

Evalueren en onderrichten van deeltaalvaardigheid: de staartdeelmachine

Karel Hurts & Kees van Putten
Universiteit Leiden

hurts@euronet.nl
putten@fsw.leidenuniv.nl

Overzicht werkgroep

- 10:15 – 10:25: Kennismaken/voorstellen
- 10:25 – 10:40: Achtergrond SDM-project + demo1 (Hurts)
- 10:40 – 11:10: Bespreking pilot study + experiment 1 + demo2 (Hurts)
- 11:10 – 11:30: Bespreking experiment 2 + demo3 (Van Putten)
- 11:30 – 11:45: Discussie (stellingen) (Van Putten)

Doelen voor referentieniveau 1 (tot 12 jaar) (domein getallen/bewerking delen)

Niveau 1F (fundament)	Niveau 1S (streef)
<p>Hoofdrekenen:</p> <ul style="list-style-type: none"> • Delingen uit de tafels (tot en met 10) uitrekenen: $45 : 5 =$, $32 : 8 =$ • uit het hoofd delen met "nullen", ook met eenvoudige decimale getallen: $3600 : 100 =$ <p>Efficiënt rekenen:</p> <ul style="list-style-type: none"> • Gebruik maken van eigenschappen van getallen en bewerking, met eenvoudige getallen <p>Delen met rest:</p> <ul style="list-style-type: none"> • Getallen met maximaal drie cijfers delen door een getal met maximaal 2 cijfers, al dan niet met een rest: $132 : 16 =$ 	<p>Hoofdrekenen:</p> <ul style="list-style-type: none"> • Delingen uit de tafels (tot en met 10) uit het hoofd kennen, ook met complexere getallen en decimale getallen: $18 : 100$ <p>Efficiënt rekenen:</p> <ul style="list-style-type: none"> • Efficiënt rekenen ook met grotere getallen <p>Delen met rest:</p> <ul style="list-style-type: none"> • Delen met rest of (afgerond) decimaal getal: $122 : 5 =$ <p>Schattend rekenen, rest in context plaatsen:</p>

(Bron: eindrapportage Expertgroep Doorlopende Leerlijnen Taal en Rekenen)

Doelen voor referentieniveau 1 (vervolg)

Niveau 1F (fundament)	Niveau 1S (streef)
<p>Schattend rekenen, rest in context plaatsen:</p> <ul style="list-style-type: none"> • In contexten de "rest" (bij delen met rest) interpreteren of verwerken • Globaal (benaderend) rekenen (schatten) als de context zich daartoe leent of als controle voor rekenen met de rekenmachine • Interpreteren van een uitkomst 'met rest' bij gebruik van een rekenmachine 	

(Bron: eindrapportage Expertgroep Doorlopende Leerlijnen Taal en Rekenen)

Kenmerken progressief schematiseren

- Delen beschouwen als basisoperatie van herhaald aftrekken
- Basisoperatie gaandeweg verkorten en schematiseren

Voordelen:

- Inzichtelijk, leerbaar, flexibel

Nadelen:

- Voor grotere sommen erg bewerkelijk
 - Veel operaties (ook: optellen, aftrekken, vermenigvuldigen)
 - Bewerkingen nemen veel ruimte in beslag
- Niet duidelijk hoe verkorten het beste geïnstrueerd/geleerd kan worden

Herdefinitie van delen als cognitieve vaardigheid

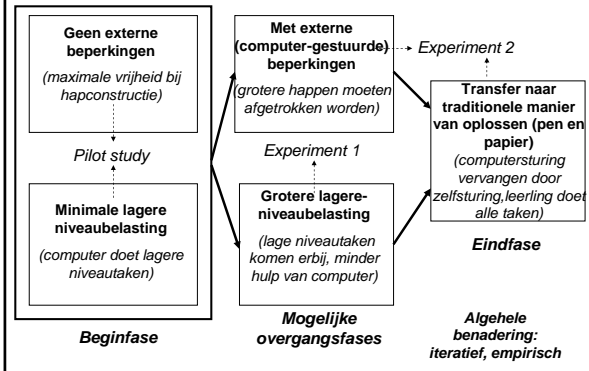
- Weten dat doel van bewerking is om *eerlijk te verdelen* (of op te delen),
-dat dit o.a. bereikt kan worden door te *splitzen en herhaaldelijk af te trekken*
- en dat bepaalde grootheden ook *afgetrokken, opgeteld en vermenigvuldigd* moeten worden,
-waarbij leerling een *supervisetaak* heeft,
 - Samenstellen van grootte van af te trekken hap (= aantal keer deler) via schatten of weten
 - Bepalen wanneer oplossing bereikt is en wat de eventuele rest is
 - Weten wanneer welke grootheden opgeteld, afgetrokken en vermenigvuldigd moeten worden
-en na afloop moet (kunnen) *reflecteren* op (inefficiëntie en kwaliteit van gekozen oplossingspad.
 - Mogelijkheid tot leren
 - Eventueel uitmondend in klassieke algoritme (staartdelen)
 - Checken oplossing via schatten

