GrafPtr;
BEGIN
  getport(saveport);
  theWindow:=WindowPtr(event.message);
  setport(thewindow);
  IF thewindow=teacher.window THEN teacher_graphics;
  IF thewindow=vorderingen.window THEN vorderingen_graphics;
  IF thewindow=grafiek.window THEN grafiek_graphics;
  IF thewindow=nieuwegrafiek.window THEN nieuwegrafiek_graphics;
  IF thewindow=oefwoord.window THEN oefwoord_graphics;
  IF thewindow=tabeloef.window THEN tabeloef_graphics;
  IF thewindow=tabelgrafeem.window THEN tabelgrafeem_graphics;
  setport(saveport);
END; (* END OF UPDATEWINDOW *)

(* PROCEDURE DIE KIJKT OF ER OP DE HELP BUTTON IS GEKLIKT *)
PROCEDURE MouseInHelp;
  VAR p:point;
BEGIN
  p:=event.where;
  GlobalToLocal(p);

```

```

IF (event.what=mousedown) AND (PtInRect(p, helprect)) THEN
BEGIN
  InvertRoundRect (helprect,10,10);
  Framaroundrect (helprect,10,10);
  HelpSelected:=TRUE;
  REPEAT
    REPEAT
      SystemTask;
    UNTIL XTgetNextEvent (everyEvent, event);
  UNTIL event.what=Mouseup;
  InvertRoundRect (helprect,10,10);
  Framaroundrect (helprect,10,10);
END;
END; (* END OF MOUSEINHELP *)

(* PROCEDURE DIE HET KLIKKEN OP BUTTONS IN HET TEACHER SCHERM AFHANDELD *)
PROCEDURE handleTEACHER;
BEGIN
  WITH WhatHappened DO BEGIN
    IF whichcontrol=TEACHER.stop THEN IkWilStoppen:=TRUE; (* programma stopt *)
    IF whichcontrol=TEACHER.vorderingen THEN
      BEGIN
        SwitchWindow(VORDERINGEN.window);
        grafiek2:=false;
      END;
  END;
  MouseInHelp; (* kijken of er op de help button is geklikt *)
  IF helpselected THEN BEGIN
    PlayASound('monkey');
    Helpselected:=false;
  END;
END;

(* HET AFHANDELEN VAN HET KLIKKEN OP DE BUTTONS EN SCROLLBARS VAN HET VORDERINGEN SCHERM *)
PROCEDURE handleVORDERINGEN;
VAR r,r1: rect;
    x1,x2,y1,y2,i,j:integer;
    p:point;
BEGIN
  IF event.what=mousedown (* als er met de muis is geklikt *)
  THEN BEGIN
    p:=event.where; (* de plaats waar geklikt is *)
    j:=1;
    globaltolocal(p); (* de coördinaten van het punt ten opzichte van de linkerbovenhoek
                       van het gedefinieerde window *)

    x1:=231;
    x2:=436;
    setrect (r,x1,69,x2,147); (* de dialoogbox met de namen *)
    setrect (r1,x1,171,x2,248); (* de dialoogbox met woorden/grafemen en oefeningen *)
    IF ptinrect(p,r) (* er is in de dialoogbox met namen geklikt *)
    THEN BEGIN
      y1:=69 + 12*(j-1);
      y2:=69 + 12*j;
      setrect (r,x1,y1,x2,y2); (* definiëren rechthoek om eerste naam in dialoogbox *)
      i:=getctlvalue (vorderingen.scrollnaam);
      WHILE y2<= 147 DO
      BEGIN
        IF ptinrect(p,r) (* kijken of er in de rechthoek is geklikt *)
        THEN BEGIN
          IF klasselijstbool[i]=false
          THEN BEGIN
            penpat (white);
            paintrect (r);
            placestring(x1+5,y1+10,klasselijst[i]); (* naam in dialoogbox printen *)
            invertrect (r);
            y2:=200;
          END;
        END;
      END;
    END;
  END;
END;

```

