



**Universiteit  
Leiden**  
The Netherlands

## **Informatie en communicatie**

Schmidt, A.H.J.; Mommers, L.

### **Citation**

Schmidt, A. H. J., & Mommers, L. (2004). Informatie en communicatie. Retrieved from <https://hdl.handle.net/1887/3249>

Version: Not Applicable (or Unknown)

License: [Leiden University Non-exclusive license](#)

Downloaded from: <https://hdl.handle.net/1887/3249>

**Note:** To cite this publication please use the final published version (if applicable).

# 1 | Informatie en communicatie

*Prof. mr. A.H.J. Schmidt en dr. L. Mommers*

## 1.1. Het uitgangspunt: technologische turbulentie

De wereld van het recht en die van de techniek zijn zeer verschillend. Nieuwe generaties van technische producten wisselen elkaar in een hoog tempo af. Het recht heeft een tragere hartslag. Het verschil in levensritme met de wereld van de informatie- en communicatietechnologie (ICT) heeft gevolgen voor het effectieve bereik van regulering. Wanneer een wettelijke regeling in werking treedt, is het technische product dat de aanleiding vormde weer verdwenen, of heeft een dusdanig ander karakter gekregen dat de nieuwe regel niet langer toepasselijk is: het recht als mosterd na de maaltijd. Door dit mechanisme is anno 2004 nog steeds onduidelijk wat de juridische status van bestandsuitwisseling (file sharing) nu precies is.<sup>1</sup> Ondertussen maken ontwikkelingen in de telecommunicatiemarkt het nog steeds bijna onmogelijk om te *reageren*, laat staan te *anticiperen* op de organisatorische en technische vernieuwing. Als voorbeeld moge dienen de rol van internet provider, die de komende jaren ongetwijfeld aan verandering onderhevig zal zijn als gevolg van de democratisering van de ICT – denk aan de opkomst van draadloos internet (WiFi), dat door elke particulier gemakkelijk aangeboden kan worden, met alle mogelijke juridische implicaties vandien.

De afstemmingsproblemen tussen de ICT en het recht zijn hardnekkig. Ook de algemene voorwaarden en het contract ontkomen (als reguleringsinstrumenten) niet aan het vraagstuk hoe dan wel een effectieve duurverbintenis over ICT *de facto* moet worden opgesteld. Vanuit dat gezichtspunt is het in de eerste plaats van belang om regelingen zoveel mogelijk onafhankelijk van de techniek te formuleren. Sinds Lawrence Lessig in 1999 heeft laten zien dat niet onbelangrijke uitingen van juridische regulering vorm krijgen in (de architectuur van) informatiesystemen heeft dit uitgangspunt iets paradoxaals gekregen. Voor dergelijke vormen van regulering is – als die onder verantwoording van juristen plaats vinden – immers kennis nodig van technische aspecten, hun ontwikkeling, hun te verwachten levenscyclus en hun vermoedelijke opvolgers. Kortom, inzicht in de achterliggende mechanismen van de informatie- en communicatietechnologie is ook voor juristen noodzakelijk.

---

<sup>1</sup> Zo mag blijken uit het arrest Buma/Kazaa, Hoge Raad 19/12/2003, LJN-nummer AN7253 Zaaknr C02/186HR, zie ook K. Koelman (2003)

In wezen gaat het bij informatietechnologie om de invariante structuren die ten grondslag liggen aan de informatie- en communicatietechniek. Het verschil tussen beide zit in het feit dat het levensritme van de technologie veel meer in overeenstemming is met dat van het recht dan het levensritme van de techniek. Om die reden bestaat dit hoofdstuk uit twee gedeelten: een historisch gedeelte dat de ontwikkelingen in de ICT schetst over de afgelopen vijftig jaar, en een gedeelte waarin deze ontwikkelingen worden geanalyseerd en een overzicht wordt gegeven van te verwachten veranderingen in technologieën, en daarmee samenhangend de noodzaak bepaalde juridische concepten te herzien.

In het eerste gedeelte van dit hoofdstuk worden per decennium aspecten van apparatuur, infrastructuur en programmatuur besproken en wordt een typering gegeven van het in dat decennium dominante maatschappelijke perspectief ten aanzien van de informatietechniek. Daarbij worden vier schema's gepresenteerd die abstracte IT-aanknopingspunten bieden voor praktische beschouwingen. Aan de hand daarvan worden aan het einde van dit hoofdstuk enkele benaderingswijzen voor het opstellen van techniekonafhankelijke regulering besproken.

Zo'n zestig jaar geleden begon de opmars van de computer. Eerst groot, log en duur, slechts weggelegd voor militaire doeleinden, daarna toegankelijk voor onderzoeksinstituten en bedrijven, en ten slotte voor consumenten. Eerder was al de basis gelegd voor communicatie over grote afstanden; via telegraaf, telefoon en radio. De laatste jaren zijn deze computer- en communicatietechnologieën grootschalig met elkaar gecombineerd, en internet vormt hiervan het meest sprekende voorbeeld. De opkomst van het internet markeert niet alleen een technologische, maar ook een functionele integratie: digitalisering van communicatiestromen leidt tot het vervagen van grenzen tussen verschillende media.

## 1.2. Rond 1945: de eerste computer

Gedurende de Tweede Wereldoorlog was er grote behoefte aan krachtige rekenmachines. In de Verenigde Staten en in het Verenigd Koninkrijk werden meerdere grote onderzoeksprojecten in die richting gefinancierd. Niet alleen voor de ontwikkeling van de kernbom was reken capaciteit een noodzakelijke voorwaarde. Ook het produceren van (individuele) correctietabellen voor het richten van kanonnen vroeg veel rekenwerk, evenals het bouwen van betere vliegtuigen en de pogingen om gecodeerde boodschappen van de vijand te ontcijferen. Er was derhalve een gunstig maatschappelijk klimaat voor het ontwikkelen van krachtig, dat wil zeggen algemeen (of programmeerbaar) rekentuig. En in dat klimaat is de computer inderdaad ontwikkeld. Hij heeft desalniettemin geen rol van betekenis gespeeld in de oorlog. De potentie van de eerste werkende computers aan Duitse en aan Amerikaanse zijde werden niet als zodanig (h)erkend zodat de eerste operationele computer zijde dateert van 1948.

We geven eerst een paar jaartallen en namen, daarna een schema met het basismodel van de (eerste) computer. Dat heeft nog niets aan actualiteitswaarde ingeboet. Het is evenzeer van toepassing op de ENIAC uit 1945 (zie hierna) als op de

hedendaagse chipcard. Het is derhalve een stabiel conceptueel aanknopingspunt in een zich overigens razendsnel ontwikkelende sector.

### 1.2.1. *De ontwikkeling van de computer: enkele jaartallen en namen*

In 1936 beschreef Turing (in Engeland) een denkbeeldige machine die hij gebruikte voor het bewijzen van een wiskundige stelling. Deze ‘Turing machine’ geldt ook nu nog als *het* theoretische model van de computer.

In 1941 bouwde Zuse (in Duitsland) zijn Z3, de eerste werkende programmeerbare rekenmachine. Deze werd ontwikkeld voor een specifiek rekenprobleem bij de Duitse luchtmacht, en als zodanig gepresenteerd. Aan die pragmatische benadering werd toen in Duitsland de voorkeur gegeven. In hetzelfde jaar liet Anastasof zijn prototype zien aan Eckert, een gebeurtenis die vooral van belang is geweest in het proces tussen Honeywell en Sperry-Rand omdat daardoor het octrooi op de computer als uitvinding (dat Sperry-Rand meende te bezitten) werd vernietigd.<sup>2</sup>

In 1943 begonnen Mauchly en Eckert aan een project bij de Moore School of Technology (in de USA). Dit leverde in 1945 de ENIAC op. Deze wordt beschouwd als het eerste prototype van een computer. In 1945 gaf Von Neumann een retrospectieve analyse van het ENIAC-project en formuleerde daarbij een logisch model van de programmeerbare computer.

In 1948 werd de eerste werkende computer gebouwd in Manchester. Turing was daar weer bij betrokken. Zijn denkbeeldige machine is equivalent aan het logische model van Von Neumann. Dat model is van een ongebruikelijke zeggingskracht en gaf aanleiding tot vele speculaties over de mogelijkheden van de computer – niet alleen, bijvoorbeeld, over de vraag of computers konden schaken, maar ook of ze konden denken. In 1950 weerlegde Turing in een ander beroemd gebleven artikel<sup>3</sup> een hele reeks van argumenten tegen de stelling dat computers zouden kunnen denken.

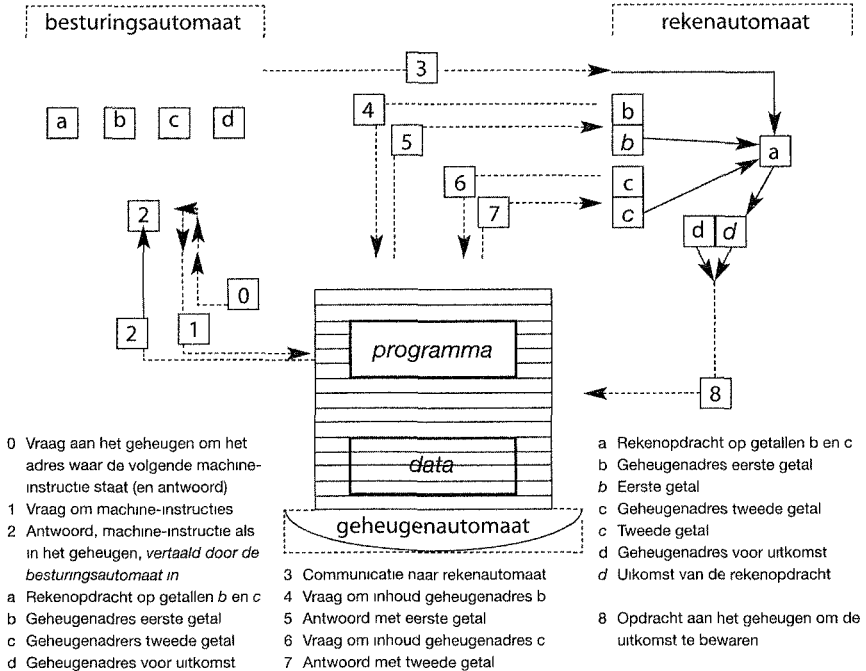
### 1.2.2. *Het basismodel van de programmeerbare computer*

De opwinding van destijds kwam door het besef dat de combinatie van drie simpele (niet-programmeerbare) automaten een machine oplevert die zo machtig is, dat over de grenzen van de mogelijkheden alleen kan worden gespeculeerd. Die drie automaten zijn: een geheugen(automaat), een besturingsautomaat en een rekenautomaat. De drie automaten hebben elk afzonderlijk zeer simpele moge-

<sup>2</sup> In deze zaak deed rechter Larson uitspraak op 19 oktober 1973. De zaak was begonnen op 26 mei 1967. Het was een mega-zaak. Er waren 135 zittingsdagen. Er werden 70 getuigen gehoord en er waren 86 schriftelijke getuigenverklaringen. Er waren uiteindelijk 32.484 stukken in het dossier, dat 91 dozen en 211 microfilms omvat en dat verspreid over drie wetenschappelijke instituten wordt bewaard. Het dossier is verspreid over drie instituten en omvat 91 dozen en 211 micro-films. Zie [http://www.archives.upenn.edu/faids/upd/eniactrial/upd8\\_10.html](http://www.archives.upenn.edu/faids/upd/eniactrial/upd8_10.html)

<sup>3</sup> A. M. Turing, ‘Computing Machinery and Intelligence’, *Mind*, 1950/59, p. 430-460

lijkheden die zich in complexiteit niet noemenswaardig onderscheiden van koffiezetmachines of broodroosters. Tezamen vormen ze evenwel een computer. In Schema 1 geven we een aanduiding van de informatieverwerking binnen en de communicatie tussen de drie automaten bij het uitvoeren van één enkele machine-instructie.



Schema 1: Het basismodel van de programmeerbare computer

Het *geheugen* is een eenvoudige automaat die twee opdrachten kan uitvoeren. De eerste is het opslaan van een van buiten aangeboden pakketje informatie op een bepaalde plaats. De tweede is het ophalen van een pakketje informatie van een bepaalde plaats en dat naar buiten aanbieden.

De *besturingsautomaat* kan drie opdrachten uitvoeren. De eerste is het opvragen van een pakketje informatie uit een vast geheugenadres. Hierin staat de plaats (of het geheugenadres) waar de volgende machine-instructie kan worden gevonden. De tweede opdracht is het ontleden van die machine-instructie in een rekenopdracht (a), in twee geheugenadressen (b,c) waar de getallen staan waarmee wordt gerekend en het geheugenadres (d) waar de uitkomst moet worden bewaard. De derde opdracht die de besturingsautomaat kan uitvoeren is doorgeven van (a), (b,c) en (d) aan de rekenautomaat.

