

# Het volle potentieel van de computer leren benutten: over informatica en computational thinking

DOOR ERIK BARENSEN EN MARTIN BRUGGINK

**H**et schoolvak informatica in de bovenbouw van havo en vwo is vernieuwd. Het vak is grondig herzien, waardoor het schoolvak beter aansluit bij het uitdagende en brede vakgebied dat informatica is. De leerlingen die nu in de 4<sup>e</sup> klas zitten zijn de eersten die het nieuwe examenprogramma volgen.

Hoe ziet het nieuwe schoolvak eruit, en wat heeft informatica met andere vakken te maken? Welke rol speelt het mysterieuze begrip *computational thinking*? Daarover gaat dit artikel.

## Informatica

Informatica is het vakgebied achter de digitale wereld. Het is een echte wetenschap, die gaat over fundamentele concepten zoals algoritmen, gegevens en netwerken, maar ook over het ontwerpen en bouwen van digitale producten en over de interactie tussen die producten en de mensen die ze gebruiken. Informatica bestaat al een tijdje, maar heeft zich door recente technologische mogelijkheden verder ontwikkeld met interessante nieuwe takken zoals *cybersecurity*, over beveiliging van digitale gegevens en computersystemen, en *data science*, over het analyseren van patronen in (vaak ongestructureerde) data en het verkrijgen van zinvolle informatie daaruit.

Het vakgebied informatica heeft verschillende gezichten. Je kunt er een funderende wetenschap met wortels in de wetenschap in herkennen, maar ook een 'construerende discipline' die concepten en methoden levert voor het maken van digitale producten. In het schoolvak staat dat *maakaspect* voorop, ook al kunnen leerlingen die dat willen zich daarnaast in meer theoretische onderwerpen verdiepen. Informatica is daarmee een

modern vak geworden, dat leerlingen uitdaagt om in samenwerking (informatici werken zelden alleen!) bij te dragen aan digitale oplossingen voor echte problemen en hen helpt begrijpen welke mogelijkheden en beperkingen computers bieden. En dat is bijzonder nuttig, in welk beroep ze later ook terechtkomen.

De breedte van het vakgebied wordt weerspiegeld in de opzet van het vernieuwde schoolvak (zie de tabel). Leerlingen die informatica kiezen, volgen een *kernprogramma* met basisconcepten en -vaardigheden zoals algoritmen en programmeren. Verder kiezen ze twee (havo) of vier (vwo) onderwerpen uit een lijst van zogenaamde *keuzethema's*. Daarmee geven ze een persoonlijke kleur aan hun programma. De keuzethema's lopen uiteen: zo zijn er technische onderwerpen zoals netwerken en *physical computing* (over het besturen van apparaten zoals robots), maar ook meer mens- en maatschappijgerichte thema's zoals *user experience* en de maatschappelijke en individuele invloed van informatica. Informatica is dan ook een vak dat door leerlingen uit alle profielen kan worden gekozen.

Samenwerken met mensen die verschillende expertises hebben speelt een prominente rol in het schoolvak. Het kerndeel van het examenprogramma is compact gehouden, en zodanig samengesteld dat alle leerlingen kunnen communiceren met collega's die zich hebben gespecialiseerd via een bepaald keuzethema. Op deze wijze kunnen leerlingen met verschillende expertises samenwerken in een ontwikkelteam.

Een programma met zo veel keuzemogelijkheden vraagt om een rijk aanbod aan leermaterialen. De bestaande informatiemethoden hebben het kernprogramma voor hun rekening

### Kernprogramma

- A. Vaardigheden
- B. Grondslagen
- C. Informatie
- D. Programmeren
- E. Architectuur
- F. Interactie

### Keuzethema's

- G. Algoritmie, berekenbaarheid en logica
- H. Databases
- I. Cognitive computing (data science)
- J. Programmeerparadigma's
- K. Computerarchitectuur
- L. Netwerken

- M. Physical Computing
- N. Security
- O. Usability
- P. User Experience
- Q. Maatschappelijke en individuele invloed van informatica
- R. Computational Science

Opzet van het vernieuwde schoolvak informatica havo/vwo.

genomen. Voor de keuzethema's is daarom gekozen voor een opzet waarin leraren, informatici en didactici samen materialen ontwikkelen, met steun vanuit het ministerie van OCW. Het materiaal is beschikbaar via i&i, de vakvereniging voor informatica en digitale geletterdheid.