```

        klasselijstbool[i]:=true;          (* de naam is geselecteerd *)
      END
    ELSE BEGIN
      invertrect(r);
      y2:=200;
      klasselijstbool[i]:=false;         (* de naam is niet geselecteerd *)
    END;
  END
ELSE BEGIN
  i:=i+1;
  j:=j+1;
  y1:=y2;
  y2:=y2+12;
  setrect(r,x1,y1,x2,y2);              (* definiëren rechthoek om volgende naam *)
END;
END;
END
ELSE IF ptinrect(p,r1)  (* er is in de dialoogbox met woorden en oefeningen geklikt *)
  THEN BEGIN
    y1:=171+12*(j-1);
    y2:=171+12*j;
    setrect(r,x1,y1,x2,y2);           (* definiëren rechthoek om eerste element in
                                       de dialoogbox *)
    i:=getctlvalue(vorderingen.scrollgrafoef);
    WHILE y2<248 DO
      BEGIN
        IF ptinrect(p,r)  (* is er in de rechthoek geklikt *)
          THEN BEGIN
            IF oefeningengeselecteerd = true  (* de dialoogbox is gevuld met
                                                de oefeningen *)
              THEN BEGIN
                IF oefeningbool[i]=false
                  THEN BEGIN
                    penpat(white);
                    paintrect(r);
                    placestring(x1+5,y1+10,oefeningen[i]);
                    invertrect(r);
                    y2:=300;
                    oefeningbool[i]:=true; (* oefening is geselecteerd *)
                  END
                ELSE BEGIN
                    invertrect(r);
                    y2:=300;
                    oefeningbool[i]:=false; (* oefening niet geselecteerd *)
                END;
              END
            ELSE BEGIN
                (* de dialoogbox is gevuld met woorden/grafemen *)
                IF woordenbool[i]=false
                  THEN BEGIN
                    penpat(white);
                    paintrect(r);
                    placestring(x1+5,y1+10,woorden[i]);
                    invertrect(r);
                    y2:=300;
                    woordenbool[i]:=true; (* woord/grafeem is geselecteerd *)
                  END
                ELSE BEGIN
                    invertrect(r);
                    y2:=300;
                    woordenbool[i]:=false;
                END;
            END;
          END;
        END
      END
    ELSE BEGIN
      i:=i+1;
      j:=j+1;
      y1:=y2;
      y2:=y2+12;
    END;
  END;

```