De *rekenautomaat* kan ook drie opdrachten uitvoeren. De eerste is het opvragen uit het geheugen van de getallen waarop de berekening moet worden uitgevoerd. De tweede is het uitvoeren van de berekening. De derde is het aanbieden

aan het geheugen van de uitkomst van de berekening met het adres waar de uitkomst moet worden bewaard. Het beschikbare repertoire aan rekenopdrachten is zeer gering: optellen, aftrekken, vermenigvuldigen, delen en vergelijken van twee getallen (dat wil zeggen bepalen of een getal groter is dan een ander).

### 1.2.3. *Enkele techniekonafhankelijke ideeën naar aanleiding van het model*

De apparatuur is in dit model niet meer toegesneden op één enkele taak. Dit was nieuw in 1945. In het geheugen wordt een reeks opdrachten opgeslagen voor een bepaalde rekentaak, en als die is voltooid kan een nieuwe reeks opdrachten voor een andere taak worden ingelezen. Oftewel: één en hetzelfde apparaat (de hardware) kan worden gebruikt voor uiteenlopende functies.<sup>4</sup> Daartoe wordt voor elke functie een *programma* geschreven, een samenhangende reeks instructies die door de automaat kan worden uitgevoerd.

Een programma bestaat derhalve uit machine-instructies die worden uitgevoerd op getallen of data die worden gekenschetst als invoer of input. Deze invoer kan eveneens in het geheugen worden gelezen en bewaard, net als de resultaten van de bewerkingen, de uitvoer of output van het programma. Deze indelingen doen de notie ontstaan van een *bestand* (Engels: *file*). Dezelfde rekentaak kan zo keer op keer worden toegepast: men gebruikt daarvoor hetzelfde programmabestand, dat gevoed wordt met verschillende databestanden.

Het meest bevrijdende idee dat Von Neumann vorm gaf in zijn basismodel is *de gelijke behandeling van programmatuur en data* in het geheugen. Het onderscheid is conceptueel van aard geworden, niet langer fysiek. Als gevolg daarvan wordt het voorstelbaar dat een programma zichzelf aanpast op basis van de data waarop het werkt: de uitkomst van een berekening kan worden bewaard en worden gebruikt als ware het een initieel onderdeel van het programma. Met name deze eigenschap heeft veel discussies over ‘denkende machines’ in gang gezet.

Het basismodel is een coöperatief model. De drie deelautomaten *communiceren* en werken samen en dat houdt in dat de uitgewisselde berichten voor de geadresseerde automaten begrijpelijk moeten zijn. De werking van de besturingsautomaat, die de machine-instructies analyseert, dicteert de voorwaarden waaraan de effectieve formulering van machine-instructies moet voldoen.<sup>5</sup>

Het schrijven van programma's in machinetaal is een moeizaam karwei. Om de rekentaak die door het programma moet worden uitgevoerd in machinetaal te kunnen formuleren, moet de taak in detail worden geanalyseerd en worden beschreven in uitdrukkingen die het menselijk begrip weinig zeggen. Echter, als instruc-

---

4 Zoals het uitvoeren van berekeningen, het bijhouden van bedrijfsadministraties, het berekenen van statistische verbanden, het administreren van ingezetenen, het ontwerpen van vliegtuigen, het herkennen van vijandige vliegtuigen, het verhandelen van aandelen en (inmiddels) het uitwisselen van post, het raadplegen van een encyclopedie, het bezichtigen van de schilderijen van een schilderjentoonstelling, het voeren van telefoongesprekken, het bekijken van films en van televisieprogramma's – het maken van films en televisieprogramma's, het spelen van spelen en het maken van spelen

5 De geschetste relatie tussen hardware en de taal voor machine-instructies is een onvermijdelijke. Elke computer heeft een *machinetaal*

ties die zijn opgesteld volgens de regels van een programmeertaal eenduidig kunnen worden vertaald naar een reeks machineopdrachten, kan in principe een vertaalprogramma worden gemaakt dat die vertaling uitvoert. Als gevolg daarvan kan de menselijke programmeur zijn opdrachten formuleren in een minder basale taal. Dit heeft geleid tot de ontwikkeling van vele verschillende programmeertalen, die voor een programmeur toegankelijker zijn dan machinetaal.

#### 1.2.4. *Enkele regelmatigheden in de technische ontwikkeling*

Hoe het basismodel wordt verwezenlijkt in een werkende computer is een uiterst techniekafhankelijke kwestie. We besteden daaraan alleen aandacht voorzover daarin weer regelmatigheden zijn te ontdekken. Zo ontwikkelt de snelheid waarmee computers machine-instructies kunnen verwerken zich onmiskenbaar in de richting van een steeds hoger aantal per seconde. Ook het geheugen van computers neemt toe. Deze ontwikkelingen leidt er toe dat computers nu kunnen worden gebruikt voor functies die enkele tientallen jaren geleden nog ondenkbaar waren, zoals de opslag en weergave van geluid en video in hoge kwaliteit. Hoewel het basismodel van de computer onveranderlijk is, maken deze technologische veranderingen nieuwe toepassingen mogelijk.

#### 1.2.5. *Maatschappelijk perspectief*

In het decennium 1940-1950 werd de computer ontwikkeld, mede gefinancierd door de bewapeningsindustrie. Het ging om extreem kostbare, in isolement opererende machines. Hun maatschappelijke betekenis lag misschien nog niet in de praktische rekenprestaties. Belangrijker is wellicht dat het visioen van computers als mechanische, algemene probleemoplossers in deze periode de denkkraft van belangrijke onderzoekers heeft geprikkeld.

### 1.3. **Rond 1955: systeemprogrammatuur**

De behoefte aan programmeerbare rekenmachines leek in de jaren vijftig minimaal. Ze waren in 1950 nog erg onbetrouwbaar. Hun bediening en het maken van programma's vergde de zeer arbeidsintensieve inzet van talentrijk, hoog opgeleid personeel. Maar in 1950 was er oorlog in Korea en het ministerie van Defensie van de Verenigde Staten had opnieuw behoefte aan rekencapaciteit. Een opdracht voor het bouwen van een computer werd aan IBM voorgelegd. Dit bedrijf achtte het commerciële potentieel voor dergelijke apparaten zeer gering, ze zouden op zijn minst \$8000 per maand huur moeten opbrengen.<sup>6</sup> Maar de werkelijkheid liet een ander beeld zien.

---

<sup>6</sup> Het gemiddelde maandinkomen van een universitair docent was toen in de Verenigde Staten \$250; in plaats van één computer konden destijds 32 universitair geschoolde medewerkers worden gefinancierd, en die 32 medewerkers waren weer ten minste nodig om die ene computer aan de praat te krijgen en aan het werk te houden.

### 1.3.1. *Programmatuur: compilers en systeemprogrammatuur*

Toen in 1953 door één bedrijf meerdere computers (IBM 701) werden afgezet, trad nadrukkelijk aan het licht hoe arbeidsintensief het maken van programma's in machinetaal en in assembleertaal eigenlijk was. In 1951 bedacht Grace Murray Hopper het concept van een programma, dat berekeningen op gebroken getallen automatisch kon omzetten in machine-instructies. Ze noemde het een *compiler*. Ze liet ook zien dat opdrachten voor randapparatuur (zoals de afdrukopdracht voor een printer) zouden kunnen worden vertaald door een dergelijke compiler. Meer in het algemeen is het zo, dat de eerder uiteengezette gelijke behandeling van programma's en data in het geheugen het mogelijk maakt om programma's van verschillende logische niveaus te maken. Bijvoorbeeld de genoemde compilers, programma's die andere programma's als data opvatten en ze vertalen in een nieuwe, in de context beter bruikbare taal.

Backus is in 1953 aan een compiler gaan werken voor een 'hogere' programmeertaal, die veel gemakkelijker te gebruiken zou zijn voor menselijke programmeurs dan de bestaande assemblers en machinetalen. In 1957 was de eerste FORTRAN-compiler gereed voor de IBM 701. Programma's doen zich sindsdien voor in twee vormen: als *broncode* (uitgedrukt in de hogere programmeertaal) en als *objectcode* (als output van een compiler, uitgedrukt in machinetaal). Sinds 1957 zijn veel compilers en interpreters gebouwd.<sup>7</sup>

Het handjevol computers dat in de praktijk werd gebruikt vertegenwoordigde een zo omvangrijke investering, dat de inspanningen vooral werden gericht op het beter benutten van de beschikbaar gekomen reken capaciteit. Dat heeft niet alleen geleid tot de ontwikkeling van compilers, maar tot de ontwikkeling van veel meer ondersteunende programma's. Zo ontstond de notie van systeemprogrammatuur die naast compilers ook wel bedrijfssystemen of *operating systems* opleverde. Dergelijke bedrijfssystemen regelen de communicatie van de centrale verwerkingseenheid met de randapparatuur, en de toegang tot en de verwerking van programma's en data.

### 1.3.2. *Randapparatuur en infrastructuur*

In de randapparatuur ontstond een ontwikkeling. Met name was er behoefte aan meer permanent geheugen dat grotere capaciteit had en minder kostbaar was dan het werkgeheugen. Naast de bestaande ponskaarten werd gebruik gemaakt van papieren ponsbanden, en later van elektromagnetische tapes en schijven. Belangrijke taak van het operating system was het ondersteunen van de archivering en de ontsluiting van bestanden in het achtergrondgeheugen. Om dat mogelijk te maken bestond één enkele computer veelal uit een heel netwerk van communicerende (rand)apparaten. Daarvoor was een lokale infrastructuur nodig (die nog geen gebruikmaakte van de openbare telecommunicatie-infrastructuur).

---

<sup>7</sup> Bijvoorbeeld voor programmeertalen als: Ada, Algol, APL, Basic, Cobol, C, C++, Fortran, Java, Lisp, PHP, PL/1, Prolog, Snobol enzovoort. Sommige programma's worden simultaan vertaald en uitgevoerd. Hun 'compilers' heten 'interpreters'.

### 1.3.3. *Maatschappelijk perspectief*

In het decennium 1950-1960 werd de computer verder ontwikkeld, aanvankelijk mede gefinancierd door de bewapeningsindustrie. Het ging nog steeds om extreem kostbare, in isolement opererende machines. In deze periode nam de betrouwbaarheid door technische vondsten enorm toe. Het basismodel werd voorts in de praktijk aangevuld met een veelheid aan randapparatuur. Om het gebruik te optimaliseren werd systeemprogrammatuur ontwikkeld. Direct, praktisch nut hadden computers in deze periode vooral door te rekenen. De productie van computers en randapparatuur was in handen van enkele ondernemingen. Elk computermodel had (en heeft nog) zijn eigen, processorafhankelijke machinetaal en systeemprogrammatuur. Er was nog geen (academische) opleiding in de informatica. Computers zijn vooral een Anglo-Amerikaans verschijnsel.

## 1.4. **Rond 1965: time-sharing, chips**

In de jaren zestig bleef de aandacht bij programmatuurontwikkeling uitgaan naar werkwijzen om de zo kostbare capaciteit van computers zo volledig mogelijk te benutten. Door de techniek werd ernaar gestreefd computercapaciteit machtiger en goedkoper te maken. Aan dat laatste is een belangrijke bijdrage geleverd door de ontwikkeling van de chip. Deze werd vanaf 1963 leverbaar en leidde tot bestendige miniaturisering, capaciteitsvergroting en kostenverlaging van de apparatuur. Daardoor werden steeds meer computers ingezet.

### 1.4.1. *Programmatuur*

Allereerst worden bedrijfssystemen geschikt gemaakt om aan meerdere gebruikers tegelijkertijd de indruk te geven dat ze de computer voor zich alleen hebben. Op deze wijze kan de kostbare capaciteit van computers vollediger worden benut en kunnen de kosten worden verdeeld over meerdere gebruikers. Via deze methode – *time-sharing* – kunnen meerdere gebruikers elk tegelijkertijd een taak laten uitvoeren.

Een tweede ontwikkeling op het gebied van de programmatuur is het ontstaan van zogenoemde programmapakketten. Het gaat dan om programma's die het inrichten van specifieke applicaties ondersteunen. Zo stellen databasemanagementsystemen organisaties in staat hun eigen administraties op maat elektronisch vorm te geven. Statistische analysepakketten ondersteunen empirische analyses. Information retrieval systemen maken het mogelijk grote hoeveelheden documenten elektronisch op te slaan en te doorzoeken op trefwoorden.

### 1.4.2. *Mainframe, randapparatuur en infrastructuur*

Time-sharing leidt ertoe, dat het basismodel wordt uitgebreid met meerdere input/output-automaten die terminals worden genoemd. Deze omvatten meestal een beeldscherm en een toetsenbord. Voor elke gebruiker een terminal, die soms

over afstand met de basiscomputer wordt verbonden. Op deze wijze wordt het netwerkarakter van het basismodel (dat immers bestaat uit communicerende automaten) meer zichtbaar, doordat de randapparatuur over afstand met de hoofdonderdelen van de computer (in deze opstelling *mainframe* genoemd) is verbonden. Bij de verbinding van terminals met mainframes wordt ook voorzichtig gebruik gemaakt van de openbare telecommunicatie-infrastructuur. De PTT's beginnen met het aanbieden van huurlijnen (*leased lines*) voor datatransmissie.

