

Het jaar-2000-probleem

Arie van Deursen

Paul Klint*

Samenvatting

Op veel computersystemen draait programmatuur die datumgegevens uit het jaar 2000 incorrect afhandelt. Worden deze fouten niet verholpen, zullen zich de meest uiteenlopende computerstoringen voordoen. De reparatie ervan wordt bemoeilijkt door de omvang van het probleem en het gebrek aan automatiseringspersoneel. In deze bijdrage bespreken we de informatica-technische aspecten van het jaar-2000-probleem, zoals de oorzaken, gevolgen, oplossingsmogelijkheden, en projectinrichting. Doel van de bijdrage is het ondersteunen van de discussie over de juridische vragen rond het jaar-2000-probleem.

Dit artikel is verschenen in *Computerrecht*, Dossier Millennium, December 1997, pp. 266–271.

1 Inleiding

In januari 1995 vervoerde een chemiebedrijf chemicaliën naar een juist-op-tijd productielijn. De chemicaliën waren vijf jaar houdbaar, en werden vervoerd in vaten waarop als uiterste datum “1/4/00” was aangegeven. Deze datum werd bij aankomst ingevoerd in het voorraadbeheersysteem, maar het systeem weigerde de vaten te accepteren en meldde dat ze teruggestuurd moesten worden naar de fabrikant. Dit gebeurde, en de productielijn moest worden stilgelegd vanwege een gebrek aan grondstoffen. De oorzaak bleek te liggen in het feit dat het voorraadbeheersysteem “00” interpreteerde als “1900”, en derhalve concludeerde dat de stof in de vaten 95 jaar oud en dus onbruikbaar was. In de bijbehorende rechtzaak is een schikking tussen de partijen getroffen [13].

Op 31 december 1996 veroorzaakte een storing in alle 660 besturingscomputers van een aluminium smeltoven in Nieuw-Zeeland een schade van 1.5 miljoen gulden. De reparatie van de software van de fabriek duurde meer dan 12 uur, tijd waarin de smeltoven oververhit raakte en onherstelbare schade opliep. Precies twee uur later (het tijdsverschil tussen de twee landen) deed hetzelfde probleem zich voor in een vergelijkbare fabriek in Australië. De oorzaak van dit alles was dat de programmatuur 1996 niet als schrikkeljaar herkende en daardoor vastliep op dag nummer 366 [19]. Het jaar 2000 is ook een schrikkeljaar, maar het jaar 1900 was dat niet.

*Dr. A. van Deursen en prof. dr. P. Klint zijn verbonden aan de afdeling Software Engineering van het Centrum voor Wiskunde en Informatica te Amsterdam. Prof. dr. P. Klint is tevens hoogleraar informatica aan de Universiteit van Amsterdam.

Hoeveel computersystemen zullen uitvallen of incorrecte resultaten opleveren rond 29 februari 2000?

Sinds eind 1996 worden veel creditcards uitgegeven met vervaldatum 11/99, wat een kortere levensduur van de kaart oplevert dan de gebruikelijke drie jaar. Vervaldatum 12/99 wordt door sommige kaartlezers als “speciale” datum beschouwd. Data met jaartal “00” leveren nog meer problemen op: begin 1997 waren er wereldwijd nog 1.3 miljoen kaartlezers die niet met het jaartal “00” konden werken. Het Visa creditcardconsortium heeft een boetesysteem opgesteld om deelnemende banken te dwingen versneld maatregelen hiertegen te nemen [21].

Het duurt nog meer dan twee jaar voor het 1 januari 2000 is, maar de nadering van het jaar 2000 veroorzaakt problemen voor veel computersystemen, zoals blijkt uit bovenstaande voorbeelden. De reparatiekosten alsmede de directe schade veroorzaakt door dit probleem kunnen hoog oplopen. De meest geciteerde schatting van de wereldwijde reparatiekosten bedraagt 600 miljard dollar [11]. Conflicten over wie deze kosten zal moeten betalen zullen zeker ontstaan, wat weer zal leiden tot talloze rechtzaken. De juridische kosten zouden hierdoor wereldwijd dan ook hoog op kunnen lopen [11].

In deze bijdrage willen we een overzicht geven van de meer technische vragen rond het jaar-2000-probleem, zoals hoe het probleem is ontstaan, welke fouten voor zullen komen, wat voor correctiemogelijkheden er zijn, en hoe een typisch jaar-2000-conversieproject eruit ziet. Doel van de bijdrage is de informatica-technische aspecten van het jaar-2000-probleem op zodanige wijze te bespreken dat een zinvolle discussie over de juridische aspecten ervan bevorderd wordt.