```

                setrect (r,x1,y1,x2,y2);
            END;
        END;
    END;
    END;
WITH whatHappened DO BEGIN
    IF whichcontrol=VORDERINGEN.GRAFIEK THEN
    BEGIN
        SwitchWindow(GRAFIEK.window);
        hidecontrol(GRAFIEK.nieuwgraf2);
        showcontrol(GRAFIEK.grafiek2);
    END;
    IF whichcontrol=vorderingen.scrollnaam    (* er is op de scrollbar geklikt, opnieuw vullen
                                                van de dialogbox *)
    THEN BEGIN
        scroll:=getctlvalue(vorderingen.scrollnaam);
        setrect (r,231,64,436,146);
        eraserect (r);
        schrijfdialoogbox (klasselijst, klasselijstbool,231,436,69,147, vulklas, scroll);
    END;
    IF whichcontrol=VORDERINGEN.TEACHER THEN SwitchWindow(TEACHER.window);
    IF (whichcontrol=VORDERINGEN.TABEL) AND OefeningenGeselecteerd THEN
    BEGIN
        SwitchWindow(TABELOEF.window);
        showcontrol(tabeloef.vorderingen);
    END;
    IF (whichcontrol=VORDERINGEN.TABEL) AND NOT(OefeningenGeselecteerd) THEN
    BEGIN
        SwitchWindow(TABELGRAFEEM.window);
        showcontrol(tabelgrafeem.vorderingen);
    END;
    IF whichcontrol=VORDERINGEN.OEFENING THEN    (* oefeningen is geselecteerd, vullen van de
                                                dialogbox met oefeningen *)
    BEGIN
        OefeningenGeselecteerd:=TRUE;
        SetCtlValue(VORDERINGEN.OEFENING,1);
        SetCtlValue(VORDERINGEN.GRAFEEM,0);
        setctlmax(vorderingen.scrollgrafoef,5);
        scroll1:=getctlvalue(vorderingen.scrollgrafoef);
        setrect (r,231,166,436,247);
        eraserect (r);
        schrijfdialoogbox (oefeningen, oefeningbool,231,436,171,248, vuloef, scroll1);
    END;
    IF whichcontrol=VORDERINGEN.GRAFEEM THEN    (* woorden/grafemen zijn geselecteerd, vullen van
                                                de dialogbox met woorden en grafemen *)
    BEGIN
        OefeningenGeselecteerd:=FALSE;
        SetCtlValue(VORDERINGEN.OEFENING,0);
        SetCtlValue(VORDERINGEN.GRAFEEM,1);
        setctlmax(vorderingen.scrollgrafoef,vulwoord-6);
        scroll1:=getctlvalue(vorderingen.scrollgrafoef);
        setrect (r,231,166,436,247);
        eraserect (r);
        schrijfdialoogbox (woorden,woordenbool, 231,436,171,248, vulwoord, scroll1);
    END;
    IF whichcontrol = VORDERINGEN.SCROLLGRAFOEF THEN    (* er is op de scrollbar geklikt, opnieuw
                                                vullen van de dialogbox *)
    BEGIN
        IF Oefeningengeselecteerd = true    (* oefeningen zijn geselecteerd *)
        THEN BEGIN
            setctlmax(vorderingen.scrollgrafoef,5);
            scroll1:=getctlvalue(vorderingen.scrollgrafoef);
            setrect (r,231,166,436,247);
            eraserect (r);
            schrijfdialoogbox (oefeningen,oefeningbool, 231,436,171,248, vuloef, scroll1);
        END
        ELSE BEGIN    (* woorden/grafemen zijn geselecteerd *)
            setctlmax(vorderingen.scrollgrafoef,vulwoord-6);

```

```

        scroll1:=getctlvalue(vorderingen.scrollgrafoef);
        setrect(r,231,166,436,247);
        erasereact(r);
        schrijfdialoogbox (woorden,woordenbool, 231,436,171,248, vulwoord, scroll1);
    END;
END;
END; (* END OF HANDLEVORDERINGEN *)

(* HET AFHANDELEN VAN HET KLIKKEN OP DE BUTTONS VAN HET GRAFIEK SCHERM *)
PROCEDURE handleGRAFIEK;
BEGIN
    WITH whatHappened DO BEGIN
        IF whichcontrol=GRAFIEK.TEACHER THEN SwitchWindow(TEACHER.window);
        IF whichcontrol=GRAFIEK.NIEUWGRAF1 THEN
            BEGIN
                SwitchWindow(NIEUWEGRAFIEK.window);
                hidecontrol(NIEUWEGRAFIEK.verwijder);
                grafiek2:=false;          (* het nieuwe grafiek scherm bevat de instellingen voor grafiek 1 *)
            END;
        IF whichcontrol=GRAFIEK.NIEUWGRAF2 THEN
            BEGIN
                SwitchWindow(NIEUWEGRAFIEK.window);
                showcontrol(NIEUWEGRAFIEK.verwijder);
                grafiek2:=true;          (* het nieuwe grafiek scherm bevat de instellingen voor grafiek 2 *)
            END;
        IF whichcontrol=GRAFIEK.GRAFIEK2 THEN
            BEGIN
                SwitchWindow(NIEUWEGRAFIEK.window);
                hidecontrol(NIEUWEGRAFIEK.verwijder);
                grafiek2:=true;          (* het nieuwe grafiek scherm bevat de instellingen voor grafiek 2 *)
                grafiektwee:=true;      (* er zijn twee grafiek geselecteerd *)
            END;
        IF WhichControl=GRAFIEK.OEF_WOORD THEN SwitchWindow(OEFWOORD.window);
        IF (whichcontrol=GRAFIEK.TABEL) AND OefeningenGeselecteerd THEN
            BEGIN
                SwitchWindow(TABELOEF.window);
                hidecontrol(tabeloef.vorderingen);
            END;
        IF (whichcontrol=GRAFIEK.TABEL) AND NOT(OefeningenGeselecteerd) THEN
            BEGIN
                SwitchWindow(TABELGRAFEEM.window);
                hidecontrol(tabelgrafeem.vorderingen);
            END;
    END;
END;
END;

```

