

# Inhoud

Voorwoord 9

**Deel I: Wat is een mens?**

- 1 Vermist; mensen 12
- 2 Een apocalyps van zelfontkenning 40
- 3 De noösfeer is slechts een andere naam voor de trol in ons allemaal 69

**Deel II: Wat zal geld betekenen?**

- 4 Digitale boerenchic 112
- 5 De stad is gebouwd op muziek 126
- 6 De heren en meesters van de clouds doen afstand van de vrije wil teneinde mateloos gelukkig te worden 135
- 7 De vooruitzichten voor een humanistische cloud-economie 143
- 8 Drie mogelijke toekomstrichtingen 154

**Deel III: De ondraaglijke dunheid van platheid**

- 9 Retropolis 170
- 10 Digitale creativiteit gaat platte plekken uit de weg 186
- 11 Leve het membraan! 193

**Deel IV: Het beste maken van de bits**

- 12 Ik ben een tegendraadse *loop* 209
- 13 Eén verhaal over hoe semantiek zou kunnen zijn geëvolueerd 216

**Deel V: Toekomstige ontwikkelingen**

- 14 Eindelijk thuis (mijn amourette met de bachelardiaanse neotenie) 244

Dankwoord 263

Register 265

## Vermist: mensen

Software drukt ideeën uit over alles, van de aard van een klank in de muziek tot die van het mens-zijn. Software is bovendien onderhevig aan een uitzonderlijk star proces van 'lock-in'. Om die reden zijn ideeën (in de huidige tijd, waarin menselijke aangelegenheden in toenemende mate door software worden gestuurd) meer aan lock-in onderhevig geraakt dan in vroeger tijden. De meeste ideeën die tot nu toe locked-in zijn, zijn niet per se slecht. Maar sommige van de zogenaamde 'web 2.0'-ideeën zijn waardeloos en die moeten we dus, zolang het nog kan, verwerpen.

Spraak is de spiegel van de ziel; zoals men spreekt, zo is men.

– Publilius Syrus

### Fragmenten zijn geen mensen

Rond de eeuwwisseling begon er iets mis te gaan met de digitale revolutie. Het wereldwijde web werd overspoeld door een stroom van triviale ontwerpen die soms web 2.0 werden genoemd. De web 2.0-ideologie propageert een radicale vrijheid op het zichtbare web, maar die vrijheid geldt, o ironie, meer voor machines dan voor mensen. Niettemin wordt er soms over gesproken als 'open cultuur'.

Anonieme blogposts, flauwe videograppen en lichtgewicht mash-ups mogen dan banaal en onschadelijk lijken, als geheel heeft deze wijdverbreide praktijk van fragmentarische, onpersoonlijke communicatie de interactie tussen mensen verarmd.

Communicatie wordt nu vaak ervaren als een bovennatuurlijk verschijnsel dat zich ver boven individuen verheft.

Er is een nieuwe generatie opgegroeid met verminderde verwachtingen over wat iemand kan zijn en wie iemand zou kunnen worden.

Het belangrijkste van een technologie is de manier waarop zij mensen verandert

Als ik met experimentele digitale gadgets werk, zoals bijvoorbeeld met nieuwe variaties op virtual reality in een laboratoriumomgeving, realiseer ik me steeds weer hoezeer kleine veranderingen in de details van een digitaal ontwerp grote en onvoorziene effecten kunnen hebben op de ervaring van de mensen die ermee werken. De kleinste verandering in een op het oog zo onbelangrijk iets als het gebruiksgemak van een knop kan soms leiden tot een complete wijziging van gedragspatronen.

Zo heeft onderzoeker Jeremy Bailenson van Stanford University laten zien hoe een wijziging in de lengte van je avatar in een immersieve virtual reality je gevoel van eigenwaarde en je sociale zelfbeeld verandert. Technologieën zijn uitbreidingen van onszelf en net als bij de avatars in Jeremy's lab kan onze identiteit door de grillen van gadgets veranderen. Het is onmogelijk om met informatietechnologie te werken zonder ook aan social engineering te doen.

Je zou kunnen zeggen: 'Hoe kan het feit dat ik veel blog, sms, twitter en wiki nou veranderen wie ik ben?' of: 'Als de *hive mind* mijn publiek is, wie ben ik dan?' Wij uitvinders van digitale technologieën zijn als stand-upcomedians of neurochirurgen in die zin dat ons werk diepfilosofische vragen oproept; helaas zijn we de laatste tijd maar povere filosofen gebleken.