2 Programmatuur met problemen

Op veel computersystemen draait programmatuur die datumgegevens van 1 januari 2000 en later (en soms ook daarvoor) incorrect afhandelt. De belangrijkste oorzaak hiervan is dat op veel systemen slechts twee cijfers worden gebruikt om jaartallen te representeren. Fouten die hierdoor in datumberekeningen kunnen optreden zijn:

- Rekenkundige bewerkingen op jaartallen leveren verkeerde resultaten op (zie Figuur 1). Te denken valt aan het uitrekenen van het interval tussen twee data of het optellen van een aantal jaren bij een datum. Bij het aftrekken van data na 2000 zullen negatieve resultaten ontstaan of zal het min-teken weg worden gelaten.
- Jaartallen worden foutief geordend. 2001 is groter dan 1997 maar in de tweecijferige representatie is 1 *kleiner* dan 97. Het sorteren op jaartal of het vergelijken van data gaat hierdoor bijvoorbeeld verkeerd, evenals het indexerend van bestanden op jaar.
- De aanname dat we in de twintigste eeuw leven is hard gecodeerd in programmatuur. In schermen of statements wordt expliciet de constante 19 vóór een tweecijferige jaar-aanduiding geplakt (op veel met de hand in te vullen formulieren staat bijvoorbeeld ook al vast “19” ingevuld bij het jaartal). Het effect is dat 2001 (gerepresenteerd door 01) geïnterpreteerd wordt als 1901.

4 cijfers	2 cijfers
1997	97
1998	98
1999	99
2000	00
2001	01
$1997 + 4 = 2001$	$97 + 4 = 01$
$2001 - 1997 = 4$	$01 - 97 = 96$
$2001 > 1997$	$01 < 97$

Figuur 1: Berekeningen met jaartallen in vier en twee cijfers.

- De dag van de week (zondag, maandag, ...) wordt fout berekend. 1 januari was een maandag in 1900, en is een zaterdag in 2000.
- De jaaraanduidingen 99 en 00 (die immers “nooit” konden voorkomen) hebben een speciale status: zij worden bijvoorbeeld gebruikt als “jaartal onbekend”, als interne foutaanduiding, of als “eeuwig houdbaar”. Dit heeft als gevolg dat al voor het jaar 2000 problemen zullen optreden.
- Het jaar 2000 wordt niet als schrikkeljaar herkend. Het jaar 1900 was geen schrikkeljaar (1900 is deelbaar door 4 en door 100, maar niet door 400), maar 2000 is dat wel (2000 is deelbaar door 400). Veel programmatuur gaat ervan uit dat “00” 1900 is, of heeft het verkeerde algoritme geïmplementeerd om te bepalen of een jaar een schrikkeljaar is.

Deze problemen doen zich vooral voor in de software, maar kunnen ook in de hardware optreden. Een systeem dat de overgang naar het jaar 2000 probleemloos aankan wordt jaar-2000-*compliant* genoemd. Een systeem dat niet compliant is kan in principe gerepareerd worden: een dergelijke reparatie heet een jaar-2000-*conversie*.

3 Gevolgen

Wat de gevolgen van foutieve datumberekeningen zullen zijn is niet precies te voorspellen. Wel staat met zekerheid vast dat alleen al in Nederland miljoenen systemen en gegevensbestanden gebruik maken van tweecijferige jaartallen. De automatisering is diep doorgedrongen in alle sectoren van onze samenleving en het jaar-2000-probleem kan dan ook tot gevolg hebben dat vele producten of diensten zullen falen: van videorecorder tot tijdslot, van auto tot vliegtuig, van bijstandsuitkering tot pensioenverzekering, van bank tot fabriek. In deze paragraaf schetsen we de mogelijke gevolgen van het jaar-2000-probleem voor administratieve systemen, procesbesturingssystemen, en voor zogenaamde *embedded* systemen.

Administratieve systemen ondersteunen in zeer ruime zin de administratieve processen bij overheid en bedrijfsleven. Jaartallen zullen daarbij een rol spelen als leeftijd van personen of als indicatie van tijdstippen voor betaling, inning of levering. Bij de overheidsautomatisering valt te denken aan het betalen van uitkeringen, of het innen van belastingen. In de financiële sector gaat het om uitbetaling van pensioenen en het innen van aflossingen en leningen.