```

(* HET AFHANDELEN VAN HET KLIKKEN OP DE BUTTONS EN SCROLLBAR VAN HET NIEUWE GRAFIEK SCHERM *)
PROCEDURE handleNIEUWEGRAFIEK;
VAR r:rect;
    p:point;
    x1,x2,y1,y2,i,j:integer;
BEGIN
    IF event.what=mousedown      (* is er met de muis geklikt *)
    THEN BEGIN
        p:=event.where;          (* punt waar geklikt is *)
        j:=1;
        globaltolocal(p);        (* punt tov de linkerbovenhoek van het gedefinieerde window *)
        i:=getctlvalue(nieuwegrafiek.scrollnaam);
        x1:=144;
        x2:=327;
        y1:=113+12*(j-1);
        y2:=113+12*j;
        setrect(r,x1,y1,x2,y2);  (* definieren rechthoek om eerste element uit dialoogbox *)
        WHILE y2<= 192 DO
            BEGIN

```

```

IF ptirect(p,r)
THEN BEGIN
  IF grafiek2=false (* er is maar een grafiek geselecteerd *)
  THEN BEGIN
    IF klasselijstbool[i]=false
    THEN BEGIN
      penpat(white);
      paintrect(r);
      placestring(x1+5,y1+10,klasselijst[i]);
      invertrect(r);
      y2:=200;
      klasselijstbool[i]:=true;
    END
  ELSE BEGIN
    invertrect(r);
    y2:=200;
    klasselijstbool[i]:=false;
  END;
  END
ELSE BEGIN (* er is een tweede grafiek geselecteerd *)
  IF klasselijst2bool[i]=false
  THEN BEGIN
    penpat(white);
    paintrect(r);
    placestring(x1+5,y1+10,klasselijst[i]);
    invertrect(r);
    y2:=200;
    klasselijst2bool[i]:=true; (* naam is geselecteerd voor grafiek 2 *)
  END
  ELSE BEGIN
    invertrect(r);
    y2:=200;
    klasselijst2bool[i]:=false;
  END;
  END;
  END
ELSE BEGIN (* definieren rechthoek om volgende element uit dialogbox *)
  i:=i+1;
  j:=j+1;
  y1:=y2;
  y2:=y2+12;
  setrect(r,x1,y1,x2,y2);
  END;
  END;
  END;
  WITH whatHappened DO BEGIN
  IF (whichcontrol=NIEUWEGRAFIEK.TABEL) AND OefeningenGeselecteerd THEN
  BEGIN
    SwitchWindow(TABELOEF.window);
    hidecontrol(tabeloef.vorderingen);
  END;
  IF (whichcontrol=NIEUWEGRAFIEK.TABEL) AND NOT(OefeningenGeselecteerd) THEN
  BEGIN
    SwitchWindow(TABELGRAFEEM.window);
    hidecontrol(tabelgrafeem.vorderingen);
  END;
  IF whichcontrol=NIEUWEGRAFIEK.VERWIJDER THEN
  BEGIN
    grafiektwee:=false; (* de tweede grafiek wordt verwijderd *)
    for i:=1 to 30 do klasselijst2bool[i]:=false; (* de selecties worden ongedaan gemaakt *)
    switchwindow(GRAFIEK.window);
    hidecontrol(GRAFIEK.nieuwgraf2);
    showcontrol(GRAFIEK.grafiek2);
  END;
  IF whichcontrol=NIEUWEGRAFIEK.GRAFIEK THEN
  BEGIN
    SwitchWindow(GRAFIEK.window);
    IF grafiektwee =true THEN BEGIN (* er is een tweede grafiek *)

```