SDM- theoretische principes



- Cognitive off-loading
- Cognitive apprenticeship/coöperatief leren
- Guided discovery/learning through exploration

SDM-gebaseerd leren delen: mogelijke leerfasen



SDM1 - kenmerken (gebruikt in pilot study)



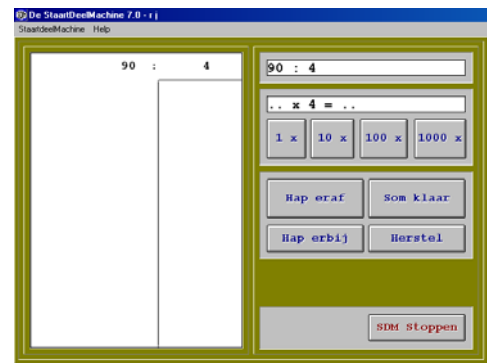
Minimale beperkingen bij hapconstructie:

- Beperking 1: maximaal 9 keer klikken op een [1x], [10x] of [100x] button
- Beperking 2: leerling ontvangt waarschuwing bij aftrekken van te grote hap
- Hulp van vermenigvuldigbalk bij construeren hap

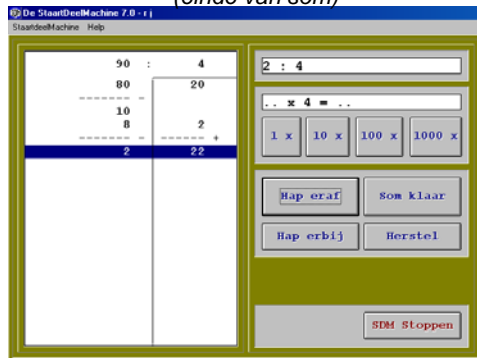
Maximale ondersteuning door computer

- Aftrekken, optellen en vermenigvuldigen wordt door SDM gedaan
- Kortste oplossing sommen kan ook gedemonstreerd worden

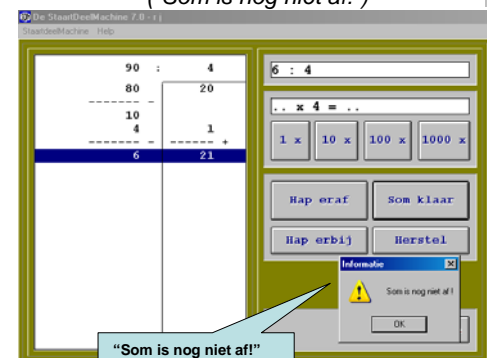
Screenshot van SDM1 (begin van som)