#### 1.4.3. *Maatschappelijk perspectief*

In de periode van 1960-1970 raakte de ontwikkeling van mainframes in een stroomversnelling. Computercapaciteit werd steeds betaalbaarder, betrouwbaarder en beter toegankelijk voor gebruikers. De organisatievorm bleef gericht op individuele computers. Het netwerk verbond de onderdelen. De bedrijfsorganisatie van mainframes was arbeidsintensief. Universitaire rekencentra kwamen van de grond. In Nederland groeiden deze uit tot organisaties met tientallen medewerkers. De bezwaren die kleven aan het gebrek aan standaardisatie werden voelbaar. Het wordt als storend ervaren dat een programma dat werkte op een computer van fabrikant X niet werkte op een computer van fabrikaat Y. Het begint duidelijk te worden, dat wanneer systeemprogrammatuur producentafhankelijk is, dit voor de computerfabrikant tot een enorme voorsprong leidde op de markt voor randapparatuur. Koppelverkoop van computers en randapparatuur werd steeds minder vanzelfsprekend gevonden. Aan de andere kant bracht de betrouwbaarheid van geheugens en processoren een groot vertrouwen in de praktische en commerciële mogelijkheden van de automatisering.

### 1.5. **Rond 1975: de softwarecrisis**

Er werden in de jaren zeventig dan ook zeer grote automatiseringsprojecten op touw gezet. Sommige daarvan waren succesvol (banken, verzekeringen), maar voor veel projecten gold dat de verwachtingen niet werden gehaald. Het ontbreken van een informaticaopleiding van niveau deed zich gelden. In 1970 waren er nog geen beproefde methoden en technieken beschikbaar om gebruikerswensen om te zetten in programma's. Evenmin is er steeds zicht op de vraag wat wel en wat niet kan worden geautomatiseerd. Enkele grote, veelomvattende automatiseringsprojecten mislukten en er werd gesproken over de softwarecrisis. Daarop werden drie antwoorden gevonden.

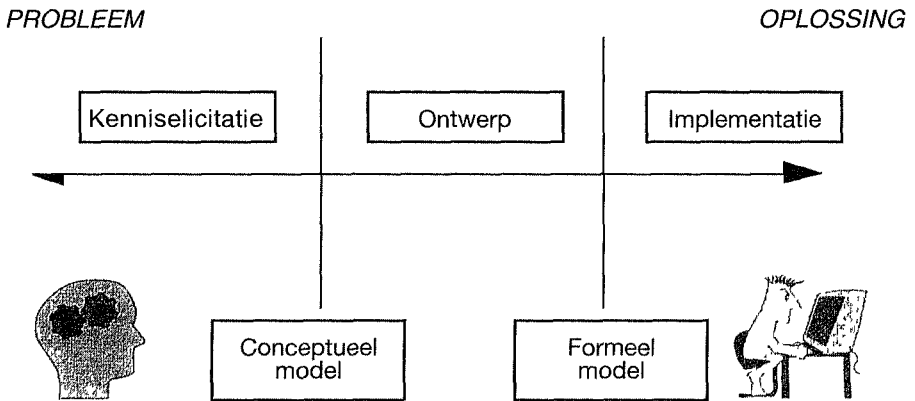
#### 1.5.1. *Ontwerpmethoden*

Het eerste antwoord bestond uit de ontwikkeling van analyse-, ontwerp- en programmeermethoden. Dijkstra ontwikkelde mede de notie van correctheidsbewijzen van programma's.<sup>8</sup> Veel ontwerpmethoden zagen rond deze tijd het licht. In

---

<sup>8</sup> O.-J. Dahl, E.W. Dijkstra & C.A.R. Hoare, *Structured Programming*, London: Academic Press 1972.

Schema 2 is een aantal kenmerken van deze methoden samengevat. Om een te automatiseren proces hanteerbaar te maken, wordt de automatisering in drie fasen gesplitst. In de eerste fase wordt een ontwerp gemaakt van de te bouwen oplossing in termen van functionaliteit. Dit ontwerp wordt gemaakt op conceptueel niveau, zodat het kan worden begrepen door de opdrachtgever. In de tweede fase wordt de oplossing op een meer formele wijze beschreven. Deze beschrijving leent zich voor formele waardering (consistentie, bereikbaarheid van een oplossing, tijdgebruik). Hier wordt al gedacht aan de wijze waarop de computer zich zal gedragen. Dit formele model kan worden omgezet (geïmplementeerd) in een computertoe-passing.



Schema 2: methodische, probleemgestuurde systeemontwikkeling

Wat als een probleem wordt gezien heeft bijvoorbeeld de dimensie van de financiële administratie van een bedrijf, of van een bibliotheek. Hetzelfde geldt voor oplossingen. Om deze problemen op te kunnen lossen worden ze ontleed in subproblemen. Subproblemen kunnen weer verder worden geanalyseerd. Deze analyse in subproblemen is nadrukkelijk onderdeel van de ontwikkelmethoden van deze tijd. Voor elk individueel te ontwikkelen informatiesysteem voor (bijvoorbeeld) een bedrijfsadministratie worden de problemen opnieuw geanalyseerd en worden subproblemen onderscheiden.

### 1.5.2. TCP/IP

Intussen bleven computercentra rond mainframes groeien. Er diende zich daarnaast een nieuw type middelgrote computer aan dat vaak in lokale netwerken onderling werd verbonden. Door de noodzaak tot (en de praktische moeilijkheden bij) het integreren van apparatuur van verschillende leveranciers in lokale netwerken traden de belemmeringen daartegen steeds nadrukkelijker op de voorgrond. In de Verenigde Staten werd geëxperimenteerd met communicerende mainframes in een computernetwerk (het ARPA-net). In de loop van het decennium groeide dit netwerk gestaag en werd in Europa een vergelijkbaar netwerk ontwikkeld

(EARN). Er ontstonden nieuwe inzichten over het verbeteren van snelheid en betrouwbaarheid van datacommunicatie. Er was behoefte aan het standaardiseren van de datacommunicatie tussen computers van verschillende leveranciers. In 1974 werd door Kahn en Cerf het ‘Transmission Control Protocol/Internet Protocol’ (TCP/IP) ontwikkeld. Het zou in de jaren negentig tot de *de facto* standaard van het internet uitgroeien. Inmiddels wordt mondjesmaat een nieuwe versie van het Internet Protocol (IPv6) geïmplementeerd, dat onder meer een enorme uitbreiding van het aantal beschikbare adressen met zich meebrengt.

### 1.5.3. *Relationele databases*

In 1975 was er nog een doorbraak in de IT: het relationele datamodel van Codd. Een deel van de softwarecrisis uit de jaren zeventig was er het gevolg van dat de gegevens die in een databasemanagementsysteem werden opgeslagen en bewerkt zodanig werden gemodelleerd, dat bij wijziging van het datamodel onvoorspelbare resultaten optraden bij gebruik van de beschikbare programmatuur. Deze conventionele, hiërarchische datamodellen konden moeilijk worden aangepast aan nieuwe eisen – en nieuwe eisen worden in een dynamische wereld steeds gesteld. Codd ontwikkelde een zeer eenvoudig, maar logisch sluitende wijze om datamodellen zodanig te ontwerpen, dat ze geschikt waren om te worden aangepast aan nieuwe tijden. In dit model is de rechthoekige datatabel de basis. Deze tabel is steeds uitbreidbaar met een andere rechthoekige datatabel, waarvan de rijen via een sleutel in verband worden gebracht met rijen uit de andere tabel. Codd ontwierp een eigen logische taal voor het beschrijven en het bevragen van uit dergelijke tabellen samengestelde databases. Een vraag aan een relationele database resulteert zo zelf in een rechthoekige tabel (die dan een *view* wordt genoemd). Het relationele model is zo krachtig dat het in de jaren zeventig een belangrijke bijdrage heeft geleverd aan de bestrijding van de softwarecrisis, terwijl het thans de noodzakelijke motor is achter vrijwel elke dynamische en enigszins complexe, van contentmanagement afhankelijke website (waarbij een selectie van gegevens uit de database wordt omgezet in webpagina’s). Een standaardtaal voor het beschrijven en bevragen van relationele databases wordt SQL (structured query language) genoemd.

## 1.6. **Rond 1985: de pc**

Miniaturisering en massaproductie van technische onderdelen (geheugen- en processorchips; schijfgeheugen) leiden in de jaren tachtig tot de pc.

### 1.6.1. *De pc*

PC is de afkorting van *personal computer*. De komst van de pc is het draaipunt in de maatschappelijke inburgering van informatietechnologie. Een universiteits-mainframe in de jaren zeventig kostte ongeveer 15 miljoen gulden. In de jaren

tachtig kwam dezelfde rekencapaciteit voor persoonlijk gebruik beschikbaar voor de prijs van een tweedehands auto.

### 1.6.2. *De gevolgen voor mainframes, randapparatuur en infrastructuur*

De eerste pc's werden aangeschaft door de (semi)professionele mainframe- en terminalgebruikers. Doordat pc's al beschikten over beeldschermen en toetsenborden en voor het overige veel machtiger waren dan terminals, werden ze al snel in de plaats van terminals ingezet. Daardoor transformeerden de mainframe-interne communicatiekanalen tussen computer en terminal naar verbindingen tussen zelfstandige computers. Daarmee nam het aantal lokale computernetwerken (*local area networks*) explosief toe, zonder dat aan de infrastructuur iets hoefde te gebeuren. De positie van mainframes werd in die zin bedreigd, dat veel van de taken leken te kunnen worden overgenomen door pc's.

### 1.6.3. *Het OSI-model*

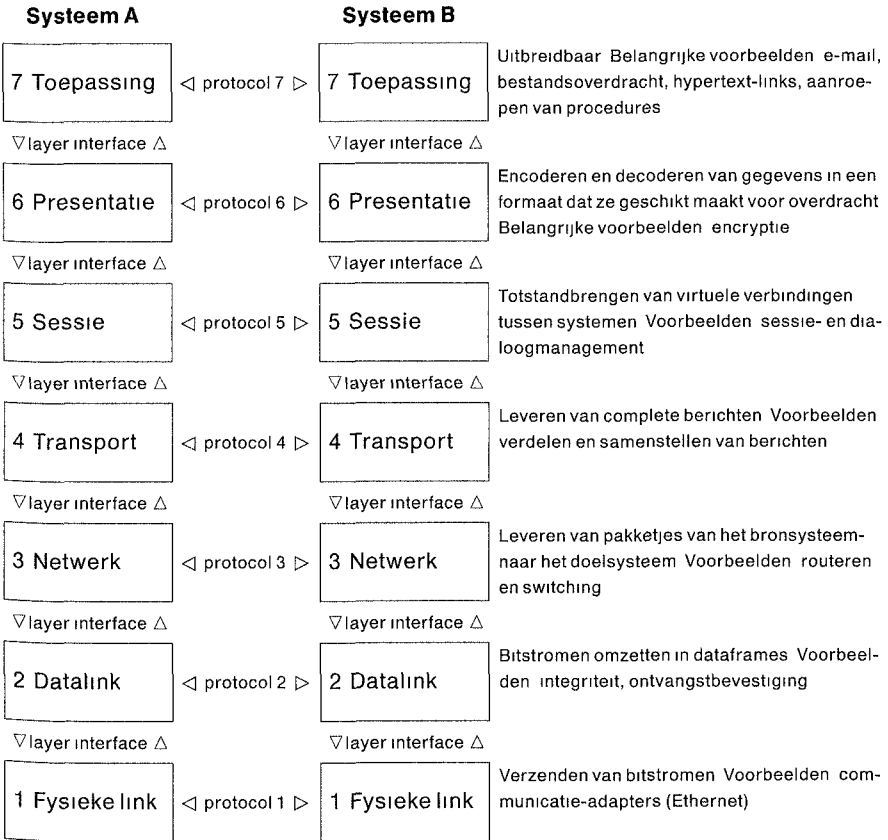
De als onnodig ervaren belemmeringen tegen effectieve communicatie tussen computers van verschillende leveranciers deden zich in toenemende mate voelen. De 'International Standardisation Organisation' (ISO) begon een project ter ontwikkeling van een referentiemodel voor 'Open System Interconnection' (OSI). Het is een poging om een gelaagd raamwerk te bieden voor communicatieprotocollen en om deze te onderscheiden van overige (computer)stelselactiviteiten.

*Protocollen* maken het mogelijk om communicatietaken te specificeren, net als programmeertalen het mogelijk maken om rekentaken te specificeren. Een uitdrukking in een bepaald protocol zorgt ervoor dat de communicatietaken die bij dat protocol hoort kan worden uitgevoerd. Net als bij programmeertalen kunnen ze van verschillend logisch niveau zijn, dat wil zeggen dat uitdrukkingen in een protocol op een lager logisch niveau deel kunnen uitmaken van een bericht uit het protocol op het hogere logische niveau.