```

        hidecontrol(GRAFIEK.grafiek2);
        showcontrol(GRAFIEK.nieuwgraf2);
    END
    ELSE BEGIN      (* er is maar een grafiek *)
        showcontrol(GRAFIEK.grafiek2);
        hidecontrol(GRAFIEK.nieuwgraf2);
    END;
END;
IF whichcontrol=NIEUWEGRAFIEK.SCROLLNAAM THEN  (* er is op de scrollbar geklikt, opnieuw
                                                vullen van de dialogbox *)
BEGIN
    scroll:=getctlvalue(nieuwegrafiek.scrollnaam);
    setrect(r,144,108,327,191);
    erasereact(r);
    IF grafiek2=true
    THEN schrijfdialogbox(klasselijst,klasselijst2bool,144,327,113,192,vulklas,scroll)
    ELSE schrijfdialogbox(klasselijst,klasselijstbool, 144,327,113,192,vulklas,scroll);
END;
END;
END;

(* HET AFHANDELEN VAN HET KLIKKEN OP BUTTONS EN SCROLLBARS IN HET SCHERM VOOR HET SELECTEREN
VAN OEFENINGEN EN WOORDEN/GRAFEMEN *)
PROCEDURE handleOEFWOORD;
VAR r:rect;
    p:point;
    x1,x2,y1,y2,i,j:integer;
BEGIN
    IF event.what=mousedown  (* is er met de muis geklikt *)
    THEN BEGIN
        p:=event.where;      (* punt waar geklikt is *)
        j:=1;
        globaltolocal(p);    (* punt tov linkerbovenhoek window *)
        i:=getctlvalue(oefwoord.scrolloef); (* uitlezen van de waarde van de scrollbar *)
        x1:=157;
        x2:=369;
        y1:=140+12*(j-1);
        y2:=140+12*j;
        setrect(r,x1,y1,x2,y2); (* definieren rechthoek om eerste element uit de dialogbox *)
        WHILE y2<= 217 DO
        BEGIN
            IF ptinrect(p,r)      (* er is in de rechthoek geklikt *)
            THEN BEGIN
                IF oefeningengeselecteerd=true
                THEN BEGIN
                    IF oefeningbool[i]=false
                    THEN BEGIN
                        penpat(white);
                        paintrect(r);
                        placestring(x1+5,y1+10,oefeningen[i]);
                        invertrect(r);
                        y2:=300;
                        oefeningbool[i]:=true;
                    END
                ELSE BEGIN
                    invertrect(r);
                    y2:=300;
                    oefeningbool[i]:=false;
                END;
            END;
        END
        ELSE BEGIN
            IF woordenbool[i]=false
            THEN BEGIN
                penpat(white);
                paintrect(r);
                placestring(x1+5,y1+10,woorden[i]);
                invertrect(r);
            END;
        END;
    END;
END;

```