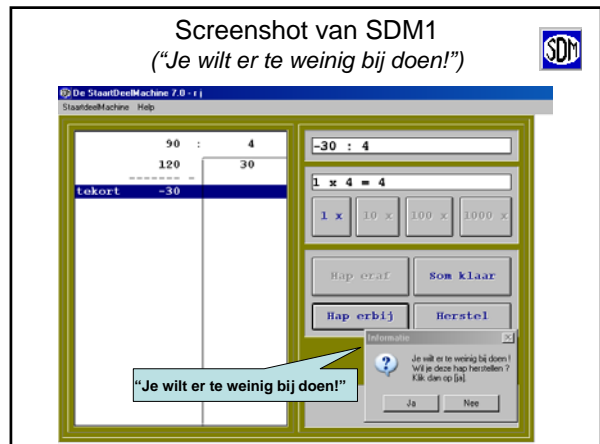
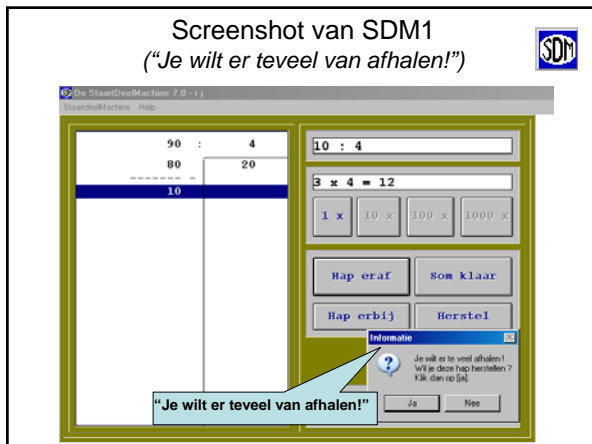


Screenshot van SDM1 (einde van som)



Screenshot van SDM1 ("Som is nog niet af!")





Vragen van pilot study
(gebruikmakend van SDM1)

- Kunnen leerlingen leren delen met SDM?
- Kunnen leerlingen SDM gebruiken?
- Hoe komen leerlingen tot oplossing? (efficiëntie, strategie)

Methode van pilot study (1)

- Zes leerlingen van klas 1 van IBO-school (leeftijd 12 - 15 jaar)
- 10 makkelijke sommen, 10 moeilijke sommen, opgelost met SDM in 2 sessies van ca. 20 min
- $6 < \text{deler} < 14$
- Sessie 1 (makkelijke sommen): $30 < \text{deeltal} < 172$
- Sessie 2 (moeilijke sommen): $100 < \text{deeltal} < 3000$
- Sommen random gegenereerd door SDM

Methode van pilot study (2)

- **Instructie:** aan begin van sessie 2 werd leerlingen gevraagd "niet te beginnen met 1x deler eraf, indien mogelijk"
- **Registratie:** SDM houdt bij waar op geklikt is en wanneer. Hieruit konden gedragsmaten afgeleid worden, o.a:
 - Oploskwaliteit**
 - Aantal oplossingsstappen per som
 - Aantal keer per som dat een te grote hap door SDM gecorrigeerd werd (*Machine Initiated Overshooting Correction of MIOC*)
 - Aantal "Niet-Klaar" fouten
 - Oplossingsstrategie**
 - Aantal keer per som dat een te grote hap door leerling zelf gecorrigeerd werd (SIOC)
 - Aantal keer per som dat een te kleine hap door leerling zelf gecorrigeerd werd (SIUC)
- **Benodigde tijd**, per klik, per hap en voor de hele som

Resultaten pilot study (1)

- Efficiëntie makkelijke sommen:
 - Gemiddeld slechts 0.16 overbodige hap per som per leerling (ca. 11% van minimum)
- Efficiëntie moeilijke sommen:
 - Gemiddeld 0.30 overbodige hap per som per leerling (ca. 12% van minimum)
- Herstel van te grote of kleine happen:
 - Bijna alle te grote happen door leerling zelf hersteld
 - Dit gebeurde in gemiddeld 60% van alle sommen, met name bij de moeilijker opgaven
 - Zelf-herstel van te kleine happen kwam nauwelijks voor

Resultaten pilot study (2)



- Leerlingstrategie: proberen zo groot mogelijke hap af te trekken en deze herstellen wanneer hij te groot is
 - Deze strategie waarschijnlijk aangemoedigd door aanwezigheid van vermenigvuldigbalk
 - Deze strategie levert wel efficiënte oplossingen op...
 - ...maar deze strategie is moeilijk te gebruiken wanneer vermenigvuldigbalk afwezig is!

Vragen van Experiment 1 (gebruikmakend van SDM2)



- Zijn oplossingen nog steeds efficiënt als vermenigvuldigbalk *niet* gebruikt kan worden?
- Wat is het voordeel van het hebben van veel vermenigvuldigkennis? Wordt daardoor de stijl/kwaliteit van oplossen beter?