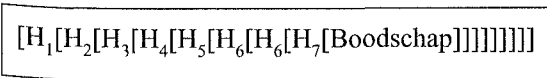
Het ISO-OSI model maakt een omgekeerde indeling in niveaus. Het meest conceptuele niveau wordt als hoogste genomen, en het meest geformaliseerde, gefragmenteerde niveau als laagste. De gedachte is dat een bericht op het hoogste niveau, het niveau van de toepassing (bijvoorbeeld elektronische brief), tussen aanhalingstekens wordt geplaatst en wordt doorgegeven aan het lagere niveau, verrijkt met de extra informatie die nodig is om de communicatiefuncties van dat niveau te kunnen uitvoeren.

Die informatie wordt (als *header*, aangeduid met een H in Schema 4) toegevoegd aan de eigenlijke boodschap (de vierkante haken dienen als aanhalingstekens/envelop). Dit proces wordt per niveau-overgang herhaald, zodat uiteindelijk, als een bericht fysiek over het net is gegaan, het uit acht enveloppen moet worden gepeld. De boodschap is in Schema 4 voor de communicatie zodanig verrijkt, dat deze zal worden herkend door het geadresseerde systeem als een bericht dat ver-

volgens per laag van een envelop en header wordt ontdaan om ten slotte aan te komen op het applicatieniveau. Dan is de verzonden elektronische brief weer als zodanig te herkennen en te lezen door de geadresseerde. De transformatie van laag naar laag wordt *layer interface* genoemd.



Schema 3: het OSI-model



Schema 4: als bericht verpakte boodschap

In feite kan de afhandeling per laag complexer zijn, doordat voor de communicatie de berichten in pakketjes van ongeveer gelijke lengte worden opgedeeld. Deze pakketjes hoeven niet hetzelfde traject te volgen gedurende de communicatie. De pakketjes moeten natuurlijk wel weer in de goede volgorde worden geplaatst alvo-

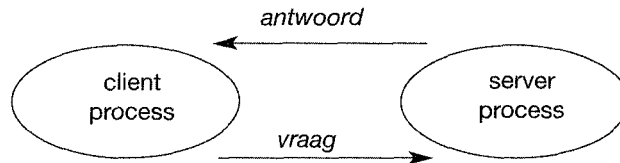
rens ze kunnen worden begrepen op het applicatieniveau, maar dat is precies wat er op de lagere niveaus van het model gebeurt.

Het OSI-model vervult zo op een abstract niveau de voorwaarden voor *interconnectiviteit* en *interoperabiliteit* van overigens diverse apparatuur en programmatuur. Het is een *de jure* standaard, die niet of nauwelijks in de praktijk is gevolgd. Wel is het een goed referentiemodel – en dus geschikt om veel ontwikkelingen, zoals de *de facto* communicatiestandaard van het internet of de interconnectiviteit tussen de diensten van verschillende mobiele telefoonaanbieders te analyseren.

#### 1.6.4. Objectoriëntatie, client-server en multi-tasking

Door terminals te vervangen door pc's groeide het aantal lokale netwerken explosief. Computers kunnen veel meer dan terminals. Lokaal kunnen berekeningen worden uitgevoerd en databases worden bijgehouden.

Eén van de problemen die vrij algemeen wordt herkend is de geringe herbruikbaarheid van programma's. Uit de erkenning van dit probleem ontstaat een belangrijke oplossingsrichting: *objectoriëntatie*. Hierbij streeft men naar het creëren van bibliotheken met herbruikbare programmaobjecten. De interne structuur van programmaobjecten is daarbij irrelevant; ze worden ingekapseld in een groter geheel (*encapsulation*). Objecten worden gekenmerkt door het gebruik van een beperkt aantal berichten (*messages*). Deze activeren processen die bij een bepaald bericht horen (*methods*). Ontwerpen en programmeren wordt dan in belangrijke mate het in een architectuur schikken van bestaande objecten. Deze objecten fungeren dan als bouwstenen.<sup>9</sup>

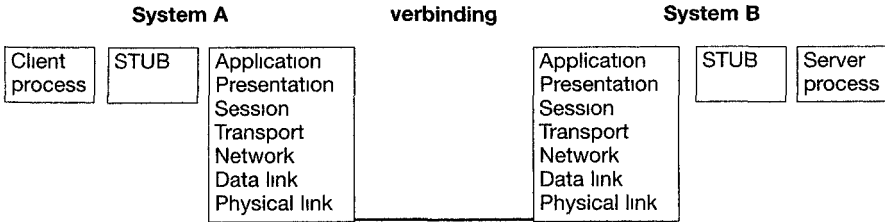


Schema 5: het client-server model

Deze benadering is op een hoger niveau van complexiteit terug te vinden in een zeer krachtig model voor *distributed processing*, voor samenwerkende computers. Daarbij houdt een bepaalde computer (de *server*) een programma ter beschikking dat vragen van andere computers (de *clients*) in het netwerk kan beantwoorden. Daartoe beschikken de clients over clientprogramma's (die clientprocessen wor-

<sup>9</sup> Auteursrechtelijk kan hier dan weer sprake zijn van een verzamelwerk.

den genoemd) en servercomputers over serverprocessen.<sup>10</sup> Deze zijn zo gemaakt, dat ze kunnen samenwerken en communiceren. Om deze zienswijze nader toe te kunnen lichten, integreren we het client-server model met het OSI-communicatiemodel. De vragen van de client aan de server zijn dan procedureaanroepen over afstand (*remote procedure calls*, afgekort als RPCs).



Schema 6: OSI en client-server gecombineerd

In Schema 6 wordt het OSI-model voor communicatie gecombineerd met het client-server model. Het resultaat is een model voor het over afstand aanroepen van programma's of procedures. Daarbij valt op dat niet alleen ten behoeve van telecommunicatie een meerlagenmodel gebruikt wordt, maar dat ook op het applicatieniveau van de client en de server een fase voor decompositie en compositie is voorzien, die *stub* wordt genoemd. Ook hier gaat het weer om de mogelijkheid om clients die op verschillende computerfabrikaten (of onder verschillende bedrijfssystemen) draaien, met dezelfde server (bijvoorbeeld in een gedistribueerd databasesysteem) te kunnen laten werken.

Omdat niet goed valt in te zien waarom samenwerkende processen alleen nuttig zouden zijn voor communicerende computers, en niet voor meerdere programma's op een enkele computer, ontstaat met de mogelijkheid van distributed processing de behoefte aan *multi-tasking*. Het operating system treedt dan op als server (bijvoorbeeld: Unix en, later, Windows) en de spread-sheet en tekstverwerkingsprogramma's als clients.<sup>11</sup> Alleen van het communicatiemodel worden vaak om efficiencyredenen (en misschien ook om mededingingsredenen) delen kortgesloten wanneer de processen op één en dezelfde computer met elkaar communiceren.

<sup>10</sup> Ik zie nu maar af van de ook hier optredende vervlechting en convergentie: het client-server model is een model van rollen. Wanneer computers A en B communiceren kan de rolverdeling nu eens zo zijn dat A vraagt en B bedient, en even later omgekeerd. Helaas heeft dit type verduidelijking vooral de neiging verwarring te scheppen.

<sup>11</sup> Een uitwerking van deze zienswijze geeft bijvoorbeeld Geert van Rossum, 'Client/Server: een nieuw concept gedefinieerd en geevalueerd', *Informatie* 1993/9.

### 1.6.5. *Peer-to-peer*

Bij het client-server model is de regie in handen van de server. De server bepaalt de volgorde waarin verzoeken van clients worden afgehandeld. De communicatiestructuur is als die van een stervormig netwerk. Als gevolg van de inspanningen die in de jaren tachtig werden ondernomen om dure 'resources' zoveel mogelijk te 'sharen' zijn bedrijfssystemen ontwikkeld die ondersteunen dat de vrije schijfruimte van alle pc's op een netwerk voor elke pc-gebruiker beschikbaar was. NFS van SUN is een dergelijk *file sharing* systeem. In dergelijke systemen is sprake van directe samenwerking tussen 'gelijken', zonder centrale communicatiemanager. Dit model ligt ten grondslag aan internetapplicaties als Kazaa en Freenet.

### 1.6.6. *Standaards*

Objectoriëntatie, client-server applicaties en software-suites zijn alleen mogelijk, wanneer modules met elkaar kunnen communiceren. Daarvoor zijn talen nodig. Softwarefabrikanten en dienstverleners hebben de overigens geheel natuurlijke impuls om idiosyncratische aspecten in de door hen gebruikte talen te definiëren. Op die wijze kunnen ze immers een eenmaal veroverde voorsprong op de markt voor informatiesystemen en voor informatiediensten beschermen. Het bedrijf Microsoft heeft een voorsprong op de markt voor programma's die gebruik maken van eigenaardigheden van het Windows-operating system, omdat bij dat bedrijf bij elke nieuwe versie eerder dan bij de concurrentie bekend zal zijn wat die eigenaardigheden zullen worden. Hier liggen problemen op het gebied van de mededinging. Het is niet denkbeeldig dat een softwarefabrikant nieuwe versies van zijn programma's uitbrengt (en lang van tevoren aankondigt) om zijn marktpositie te verstevigen.

### 1.6.7. *Kunstmatige intelligentie*

Kunstmatige intelligentie is één van de begrippen die (vooral in de jaren tachtig) een groot publiek aanspreekt. De term verwijst enerzijds naar een filosofisch getinte discussie over de vraag of computers intelligent gedrag kunnen vertonen.<sup>12</sup> Maar anderzijds, en meer praktisch, gaat het om een tak van de informatica die zich bedient van technieken die zijn ontleend aan modellen voor probleemoplossen. Daarbij zijn belangrijke aspecten onderkend die een rol behoren te spelen bij het maken van computerondersteuning voor het gezamenlijk met mensen uitvoeren van moeilijke taken. Ik noem enkele kernbegrippen. *Kenniselicitatie* is het proces waarbij de voor het uitvoeren van moeilijke taken benodigde kennis wordt opgespoord en bespreekbaar gemaakt.

---

<sup>12</sup> Zie in dit verband Hofstadter's *Gödel, Escher, Bach*, Amsterdam: Contact 1985 en ook: H.J. van den Herik, *Kunnen computers rechtspreken?*, Arnhem: Gouda Quint 1991.

Dat dit problematisch kan zijn valt te illustreren aan de hand van expertgedrag (zoals van de ervaren automobilist) en diens onvermogen om de kennis die hij kennelijk gebruikt volledig en eenduidig te beschrijven. *Kennisrepresentatie* is de manier waarop de benodigde kennis voor het computerprogramma zal worden weergegeven. Hier zijn verschillende representaties nodig, die lopen van conceptueel (voor mensen begrijpelijk) naar formeel (voor computers uitvoerbaar, bewerkbaar). Een karakteristiek representatievraagstuk betreft vage begrippen, waardeoordelen, intuïties en overtuigingen.

Van betekenis voor de AI<sup>13</sup>-benadering is ook de scheiding die wordt gemaakt tussen de statische kennis en de kennis over de wijze waarop met die kennis wordt omgegaan, wordt geredeneerd. Opgemerkt moet wel worden dat de methoden die werden bedacht voor het ontwerpen van kennissystemen (zoals in Nederland: KADS<sup>14</sup> en DESIRE<sup>15</sup>) met het verloop van de jaren steeds dichter bij de conventionele methoden voor het ontwikkelen van informatiesystemen (zoals SDM<sup>16</sup> en Requirements analysis<sup>17</sup>) zijn komen te liggen.<sup>18</sup>

De kunstmatige intelligentie is van groot belang geworden voor de rechtsinformatica.<sup>19</sup> Aan de relevante werkwijzen en methoden heeft Anja Oskamp een apart hoofdstuk (hoofdstuk 14) van dit boek gewijd.