Administratieve systemen bevatten ook tijdsafhankelijke informatie die bedoeld is om de algehele betrouwbaarheid van de systemen te vergroten. Veel systemen zijn gebaseerd op transactieverwerking: pas als een volledige transactie—bestaande uit een met elkaar samenhangend aantal handelingen—is verricht worden de eindresultaten in het informatiesysteem opgeslagen. Als er halverwege iets fout gaat worden alle neveneffecten van de transactie ongedaan gemaakt (“roll back”) met behulp van (tijdsafhankelijke) gegevens over deeltransacties die in een systeembestand worden bijgehouden. Een andere karakteristiek van administratieve systemen zijn de maatregelen die getroffen zijn om bij het uitvallen van een computer alle databestanden later weer te kunnen herstellen. Hiertoe worden “backups” gemaakt: kopieën van de voornaamste databestanden met tijdsaanduiding. In het geval van uitval bepalen de tijdsaanduidingen welke databestanden hersteld moeten worden uit het backup bestand. Alle hier genoemde systeeminformatie is gevoelig voor het jaar-2000-probleem en kan er oorzaak van zijn dat nieuwe fouten geïntroduceerd worden zoals, bijvoorbeeld, het incorrect ongedaan maken van een transactie of het herstellen van bestanden uit foutief gedateerde backups.

Tenslotte noemen we nog twee kenmerkende voorbeelden. Ten eerste, speelt bij het automatisch beheren van voorraden (medicijnen, bloed, levensmiddelen) de houdbaarheidsdatum een grote rol, zoals we gezien hebben in het voorbeeld van de beperkthoudbare chemicaliën uit de inleiding. Mogelijke gevolgen van het jaar-2000-probleem zijn het bewaren van allang bedorven voorraden of het vernietigen van verse voorraden die na 2000 geproduceerd zijn.

Ten tweede worden *spreadsheet* programma's veelvuldig gebruikt voor allerlei berekeningen. Een weinig onderkend probleem vormen de berekeningsvoorschriften (macro's) die eindgebruikers zelf hebben toegevoegd: ook deze kunnen jaar-2000-problemen bevatten.

Procesbesturingssystemen besturen een fysiek apparaat of proces zoals een lift, beveiligingssysteem, raffinaderij of vliegveld. De schade die in deze gevallen kan optreden is zeer gevarieerd: liften die vast komen te zitten (omdat het onderhoud te lang geleden plaatsgevonden zou hebben), het wegvallen van de beveiliging van een bank, beschadiging van machines of fabrieksonderdelen danwel voor het milieu schadelijke lozingen, of het falen van de verkeersleiding van een vliegveld.

Embedded systemen zijn computersystemen die onderdeel vormen van een groter geheel zoals, bijvoorbeeld, de hardwareklok in een personal computer, de besturing van een videorecorder, de boordcomputer van een auto, of een doseringsklep in een pijpleiding.

Schade die in deze gevallen kan optreden is, bijvoorbeeld, het niet functioneren van een backupstelsel met als gevolg het overschrijven van bestanden door een “nieuwere” (maar in werkelijkheid veel oudere) versie, het niet functioneren van beveiligingscamera's, het niet starten van auto's (alweer omdat het onderhoud te lang geleden plaatsgevonden zou hebben) of het

overbelast raken van een pijpleiding.

Een extra complicatie bij embedded systemen is dat hun aanwezigheid of herkomst soms niet meer bekend is of dat zij moeilijk te bereiken zijn zoals een sensor op de oceaanbodem of een boordcomputer van een auto die in miljoenen exemplaren wereldwijd verkocht is.

4 Oorzaken

Er vallen een aantal oorzaken aan te wijzen voor het ontstaan van het jaar-2000-probleem in de jaren 60 en 70.

- Harde schijven en intern geheugen waren duur en ponskaarten waren traag. Elke besparing op de omvang van de te verwerken gegevens leverde geld en snelheid op.¹
- Het was moeilijk voor te stellen dat de levensduur van systemen zich zou uitstrekken tot in het volgende millennium.
- Programmeertalen ondersteunden geen ingebouwde representaties voor gegevens van het type “datum”.
- Tweecijferige jaartallen zijn voor mensen heel natuurlijk (geen lezer zal in verwarring geraakt zijn over welke “jaren 60” deze paragraaf handelt). Programmatuur uit die tijd automatiseerde direct de jaarrepresentatie die voor mensen intuïtief is.
- Het bespaart tijd om datatypistes niet voortdurend “19” in te laten typen.

Hoewel er indertijd ongetwijfeld ook systemen zijn geweest die viercijferige jaartallen hanteerden, is de tweecijferige representatie in die periode de *de facto* standaard geworden waarop latere systemen zich baseerden.

Het gebruik van twee cijfers voor jaartallen heeft om de hierboven genoemde redenen waarschijnlijk veel geld bespaard in de jaren 60 en 70. Bij het ontwikkelen van latere systemen in de jaren 70 en 80, kwamen er naast bovenstaande argumenten nieuwe bij:

- Een systeem moet mogelijk datumgegevens uitwisselen met andere systemen, die vaak gebruik maken van tweecijferige jaartallen. Dit gaat makkelijker als het systeem zelf ook tweecijferige jaartallen gebruikt.
- Een nieuw systeem wordt vaak ter vervanging van een oud systeem ontwikkeld. Over het algemeen zal het zo zijn dat het nieuwe systeem moet kunnen werken met gegevensbestanden van het oude systeem, waarmee het de tweecijferige jaarrepresentatie erft.