Screenshot van SDM2

(zonder vermenigvuldigbalk, rest als bij SDM1)



Uitkomst van vermenigvuldiging wordt niet weergegeven

Methode van Experiment 1 (1)



- 36 leerlingen uit groep 7 van basisschool (leeftijd 10 - 12 jaar)
- 16 leerlingen in "stille" groep, 20 leerlingen in "hardop-denken" groep
- Stille groep: leerlingen werkten individueel
- In hardop-denken-groep (na hapconstructie, maar voor aftrekken hap): "Hoeveel blijft er over van deeltal na aftrekken van deze hap?"
 - Antwoord van voorspelling opgeschreven door proefleider (zonder feedback)
 - Voorspelling omgezet in score voor "vermenigvuldigkennis"

Methode van Experiment 1 (2)



- Eerst 10 makkelijke sommen, daarna 10 moeilijke sommen opgelost met SDM in 2 sessies van ca. 20 min
- Makkelijke sommen: $6 < \text{deler} < 9$, $78 < \text{deeltal} < 162$
- Moeilijke sommen: $11 < \text{deler} < 19$, $96 < \text{deeltal} < 293$
- Alle leerlingen begonnen met 4 oefensommen; pas als ze alle begrepen hadden kregen ze de eerste "echte" som

Resultaten Experiment 1 (1)



Kwaliteit oplossing:

- Geen verschillen tussen stille en hardop-denkgroep ten aanzien van kwaliteit en stijl van oplossen
- Gemiddeld aantal overbodige oplossingsstappen per som per leerling was 0.27 (ca. 18% van minimum)
 - Dit is slechter dan in pilot study
- Bij gemiddeld 27% van alle pogingen een hap af te trekken was deze te groot → hersteld door SDM
 - Dit is vaker dan in pilot study

Strategiegebruik:

- Leerlingen herstelden zichzelf nauwelijks voor overshooting (hap te groot), maar wel regelmatig voor undershooting (hap te klein)

Resultaten Experiment 1 (2)



Strategiegebruik (vervolg)

- Interpretatie leerlingstrategie: voorzichtig beginnen met kleine happen, maar wel proberen deze zo groot mogelijk te maken; je laten herstellen door SDM wanneer ze te groot zijn
 - Deze strategie leidt tot minder efficiënte oplossingen dan in pilot study
 - Maar deze strategie bereidt beter voor op traditionele situatie (pen en papier, geen computer)!
 - Deze strategie waarschijnlijk aangemoedigd door afwezigheid van vermenigvuldigbalk

Resultaten Experiment 1 (3)



Vermenigvuldigkennis:

- Bij moeilijker opgaven nam vermenigvuldigkennis af
- Leerlingen met grotere vermenigvuldigkennis bereikten efficiëntere oplossingen en probeerden minder vaak een te grote hap af te trekken
- Geen verband tussen aantal oplosstappen en aantal keren een te grote hap willen aftrekken → oplossingen zijn kennisgestuurd en niet het gevolg van bewust gebruik van overshooting-strategie

Resultaten Experiment 1 (3)



	Sol. steps		MIOC	
	First session	Second session	First session	Second session
<i>Silent</i>				
Boys ($n = 9$)	0.17 (0.15)	0.28 (0.25)	0.18 (0.50)	0.23 (0.41)
Girls ($n = 7$)	0.27 (0.41)	0.31 (0.33)	0.24 (0.31)	0.44 (0.43)
<i>Think-aloud</i>				
Boys ($n = 12$)	0.19 (0.15)	0.23 (0.37)	0.06 (0.10)	0.14 (0.43)
Girls ($n = 8$)	0.15 (0.18)	0.26 (0.72)	0.13 (0.24)	0.10 (0.21)

Sol steps: gemiddeld aantal overbodige oplosstappen per som per leerling

MIOC: gemiddeld aantal *machine-initiated overshooting corrections* per som per leerling

Resultaten Experiment 1 (4)



	SIUC		SIOC	
	First session	Second session	First session	Second session
<i>Silent</i>				
Boys ($n = 9$)	0.02 (0.04)	0.02 (0.04)	0.01 (0.03)	0.01 (0.03)
Girls ($n = 7$)	0.01 (0.04)	0.03 (0.05)	0.00 (0.00)	0.01 (0.04)
<i>Think-aloud</i>				
Boys ($n = 12$)	0.13 (0.34)	0.14 (0.31)	0.00 (0.00)	0.01 (0.03)
Girls ($n = 8$)	0.11 (0.15)	0.08 (0.10)	0.05 (0.08)	0.03 (0.05)