### 1.6.8. *Maatschappelijk perspectief*

De transformatie van rekencentra rond mainframes naar schakelcentra voor communicatie tussen vooral personal computers zette in. Computercapaciteit kwam ter beschikking van niet-professionele gebruikers. Er was een enorme uitbreiding van de markt voor software. Mede daardoor werd gebruiksvriendelijkheid een belangrijk mededingingsaspect voor programmaleveranciers. Aanvankelijk werd gedacht dat wanneer pc's voor individueel gebruik zijn, er geen behoefte zou bestaan aan time-sharing. De eerste bedrijfssystemen voor pc's betekenden in dit opzicht dan ook een grote stap terug. Voor operating systems leek de geschiedenis opnieuw te beginnen. Pas aan het einde van het decennium kwam de time-sharing technologie van de jaren zeventig als 'multi-tasking' beschikbaar voor de pc van de jaren tachtig. Kunstmatige intelligentie werd een belangrijk trefwoord in de IT. Computers werden in netwerken nu ook gebruikt voor het sluiten van handelstransacties over afstand (EDI). Overheden begonnen de informationele krachten te bundelen en gebruikten netwerken om informatie langs elektronische weg uit te

<sup>13</sup> AI is de afkorting van 'Artificial Intelligence' Deze is om begrijpelijke redenen meer ingeburgerd dan de Nederlandstalige 'Kunstmatige Intelligentie'

<sup>14</sup> G Schreiber, B Wielinga & J Bruker, *KADS*, London Academic Press 1993

<sup>15</sup> W Kowalczyk & J Treur, 'On the use of a formalised generic task model', in B Wielinga e a (red), *Knowledge Acquisition*, Amsterdam IOS Press 1990

<sup>16</sup> H B Eilers, *Systeemontwikkeling volgens SDM*, Schoonhoven Academic Service 1991

<sup>17</sup> R J Wieringa, *Requirements Engineering, Frameworks for understanding*, New York Wiley and Sons 1996

<sup>18</sup> Of *vice versa*, natuurlijk Zie ook Schreiber, G, H Akkermans, A Anjewierden, R de Hoog, N Shadbolt, W Van de Velde en B Wielinga, *Knowledge Engineering and Management The CommonKADS methodology*, Cambridge, MA The MIT Press

<sup>19</sup> Deze onderzoeksgemeenschap is al meer dan een decennium verenigd in de Stichting Juridische Kennissystemen (Jurix), <http://www.jurix.nl>

wisselen. Hier werden dezelfde standaards ingezet als bij EDI. Er wordt in dit verband wel gesproken van publiekrechtelijke EDI.

### 1.6.9. *Twee vormen van electronic data interchange (EDI)*

EDI is de term die wordt gebruikt voor de uitwisseling van gestructureerde berichten tussen onafhankelijke computers van organisaties die daarmee beogen rechtsgevolgen in het leven te roepen. In termen van het gecombineerde model voor communicatie en client/server-toepassingen (Schema 6) zijn de onafhankelijke computers van organisaties uit bovengenoemde definitie client/server-programma's op applicatieniveau. De taal waarvan deze programma's gebruik maken moet onafhankelijk zijn van het computerfabrikaat. Het is niet goed wanneer een kruideniersketen die EDI gebruikt zijn toeleveranciers daarmee dwingt tot de aanschaf en het gebruik van een bepaald fabrikaat computer. Het gaat dan ook weer om een techniekonafhankelijke standaardtaal voor geautomatiseerde berichtenuitwisseling die rechtsgevolgen tot stand kan brengen.

#### *Gesloten EDI*

In de jaren tachtig werd EDI vooral toegepast voor bestaande verhoudingen tussen handelspartners die elkaar vertrouwen. De risico's waren dan minder zwaarwegend en daardoor waren de voordelen duidelijk. Dergelijke EDI-systemen zijn tamelijk statisch. De deelnemers liggen vast. De infrastructuur waarvan gebruik wordt gemaakt (huurlijnen), is alleen voor het EDI-verkeer bestemd. Het gaat om *gesloten EDI*. Deze variant van EDI lijkt volkomen beheersbaar. Achterliggende, voorwaardenscheppende afspraken tussen de deelnemers gaan over technische aspecten (zoals beschikbaarheid en betrouwbaarheid), over organisatorische aspecten (als: beveiliging, de standaard die wordt gebruikt voor de berichten) en over juridische aspecten (zoals risicoverdeling). Deze achterliggende afspraak wordt de *interchange agreement* genoemd. Het gaat om een afspraak met meer partijen (de deelnemers). De toetredende partijen worden als het ware lid van een 'EDI-club'. De interchange agreement kan worden uitgebreid in de richting van gespecialiseerde dienstverleners (van bijvoorbeeld: telecomcapaciteit, beveiliging, archieven voor bewijs en bewaring). Deze interchange agreement ondersteunt de *transaction agreements*; dit zijn de overeenkomsten die totstandkomen door het EDI-berichtenverkeer zelf. Bij gesloten EDI zijn de deelnemers over het algemeen bedrijven. De dienstverleners die vertrouwelijke en/of beveiligingsdiensten leveren worden over het algemeen Trusted Third Parties (TTP's) genoemd.

#### *Open EDI (electronic commerce)*

De ontwikkeling van wereldwijde communicatienetwerken heeft nieuwe vergezichten doen ontstaan. Gedacht wordt wel aan elektronische markten waar door slimme EDI-programma's het gewenste product met de beste prijs/prestatieverhouding wordt opgespoord en aangekocht. Voor dergelijke EDI-systemen is het nodig dat deelnemers vrij en *ad hoc* kunnen participeren aan het systeem (en hun participatie beëindigen). Algemeen wordt aangenomen dat het vertrouwen tussen de deelnemers aanmerkelijk sterker dient te worden ondersteund door beveili-

gingsmaatregelen dan bij gesloten EDI. In dit soort systemen kan het immers voor de individuele gebruiker de moeite lonen het vertrouwen van de andere deelnemers te misbruiken.

Hierdoor dienen zich talloze vragen aan. Bijvoorbeeld de identificatievraag (hoe weet ik dat dit bericht echt van afzender X afkomstig is?), de authenticatievraag (hoe weet ik dat afzender X deze verplichting aan mag gaan?), de vragen over juridische vormgeving (hoe zorg ik ervoor dat de risico's en aansprakelijkheden worden verdeeld op een aanvaardbare wijze, welke contractmogelijkheden zijn hierbij mogelijk?). Dergelijke EDI-systemen zijn dynamisch: de deelnemers wisselen bij voortdurend. De infrastructuur waarvan gebruik wordt gemaakt is multifunctioneel.

De vraag is in hoeverre behoefte is aan dergelijke open EDI; de elektronische marktplaatsen voor bedrijven, die vandaag de dag sterk in opkomst zijn, drijven weliswaar op een vergelijkbare infrastructuur, maar zullen veelal openstaan voor menselijk ingrijpen bij het bevestigen van transacties. Wel werd in de afgelopen jaren duidelijk dat het (open) internet langzamerhand geaccepteerd wordt als medium voor dergelijke elektronische handel. De elektronische marktplaatsen zijn combinaties van bedrijven, die zich gezamenlijk manifesteren om op die wijze schaalvoordelen te behalen bij de inkoop van goederen.

#### *Karakteristieke verschillen tussen gesloten en open EDI*

De evolutie van gesloten naar open EDI wordt al een tijd verwacht, maar is nog niet verwezenlijkt. Genoemde evolutie heeft een krachtig reorganiserende potentie, en misschien laat hij daarom op zich wachten. De kenmerken zijn evenwel al enige tijd bekend.<sup>20</sup> We geven ze hier omdat de vraagstukken zeer verwant zijn aan die welke in verband met eventuele regulering van het internet opnieuw worden gesteld. Als gesloten EDI de structuur heeft van samenwerkende bedrijven die in een min of meer vaste 'club' samenwerken om hun bedrijf beter te kunnen uitvoeren, dan wordt hieraan bij open EDI een organisatorische laag toegevoegd: de klanten van de bedrijven worden ook lid van een 'club' (nu niet via een interchange agreement, maar via de constructie van algemene voorwaarden).

Eén van de redenen waarom de beweging van gesloten naar open EDI moeilijker verliep dan aanvankelijk werd verwacht is zeker ook het probleem van de algemene beschikbaarheid van een communicatie-infrastructuur. Deze belemmering is met de doorbraak van het internet deels weggenomen. Maar daarin schuilt ook een probleem: de infrastructuur en het protocol waarop internet draait (TCP/IP) zijn onveilig. Het veiligheidsprobleem moet worden ondervangen met het gebruik van encryptie en TTP's. Daarnaast moet het vertrouwen van consumenten en bedrijven aanwezig zijn. Tot op heden is dit vertrouwen verre van optimaal; een aankoop op het internet wordt nog al te dikwijls en meestal ten onrechte gezien als een vorm van Russische roulette.

---

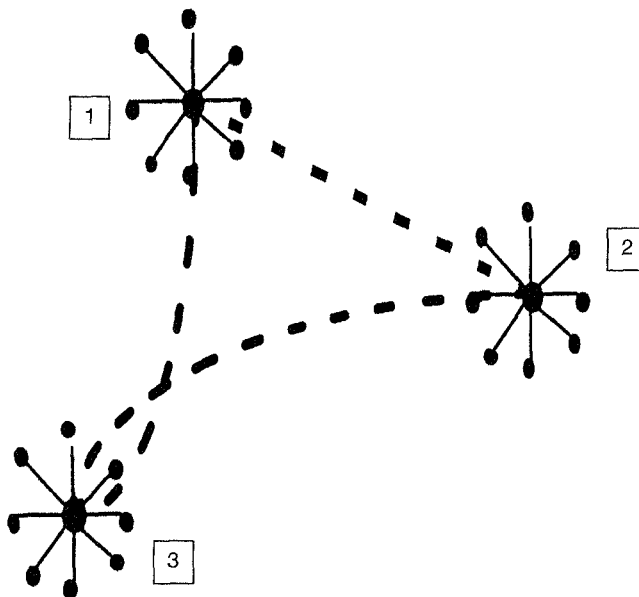
<sup>20</sup> A.H.J. Schmidt, 'TEDISH – EDICON. Final Report – TEDIS Phase II Task F4: Legal Aspects', in: *The EDI Law Review* 1997, 4/1, p. 5-49.

## 1.7. Rond 1995: het internet

Het internet, ontstaan toen militaire en universitaire netwerken met elkaar verbonden werden, begon halverwege jaren negentig aan een opmars buiten de academische wereld.

### 1.7.1. Infrastructuur en organisatie van het internet

In de jaren zeventig waren mainframes toegankelijk gemaakt voor grotere groepen gebruikers door het inzetten van terminals. Hierbij was meestal gebruik gemaakt van huurlijnen. In de jaren tachtig waren de terminals vervangen door pc's. Intussen werden de mainframes van de jaren zeventig en tachtig in toenemende mate onderling verbonden door communicatieverbindingen.



Schema 7: een netwerk van netwerken van ...

Het gevolg is in de jaren negentig zichtbaar geworden: een wereldwijde communicatie-infrastructuur met onderling verbonden computers was beschikbaar. En deze infrastructuur is in belangrijke mate gebaseerd op lokale bekabeling en op gehuurde datatransmissielijnen, dat wil zeggen dat belangrijke delen van deze infrastructuur buiten de openbare communicatie-infrastructuur liggen en daarmee: buiten de zeggenschap van de al dan niet publieke telecom-operators (voorheen: PTT's).

In Schema 7 zijn drie lokale netwerken weergegeven die zijn ontstaan uit drie mainframes. Elke server is met de twee andere servers verbonden door een data-transmissielijn, zodat er een netwerk van netwerken is ontstaan. De datatransmissielijnen worden betaald door de beheerders van de lokale netwerken die met elkaar zijn verbonden. Het geschetste netwerk kan zich uitbreiden doordat een van de deelnemers aansluiting zoekt bij een andere deelnemer aan een ander netwerk van netwerken. Op deze wijze is een wereldomspannende communicatie-infrastructuur ontstaan van netwerken van netwerken van communicerende computers. Deze infrastructuur noemen we het internet.

### 1.7.2. De diensten die over het internet worden aangeboden

De diensten die achtereenvolgens zijn ontstaan weerspiegelen de ontwikkelingen in de loop van de tijd. Eerst de diensten die voor professionals van belang waren die gebruik maakten van dure mainframecomputers (ftp, telnet, e-mail). Vervolgens een standaardisering van het gebruik van e-mail in de richting van nieuwsbulletins en discussiegroepen voor de toegenomen communicatiebehoefte van niet-professionele IT-gebruikers. Daarna de ontwikkeling van kwalitatief hoogwaardig weer te geven informatie met behulp van door de informatieaanbieders aan te leggen verwijzingen (multimediale webpages).

Hiermee is een ontwikkeling in gang gezet die spoedig leidde tot een gigantische, multimediale, wereldbibliotheek als informatienetwerk. Daarin is behoorlijk wat persoonlijke en vooral heel veel commerciële informatie terechtgekomen, die toegankelijk wordt gemaakt met behulp van *browsers*. Vervolgens de ontwikkeling naar zoeksystemen, naar *crawlers* en *search engines*. Ten slotte de ontwikkeling van een standaard voor applicatie-uitwisseling (Java). We bespreken de ontwikkelde (veelal geautomatiseerde) diensten heel in het kort. We maken daarbij wederom gebruik van het OSI-model om de diensten een plaats te geven in het communicatieproces. Ik ga nu van laag naar hoog.

#### 1.7.2.1. DE EERSTE LAAG: TRANSPORT VAN BITSTREAMS

Het gaat om de meest primitieve van de telecomdiensten: het verschaffen van capaciteit voor het transporteren van elektronische eenheden, van bits. Een *bridge* is een apparaat dat op dit niveau werkt, omdat het twee lokale netwerken met elkaar verbindt die gebruik maken van dezelfde protocol-implementaties. Tussen computer A en computer B van Schema 3 gedacht, komt een bridge niet verder dan niveau 1.