Zelfs in de jaren 80 en 90 zullen deze argumenten voor tweecijferige jaaraanduiding voor vele systemen zijn gebruikt. Wel mag worden aangenomen dat meer en meer systemen jaar-2000-compliant werden opgeleverd, zeker vanaf 1995. Immers:

¹Hoe belangrijk efficiënt geheugengebruik is blijkt ook uit het hoofdstuk dat F. Brooks, projectmanager bij IBM OS/360, aan dit onderwerp wijdt in zijn bekende boek uit 1975, *The Mythical Man-Month* [2].

- Het werd steeds waarschijnlijker dat de software ook na 2000 in gebruik zou blijven.
- Vanaf 1993 heeft het jaar-2000-probleem door publicaties in de vakpers ruime aandacht gekregen (zie ook § 7).
- Geheugen is steeds goedkoper en groter geworden: het argument van minder geheugen-gebruik speelt daarom slechts voor systemen waarbij in beperkte tijd enorme volumes aan gegevens verwerkt moeten worden.
- De introductie van een nieuw systeem is een goed moment om de oude datumgegevens te converteren van twee naar vier cijfers.
- Interfaces naar externe systemen met tweecijferige jaartallen kunnen worden voorzien van een extra rekenlaag om deze om te zetten in de door het nieuwe systeem te gebruiken viercijferige representatie.
- Tweecijferige representaties bij het bouwen van een nieuw systeem kunnen worden voorzien van een zogenaamd *venster* (zie §5) om berekeningen na 2000 ook correct uit te voeren.

5 Technische oplossingen

Een los programma met een jaar-2000-probleem is redelijk eenvoudig te repareren. Bij een dergelijke reparatie wordt doorgaans gebruik gemaakt van een van drie mogelijke technische oplossingen: *expansie*, *vensters*, of *codering*. Hier vatten we deze oplossingen kort samen; een meer uitgebreide beschrijving is te vinden in [8, 5].

Expansie Het expanderen (verbreden) van jaarvelden naar 4 cijfers is conceptueel de eenvoudigste oplossing van het probleem. Als in het oude programma het jaartal met twee cijfers werd gerepresenteerd (YY), gebruik daarvoor dan in het geconverteerde programma vier cijfers (YYYY).

Deze simpele methode heeft echter twee grote nadelen. Ten eerste moeten alle programma's *en* alle gegevensbestanden geconverteerd worden, terwijl bij de andere hierna nog te bespreken methode's alleen de programma's geconverteerd hoeven te worden. Naarmate de gegevensbestanden omvangrijker zijn telt deze overweging zwaarder. Ten tweede treedt bij het gebruik van deze methode een *sneeuwbal* effect op dat het moeilijk maakt om een conversie te plannen. Zodra een gegevensbestand geconverteerd is zullen alle programma's die ervan gebruik maken ook geconverteerd moeten zijn. Deze nieuwe programma's gebruiken weer andere gegevensbestanden die ook weer geconverteerd moeten worden. Op deze manier zit er weinig anders op dan het hele systeem in één keer te converteren maar dit zal, in het algemeen, gezien de omvang weer niet mogelijk zijn.

Vensters (windowing) Een venster heeft tot doel om door het invoeren van een zogenaamd “breekjaar” de gegevensbestanden ongewijzigd te laten en alleen de programma’s aan te passen. Men kan, bijvoorbeeld, 1960 als breekjaar invoeren. Alle tweecijferige jaartallen in het interval 0–60 representeren dan jaren in de volgende eeuw (2000–2060), terwijl tweecijferige jaartallen in het interval 61–99 juist in deze eeuw liggen (1961–1999). Uiteraard werkt deze methode alleen als de in de gegevensbestanden voorkomende jaartallen minder dan 100 jaar uit elkaar liggen. In de praktijk is dit de meest gebruikte oplossing voor het jaar-2000-probleem.

De aanpassing van de programma’s bestaat uit het toevoegen van extra instructies die, bijvoorbeeld, ervoor zorgen dat rekenkundige bewerkingen en vergelijkingen rekening houden met het breekjaar. Deze extra instructies kunnen een klein snelheidsverlies van het systeem ten gevolge hebben. Merk verder op dat het breekjaar niet vast gekozen hoeft te worden: het is mogelijk een *schuivend venster* te gebruiken waarbij het breekjaar elk jaar met één wordt opgehoogd.