SIUC: gemiddeld aantal *student-initiated undershooting corrections* per som per leerling

SIOC: gemiddeld aantal *student-initiated overshooting corrections* per som per leerling

Resultaten Experiment 1 (5)



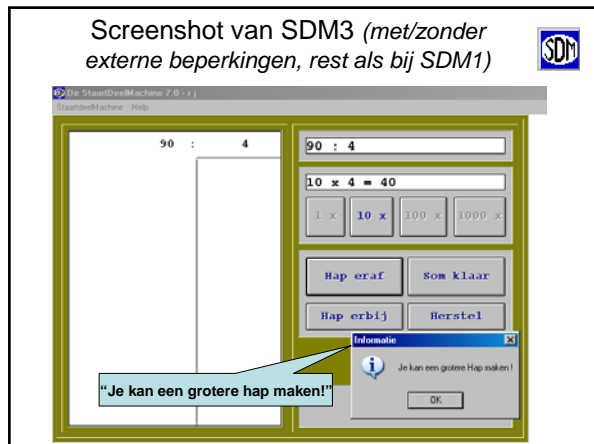
	Pred. acc.	
	First session	Second session
<i>Silent</i>		
Boys ($n = 9$)	–	–
Girls ($n = 7$)	–	–
<i>Think-aloud</i>		
Boys ($n = 12$)	8.94 (1.62)	8.18 (2.11)
Girls ($n = 8$)	9.70 (0.49)	8.19 (1.82)

Pred. acc.: *prediction accuracy* (op schaal van 1 -10), gemiddelde nauwkeurigheid van voorspellingen over hapgrootte

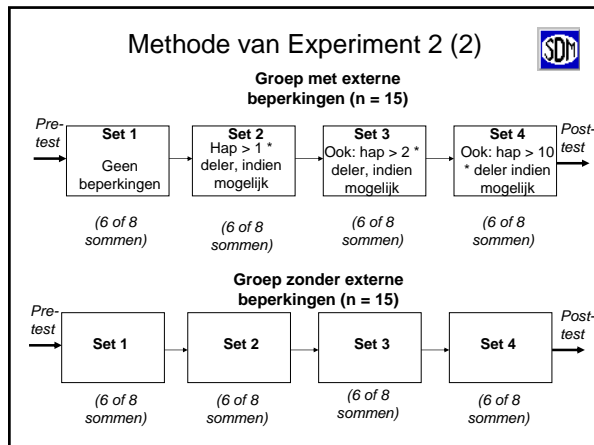
Vragen van Experiment 2 (gebruikmakend van SDM3)



- Kunnen leerlingen leren om efficiëntere oplossingen te maken (grotere happen maken) als ze daartoe (tijdelijk) gedwongen worden?
- Hoeveel en hoe snel kunnen leerlingen leren door herhaaldelijk te werken met de SDM?
- Kan SDM-ervaring ten goede komen aan het oplossen van sommen op traditionele manier (pen en papier)? (vraag naar *transfer of learning*)



- ### Methode van Experiment 2 (1)
- 29 leerlingen uit groepen 5 en 6 van basisschool
 - Meer dan 50% leerlingen van allochtone afkomst
 - Helft leerlingen werden in "externe beperkingen" (EB) - groep geplaatst, overige leerlingen in "geen externe beperkingen" (GEB) - groep
 - Alle sommen gemaakt in 3 sessies van 20 min
 - Pre- en posttest gebruikt om leereffecten te meten. Beide testen bestonden uit 22 opgaven
 - Opgaven vergelijkbaar maar niet identiek in pretest en posttest
 - Groepen gematcht ten aanzien van algemene rekenvaardigheid
 - Vermenigvuldigbalk weer aanwezig in SDM-interface