#### 1.7.2.2. DE TWEDE LAAG: DATATRANSPORT

Op dit niveau worden de bits al samengenomen in *packets*, die voor verzending als een geheel worden gezien. Een bericht kan worden opgedeeld in packets. Hiervoor is geen afzonderlijke geautomatiseerde dienst. Het gaat om een input/output taak die wordt gerealiseerd door de communicatie-adapter, die als een onderdeel van de communicerende computer wordt gezien.

### 1.7.2.3. DE DERDE LAAG: TRANSPORT VAN PACKETS OVER EEN NETWERK

Bij communicatie speelt adressering een rol: de verschillende packets die samen een bericht uitmaken kunnen langs verschillende wegen worden verzonden. Voor het bepalen van de te nemen route wordt (automatisch) gezorgd door *routers*. Een router is een apparaat dat bijdraagt aan de keuzen die gemaakt worden over het communicatietraject. Hierbij is kennis over de eerste drie lagen nodig: een router moet tot niveau 3 in de berichten kunnen kijken. Een belangrijke taak in deze laag is ervoor te zorgen dat de packets ongeschonden overkomen. Daarvoor zijn verschillende methoden, waarvan er sommige berusten op meervoudige vraag-en-antwoordschema's. Merk op dat deze vorm van betrouwbaarheid haast onvermijdelijk botst met de (elektronische) anonimiteit.

### 1.7.2.4. DE VIERDE LAAG: TRANSPORT VAN BERICHTEN OVER EEN NETWERK

In deze laag worden de packets samengenomen in *berichten* die voor de hogere lagen als een eenheid van (getransporteerde) informatie worden beschouwd. Deze dienst wordt geleverd op het niveau van de applicaties, die in staat zijn bij elkaar behorende packets adequaat te presenteren aan het systeem of aan de gebruiker.

### 1.7.2.5. DE VIJFDE LAAG: SESSIES

In de vijfde laag is de eenheid: een sessie, of een reeks van uitgewisselde berichten (als in een conversatie twee gebruikers via internet, zoals in een *chatbox*). De diensten die hier worden geleverd, betreffen vooral de verrekening van gebruikte capaciteit. Op dit niveau is het instandhouden van een verbinding tussen twee communicerende computers gedurende een conversatieachtige berichtenuitwisseling aan de orde. Doordat voor de belangrijkste communicatietoepassingen niet meer wordt uitgegaan van de noodzaak van vaste verbindingen, maar wordt vertrouwd op het simuleren daarvan met behulp van fragmentarische, samenwerkende communicatietechnieken op lagere niveaus, is een sessie op internet dikwijls volstrekt virtueel (dat wil zeggen: niet als zodanig aan te wijzen door kenmerken van de communicatie op lagere niveaus). Wel komt de sessie nog voor als een eenheid bij de communicatie tussen bijvoorbeeld een bank en een klant van die bank. Dergelijke sessies verlopen via een door encryptie beveiligde 'virtuele verbinding', die na een bepaalde tijd verloopt. De sessie is dan afgelopen, en als de klant verder wil communiceren met de bank, dan moet hij opnieuw zijn gebruikersnaam en wachtwoord opgeven.

Omdat sessies kunnen zijn opgebouwd uit gefragmenteerde elementen is het soms praktisch vanuit het sessiebeheer om paden en adressen die bij een sessie worden gebruikt expliciet te maken en te onthouden voor verder gebruik. Hiervoor zijn de zogenoemde *cookies* in zwang geraakt: kleine berichten die de server aan de client aanbiedt om op te slaan (bijvoorbeeld om diens internetidentiteit te kunnen bevestigen). Op deze wijze kunnen servers zich bovendien weren tegen sessies met anonieme clients. Maar ook: op deze wijze kunnen servers sporen bij

clients achterlaten die (bijvoorbeeld) weer kunnen worden benut bij het vervaardigen van profielen van gebruikers.

Een belangrijk aspect van een sessie op het internet (zoals deze wordt ervaren door de gebruiker) is, dat deze kan bestaan uit meerdere deelsessies met verschillende servers. Dat komt omdat bij het bekijken van webpagina's, de browser kan worden gebruikt om via een *hyperlink* verder te kijken in een gerelateerd document dat op een geheel andere plaats op het internet beschikbaar is. Op deze manier verliest de gebruiker nog verder de controle over de fysieke plaats van de drager van de door hem geraadpleegde informatie.

Deze laatste functie heeft negatieve connotaties in verband met (zelf)censuur en privacyrisico's: door dergelijke profielen is immers niet meer alle informatie beschikbaar en aan het voorkeursprofiel kan informatie worden ontleend over de persoon voor wie het profiel geldt. In de Verenigde Staten worden dergelijke makelaarsdiensten wel aangeboden om (al dan niet computeranalfabete) ouders de middelen aan te reiken waarmee zij kunnen voorkomen dat hun internetsurfende kinderen kennis nemen van pornografisch materiaal.

Ook bij nieuws- en discussiegroepen komt het voor dat er gebruik wordt gemaakt van autoriteiten die als redacteurs optreden en de inhoud van de discussie/nieuwsvoorziening regelen. Dergelijke functionarissen worden 'moderators' genoemd.

Bij de informatiemakelaardij wordt gebruikgemaakt van *labeling authorities*. Dit zijn instanties die informatie scannen en voorzien van labels die worden uitgedrukt in termen van het eerdergenoemde gebruikersprofiel en waaraan de *browsers* en *search-engines* zich iets gelegen laten liggen.

#### 1.7.2.6. DE ZESDE LAAG, PRESENTATIE – 'HTML-BROWSERS'

Hier is de eenheid: de presentatie, de wijze waarop de informatie (het bericht, het bestand) wordt gepresenteerd. Tegenwoordig wil dat zeggen: afgebeeld, ten gehore gebracht (filmisch), afgespeeld. Voor het omzetten van bestanden in beeld, geluid en film zijn standaarden ontwikkeld. In de zesde laag worden die standaarden toegepast. Hier opereren *viewers* en *players*, maar vooral *browsers*. Dit zijn de programma's die bestanden die zijn vastgelegd met behulp van de HTML-standaard (taal), in beeld kunnen brengen. HTML is de *techniekonafhankelijke* standaardtaal voor het opmaken van documenten. Deze standaard is gebaseerd op SGML, een veel uitgebreidere *mark up language*, waarmee men niet alleen de stijlkenmerken van een tekst of afbeelding kan definiëren, maar ook betekenis kan hechten aan bepaalde elementen. Mark-up talen zoals XML, dat gebaseerd is op SGML, vormen een uitbreiding op HTML. Deze uitbreiding is hard nodig, nu de wens is ontstaan om gegevens op het internet op snelle en effectieve wijze met elkaar te vergelijken. XML is hard op weg de grammatica te worden, waarbinnen veel verschillende kennisculturen hun eigen formele vaktalen ontwikkelen. Deze talen zijn weer behulpzaam bij het techniekonafhankelijk specificeren van kennis en informatie die met behulp van IT kunnen worden verwerkt en bewerkt. In de gebruikerscontext kunnen daarmee IT-gerelateerde culturen ontstaan die een

zodanige geordende conceptuele ontwikkeling doormaken, dat ze daadwerkelijk gebruik kunnen blijven maken van eerder en elders ontwikkelde IT-objecten.

### 1.7.2.7. DE ZEVENDE LAAG: APPLICATIONS

De zevende laag is de laag waarop de informatie wordt teruggebracht in een gedaante die door de applicatie<sup>21</sup> kan worden begrepen. Het gaat hier bijvoorbeeld om de programma's die de post afhandelen. Niet iedereen gebruikt hetzelfde post-programma/systeem. Een *gateway* is een apparaat dat de communicatie verzorgt tussen twee netwerken met verschillende implementaties van communicatieapplicaties. Een gateway kijkt tot het applicatieniveau en vertaalt tussen standaarden (bijvoorbeeld van verschillende elektronische postsystemen).

De bekendste applicaties op het internet zijn respectievelijk *e-mail*, *news*, *discussiegroepen*, *www* en *search engines*. Ze worden kort beschreven in de verklarende begrippenlijst in dit boek. Recent zijn hieraan file-sharing applicaties toegevoegd als Kazaa, en de vele varianten daarvan, die het mogelijk maken om informatie van verschillende aard (van Word-documenten tot muziek, van programmabestanden tot films) met andere internetgebruikers te delen. Het enkele jaren geleden al ter ziele gegane Napster maakte daarbij gebruik van een centrale verwijzindex, maar Kazaa heeft een dergelijke verwijzindex niet nodig. Daardoor zijn inbreuken op het auteursrecht door middel van Kazaa ook bijzonder lastig aan te pakken – waar dit bij Napster nog wel mogelijk was. Het netwerkprogramma Freenet maakt het zelfs bijna onmogelijk om de plaats van individuele bestanden te herleiden tot individuele gebruikers. Vertaald naar de praktijk: ook bestanden die bij verspreiding een uitingsdelict opleveren, zoals kinderporno, zijn in opgeslagen vorm binnen Freenet niet te traceren tot individuele gebruikers.

#### *De dienstverleners*

Er treedt voor de eerste vier lagen geen differentiatie op van dienstverleners, althans niet per laag in het OSI-model. De onderste vier lagen zijn meestal in één organisatorische hand. Voor lokale netwerken is dat meestal de internetprovider. Voor het transport tussen netwerken, maar ook tussen gebruikers en internetproviders kan zeer wel gebruik worden gemaakt van de openbare telecommunicatie-infrastructuur. De spelers in dat veld zijn de houders van telecomlicenties. Het zijn de aanbieders van de infrastructuur, de 'telecom-providers'. Daartussenin (dat wil zeggen: tussen de telecomlicentiehouders en de internetproviders in) kunnen weer regionale, nationale, en supranationale aanbieders een rol spelen.

### 1.7.3. *Open source vs. proprietary software*

Richard Stallman is een informaticus. Hij kreeg in 1987 ernstig last van verstoorde wederkerigheidsgevoelens toen een printerfabrikant hem weigerde de bijbehorende programmatuur in leesbare vorm te verstrekken, zodat hij de daarin voor-

---

21 Een applicatie is hier een communicatietoepassingsprogramma.

komende fouten zou kunnen verbeteren. Hij verzong een contract (de *General Public Licence*) en een beweging (de *open source* beweging), die de deelnemers in staat stellen van de broncode van elkaars programma's gebruik te maken en de verbeterde of aangevulde programmatuur opnieuw vrij ter beschikking te stellen.

Interessant hieraan is, dat in de cultuur van de informatica twee bedrijfsmodellen zijn ontstaan, een open en een gesloten variant. Met name de open variant neemt in onze economie vooralsnog een uitzonderingspositie in, waarvan het niet zeker is dat hij in zijn consequenties wordt voorzien. Organisaties die kiezen voor het open bedrijfsmodel zijn in verschillende sectoren zeer succesvol, met name bij software-ontwikkeling, in de wetenschap en binnen grote internationale bedrijven als exponent van wat kennismanagement wordt genoemd. Het gaat wél ergens over. De helft van alle internet servers in de wereld draait op programmatuur die is gemaakt en wordt onderhouden in netwerken die kozen voor het open bedrijfsmodel.

De opkomst van de pc, en de standaardisering die daarmee gepaard ging, maakte dat softwarefabrikanten zich meer gingen bekommeren om hun auteursrecht. Daarvoor waren bedrijfssystemen slechts te gebruiken op de mainframes waar zij bij geleverd werden. Toepassingen werden door bedrijven en instellingen veelal zelf geschreven. De mogelijkheid van grootschalig gebruik bracht softwarefabrikanten ertoe beveiligingen toe te passen en de broncode geheim te houden. Het internet heeft de verspreiding van software alleen maar gemakkelijker gemaakt. Fabrikanten van *proprietary* software bestrijden deze mogelijkheid. Het internet biedt echter ook de mogelijkheid om met vele personen over de hele wereld te werken aan hetzelfde programma.<sup>22</sup>

#### 1.7.4. *Maatschappelijk perspectief*

Het internet is vooral een massaal verschijnsel. Het is ontstaan uit een overzichtelijk, beperkt gebruik voor specifieke doeleinden door specialisten. Het is heel moeilijk om te voorzien wat de maatschappelijke gevolgen zullen zijn van de intredende massaliteit van het internetgebruik. De geschetste ontwikkeling en organisatie heeft echter enkele belangrijke kenmerken die mogelijk houvast bieden:

##### (a) *Lokale controle*

De zeggenschap over het al dan niet aanleggen van een verbinding tussen twee lokale netwerken is eveneens lokaal. Om deel te nemen aan het internet is niet meer nodig dan het onderschrijven van enkele afspraken zoals: het gebruik van het TCP/IP-communicatieprotocol, het gebruik van bepaalde adresseringsconventies, de bereidheid de huurlijnen toe te laten voor intermediair dataverkeer, dat wil zeg-

---

<sup>22</sup> L. Lessig (2004), *Free Culture: How Big Media Uses Technology and the Law to Lock Down Culture and Control Creativity*, The Penguin Press. p. 264 e.v.

gen voor berichtenverkeer over het internet, ook tussen afzenders en ontvangers van buiten de lokale netten. Deze functies worden gedelegeerd aan geautomatiseerde diensten als domain name servers, bridges, routers en gateways.<sup>23</sup> Het toedelen van namen en adressen wordt vanuit een wereldomspannende organisatie<sup>24</sup> gedelegeerd aan organisaties die per land/regio opereren.<sup>25</sup> De kosten voor het berichtenverkeer worden lokaal gedragen, enerzijds door de verbinding tussen individuele gebruiker en lokale netwerk-server en anderzijds door de lokale netwerken die met elkaar zijn verbonden (die de kosten die ze maken als internet-providers wel weer doorberekenen aan hun klanten). Hoe dan ook, de tarifiering voor berichtenuitwisseling over het internet is volkomen anders dan vooralsnog gebruikelijk is in het telefoonverkeer en onafhankelijk van de te overbruggen fysieke afstand.