Ontwerpen ontwikkelaars van digitale technologieën een programma dat je vraagt met een computer te interacteren alsof het een mens is, dan vragen ze je eigenlijk om in een hoekje van je hersenen te accepteren dat jijzelf ook als een

programma zou kunnen worden beschouwd. Ontwerpen ze een internetdienst die door een immense anonieme menigte wordt bewerkt, dan suggereren ze eigenlijk dat een willekeurige menigte mensen een organisme met een legitiem standpunt is.

Verschillende mediaontwerpen stimuleren verschillende capaciteiten in de menselijke natuur. We zouden er niet naar moeten streven de kuddegeest zo efficiënt mogelijk te maken. Integendeel, we zouden moeten proberen het fenomeen van de individuele intelligentie te stimuleren.

‘Wat is een mens?’ Als ik het antwoord op die vraag wist zou ik misschien wel een kunstmatig mens in een computer kunnen programmeren. Maar dat kan ik niet. Iemand zijn is geen pasklare formule, maar een zoektocht, een mysterie, een kwestie van geloof.

## Optimisme

Het zou iedereen, en een technoloog al helemaal, moeilijk vallen om 's ochtends op te moeten staan zonder het vertrouwen dat de toekomst beter kan zijn dan het verleden.

In de jaren tachtig van de vorige eeuw, toen internet slechts voor een klein aantal pioniers beschikbaar was, kwam ik vaak mensen tegen die vreesden dat de vreemde technologieën waar ik aan werkte, zoals virtual reality, de demonen van de menselijke natuur zouden kunnen ontketenen. Zouden de mensen bijvoorbeeld aan virtual reality verslaafd kunnen raken alsof het een drug was? Zouden ze erin opgesloten raken, niet in staat terug te vluchten naar de fysieke wereld waar de rest van ons leeft? Sommige vragen waren onnozel, andere vooruitziend.

## Hoe de politiek van invloed is op de informatietechnologie

Ik maakte in die jaren tachtig deel uit van een vrolijk troepje idealisten. Zou je toen bijvoorbeeld bij mij en John Perry Barlow (later medeoprichter van de Electronic Frontier Foundation) of Kevin Kelly (later grondlegger en hoofdredacteur van het tijdschrift *Wired*) voor een lunch zijn langsgekomen, dan is dit het soort ideeën waarmee we stoeiden en waarover we debatteerden. In de wereld van de technologie zijn idealen belangrijk, maar het mechanisme waarmee idealen gebeurtenissen beïnvloeden, verschilt van dat in andere levenssferen. Technologen maken geen gebruik van overreding om je te beïnvloeden – en als we dat wel doen, doen we het in elk geval niet erg goed. Je vindt bij ons wel een paar mensen die voortreffelijk communiceren (zoals Steve Jobs), maar meestal zijn we niet bepaald overtuigend.

We bedenken extra's voor je *zijn*, zoals afstandsogen en -oren (webcams en mobiele telefoons) en *expanded memory*, extra geheugen (de enorme hoeveelheid details waar je online naar kunt zoeken). Dat worden de structuren via welke je verbonden bent met de wereld en met andere mensen. Deze structuren kunnen op hun beurt de manier veranderen waarop je jezelf en de wereld ziet. We prutsen met je filosofie door rechtstreekse manipulatie van je cognitieve ervaring, niet indirect door middel van discussie. Er is maar een klein groepje ingenieurs nodig om technologie te creëren die de hele toekomst van de menselijke ervaring met een ongelofelijke snelheid kan bepalen. Om die reden zouden er voordat dergelijke rechtstreekse manipulaties worden ontworpen, tussen ontwikkelaars en gebruikers belangrijke discussies over de menselijke relatie met technologie moeten worden gevoerd. Over die discussies gaat dit boek.

Het ontwerp van het web zoals dat zich vandaag de dag voordoet, was niet onvermijdelijk. Begin jaren negentig

was er sprake van misschien wel tientallen geloofwaardige pogingen om met een ontwerp te komen voor de weergave van genetwerkte digitale informatie op een manier die meer populair gebruik zou aantrekken. Bedrijven als General Magic en Xanadu ontwikkelden alternatieve ontwerpen met wezenlijk andere kwaliteiten, die nooit de markt bereikten.