Het vak informatica wordt momenteel aangeboden op ongeveer de helft van alle havo- en vwo-scholen. De populariteit van het schoolvak lijkt toe te nemen: op deze scholen kiezen steeds meer leerlingen ervoor informatica in hun vakkenpakket op te nemen. Dat niet meer scholen het vak aanbieden of het na een tijd zelfs afschaffen is voor een belangrijk deel toe te schrijven aan het tekort aan informaticaleraren. Het oplossen daarvan is voor universiteiten, hogescholen, de IT-sector en de overheid een weerbarstigste uitdaging.

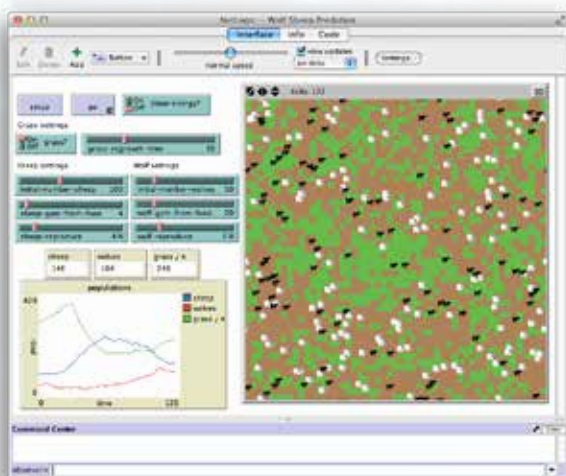
### Computational thinking

Veel onderzoekers kunnen niet zonder computermodellen: denk aan klimaatonderzoekers die voorspellingen doen over de zeespiegelstijging of wetenschappers die de mechanismen achter kanker proberen te ontfaan. Ook in de beroepspraktijk is de computer niet meer weg te denken. De computer heeft onze manier van werken en leren fundamenteel veranderd.

Toch maken we lang niet altijd ten volle gebruik van de potentie die de computer heeft.

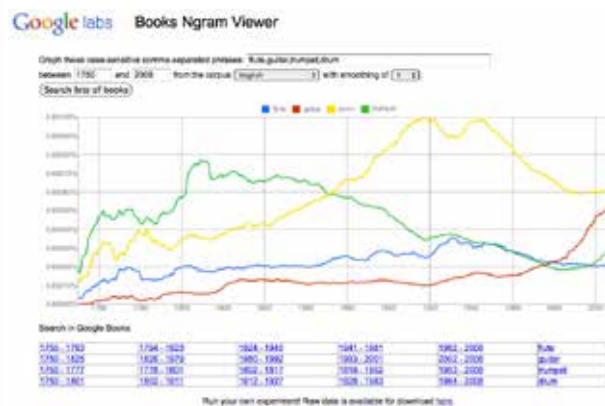
Daarvoor heb je namelijk vaardigheden nodig die verder gaan dan het bedienen van een programma als Word of het beheren van een Facebookpagina. Moet je dan een informaticaspecialist zijn? Dat hoeft nu ook weer niet, maar basisconcepten uit de informatica komen wel van pas om de verbinding te leggen tussen IT en een ander vakgebied. En daarbij speelt computational thinking een voorname rol.

Neem bijvoorbeeld een simulatie van een ecosysteem met schapen en wolven. Daarmee kun je voorspellingen doen over populaties en inzicht krijgen in biologische mechanismen. In de afbeelding hieronder staat een schermafbeelding van NetLogo, een programma waarin je zo'n simulatie kunt maken. Je moet een beetje kunnen programmeren om de simulatie te maken. En je moet weten, of leren, wat belangrijke aspecten zijn die je in het model wilt opnemen.