### *(b) Robuuste verbindingen*

Doordat het internet een netwerk van netwerken is, zijn veel verbindingen redundant, althans gezien vanuit de individuele deelnemer aan het berichtenverkeer. In Schema 7 komt dat tot uiting doordat lokaal netwerk 1 direct met lokaal netwerk 3 kan communiceren, maar ook via lokaal netwerk 2. Deze ‘overtollige’ mogelijkheden komen niet alleen van pas wanneer een grondwerker in het kader van rioolwerkzaamheden één van de datatransmissielijnen stuk trekt. Ook als één van deze verbindingen overbelast raakt, kan een alternatief traject soelaas bieden. Het internet is zo een *robuust* netwerk geworden, dit in tegenstelling met de *broosheid* die veel doelgericht gemodelleerde communicatienetwerken (en computertoepassingen) kenmerkt: deze vallen geheel uit zodra ook maar iets tegenzit. Daarbij moet wel worden opgemerkt dat internet (net als ieder ander computernetwerk) gevoelig is voor aanvallen van *hackers*. Door doelbewuste overbelasting van individuele websites kan het internetverkeer in grote regio's danig ontregeld raken. Overigens kan, ondanks het inherent-robuste karakter van internet, de infrastructuur sowieso erg kwetsbaar worden door de aanwezigheid van slechts één of enkele hoofdverbindingen en toegangspunten (*backbones*, *internet exchanges*) in een land. Het systeem dat voor de vertaling van domeinnamen naar IP-adressen zorgt, kan eveneens onderwerp worden van aanvallen door hackers.

### *(c) De aanwezigheid van risico's*

Lokale controle en robuustheid wekken vertrouwen, maar ook onzekerheid. Niemand garandeert dat het lokale netwerk in Evanston er volgend jaar nog zo zal bijliggen. En hoewel er letterlijk miljoenen communicatiepaden feitelijk beschikbaar zijn tussen Leiden en Evanston, kan geen enkele organisatie over de beschikbaarheid van zelfs maar één daarvan een garantie geven.

23 Zie de verklarende begrippenlijst voorin dit boek voor een nadere omschrijving van dit jargon.

24 ICANN, The Internet Corporation for Assigned Names and Numbers (<http://www.icann.org>).

25 In Nederland thans de Stichting Internet Domeinregistratie Nederland (<http://www.domain-registry.nl>).

*(d) De noodzaak van vertrouwen*

De bereikbaarheid van partijen via internet gaat niet hand in hand met onafhankelijke kennis over de betrouwbaarheid van die partijen. Naar analogie hiervan kan de gebruiker weliswaar met groot gemak een bestelling plaatsen in de Verenigde Staten, maar is hij er ook zeker van dat zijn bestelling geleverd wordt? Misbruik van persoonsgegevens ligt op de loer. TTP's spelen een rol in het vestigen van vertrouwen: is de partij 'aan de andere kant van de lijn' ook werkelijk degene die hij claimt te zijn? Veilige communicatie, mogelijk gemaakt door asymmetrische encryptie, laat de consument op een (relatief) veilige manier betalen.

*(e) Het verlies van specifieke betekenis van de fysieke informatiedragers ('convergentie')*

Door de beschikbaarheid en de spreiding van de communicatie-infrastructuur, in samenhang met het vermogen van computers om bepaalde functies te vervullen (bijvoorbeeld telefoon of televisie), worden de verschillen tussen traditionele informatiedragers van minder belang. Telefonie en omroep zijn mogelijk over het internet. Daarmee treedt een belangrijke vervang op tussen traditionele partijen als omroepen en telecomaانبieders, en nieuwe partijen als internetproviders. De regulering van deze partijen wordt daardoor gecompliceerd.<sup>26</sup> Zo kan de OPTA, de toezichthouder voor de telecommarkt, tot de invoering van de nieuwe Telecommunicatiewet vrijwel niets zeggen over de internetactiviteiten van telecombedrijven, hoe nauw die ook gerelateerd zijn aan de traditionele diensten van deze bedrijven. Hierin komt met het nieuwe Europese kader voor elektronische communicatie verandering. Zo valt bijvoorbeeld Voice over IP (VoIP), dat gebruik maakt van het Internet Protocol om spraaktelefonie mogelijk te maken, onder de reikwijdte van de nieuwe wet.

*(f) Het terugtreden van de individuele, ongedeelde verantwoordelijkheid (of: de fragmentering daarvan)*

Zeggenschap en wetenschap over waar wat in fysieke zin gebeurt, raken gefragmenteerd (bijvoorbeeld bij elektronische handel) en traditionele verantwoordelijkheden worden daarmee diffuus: wie is verantwoordelijk voor de beschikbaarheid van bepaalde informatie? Wie sluit de overeenkomst? Het is in de context van informatie- en communicatietechnologie gemakkelijk verantwoordelijkheden te laten 'verdampen', doordat niet direct duidelijk is wie verantwoordelijkheid draagt voor de werking van systemen.

*(g) Het verlies van de fysieke plaats als aanknopingspunt voor regulering*

Communicatie via het internet kan vanuit verschillende gezichtspunten juridische betekenis hebben. Overeenkomsten kunnen worden gesloten, betalingen verricht, openbaarmaking verwezenlijkt, veelevoudigingen gemaakt en strafbare uitingen gedaan – alle op of via het internet.

<sup>26</sup> Als voorbeeld van convergentie moge dienen de wens van KPN van een groter belang in Digitenne, de distributeur van digitale televisiesignalen door de ether. Deze zou door KPN gezien worden als aanvulling op het draadloze breedbandaanbod via UMTS.

*(h) Lex ex machina*

Lawrence Lessig heeft in een aantal boeken de aandacht gevestigd op de invloed van ‘cyberspace’ op de rol van het recht.<sup>27</sup> Een van de lessen die uit deze boeken kunnen worden getrokken, is dat regulering van informatie- en communicatie-technologie niet eenzijdig vanuit het recht kan plaatsvinden. Sterker, de wisselwerking wordt veeleer geregisseerd door technologische ontwikkelingen, die zich in de maatschappij zichtbaar maken door veranderende mogelijkheden en beperkingen. Als voorbeeld daarvan kan dienen het toenemend belang van digital rights management technieken, die gebruikmaking van ‘fair use’-uitzonderingen uit het auteursrecht onmogelijk maken.

**1.8. Uitdagingen voor het recht rond 2005<sup>28</sup>**

Het decennium rond 2005 wordt gekenmerkt door de democratisering en commercialisering van het internet. Het wordt daarmee aannemelijk dat de uitdagingen van de ICT aan het recht pas in deze tijd werkelijk op grote schaal zichtbaar worden. In deze laatste paragraaf wenden we de blik vooruit om enkele van die uitdagingen te schetsen aan de hand van een paar maatschappelijke verschijnselen die de democratisering en de commercialisering van het internet met zich mee hebben gebracht.

*1.8.1. Computerregulering*

Roel Wieringa – een informaticus – besteedde in zijn inaugurele rede aandacht aan twee complementaire verschijnselen:

- De toenemende socialisering van informatiesystemen, waarmee hij bedoelt dat informatiesystemen steeds meer handelingen uitvoeren waarnaar wij, mensen, ons te voegen hebben, en
- het verschijnsel dat steeds meer objecten en handelingen, die voorheen alleen in de fysieke wereld bestonden en plaatsvonden, nu uitsluitend binnen een informatiesysteem bestaan en plaatsvinden.

Wieringa is vooral filosofisch gefascineerd door het verschijnsel dat modellen in informatiesystemen zo zeer een eigen leven kunnen gaan leiden, dat wij worden genormeerd door die informatiesystemen. Daarbij valt bijvoorbeeld te denken aan het betalingsverkeer zoals dat verloopt via de chipknip. Geld bestaat in dat circuit niet meer in fysieke vorm, maar het *is* er wel. Het opladen van de chipknip wordt genormeerd door uw banksaldo, en ook dat bestaat niet meer anders dan in een informatiesysteem, in de databank van uw bank. Uw chipknipgedrag wordt daarvoor gereguleerd: het oplaadpunt weigert uw opdracht uit te voeren, wanneer het

---

27 Zie onder meer Lessig (1999, 2004).

28 Grotendeels ontleend aan Schmidt (2004).

van ‘oordeel’ is dat uw banksaldo dat niet toelaat. Ook voor juristen is dit verschijnsel fascinerend. Wat is de rol van recht en jurist bij het tot stand brengen van regulering met behulp van dergelijke virtuele objecten en normen? Standaards spelen daarbij een belangrijke rol. Ze ondersteunen niet alleen samenwerking – ze kunnen ook leiden tot het gevoel ernstig in vrijheid te worden beknot. Dat gevoel overvalt ons allemaal wel eens wanneer we een formulier invullen dat geen ruimte biedt voor het meest wezenlijke dat we willen noteren. Het resultaat is formulierfrustratie. Anders gezegd: formulierfrustratie is er een aanwijzing voor dat we door een informatiesysteem worden gereguleerd. In 1999 beschreef Lawrence Lessig dit verschijnsel in zijn global bestseller: *Code and other laws in Cyberspace* onder de noemer *regulation by architecture*.

Het algemene streven naar efficiency leidt er toe dat de normering van gedrag in de fysieke wereld waar mogelijk, en derhalve in toenemende mate, wordt overgelaten aan informatiesystemen. We kennen allemaal de ontwikkeling van baliemedewerker via *call centre* naar ingeblikte stemmen die u opdragen om ‘toets 1’ in te drukken. En die uiteindelijk laten weten dat ‘al onze medewerkers in gesprek zijn.’ Om ons daarna aan te bevelen onze zaken af te handelen via een formuliertje op internet. Wanneer we daarbij door publiekrechtelijke informatiesystemen worden gereguleerd zou dat volgens democratisch gelegitimeerde regels moeten gebeuren. En wanneer we door civielrechtelijke informatiesystemen worden gereguleerd dient dat te gebeuren op een wijze die werd gelegitimeerd door kenbare wilsovereenstemming.

Dat vraagt nogal wat.

Het overlaten van handelingen aan automaten en lopende banden is natuurlijk sinds twee eeuwen gemeengoed in de industriële wereld, en is gedurende die tijd tot ontwikkeling gekomen als het specifieke domein van *science & industry*. De informatica behoort tot dat domein. Het fixeren van technische kennis in automaten levert dan ook weinig problemen in de communicatie tussen ingenieurs en informatici. De ingenieurs maken de modellen van de technische processen die worden geautomatiseerd. De informatici vertalen die modellen naar informatiesystemen. Dat gaat vaak goed. Ingenieurs en informatici spreken dialecten van dezelfde taal. Dat is anders, wanneer we het gedrag van mensen gaan normeren met behulp van informatiesystemen. Want dan is het immers wenselijk dat die systemen zodanig in elkaar zijn gezet dat de regels die ze verwezenlijken gelegitimeerd zijn. Dat is niet gemakkelijk vast te stellen voor een ingenieur. We moeten derhalve aannemen dat voor het normeren van menselijk gedrag via informatiesystemen juridische kennis nodig is. En dat houdt weer in dat er juristen beschikbaar moeten zijn om de *legitieme* normatieve modellen te maken die door informatici kunnen worden vertaald naar informatiesystemen. Maar juristen en informatici spreken geenszins dezelfde taal. Het is aannemelijk dat wanneer meeregulerende informatiesystemen operationeel zijn, zij onderdeel uit gaan maken van ons rechtssysteem. En dat, als er een legitimiteitvereiste geldt voor de regels van ons rechtssysteem dat vereiste ook moet worden gesteld aan de informatiesyste-

men die daar een onderdeel van zijn. Als dat niet gebeurt – en dat gebeurt niet of nauwelijks zolang we het maken van modellen aan ingenieurs overlaten – dan kunnen die informatiesystemen ons rechtssysteem van binnenuit aantasten. We zien dat dan ook overal gebeuren. De grootste drama's doen zich voorlopig voor in bedrijven, waar normen worden gemodelleerd in administraties. Denk aan de schandalen bij Enron, Albert Heijn en Shell – in al deze gevallen gaat het om de legitimiteit van administraties waarbij de regulering in belangrijke mate is overgedragen aan de informatica. En dat is niet alles. In de jaren negentig zijn de eerste systemen ontwikkeld die werden ingezet om erop toe te zien dat ambtenaren van de sociale diensten de uitvoeringsregels van de sociale zekerheid correct zouden toepassen. Bij de evaluatie van die systemen kwam aan het licht dat de officials die er gebruik van moesten maken zich nogal eens niet gebonden voelden aan de uitkomsten. Wat deden ze? Ze veranderden de input zodanig, dat er de door hen gewenste uitkomst uitkwam. Dit gebeurt niet alleen in de sociale zekerheid, het is vermoedelijk ook aan de hand bij bovengenoemde schandalen. Computerregulering vormt, als maatschappelijk verschijnsel, in de nabije toekomst een belangrijke uitdaging voor ons recht.