Codering Het coderen van jaarvelden is gebaseerd op de waarneming dat er nog enige verborgen ruimte over is in de velden die gebruikt worden om het jaartal op te slaan. Dit komt doordat de cijfers van het jaartal meestal *decimaal* opgeslagen worden, d.w.z. als één van de cijfers 0, ..., 9. Per positie zijn er dus tien mogelijkheden nodig. Voor de opslag van elke positie wordt echter standaard een 8-bits veld (“byte”) gebruikt dat $2^8=256$ mogelijkheden kan voorstellen! Door van deze extra ruimte gebruik te maken kan het volledige jaartal ook binnen een 6-plaatsig datumveld van de vorm YYMMDD (jaar, maand, dag) gerepresenteerd worden.

Er zijn diverse manieren om een dergelijke codering te bereiken. Al deze methoden hebben echter als nadeel dat ook weer alle programma’s en gegevensbestanden geanalyseerd en geconverteerd moeten worden. Bovendien is aan de gegevens zelf niet (altijd) te zien of deze geconverteerd zijn of niet. Tenslotte leiden deze schema’s tot grotere complexiteit van de geconverteerde code. Dit is zowel een nadeel tijdens de conversie als bij toekomstig onderhoud.

6 Aanpak jaar-2000 projecten

Een organisatie als geheel jaar-2000-compliant maken vergt meer dan het aanpassen van een enkel programma. De eerste fases zijn *bewustwording* en *inventarisatie*; daarna zal voor zelfontwikkelde software een renovatie uitgevoerd moeten worden. Hieronder worden de verschillende stappen besproken.

Bewustwording De allereerste stap moet de *bewustwording* binnen de organisatie zijn. Bewustwording omvat activiteiten die tot doel hebben dat alle geledingen van een organisatie het belang van het jaar-2000-probleem onderkennen. Veel bedrijven richten hiertoe expertisecentra of discussieplatforms in. Bewustwordingscampagnes moeten bagatellisering van het probleem bestrijden en de (financiële) steun van het topmanagement bevorderen: de hoge investeringen om het jaar-2000-probleem op te lossen lijken immers geen toegevoegde waarde voor de organisatie te hebben. De Nederlandse overheid is voornemens (juli 1997) om een Nationaal Millennium Platform te creëren dat de bewustwording op nationaal niveau moet bevorderen.

In de VS werkt 35% van de bedrijven aan een oplossing terwijl dit percentage in Europa 10% is (Bron: Fenit). In Engeland werkt 35% aan een oplossing terwijl 9% een volledige analyse van zijn systemen heeft gemaakt (bron: PA Consultants). In Nederland heeft 20% nog nooit van het probleem gehoord en 36% heeft er nog niet over nagedacht (bron: NIPO).

Inventarisatie Is men zich eenmaal bewust dat er een probleem is, dan kan begonnen worden met een *inventarisatie* van alle manieren waarop de organisatie afhankelijk is van software die een jaar-2000-probleem kan hebben. Hierbij valt te denken aan:

- Zelfontwikkelde software.
- Gebruikte standaardpakketten zoals tekstverwerkers of spreadsheets. Welke versie van de pakketten wordt gebruikt? Is deze versie jaar-2000-compliant? Moet een nieuwe versie worden aangeschaft? Worden de pakketten op de juiste manier gebruikt of introduceren gebruikers toch hun eigen jaar-2000 problemen?
- Gebruikte hardware. Nog vrij recente PC's zoals bijvoorbeeld sommige modellen van de 486, zetten na 2000 de klok terug naar 4 januari 1980 [18].
- Gebruikte embedded systemen of procesbesturingsprogrammatuur.
- Gebruik van door derden elektronisch aangeleverde datumgevoelige informatie.
- Afhankelijkheid van strategische toeleveranciers en hun aanpak van het jaar-2000-probleem.

Het maken van een dergelijke inventarisatie is tijdrovend en arbeidsintensief. Wel zijn er bedrijven die ondersteuning kunnen bieden door kennis over welke standaardsystemen wel en niet compliant zijn, en door te helpen met het opsporen van alle programma's die op een bepaalde computer gebruikt worden.

Evaluatie en planning Per gevonden systeem zal besloten moeten worden hoe de daarin aanwezige jaar-2000-problemen aan te pakken. Worden de systemen na 2000 nog gebruikt? Is het mogelijk het systeem te vervangen door een reeds compliant alternatief systeem? Wegen de kosten van een correctie op tegen de verwachte schade? Hoe worden compliance claims van softwareleveranciers beoordeeld en getest? In welke volgorde dienen de systemen geconverteerd te worden, bekeken zowel vanuit strategisch als technisch perspectief? Welke programmeertalen worden gebruikt in de diverse systemen? Welke tools of externe diensten kunnen worden ingezet om conversies te ondersteunen?