Eén enkel iemand, Tim Berners-Lee, vond uiteindelijk het specifieke ontwerp van het web van nu uit. Het web was bij zijn introductie minimalistisch in de zin dat het zo ongeveer zo weinig mogelijk aannam over hoe een webpagina eruit zou zien. Het was ook open in de zin dat de architectuur geen enkele pagina boven een andere verkoos, en alle pagina's waren voor iedereen toegankelijk. Het legde ook de nadruk op verantwoordelijkheid, omdat alleen de eigenaar van een website ervoor kon zorgen dat zijn site bezocht kon worden.

Berners-Lee's aanvankelijke drijfveer was een gemeenschap van natuurkundigen van dienst te zijn, niet de hele wereld. Niettemin was de sfeer waarin het ontwerp van het web door de eerste gebruikers werd omarmd er een van idealistische discussies. In de periode voor het ontstaan van het web waren de ideeën die er omgingen radicaal optimistisch en kregen deze eerst voet aan de grond in de gemeenschap en vervolgens in de wereld als geheel.

Als we bij het bouwen van informatietechnologieën dan zoveel van het begin af aan moeten verzinnen, hoe kunnen we dan verzinnen welke de beste zijn? Het soort radicale vrijheid dat we in digitale systemen vinden, gaat gepaard met een desoriënterend moreel probleem. We verzinnen het allemaal – dus wat zullen we eens gaan verzinnen? Helaas, dat dilemma – zo veel vrijheid hebben – is een tweesnijdend zwaard.

Naarmate een programma in omvang en complexiteit toeneemt, kan de software een sadistisch labyrint worden. Als er andere programmeurs bij betrokken raken kan het als een

doolhof gaan voelen. Wie slim genoeg is, kan willekeurig welk klein programmaatje uit het niets schrijven, maar het vergt een enorme inspanning (en meer dan een beetje geluk) om een groot programma met succes aan te passen, vooral als andere programma's daar al van afhankelijk zijn. Zelfs de beste ontwikkelgroepen van software raken bij tijd en wijle verstrikt in een zwerm bugs en ontwerpraadsels.

Kleine programma's zijn heerlijk om in je eentje te schrijven, maar het onderhouden van grootschalige software is altijd een ellende. Om die reden verleidt de digitale technologie de psyche van de programmeur tot een soort schizofrenie. Voortdurend worden echte en ideale computers door elkaar gehaald. Technologen hopen dat elk programma zich als een splinternieuw, speels programmaatje zal gedragen en zullen alle beschikbare psychologische strategieën inzetten om te vermijden dat ze op een realistische manier over computers moeten gaan nadenken.

Het broze karakter van computerprogramma's-in-ontwikkeling kan tot gevolg hebben dat digitale ontwerpen 'vastgevroren' raken door een proces dat bekendstaat als lock-in. Dat gebeurt als veel softwareprogramma's zo worden ontworpen dat ze met een bestaand programma moeten samenwerken. Het moeilijkste is het ingrijpend wijzigen van software in een situatie waarin heel veel andere software daarvan afhankelijk is. En dat gebeurt dus vrijwel nooit.

Bij tijd en wijle doet zich een digitaal paradijs voor

Begin jaren tachtig van de vorige eeuw verzoon een ontwerper van synthesizers, Dave Smith, terloops een systeem om muzikale klanken weer te geven. Het systeem kreeg de naam MIDI. Hij benaderde de muziek vanuit het gezichtspunt van een keyboardspeler. MIDI bestond uit digitale patronen die klavieracties als 'een toets lager' en 'een toets hoger' voorstelden.

Dat hield in dat het systeem niet de afgeronde, vloeiende



expressie van een zanger of een saxofonist kon weergeven. Het kon slechts de mozaïeksteentjes van de keyboardspeler beschrijven, niet de aquarellen van de violist. Maar er was geen enkele reden waarom MIDI zich om het geheel van de muzikale expressie zou bekommeren – Dave wilde alleen maar een paar synthesizers met elkaar verbinden zodat hij een groter palet aan klanken ter beschikking zou hebben, ook al bespeelde hij maar één enkel keyboard.