- ### Resultaten Experiment 2 (1)
- Oplossingsgedrag overall:
- Gemiddeld aantal overbodige oplossingsstappen per som = 2.5 (~100% van minimum aantal stappen)
 - Dit is slechter dan in Experiment 1
 - Zelf-herstel van fouten gebeurde relatief vaak (gemiddeld 1.3 keer per som per leerling)
 - Bij 14% van de sommen werden "Niet-Klaar" fouten gemaakt.
 - Gemiddeld probeerden leerlingen 1 keer per som een te grote hap af te trekken
 - Dit is vaker dan bij Experiment 1
- Transfer:
- Leerlingen presteerden significant beter op posttest dan op pretest
 - Antwoorden werden sneller gegeven op posttest
 - Minder foutieve antwoorden gegeven op posttest

- ### Resultaten Experiment 2 (2)
- Effecten van oefening:
- Binnen elke set werden sommen beter gemaakt naarmate meer geoefend was
 - Minder tijd nodig om oplossing te vinden
 - Minder oplosstappen per som
 - Minder zelf-herstel van fouten (vooral in set 1 en 2)
 - Minder geneigd te grote hap af te trekken, behalve in set 4
 - Over hele experiment ging gemiddeld aantal overbodige oplosstappen van 130% van minimum (set 1) naar 86% van minimum (set 4), ondanks dat sommen moeilijker waren in set 4!

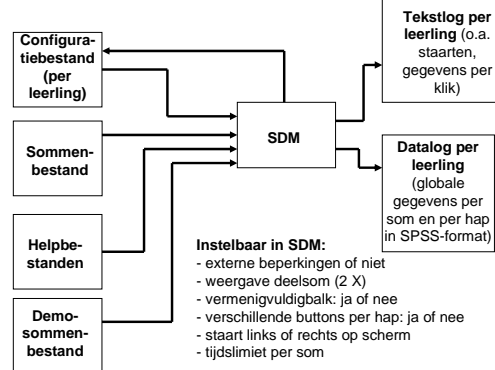
- ### Resultaten Experiment 2 (3)
- Effecten van externe beperkingen
- Geen verschil in transfer tussen EB-groep en GEB-groep
 - Sommen werden bij EB-groep efficiënter gemaakt (minder oplosstappen), en met meer zelf-herstel van fouten dan bij GEB-groep
 - Dit effect niet alleen het gevolg van het feitelijk gecorrigeerd worden door SDM → de "dreiging" hiervan werkt ook al!
 - Effecten van oefening niet verschillend voor EB-groep en GEB-groep

Conclusies



- Leerlingen van pilot study het "slimst": kleinste aantal oplosstappen, kleinste aantal SDM-correcties
 - Oudere leerlingen
 - Ook mogelijk door slim gebruik van vermenigvuldigbalk
- Leerlingen van experiment 2 het "domst": grootste aantal oplosstappen, grootste aantal SDM-correcties, grootste aantal "Niet-Klaar" fouten
 - Ondanks aanwezigheid van vermenigvuldigbalk!
 - Relatief jonge leerlingen
 - Veel leerlingen van allochtone afkomst
- Zelf-herstel gebeurde ook het vaakst in experiment 2
 - Deze leerlingen zijn zich wel bewust wanneer hap te groot of te klein is
- Zelf-herstel van te kleine happen kwam nauwelijks voor in pilot study, vaker in experiment 1
 - Mogelijk effect van aan- of afwezigheid van vermenigvuldigbalk

Schema gebruik SDM



Stellingen voor discussie



1. Cijferend delen is een belangrijke vaardigheid voor diverse praktische beroepen waarin veel leerlingen terecht komen. De gebrekkige aandacht voor cijfermatig delen (incl. klassiek staartdelen) in veel huidige rekenmethoden is dus onterecht.
2. Om meer tijd in het curriculum vrij te maken voor cijferend delen zouden andere onderdelen die nu veel tijd kosten misschien minder zwaar benadrukt kunnen worden, zoals ...?
3. De SDM is in principe geschikt om het maken van kale, kolomsgewijze deelsommen sneller te leren, en daardoor de overgang naar het klassieke staartdeelalgoritme te bespoedigen.
4. Open discussiepunt: Hoe zouden didactische verbeteringen aan de SDM aangebracht kunnen worden?
 - Meer individuele benadering bij opleggen extrinsieke beperkingen
 - Hulp bij vergroten inzicht getalstelsel
 - Meer uitleg bij reflectie op efficiëntie van oplossing
 - Anders, nl.....