Conversie eigen programmatuur Bij het jaar-2000-compliant maken van de eigen programmatuur worden de volgende stappen doorlopen:

- **Detailinventarisatie:** localiseer alle werkelijk nodige source code. Is deze niet te vinden (gemiddeld is 5% van de sources "kwijt") dan moet decompilatie of herimplementatie overwogen worden.

- **Impact-analyse:** localiseer alle plaatsen in de source code waar potentieel gevaarlijke datummanipulaties voorkomen, en analyseer hoe deze manipulaties doorwerken in de rest van de programmatuur.
- **Detailplanning:** bepaal de volgorde waarin de deelprogramma's worden geconverteerd, selecteer tools en de te gebruiken technische oplossing (expansie, vensters of codering).
- **Renovatie:** breng de feitelijke wijzigingen in programma's en bestanden aan.
- **Validatie en testen:** onderzoek of de gewijzigde programma's enerzijds jaar-2000-compliant zijn en anderzijds geen veranderd gedrag vertonen.
- **Overdracht:** voer het gewijzigde systeem stapsgewijs in de productieomgeving in.

Elk van deze stappen is technisch gecompliceerd, arbeidsintensief en risicovol. Een conversie wordt verder bemoeilijkt door de hoge tijdsdruk (er is geen uitstel mogelijk) en door het grote aantal programma's dat geconverteerd moet worden. Hierdoor zullen in de praktijk concessies gedaan moeten worden aan de kwaliteitscriteria die normaal gesproken gelden voor bijvoorbeeld testen en overdracht. Tools en serviceproviders kunnen bij deze stappen kostenbesparend en kwaliteitsverhogend werken, maar desalniettemin zullen de totale kosten en de benodigde doorlooptijd hoog blijven.

De meest geciteerde schatting (afkomstig van de Gartner Group) voor de kosten van het converteren van programmatuur bedraagt iets meer dan \$1,- per regel code, waarbij dit bedrag oploopt naarmate het jaar 2000 dichterbij komt [14]. Andere schattingen gaan uit van \$0,75–\$1,70 per regel code tot zelfs \$8,52 per regel code voor procesbesturingssystemen [17]. Een uitgebreide bespreking van de factoren die de kosten beïnvloeden is te vinden in [11].

7 Juridische vragen

De juridische kwesties rond het jaar-2000-probleem zijn talrijk, en liggen buiten de expertise van de auteurs. Wel willen we proberen enkele vragen en feiten aan te dragen die kunnen helpen bij het bestuderen van de juridische aspecten.

Standaards Uit de omvang van het jaar-2000-probleem blijkt wel dat het gebruik van tweecijferige jaartallen zeker niet als afwijkend gedrag kan worden bestempeld. Er is weliswaar een ISO standaard voor datumrepresentaties die het formaat YYYY-MM-DD aanbeveelt, maar deze is uit 1988, en staat ook toe de eeuw weg te laten als deze niet echt nodig is [9]. Men mag dus zeker niet stellen dat deze standaard het gebruik van tweecijferige jaartallen noodzakelijk heeft gemaakt.

Wel is het zo dat sommige programmeersystemen, zoals bijvoorbeeld de meeste spreadsheets, ingebouwde datumtypes hebben. Als een taal deze ondersteunt, mag men redelijkerwijs verwachten dat een programmeur of ontwerper deze ook gebruiken en niet zelf alsnog tweecijferige getallen introduceren om jaren te representeren.

Bewustwording Men zou kunnen stellen dat het jaar-2000-probleem zich voor het eerst manifesteerde in 1969 toen systemen met gegevens over onder meer hypotheek met een looptijd van 30 jaar vastliepen [18]. Deze referentie noemt ook diverse andere gevallen (1974, 1989) waarin bij computers van verschillende leveranciers datumproblemen opgetreden zijn doordat er onvoldoende ruimte gebruikt werd voor de opslag van het jaartal.

Hoewel al voor 1990 artikelen verschenen die berichtten over de omvang van het probleem, was het Peter de Jager's artikel uit 1993 [4] dat de bewustwording op gang bracht in de Verenigde Staten. Zijn WWW site [3] bevat veel actuele informatie over het jaar-2000-probleem. Hoewel in Nederland de bewustwording later kwam, verschijnen nu ook regelmatig artikelen in kranten en tijdschriften, jaar-2000-probleem, onder meer in de *Automatisering Gids* [7, 1, 12].

Omschrijving jaar-2000-compliance Een waterdichte definitie van jaar-2000-compliance, bijvoorbeeld voor gebruik in jaar-2000-conversiecontracten, is moeilijk te geven. Een voorbeeldlijst van criteria is te vinden in [6]. Naast meer voor de handliggende criteria zoals de mogelijkheid tot eenduidige eeuwbepaling, dienen ook afspraken gemaakt te worden over risico's gerelateerd aan potentieel performance verlies vanwege een conversie of aan verminderde onderhoudbaarheid van de code na de conversie.