```

        y2:=300;
        woordenbool[i]:=true;
    END
ELSE BEGIN
    invertrect(r);
    y2:=300;
    woordenbool[i]:=false;
    END;
    END;
END
ELSE BEGIN
    i:=i+1;
    j:=j+1;
    y1:=y2;
    y2:=y2+12;
    setrect(r,x1,y1,x2,y2); (* definieren rechthoek om volgende element uit dialogbox *)
    END;
    END;
END;
WITH whatHappened DO BEGIN
    IF (whichcontrol=OEFWOORD.TABEL) AND OefeningenGeselecteerd THEN
    BEGIN
        SwitchWindow(TABELOEF.window);
        hidecontrol(tabeloef.vorderingen);
    END;
    IF (whichcontrol=OEFWOORD.TABEL) AND NOT(OefeningenGeselecteerd) THEN
    BEGIN
        SwitchWindow(TABELGRAFEEM.window);
        hidecontrol(tabelgrafeem.vorderingen);
    END;
    IF whichcontrol=OEFWOORD.GRAFIEK THEN SwitchWindow(GRAFIEK.window);
    IF whichcontrol=OEFWOORD.OEFENING THEN
    BEGIN
        OefeningenGeselecteerd:=TRUE;
        SetCtlValue(OEFWOORD.OEFENING,1);
        SetCtlValue(OEFWOORD.oef_woord,0);
        setctlmax(oefwoord.scrolloef,5);
        scroll1:=getctlvalue(oefwoord.scrolloef);
        setrect(r,157,135,369,216);
        eraserect(r);
        schrijfdialogbox(oefeningen,oefeningbool,157,369,140,217,vuloeff,scroll1);
    END;
    IF whichcontrol=OEFWOORD.OEF_WOORD THEN
    BEGIN
        OefeningenGeselecteerd:=FALSE;
        SetCtlValue(OEFWOORD.OEFENING,0);
        SetCtlValue(OEFWOORD.oef_woord,1);
        setctlmax(oefwoord.scrolloef,vulwoord-6);
        scroll1:=getctlvalue(oefwoord.scrolloef);
        setrect(r,157,135,369,216);
        eraserect(r);
        schrijfdialogbox(woorden,woordenbool,157,369,140,217,vulwoord,scroll1);
    END;
    IF whichcontrol = OEFWOORD.SCROLLOEF THEN
    BEGIN
        IF oefeningengeselecteerd=true
        THEN BEGIN
            setctlmax(oefwoord.scrolloef,5);
            scroll1:=getctlvalue(oefwoord.scrolloef);
            setrect(r,157,135,369,216);
            eraserect(r);
            schrijfdialogbox(oefeningen,oefeningbool,157,369,140,217,vuloeff,scroll1);
        END
        ELSE BEGIN
            setctlmax(oefwoord.scrolloef,vulwoord-6);
            scroll1:=getctlvalue(oefwoord.scrolloef);
            setrect(r,157,135,369,216);
            eraserect(r);
        END
    END

```

```

        schrijfdialoogbox(woorden,woordenbool,157,369,140,217,vulwoord,scroll1);
    END;
END;
END;

(* HET AFHANDELEN VAN HET KLIKKEN OP DE BUTTONS IN HET SCHERM VAN DE TABEL MET DE OEFENINGEN *)
PROCEDURE handleTABELOEF;
BEGIN
    WITH whatHappened DO BEGIN
        IF whichcontrol=TABELOEF.TEACHER THEN SwitchWindow(TEACHER.window);
        IF whichcontrol=TABELOEF.GRAFIEK THEN
            BEGIN
                SwitchWindow(GRAFIEK.window);
                if grafiek2 =true THEN BEGIN
                    hidecontrol(GRAFIEK.grafiek2);
                    showcontrol(GRAFIEK.nieuwgraf2);
                END
                ELSE BEGIN
                    showcontrol(GRAFIEK.grafiek2);
                    hidecontrol(GRAFIEK.nieuwgraf2);
                END;
            END;
        IF whichcontrol=TABELOEF.VORDERINGEN THEN SwitchWindow(VORDERINGEN.window);
    END;
END;