Ondanks die beperkingen werd MIDI het standaardstelsel voor de representatie van muziek in software. Muziekprogramma's en synthesizers werden zo ontworpen dat ermee gewerkt kon worden en het bleek al gauw onpraktisch om al die software en hardware te veranderen of weg te doen. MIDI werd onontkoombaar en in weerwil van herhaalde herculische pogingen van een tientallen jaren omspannende stoet van machtige internationale commerciële, academische en professionele organisaties om het programma te hervormen, is het dat nog altijd.

Standaarden en hun onvermijdelijke gebrek aan vooruitziendheid waren natuurlijk al vóór computers een lastig fenomeen. Een goed voorbeeld is spoorbreedte, de afstand tussen de sporen. De Londense ondergrondse werd ontworpen met smalle sporen en bijbehorende tunnels; op een aantal lijnen kan daarom geen airconditioning worden aangebracht omdat er geen ruimte is om de warme lucht uit de treinen te ventileren. Om die redenen moeten tienduizenden hedendaagse inwoners van een van de rijkste steden ter wereld als gevolg van een meer dan honderd jaar geleden genomen inflexibel ontwerpbesluit een verstikkend hete reis naar en van hun werk ondergaan.

Maar software is erger dan spoorwegen omdat het zich altijd met volledige precisie aan een oneindig specifieke, willekeurige, in de knoop geraakte, hardnekkige warwinkel moet houden. De ontwerpeisen zijn zo stringent en pervers dat aanpassing aan zich wijzigende standaarden een einde-

loze worsteling kan zijn. Mag lock-in dan een gangster in de wereld van de spoorwegen zijn, in de digitale wereld is het een absolute tiran.

### Het leven op het gewelfde oppervlak van de Wet van Moore

Een noodlottig, zenuwslopend aspect van informatietechnologie is dat een bepaald ontwerp zo nu en dan toevallig een niche blijkt te vullen die later een *onneembare vesting* blijkt. Vanaf dat ogenblik wordt het een permanent iets, ook al had een beter ontwerp die plek, vóór het moment dat het andere zich ingroef, net zo goed kunnen innemen. Een lichte ergernis groeit vervolgens uit tot een rampzalig probleem doordat de brute rekenkracht van computers exponentieel toeneemt. In de computerwereld staat dit bekend als de Wet van Moore.

Sinds het begin van mijn loopbaan – wat nog niet zo heel lang geleden is – zijn computers miljoenen malen krachtiger en oneindig veel gangbaarder geworden, en veel meer onderling verbonden geraakt. Het is alsof je op de grond knielt om een zaadje van een boom te planten en dat zaadje vervolgens zo snel groeit dat het, nog voor je je ook maar weer hebt kunnen oprichten, je hele dorp verzwelgt.

Software stelt technologen dus voor een grote, vaak als oneerlijk ervaren verantwoordelijkheid. Omdat computers in exponentieel tempo krachtiger worden, moeten de technologieontwerpers en -programmeurs bij het maken van ontwerpkeuzen uitermate voorzichtig zijn. De gevolgen van piepkleine, in eerste instantie onbeduidende beslissingen worden vaak zozeer versterkt dat het bepalende, onveranderlijke regels voor ons leven worden.

MIDI zit nu in je telefoon en in miljarden andere apparaten. Het is het raamwerk waarop bijna alle populaire muziek die je hoort is gebouwd. Veel geluiden in onze omgeving – muzikaal behang en audiopiepjes, beltonen en waarschuwingssigna-

len – zijn in MIDI uitgedrukt. Het geheel van de menselijke auditieve beleving is inmiddels gevuld met losse tonen die in een rasterpatroon passen.

Op een dag zal een digitaal ontwerp voor de beschrijving van spraak, een ontwerp dat computers in staat stelt om als ze tegen ons praten beter dan nu te klinken, locked-in raken. Vervolgens wordt dat ontwerp wellicht aangepast voor muziek en misschien wordt er dan wel een soepelere en expressievere vorm van digitale muziek ontwikkeld. Maar als dat al gebeurt, over duizend jaar of zo, als een afstammeling van ons op relativistische snelheid reist om een nieuw sterrenstelsel te verkennen, zal hij zich waarschijnlijk ergeren aan een paar vreselijke door MIDI voortgebrachte piepjes muziek om hem te waarschuwen dat het antimateriefilter opnieuw moet worden aangepast.