Kwaliteit van de conversie De kwaliteit van een conversie zelf hangt van veel factoren af: de gebruikte tools, de ervaring van de medewerkers, de systematiek van het proces, de beschikbare tijd, de mate van testen enz. — zie [5] voor een bespreking van kwaliteitsaspecten van de fases in een jaar-2000-conversie. Bij een (gerechtelijke) beoordeling of een softwareproducent een zorgvuldige poging heeft gedaan zijn software compliant te krijgen zal een kwaliteitsinspectie van het conversieproces een belangrijke rol spelen. Van belang hiervoor is een goede documentatie van het plan van aanpak, de gemaakte inventaris, de te volgen conversiemethode, de beschikbare tijd, enz.

Garanties De vraag doet zich voor hoeveel waarde gehecht moet worden aan uitspraken van bedrijven die beweren jaar-2000-compliant software te leveren. Betekent dit dat de nieuwste versie van een pakket compliant is, en dat de klanten tegen betaling moeten overstappen naar deze versie? Wat betekent een belofte dat medio 1999 een compliant versie zal verschijnen? Blindelings erop vertrouwen dat dit daadwerkelijk zal gebeuren brengt aanzienlijke risico's met zich mee.

Naast gebruikers zullen ook aandeelhouders garanties willen hebben dat de bedrijven waarin zij beleggen jaar-2000-compliant zijn. Kan een aandeelhouder schadevergoeding eisen als een bedrijf dat beweerde compliant te zijn toch in de problemen komt in het jaar 2000?

Het lijkt erop dat garanties met betrekking tot jaar-2000-compliance alleen zinvol zijn als zij gepaard gaan met openheid van zaken over de manier waarop deze compliance is of zal worden bereikt.

Eigendom programmatuur De toch al ingewikkelde aansprakelijkheidsketens die gebruikelijk zijn in de automatiseringswereld worden er door jaar-2000-conversie niet eenvoudiger op.

Wie is verantwoordelijk voor fouten in software die in opdracht van bedrijf *A* is ontwikkeld door bedrijf *B*, geconverteerd door bedrijf *C* met behulp van een door *D* ontwikkeld tool, en op compliance getest door bedrijf *E*?

Overige kwesties De lijst met aan het jaar 2000 gerelateerde juridische thema's kan moeiteloos verder uitgebreid worden: het potentieel verlies van data (rekeninggegevens, patientgegevens, boekhouding) en de gevolgen daarvan, geschillen met buitenlandse bedrijven, de rol van onderhoudscontracten, verzekeringen, enz. Verdere informatie over deze en andere juridische kwesties is te vinden op het World Wide Web, onder meer [10, 16].

8 Tot besluit

Het jaar-2000-probleem is een intrigerend informaticaprobleem met verstrekkende maatschappelijke gevolgen. Tenzij de diverse computersystemen op tijd geconverteerd worden, zal het jaar-2000-probleem zeker aanzienlijke schade veroorzaken.

Het op grote schaal uitvoeren van conversieprojecten wordt echter ernstig belemmerd door een aantal factoren. Allereerst betreft het vaak (maar zeker niet altijd) oude programmatuur. De oorspronkelijke ontwikkelaars hiervan zijn niet meer werkzaam in de organisatie, de gebruikte programmeertaal wordt door weinigen begrepen, en de oorspronkelijk gebruikte compiler wordt niet meer ondersteund.

Ten tweede is software-onderhoud het stiefkind van de automatisering. Onderhoud wordt primair gezien als kosten- en bezuinigingspost, en veel minder als investering in toekomstige flexibiliteit. Gebrek aan onderhoud maakt het moeilijk programmatuur te begrijpen, wat weer leidt tot een grotere kans op het introduceren van nieuwe fouten bij het uitvoeren van "correcties".

Een laatste factor die het adequaat aanpakken van het jaar-2000-probleem in de weg staat is het steeds groter wordende tekort aan personeel in de automatisering. Een jaar-2000-conversie is een organisatorisch complex proces dat veel menskracht vergt. Te verwachten valt dat er in 1999 niet genoeg programmeurs zullen zijn om alle software met jaar-2000-problemen te corrigeren.

Deze factoren hebben tot gevolg dat niet alle datumproblemen op tijd kunnen worden opgelost en dat computersystemen werkelijk foute resultaten zullen opleveren in het jaar 2000. Wat de juridische consequenties hiervan zullen zijn is een open vraag. In deze bijdrage hebben we geprobeerd een discussie over de juridische consequenties te ondersteunen door een overzicht te geven van de kenmerkende foutieve datumberekeningen, de mogelijke gevolgen en vermoedelijke oorzaken, en de opzet van een conversieproject. Daarnaast hebben we een aantal juridische kwesties aangestipt.