### 1.8.2. *Het verantwoordelijkheidslek*

Het tweede relevante maatschappelijke verschijnsel is een alledaags syndroom: het de-computer-heeft-het-gedaan-syndroom. Wanneer we onderling communicerende computertoepassingen 'agenten' noemen, en we erkennen dat ze zelfstandig kunnen handelen, dan kunnen we misschien ook – naar analogie met rechtssubjecten – gaan denken aan de normatieve aspecten van de keuzes die ze maken en van de handelingen die ze verrichten. Misschien is het zelfs mogelijk te spreken over de gerechtvaardigde verwachtingen die ze ten opzichte van elkaar zouden kunnen hebben. En over de zorgvuldigheid die hen op grond van ongeschreven spelregels in hun onderlinge maatschappelijke verkeer betaamt. Deze gedachte roept enerzijds het visioen op van een openbare orde in Cyberspace. Maar het roept ook de vraag op of dergelijke systemen een belangrijk nieuw maatschappelijk verschijnsel inhouden: de vervaging van persoonlijke verantwoordelijkheid door de informatica, ofwel: het verschijnsel dat niet de dader, maar 'de computer' het heeft gedaan.

We geven een voorbeeld dat we vermoedelijk allemaal kennen. Soms krijgen we, wanneer we onze computer aanzetten van Microsoft het bericht dat er ergens een verbetering zal worden aangebracht. Geen idee wat er dan in feite gebeurt. Het gevolg is, dat we min of meer gedwongen worden om delen van wat we ervaren als onze handelingsbekwaamheid over te dragen aan de systemen die ons door de *system level* bureaucraten van een multinational worden voorgeschoteld.

Een ander voorbeeld. In 2003 werden de activiteiten van de Nederlands organisatie voor Wetenschappelijk Onderzoek (NWO) een week opgeschort doordat een computervirus het systeem had platgelegd. Een hele week! Ongeveer 200 medewerkers met de armen over elkaar! Niemand is bij ons weten op de gedachte gekomen om de verantwoordelijkheid van de leverancier of de beheerder van

dit systeem in te roepen, en niemand heeft bij on weten getracht NWO aansprakelijk te stellen. Een hele subsidieronde is bedorven en de computer heeft het gedaan. Niemand is verantwoordelijk – of het zou de maker van het virus moeten zijn. Maar we weten al een tijd dat die dreiging bestaat, en we weten ook al een tijd hoe we bedrijfskritische systemen daarvan moeten vrijwaren.

NWO is een zelfstandig bestuursorgaan. NWO zet computers in met programma's waarvan NWO de werking niet kent, zelfs niet kan kennen. NWO laat een deel van zijn bestuurlijk handelen uitvoeren door een systeem dat niemand kan controleren. En NWO is absoluut geen uitzondering. Het is de vraag hoe we in de toekomst met dergelijke verantwoordelijkheidslekken moeten omgaan.

### 1.8.3. *Informatiemonopolies*

Wie bij Albert Heijn wil profiteren van de aanbiedingen, gebruikt een bonuskaart die bij de kassa wordt gelezen. De kruidenier kan met behulp van die bonuskaart in zijn databases opslaan wat u waar en wanneer heeft gekocht. Albert Heijn doet dat ook. Het voordeel is dat daardoor in een afpersingskwesitie de vergiftigde toetjes zo nodig bij u thuis kunnen worden teruggehaald. Een ander voorbeeld. Google houdt een register bij van alles waarnaar u ooit via Google heeft gevraagd. Wereldwijd gaat het om de helft van alle vragen die aan internet worden gesteld. We geven een citaat uit de *privacy policy* van Google:

“Google verzamelt geen unieke informatie over u (zoals uw naam, e-mailadres. etc.) behalve als u die informatie uitdrukkelijk en bewust verstrekt. Google noteert en bewaart informatie zoals tijdstip ... en IP-adres van elke zoekvraag”

IP betekent internet protocol. Het IP-adres identificeert niet *u*, maar de computer die u gebruikt. Google bewaart alle vragen en creëert daarmee, net als Albert Heijn met zijn bonuskaart, een informatiemonopolie. Google en Albert kunnen deze inzetten om aan u betere diensten te verlenen. Aan *u*, niet aan de computer die u gebruikt.

En dit gaat maar zo door. De overheid heeft zo zijn eigen informatiemonopolies. De maatschappij die uw mobiele telefoon van berichten voorziet registreert van minuut tot minuut waar uw mobieltje zich bevindt. En sinds een aantal jaren heeft de industrie zogenoemde RFID-chips beschikbaar. RFID staat voor radio frequentie identificatie. Het gaat om piepkleine computertjes die hun energie halen uit een magnetisch veld en die vervolgens een radiosignaal afgeven met een uniek identificatienummer. Er zijn plannen om bankbiljetten van deze chips te voorzien.

Kortom, er komen maatschappelijke spelers die over gigantische informatieverzamelingen over het gedrag van burgers en bedrijven beschikken. Die informatie kan voor alle mogelijke doeleinden worden gebruikt, niet in de laatste plaats voor het bestrijden van internationaal terrorisme. Misschien is het wel wenselijk dat die informatie daarvoor wordt gebruikt. Maar het is ook mogelijk dat die kennis wordt gebruikt voor minder vanzelfsprekende doeleinden.

De vraag hoe we informatieposities reguleren is van belang voor de kwaliteit van onze rechtsstaat. In die zin vormen de *mega* informatieposities van de kennisbovenbazen zowel kansen voor bescherming als voor bedreiging. Het op de individu gerichte grondrecht op privacy komt in dit spanningsveld niet erg goed uit de verf. Het lijkt dan ook voor de hand te liggen om de mogelijkheden van kennis-is-macht *evenwichten* te onderzoeken. Tegenover de informatiepositie van de overheid ten behoeve van zijn wettelijke taakuitoefening wordt dan een registratieverplichting van het gebruik ervan gesteld ten behoeve van parlement, wetenschap, pers en burger. Iets soortgelijks kan worden bedacht voor de controle op de informatieposities van bedrijven ten behoeve van de informatieposities van aandeelhouders, werknemers, klanten, banken en fiscus. Dit alles wekt evenwel visioenen van ongebreidelde registratie- en administratiedrift. In dat visioen zijn we hollend op weg naar een Big Brother maatschappij. En om de ene Big Brother in bedwang te houden moeten we er dan nog een maken, en nog een ..., totdat er een hele gemeenschap van kennisbovenbazen is die elkaar administratief in bedwang houden. Er is hier maar één vraag van belang: wat gebeurt er met ons rechtssysteem als we *nalaten* die evenwichten tot stand te brengen?

#### 1.8.4. *Unsollicited commercial e-mail (UCE of SPAM)*

Een voorproefje van wat er dan gebeuren kan levert de vloed van *unsolicited commercial e-mail* of SPAM), die sinds 2002 een zo grote vlucht geeft genomen, dat daarmee een aanmerkelijk beslag wordt gelegd, niet alleen op de beschikbare telecommunicatiecapaciteit, maar ook op de beschikbare tijd van degenen die e-mail ontvangen. SPAM is reclame via e-mail. E-mail adressen kunnen worden verzameld door *crawlers* (agenten die de publieke informatie op het internet 'afgrazen'), en aan de hand van uw surfgedrag kan worden afgeleid of u tot de betreffende doelgroep behoort. Uw surfgedrag wordt, als gezegd, geregistreerd door Google, maar het kan ook worden vastgelegd via een *worm* (een computerprogrammaatje dat zich, meestal onrechtmatig, op uw computer heeft genesteld en dat aspecten van uw computergebruik doorzendt via internet). De teugelloze commercialisering van internet leidt zo in toenemende mate naar vormen van virtuele anarchie en corruptie, die ook in de toekomst het recht zullen uitdagen.

#### 1.8.5. *Wie is nu eigenlijk een internet service provider?*

De democratisering van het internet bracht ook nieuwe actoren, waarvan de internet service providers het meest in het oog springen. Internet service providers kunnen verschillende diensten verlenen, waaronder toegangsverlening (het bieden van toegang tot het internet; deze toegang speelt zich af op het niveau van de eerste drie lagen van het ISO/OSI-model), hosting (server-ruimte en bandbreedte voor het aanbieden van webdiensten; deze dienst speelt zich af tot en met de vijfde laag), content (bieden van inhoudelijke diensten, zoals nieuwsberichten; deze vallen buiten het ISO/OSI-model, maar hun presentatie en gebruik spelen zich af in de zesde en zevende laag) en andere services (zoals e-mail; hiervoor geldt dezelfde opmerking m.b.t. het ISO/OSI-model). Internet service providers zijn tegen-

woordig voornamelijk commerciële bedrijven, een enkele instelling daargelaten, zoals Surfnet, dat de internetverbindingen voor universiteiten en hogescholen verzorgt. Met de opkomst van breedband- en van draadloos internet via *WiFi* lijken particulieren (of consumenten) de functie van internet service provider te krijgen, terwijl de diensten van de 'echte' internet service provider dan als 'mere conduit' moeten worden gekwalificeerd. Het is de vraag of de huidige regulering, die met name is gericht op de professionele Internet service provider, daarop voldoende is toegesneden.

#### 1.8.6 *Ten slotte*

Juristen die hun oren te luisteren leggen bij de technologiegemeenschap kunnen misschien anticiperen op ontwikkelingen. Het is zo tenminste mogelijk om *mogelijke* obstakels te *onderzoeken*. Wie de huidige inspanningen bekijkt met betrekking tot de ontwikkeling van protocollen en standaards (er is zelfs een gemeenschap die zich bezighoudt met de ontwikkeling van talen die verdergaande uitwisseling van juridische documenten mogelijk maakt (*legal XML*), zal zich niet aan de indruk kunnen onttrekken dat op alle gebieden waar dergelijke vaktalen ontstaan (dus ook in de rechtspraktijk) de IT binnen afzienbare tijd zulke ingrijpende ondersteuning zal gaan bieden dat de traditionele organisatievormen destabiliseren. Ook wie met open ogen de technologie tegemoet treedt, kan door haar voortbrengselen onthutst en verrast worden. Wie afwacht tot nieuwe technologie en techniek hun beslag krijgen in een maatschappelijke context, en wie nalaat te onderzoeken wat daarvan de normatieve consequenties zijn, neemt de grootste risico's.

#### *Geraadpleegde en aanbevolen literatuur*

Franken, H., *Kanttekeningen bij het automatiseren van beschikkingen*, VAR-reeks, Alphen aan den Rijn: Samsom H.D. Tjeenk Willink 1993.

Herik, H.J. van den, *Kunnen computers rechtspreken?*, Arnhem: Gouda Quint 1991.

Hofstadter, D., *Gödel, Escher, Bach*, Amsterdam: Contact 1985.

Koers, A.W., *Rechten en plichten van het individu op de elektronische snelweg*, *Aanzet tot een handvest*, Den Haag: Rathenau Instituut 1995.

Lessig (1999), *Code and Other Laws of Cyberspace*, Basic Books.

Lessig (2004), *Free Culture: How Big Media Uses Technology and the Law to Lock Down Culture and Control Creativity*, The Penguin Press.

Rossum, G. van, 'Client/Server: een nieuw concept gedefinieerd en geëvalueerd', in: *Informatie* 1993/9.

Schmidt, A.H.J., 'Bedreigen computers ons rechtssysteem? Oratie Leiden 2004.

Schreiber, G., H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde en B. Wielinga (1999), *Knowledge Engineering and Management. The CommonKADS methodology*, Cambridge, MA: The MIT Press.

Turing, A.M., 'Computing Machinery and Intelligence', *Mind*, 1950/59, p. 430-460.

Wieringa, R.J., *Requirements Engineering, Frameworks for understanding*, New York: Wiley and Sons 1996.

Wieringa, R.J., *De onzichtbare wereld van de informaticus: Conceptuele modellen en virtuele objecten*, Universiteit Twente, 1998 (inaugurele rede).