### Lock-in verandert gedachten in feiten

Vóór MIDI was een muzikale toon een onuitputtelijk idee dat een absolute definitie oversteeg. Voor een musicus was het een manier om na te denken over muziek, of een manier om muziek te doceren en te documenteren. Het was een mentaal stuk gereedschap dat te onderscheiden was van de muziek zelf. Zo konden verschillende mensen bijvoorbeeld transcripties van dezelfde muziekopname maken en met net iets andere partituren voor de dag komen.

Na MIDI was een muzikale klank niet langer zomaar een idee, maar een rigide, verplichte structuur waaraan in die aspecten van het leven die digitaal waren geworden, niet te ontkomen viel. Het lock-inproces is als een golf die het handboek van ons leven allengs overspoelt en de ambiguïteit van flexibele gedachten uitzift naarmate steeds meer gedachtestructuren stollen tot wat in feite een blijvende realiteit is.

Lock-in is te vergelijken met een wetenschappelijke me-

thode. De filosoof Karl Popper had gelijk met zijn betoog dat wetenschap een proces is dat, naarmate het voortschrijdt, ideeën uitsluit – we kunnen bijvoorbeeld niet langer met reden geloven in een platte aarde die een paar duizend jaar geleden plotseling is ontstaan. Wetenschap elimineert ideeën op empirische basis en gefundeerd. Lock-in elimineert ontwerpmogelijkheden echter op basis van wat gemakkelijk te programmeren is, wat politiek haalbaar is, wat mode is of wat toevallig wordt gecreëerd.

Lock-in elimineert ideeën die niet in het succesvolle digitale representatieprogramma passen. Tegelijkertijd reduceert of verengt het ook de ideeën die het onsterfelijk maakt, door de onpeilbare halfschaduw van betekenis die een woord in de natuurlijke taal onderscheidt van een commando in een computerprogramma, weg te snijden.

De criteria die als leidraad voor de wetenschap gelden mogen dan bewonderenswaardiger zijn dan die voor lock-in, maar als we niet een geheel nieuwe manier bedenken om software te maken, zijn verdere lock-ins gegarandeerd. Wetenschappelijke vooruitgang daarentegen vergt altijd vastbeslotenheid en kan tot stilstand komen dankzij de politiek, gebrek aan subsidie of aan nieuwsgierigheid. Er doet zich een interessant probleem voor: hoe kan een musicus het bredere, minder gedefinieerde begrip van een toon vóór MIDI koesteren als hij de hele dag door van MIDI gebruikmaakt en met andere musici via het filter van MIDI in contact staat? Is het zelfs maar de moeite van het proberen waard? Moet een digitaal artiest maar gewoon aan lock-in toegeven en het oneindig expliciete, eindige concept van een MIDI-toon maar accepteren?

Als het van belang is de grens van het raadsel op te zoeken, na te denken over de dingen die zich niet helemaal laten definiëren – of in een digitale standaard kunnen worden omgezet – moeten we permanent op zoek gaan naar volkomen nieuwe ideeën en objecten, en oude, zoals muzikale tonen,

laten varen. Aan het eind van dit boek zal ik onderzoeken of mensen bezig zijn een soort MIDI-tonen te worden: overgedefinieerd en in de praktijk beperkt tot wat in een computer kan worden weergegeven. Dit heeft enorme implicaties: we kunnen het misschien wel zonder muzikale klanken stellen, maar niet zonder onszelf.

Toen Dave MIDI maakte, was ik razend enthousiast. Een stel vrienden van me uit het oorspronkelijke team van Macintosh bouwde vlug een hardware-interface zodat een Mac gebruik van MIDI kon maken om een synthesizer aan te sturen, en ik zette een snel programma om muziek te creëren in elkaar. We voelden ons ontzettend vrij – maar we hadden beter moeten opletten.

Inmiddels is MIDI veel te moeilijk te veranderen en is de cultuur dus veranderd om het systeem completer te laten lijken dan oorspronkelijk de bedoeling was. Om de technologie geschikt te maken hebben we onze verwachtingen van de allergewoonste vormen van muzikaal geluid naar beneden bijgesteld. Het was niet de schuld van Dave. Hoe had hij het moeten weten?

### Digitale materialisatie: lock-in maakt filosofie tot realiteit

Veel locked-in ideeën over hoe software wordt samengesteld, zijn afkomstig uit een oud besturingssysteem, UNIX. Het vertoont enkele kenmerken die verwant zijn met MIDI.