Meer informatie over het jaar-2000-probleem is te vinden in diverse boeken, zoals [15, 20]. Een overzicht van de informatica-technische aspecten van het jaar-2000-probleem (inclusief voorbeeldprogramma's) is te vinden in [5]. Enkele zeer nuttige, op het Internet beschikbare, artikelen zijn: het rapport van Capers Jones over de wereldwijde economische gevolgen [11], de gids van IBM over de technische aspecten [8], het rapport van William Rabin over de gevolgen voor de financiële markten [14], en het artikel van Jeff Jinnett met zeer veel informatie over de juridische aspecten [10].

Referenties

- [1] G. Bakker. Millenniumprobleem dwingt tot verandering onderhoudsaanpak. *Automatisering Gids*, 14 maart 1997.
- [2] F. Brooks, Jr. Ten pounds in a five-pound sack. In *The Mythical Man-Month*, chapter 9. Addison Wesley, 1975. See also the *Anniversary Edition with Four New Chapters*, 1995.
- [3] P. de Jager. The year 2000 information center. URL <http://www.year2000.com/>.
- [4] P. de Jager. Doomsday 2000. *Computerworld*, 27(36):105–109, September 1993.
- [5] A. van Deursen, P. Klint, and A. Sellink. Validating year 2000 compliance. Technical Report SEN-R9713, CWI – Centrum voor Wiskunde en Informatica, 1997. URL: <http://www.cwi.nl/cwi/publications/reports/abs/SEN-R9713.html>.
- [6] GTE. Proposed criteria for “century compliance”, 1996. GTE Technology & Systems, MA. URL <http://www.year2000.com/archive/gte-article/index.html>.
- [7] P. Hanselman. Bedrijfsleven krijgt oog voor eeuwwisseling. *Automatisering Gids*, 24 mei 1996.
- [8] IBM. The year 2000 and 2-digit dates; a guide for planning and implementation, 1997. Seventh edition. URL: <http://www.software.ibm.com/year2000/>.
- [9] ISO. ISO 8601: 1988 date/time representations, 1988. See also M. Kuhn’s comments at URL <http://www.ft.uni-erlangen.de/~mskuhn/iso-time.html>.
- [10] J. Jinnet. Legal issues confronting the federal and state governments due to the year 2000 “millennium bug” problem. Technical report, LeBoeuf, Lamb, Greene & MacRae, L.L.P, 1997. URL: <http://www.comlinks.com/legal/jjin3.htm>.
- [11] C. Jones. The global economic impact of the year 2000 software problem. URL: http://www.spr.com/html/year_2000_problem.htm, 1997. Software Productivity Research, Inc.
- [12] J. Kloeze. Operatie 2000 in handen van ongeschoolden? *Automatisering Gids*, 7 maart 1997.
- [13] P. Michael. Disastrous tales of year 2000 flops. URL: http://www.itanz.org.nz/year2000/horror/horror_toc.html, 1997. See also GartnerGroup publications.
- [14] W. D. Rabin. The year 2000 problem; ready or not, here it comes. Technical report, J. P. Morgan Securities Inc, 1997. URL <http://www.jpmorgan.com/MarketDataInd/Research/Year2000/index.html>.

- [15] B. Ragland. *The Year 2000 Problem Solver: A Five Step Disaster Prevention Plan*. McGraw-Hill, 1997.
- [16] W. S. Reid and S. Brower. *Beyond Awareness: Ten Management and Ten Legal Pitfalls Regarding the Year 2000 Computer Problem that you may not have considered yet*, 1996. URL <http://www.year2000.com/archive/beyond.html>.
- [17] J. F. Roberts. *Cost Estimation for Year 2000 Efforts*. MITRE Corporation, 1997. URL http://www.mitre.org/research/y2k/docs/COST_EST.html.
- [18] R. J. Sandler. Frequently asked questions about the year 2000 computer crisis. URL: <http://www.year2000.com/y2karchive.html>, 1997.
- [19] J. Towler. Leap-year software bug gives “million-dollar glitch”. *The Risks Digest*, 18(74), 1997. URL: <http://catless.ncl.ac.uk/Risks/18.74.html#subj5>.
- [20] W. Ulrich and I. Hayes. *The Year 2000 Software Crisis — Challenge of the Century*. IEEE Computer Society Press, 1997.
- [21] L. Wood. VISA fines banks with Y2K problems. *The Risks Digest*, 18(74), 1997. URL: <http://catless.ncl.ac.uk/Risks/18.74.html#subj6>.