(* HET AFHANDELEN VAN HET KLIKKEN OP DE BUTTONS IN HET SCHERM VAN DE TABEL MET DE WOORDEN *)
PROCEDURE handleTABELGRAFEEM;
BEGIN
    WITH whatHappened DO BEGIN
        IF whichcontrol=TABELGRAFEEM.TEACHER THEN SwitchWindow(TEACHER.window);
        IF whichcontrol=TABELGRAFEEM.GRAFIEK THEN
            BEGIN
                SwitchWindow(GRAFIEK.window);
                IF grafiek2 =true THEN BEGIN
                    hidecontrol(GRAFIEK.grafiek2);
                    showcontrol(GRAFIEK.nieuwgraf2);
                END
                ELSE BEGIN
                    showcontrol(GRAFIEK.grafiek2);
                    hidecontrol(GRAFIEK.nieuwgraf2);
                END;
            END;
        IF whichcontrol=TABELGRAFEEM.VORDERINGEN THEN SwitchWindow(VORDERINGEN.window);
    END;
END;

(* HET INITIALISEREN VAN VARIABELEN *)
PROCEDURE Init;
BEGIN
    IkWilStoppen:=FALSE;
    OefeningenGeselecteerd:=TRUE;
    SetRect(helprect,458,0,497,51);
    SetRect(routerect,17,0,54,51);
    helpselected:=false;
    routeselected:=false;
END; (* END OF INIT *)

(* HET INLEZEN VAN DE FILE MET LEERLINGEN EN DE WOORDEN/GRAFEMEN EN OEFENINGEN *)
PROCEDURE Inlezen (VAR klasselijst, woorden: arstring; VAR vulklas, vulwoord, vuloef: integer;
    VAR oefeningen:arstring);
VAR infile, infile1, infile2: text;
    i      : integer;

```

```

BEGIN
  reset(infile, 'user');      (* inlezen van de file met leerlingnamen *)
  i := 1;
  REPEAT
    readln(infile, klasselijst[i]);
    readln(infile);
    i:=i+1;
  UNTIL EOF(infile);
  close(infile);
  vulklas:=i-1;
  reset(infile1, 'oefeningen');      (* inlezen van de file met oefeningen *)
  i := 1;
  REPEAT
    readln(infile1, oefeningen[i]);
    i:=i+1;
  UNTIL EOF(infile1);
  close(infile1);
  vulloef :=i-1;
  reset(infile2, 'woorden');      (* inlezen van de file met woorden en grafemen *)
  i := 1;
  REPEAT
    readln(infile2, woorden[i]);
    i:=i+1;
  UNTIL EOF(infile2);
  close(infile2);
  vulwoord:=i-1;
END; (* END OF INLEZEN *)

(* ----- *)
(*           MAIN PROGRAM           *)
(* ----- *)

BEGIN
  xtendinit;
  MakeWindow;
  grafiek2:=false;
  for i:=1 to 30 do klasselijstbool[i]:=false; (* lijst met leerlingen voor grafiek 1 *)
  for i:=1 to 30 do klasselijst2bool[i]:=false; (* lijst met leerlingen voor grafiek 2 *)
  for i:=1 to 30 do oefeningbool[i]:=false; (* lijst met geselecteerde woorden en grafemen *)
  for i:=1 to 30 do woordenbool[i]:=false; (* lijst met geselecteerde oefeningen *)
  Inlezen(klasselijst, woorden, vulklas, vulwoord, vulloef, oefeningen);
  Init;
  InstallButtons;
  REPEAT
    REPEAT
      SystemTask;
    UNTIL XTgetNextEvent(everyEvent, event);
    HandleEvent(event, WhatHappened);
    If event.what=updateevt THEN UpdateWindows;
    IF FrontWindow=TEACHER.window THEN HandleTEACHER;
    IF FrontWindow=VORDERINGEN.WINDOW THEN handleVORDERINGEN;
    IF FrontWindow=GRAFIEK.window THEN handleGRAFIEK;
    IF FrontWindow=NIEUWEGRAFIEK.window THEN handleNIEUWEGRAFIEK;
    IF FrontWindow=OEFWOORD.window THEN handleOEFWOORD;
    IF FrontWindow=TABELOEF.window THEN handleTABELOEF;
    IF FrontWindow=TABELGRAFEEM.window THEN handleTABELGRAFEEM;
  UNTIL ExitRequest(WhatHappened) or IkWilStoppen;
END.

```