Terwijl MIDI muzikale expressie door een beperkend model van de actie van toetsen op een muziektoetsenbord perst, doet UNIX hetzelfde voor alle computerverwerking maar gebruikt daarvoor de actie van de toetsen op een schrijfmachineachtig toetsenbord. Een UNIX-programma heeft vaak veel weg van een simulatie van iemand die snel aan het typen is.

UNIX heeft als belangrijkste ontwerpkenmerk een ‘opdrachtregelinterface’. In dit systeem typ je opdrachten, je

drukt op ‘enter’ en de opdrachten worden uitgevoerd.\* Een verbindend ontwerpprincipe van UNIX is dat een programma niet kan zeggen of het een mens is die de entertoets heeft aangeslagen, of een programma. Aangezien echte mensen toetsenborden trager bedienen dan gesimuleerde mensen, vervalft door dit specifieke idee het belang van exacte timing. Daardoor is UNIX gebaseerd op zogeheten *discrete events*, ‘losse’ acties, die niet op een specifiek ogenblik in de tijd hoeven plaats te vinden. Het menselijk organisme is daarentegen gebaseerd op doorlopende zintuiglijke, cognitieve en motorische processen die exact tegelijk moeten worden gesynchroniseerd. (MIDI, gebaseerd op discrete events die zich op bepaalde momenten voordoen, zit ergens tussen het concept van tijd zoals dat belichaamd is in UNIX en dat in het menselijk lichaam in.)

UNIX geeft blijk van een al te groot geloof in afzonderlijke abstracte symbolen, en van een tekort aan geloof in de temporele, doorlopende, niet-abstracte werkelijkheid; het heeft meer weg van een schrijfmachine dan van een danspartner. (Misschien zouden schrijfmachines of tekstverwerkers altijd onmiddellijk moeten reageren, net zoals een danspartner – maar dat is nog niet het geval.) UNIX heeft de neiging zich te ‘willen’ verbinden met de werkelijkheid alsof die werkelijkheid een netwerk van snelle typistes is.

Als je de hoop hebt dat computers worden ontworpen ten dienste van zowel echte mensen als mogelijke mensen, moet UNIX als een slecht ontwerp worden beschouwd. Daar kwam ik achter in de jaren zeventig, toen ik probeerde er responsieve muziekinstrumenten mee te maken. Ik probeerde te doen wat MIDI niet doet, namelijk werken met vloeiende, moeilijk

\* De stijl van UNIX-opdrachten is, hoe ongelofelijk dat ook klinkt, inmiddels onderdeel van de popcultuur geworden. Zo zijn bijvoorbeeld de URL's (Universal Resource Locators) die we tegenwoordig gebruiken om internetpagina's te vinden, zoals <http://www.jaronlanier.com/>, voorbeelden van het soort toetsaanslagreeksen die in UNIX alomtegenwoordig zijn.

te noteren aspecten van muziek en ontdekte dat de onderliggende filosofie van UNIX daar te afstandelijk en onhandig voor was.

De argumenten pro UNIX draaiden om de manier waarop computers de komende decennia letterlijk miljoenen malen sneller zouden worden. Het idee was dat de toename van de snelheid de timingproblemen waar ik mee zat, teniet zou doen. En inderdaad, de computers van nu zijn miljoenen malen sneller en UNIX is een alomtegenwoordig deel van het leven geworden. Er zijn een paar redelijk expressieve tools met UNIX erin, en de toename in snelheid is inderdaad voldoende om de problemen van UNIX in sommige gevallen te compenseren. Maar niet in alle.

Ik heb een iPhone in mijn zak en jawel, het ding heeft wat in wezen UNIX is in zich. Een zenuwslopende factor van dit apparaat is dat het geplaagd wordt door een eigenaardige reeks onvoorspelbare vertragingen in de gebruikersinterface. Je verstand wacht op de reactie op het indrukken van een virtuele knop, maar die komt nog even niet. Op dat moment wordt er een merkwaardige spanning opgebouwd en je relaxte intuïtie maakt plaats voor nervositeit. Het is de geest van UNIX, die na al die jaren nog steeds weigert tegemoet te komen aan het ritme van mijn lichaam en mijn geest.