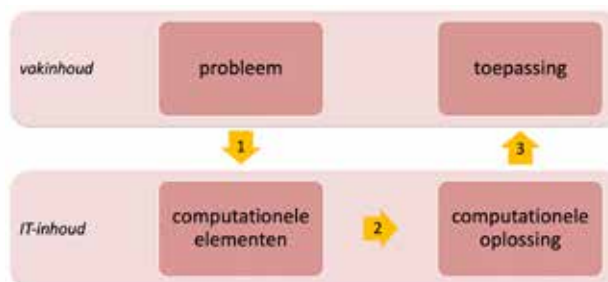


Simulaties hebben hun nut bewezen in allerlei domeinen: om files te voorspellen, consumentengedrag te analyseren, winstprognoses te bepalen, etc.

Maar programmeren is lang niet altijd nodig. Een voorbeeld van het gebruik van standaard digitaal gereedschap is de toepassing van analysetools zoals Google Ngram viewer. Met die online toepassing kun je het voorkomen van woorden in geschreven bronnen analyseren, wat veel mogelijkheden biedt in literatuur- of geschiedenisonderwijs.



In deze voorbeelden gaat het om computational thinking, waarin kennis van vakinhoud en kennis uit de informatica samen komen. De drie pijlen in het schema hieronder laten dat zien.



Het gaat erom (1) het probleem of de vraag te vertalen in termen waar de computer iets meer kan, zoals gegevens en processen. Vervolgens kun je (2) met hulp van bestaande applicaties of met zelfgemaakte algoritmen en programma's een oplossing construeren, om die (3) vervolgens terug te vertalen naar het specifieke vakgebied of domein.

De manier waarop de computer wordt ingezet kan dus variëren: van het gebruiken van bestaande apps tot het aanpassen of zelf programmeren van zo'n computerprogramma.

Soms zijn de data al voor handen, alleen is het moeilijk om daar zinvolle informatie uit te halen. Een journaliste wil graag weten hoe vaak bepaalde woorden zijn gebruikt in nieuwsitems. De data liggen er wel, het is echter onmogelijk om ze handmatig te analyseren. Als deze journalist computational thinking vaardigheden heeft geleerd, ziet ze de potentie die de data hebben, en weet ze hoe ze de informatie die ze nodig heeft automatisch kan laten genereren. Ze weet hoe ze de juiste programma's kan gebruiken of ze kan zelf hier een programma voor schrijven.

Jeannette Wing (2006) heeft het begrip computational thinking gepropageerd. Zij karakteriseerde het als een denkproces om problemen te formuleren en oplossingen te beschrijven, zodanig, dat die met behulp van een computer kunnen worden uitgevoerd. Wing betoogde dat computational thinking belangrijk wordt in elk beroep en vakgebied.

Sindsdien is computational thinking door verschillende mensen nader geconcretiseerd. Deze definities lopen op het eerste gezicht erg uiteen, maar dat valt erg mee: ze komen ruwweg op hetzelfde neer, maar benadrukken verschillende stappen in het denkproces. Zo legt de veelgebruikte definitie door Shelby en Woollard (2013) het accent op de verbinding tussen vakinhoud en IT-inhoud (pijlen 1 en 3), terwijl de beschrijving door de Amerikaanse vereniging van informaticaleeraren CSTA (2011) juist pijl 2 gedetailleerder uitwerkt.

Essentieel aan probleemoplossen via computational thinking is dat het een *uitvoerbare* oplossing oplevert: iets dat je een computer (of eventueel een mens) kunt laten *doen*. Zo'n oplossing is bijvoorbeeld een algoritme, een programma of een simulatie.