```
UNIT TEACHERdefs;
```

```
INTERFACE  
TYPE
```

```
TeacherWindowRecord=RECORD  
  rec   : WindowRecord;  
  wrect : Rect;  
  window: WindowPtr;  
  (* BUTTONS *)  
  stop,  
  vorderingen,  
  agenda,  
  klasselijst,  
  demonstratie,  
  training_starten,  
  instructie,  
  help : ControlHandle;  
  (* AFMETINGEN VAN BUTTONS *)  
  stopR,  
  vorderingenR,  
  agendaR,  
  klasselijstR,  
  demonstratieR,  
  training_startenR,  
  instructieR,  
  helpR: Rect;  
END;
```

```
vorderingenwindow=RECORD  
  rec   : Windowrecord;  
  wrect : Rect;  
  window: WindowPtr;  
  (* BUTTONS EN SCROLLBARS *)  
  tabel,  
  grafiek,  
  teacher,  
  oefening,  
  grafeem,  
  scrollnaam,  
  scrollgraoef,  
  help,  
  route : ControlHandle;  
  (* AFMETINGEN VAN BUTTONS EN SCROLLBARS *)  
  tabelR,  
  grafiekR,  
  teacherR,  
  oefeningR,  
  grafeemR,  
  scrollnaamR,  
  scrollgraoefR,  
  helpR,  
  router : Rect;  
END;
```

```
grafiekwindow=RECORD  
  rec   : Windowrecord;  
  wrect : Rect;  
  window: WindowPtr;  
  (* BUTTONS *)  
  graf1,  
  graf2,  
  nieuwgraf1,  
  nieuwgraf2,  
  grafiek2,  
  tabel,  
  oef_woord,  
  teacher,
```

```
    help,
    route : ControlHandle;
    (* AFMETINGEN *)
    graf1R,
    graf2R,
    nieuwgraf1R,
    nieuwgraf2R,
    grafiek2R,
    tabelR,
    oef_woordR,
    teacherR,
    helpR,
    router : Rect;
END;

nieuwgrafiekwindow=RECORD
    rec : Windowrecord;
    wrect : Rect;
    window: WindowPtr;
    (* BUTTONS *)
    tabel,
    grafiek,
    scrollnaam,
    help,
    route,
    verwijder: ControlHandle;
    (* AFMETINGEN *)
    tabelR,
    grafiekR,
    scrollnaamR,
    helpR,
    router,
    verwijderR: Rect;
END;

oefwoordwindow=RECORD
    rec : Windowrecord;
    wrect : Rect;
    window: WindowPtr;
    (* BUTTONS *)
    tabel,
    grafiek,
    demonstratie,
    oefening,
    oef_woord,
    scrolloef,
    help,
    route : ControlHandle;
    (* AFMETINGEN *)
    tabelR,
    grafiekR,
    demonstratieR,
    oefeningR,
    oef_woordR,
    scrolloefR,
    helpR,
    router : Rect;
END;

tabeloefwindow=RECORD
    rec : Windowrecord;
    wrect : Rect;
    window: WindowPtr;
    (* BUTTONS *)
    grafiek,
    teacher,
    vorderingen,
    help,
```

```
route : ControlHandle;
(* AFMETINGEN *)
grafiekR,
teacherR,
vorderingenR,
helpR,
router: Rect;
END;
```

```
tabelgrafeemwindow=RECORD
  rec  : Windowrecord;
  wrect : Rect;
  window: WindowPtr;
  (* BUTTONS *)
  grafiek,
  teacher,
  vorderingen,
  help,
  route : ControlHandle;
  (* AFMETINGEN *)
  grafiekR,
  teacherR,
  vorderingenR,
  helpR,
  router : Rect;
END;
```

```
CONST
```

```
  dummy=0;
```

```
IMPLEMENTATION
END.
```