Ik wil niet specifiek afgeven op de iPhone (waarover ik later, in een andere context, de loftrompet zal steken). Ik had net zo gemakkelijk elke hedendaagse personal computer kunnen kiezen. Windows is dan wel geen UNIX, maar het deelt het idee van UNIX dat een symbool belangrijker is dan het vloeiende verloop van tijd en de onderliggende continuïteit van ervaring.

De schoorvoetende relatie tussen UNIX en de temporele wereld waarin het menselijk lichaam zich beweegt en de menselijke geest denkt, is een teleurstellend maar niet rampzalig voorbeeld van lock-in. Misschien zal het, naarmate virtual reality beter wordt, mensen zelfs helpen de ouderwetse fysie-

ke wereld te waarden. Als dat gebeurt, zal het een geluk bij een ongeluk blijken te zijn.

Verschanste softwarefilosofieën worden dankzij hun alomtegenwoordigheid onzichtbaar

Een nog dieper locked-in idee is dat van de file, het bestand. Er was eens een tijd, nog niet zo heel lang geleden, dat nogal wat computerwetenschappers de bestandsgedachte niet zo heel geweldig vonden.

Het eerste ontwerp voor zoiets als het wereldwijde web, Xanadu van Ted Nelson, stelde zich bijvoorbeeld een reusachtig groot, mondiaal bestand voor de hele wereld voor. De eerste versie van de Macintosh, die nooit in de handel is gekomen, kende geen bestanden. Integendeel, de totale productie van een gebruiker verzamelde zich tot één grote structuur, zoiets als een enkele persoonlijke webpagina. Steve Jobs nam het Mac-project over van de man die ermee begon, wijlen Jef Raskin, en al gauw maakten de bestanden hun opwachting.

UNIX had bestanden, de Mac had bestanden toen hij op de markt kwam, Windows had bestanden. Bestanden vormen nu een deel van het leven; we brengen informaticastudenten het idee van een bestand bij alsof het een onderdeel van de natuur zou zijn. Sterker nog, ons idee van bestanden is misschien wel permanenter dan onze ideeën over de natuur. Ik kan me voorstellen dat natuurkundigen ons op een dag vertellen dat het tijd is om op te houden met te geloven in fotonen omdat er een betere manier is ontdekt om over licht na te denken – maar het bestand zal hoogstwaarschijnlijk wel voortleven.

Het bestand is een serie filosofische ideeën die tot eeuwige waarheid zijn verheven. De ideeën die door het bestand worden uitgedrukt omvatten de notie dat de menselijke expressie zich voordoet in deelbare brokken die als bladeren aan een abstracte boom kunnen worden georganiseerd – en dat die brokken versies hebben en moeten worden samengevoegd tot compatibele applicaties.



Wat betekenen bestanden voor de toekomst van de menselijke expressie? Deze vraag is moeilijker te beantwoorden dan de vraag ‘hoe beïnvloedt de Engelse taal de gedachten van Engelse moedertaalsprekers?’ Engelssprekenden kun je tenminste nog vergelijken met Chineessprekenden, maar bestanden zijn universeel. Het idee van het bestand is zo groot geworden dat we niet in staat zijn ons een kader voor te stellen dat groot genoeg is om eromheen te passen zodat we het empirisch kunnen beoordelen.

Wat er met treinen, bestanden en muzikale klanken is gebeurd, kan op korte termijn zomaar met de definitie van een mens gebeuren

Het loont de moeite te zien wanneer filosofieën tot locked-in software stollen. Is alomtegenwoordige anonimiteit of pseudonimiteit een goede zaak? Dat is een belangrijke vraag, omdat de overeenkomstige opvattingen over de manier waarop mensen betekenis kunnen uitdrukken zo zijn ingeworteld in de met elkaar samenhangende softwareontwerpen van internet dat we er misschien wel nooit helemaal van af komen, of ons zelfs maar kunnen herinneren dat het ook anders had kunnen zijn.

We zouden op z'n minst moeten proberen dit bijzonder lastige voorbeeld van dreigende lock-in te vermijden. Lock-in doet ons de verloren vrijheden vergeten die we in het digitale verleden hadden. Dat kan het moeilijker maken de vrijheden te zien die we in het digitale heden wel hebben. Gelukkig kunnen we, hoe moeilijk dat ook is, nog altijd proberen sommige expressies van de filosofie die op het punt staan locked-in te raken in de hulpmiddelen die we gebruiken om elkaar en de wereld te begrijpen, te veranderen.