Dit operationele aspect voegt ook een nieuw perspectief toe aan de traditionele, meer analytische manier om naar leerinhoud te kijken. Zie bijvoorbeeld de figuur hieronder, afkomstig uit een Engelse taalklas. We zien hier een zogenaamde eindige automaat, een bekend model uit de informatica. Door links te beginnen en de pijlen te volgen tot je op de rechter stip uitkomt, kun je verschillende Engelse zinnen maken. Het is dan niet de computer die zinnen genereert (hoewel dat prima zou kunnen met behulp van deze automaat), leerlingen kunnen dit ook handmatig doen. In plaats van een klassieke analytische aanpak (ontleden van zinnen) wordt grammatica hier synthetisch (genereren van zinnen) benaderd. Dat levert nieuwe leeractiviteiten op: leerlingen kunnen met toevoegen en weglaten van pijlen experimenteren met taal en grammatica.



Genereren van Engelse zinnen (met dank aan Simon Peyton Jones).

Deze voorbeelden van 'computational thinking zonder de computer' zijn interessante bijproducten van het vertalen van vakinhoud naar computationele termen.

Om leerlingen meer de potentie van de computer te laten benutten, is onderwijs in CT nodig.

Computational-thinking-pionier Seymour Papert (1980) introduceerde programmeerlessen voor kinderen. Het idee was dat die kinderen daardoor nieuwe probleemoplossingsvaardigheden zouden ontwikkelen die ze in nieuwe situaties zouden toepassen.

Hoewel Paperts aanpak tot waardevolle innovaties heeft geleid, bleek de beoogde *transfer* tegen te vallen. Het is dus niet handig om computational thinking geheel in een losstaand 'vak' te stoppen en dan te verwachten dat leerlingen hierdoor spontaan de computer zullen inzetten bij het oplossen van problemen in andere vakgebieden. Het verwerken van computational thinking in bestaande vakken is het andere uiterste; waarschijnlijk leren de

## Essentieel aan probleemoplossen via computational thinking is dat het een uitvoerbare oplossing oplevert.

leerlingen dan onvoldoende de basisvaardigheden die hiervoor vereist zijn. Het lijkt erop dat een gecombineerde benadering de beste kansen heeft: aandacht voor computational thinking in bestaande vakken, en daarnaast de computationele vaardigheden (pijl 2), die een specifieke didactiek vereisen, apart.

Het inbouwen van computational thinking in bestaande vakken en leergebieden vergt wel het nodige van de leraar. Daarom is onlangs een interessant EU-project *TeaEdu4CT* gestart waaraan onderwijsontwikkelaars uit 10 landen deelnemen. In dit project worden computational-thinking-modules voor lerarenopleidingen ontwikkeld. De beschikbaarheid van zulke modules zal helpen om leraren in opleiding vertrouwd te maken met leren en onderwijzen van computational thinking. ●

### ► Over de auteurs

**Erik Barendsen** is hoogleraar bètadidactiek aan de Radboud Universiteit en hoogleraar informaticadidactiek aan de Open Universiteit. Hij was voorzitter van de vernieuwingscommissie informatica havo-vwo en is momenteel voorzitter van de regiegroep die de invoering van de vernieuwingen begeleidt. Hij leidt het Nederlandse team in het Europese project *TeaEdu4CT*.

**Martin Bruggink** is vakdidacticus informatica in de lerarenopleiding aan de Technische Universiteit Delft. Hij was lid van de vernieuwingscommissie informatica havo-vwo en is momenteel projectleider voor de ontwikkeling van lesmodules en nascholing voor het vernieuwde programma.

### Verwijzingen

- CSTA. (2011). *Computational thinking teacher resources, second edition*. [https://id.iste.org/docs/ct-documents/ct-teacher-resources\\_2ed-pdf.pdf](https://id.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf)
- Selby, C., & Woollard, J. (2013). *Computational thinking: the developing definition*. Verkregen via <http://eprints.soton.ac.uk/356481>.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.



Erik Barendsen



Martin Bruggink