

Vertaling uit Int. Journal of Heat and Mass Transfer, Vol. 36, no. 5 'Determination of moisture diffusivity in porous media using scanning neutron radiography', L. Pel, A.A.J. Ketelaars, O.C.G. Adan, A.A. van Well, p. 1261-1267, Copyright 1993, met toestemming van Pergamon Press Ltd, Headington Hill Hall, Oxford OX3 0BW, UK.

Dankwoord

De auteurs zijn grote dank verschuldigd aan P. v.d. Ende, A.W.B. Theuws en H. Smulders voor hun onmisbare hulp bij het bouwen van de meetopstelling en het uitvoeren van de experimenten. Aan dit project werd financieel bijgedragen door de KNB en de NOVEM.

Referenties

1. Philip, J.R., de Vries, D.A., Moisture movement in porous materials under temperature gradients, *Trans. Am. Geophys. Un.*, 38, 222-232 (1957).
2. Bear, J., Bachmat, Y., Introduction to modeling of transport phenomena in porous media, Vol. 4, Kluwer Dordrecht (1990).
3. Withaker, S., Simultaneous heat, mass and momentum transfer in porous media. A theory of drying porous media, *Adv. Heat Transfer*, 13, 119-200 (1977).
4. Coumans, W.J., Power law diffusion in drying processes, Ph.D. thesis, Eindhoven University of Technology (1987).
5. Platten, A.K., A study of evaporation and drying in porous building materials, Ph.D. thesis, University of Manchester (1985).
6. McLane, V., Dunford, C.L., Rose, P.F., Neutron cross sections, Vol. 4, Academic Press London (1988).

Simulatie Energie in Gebouwen: internationale samenwerking noodzakelijk

J.L.M. Hensen

Technische Universiteit Eindhoven

Thans: University of Strathclyde, Energy Systems Division, GLASGOW G1 1XJ, Scotland

Computers worden alsmaar krachtiger en er is een groeiende noodzaak tot meer gedetailleerde thermische evaluaties. Daarom wordt het toepassen van vereenvoudigde berekeningsmethoden steeds meer vervangen door het gebruik van uitgebreide, dynamische thermische evaluatieprogramma's. Deze zijn wel in staat om de complexiteit van het (bouwkundige) ontwerpen mee te nemen. Vanuit de academische optiek, ligt de tijd ver achter ons dat een individuele onderzoeker of zelfs een klein team nog in staat was om een dergelijk produkt te onderhouden en verder te ontwikkelen. Degene die de state-of-the-art wil verbeteren van een bepaald facet van simulatie, heeft de optie om vanuit niets iets nieuws te ontwikkelen of om uit te gaan van een bestaand systeem en dat vervolgens uit te breiden of te veranderen. In dit artikel wordt het standpunt verdedigd dat de laatste optie belangrijke voordelen biedt voor zowel de individuele onderzoeker als voor de simulatie-gemeenschap als geheel. De laatste optie kent echter ook een aantal valkuilen, indien we niet zorgen voor een goede ondersteuning en een efficiënte manier van samenwerking. Dit artikel beschrijft de ontwikkeling van een specifieke simulatie-omgeving om van daaruit mogelijke toekomstige ontwikkelingen aan te geven.

Inleiding

Voor zowel milieu-effecten als economische aspecten, is het uitermate belangrijk dat verstandige en goed gefundeerde beslissingen kunnen worden genomen bij het ontwerpen van gebouwen en de daarbij behorende verwarmings-, koelings- en/of ventilatie-installaties.

Met onze huidige generatie computers, is gebouwprestatie-evaluatie door middel van simulatie van gecompliceerde gebouwen installatieconfiguraties mogelijk geworden voor

een groot gedeelte van de onderzoeksweld. Men zou zelfs kunnen stellen dat de dagen van de vereenvoudigde berekeningsmethoden geteld lijken te zijn. Maar dat laatste geldt evenzeer voor de kans dat een individuele onderzoeker of een kleine groep (zoals FAGO) tijd en middelen zal hebben om nieuwe produkten (ontwerphulpmiddelen) voor de ontwerp-praktijk te produceren.

Het opnieuw 'uitvinden' van bestaande faciliteiten is zeer inefficiënt. Bij onze huidige programmatuur worden we erg beperkt in het onderzoeken van nieuwe aspecten door de gigantische hoeveelheid programma-tekst die nodig is om de warmtehuishouding van gebouw en installatie te beschrijven. Oplossingen voor dit soort problemen worden onderzocht in de recente EKS (Energy Kernel System) onderzoeken in de Verenigde Staten en in het Verenigd Koninkrijk. Het is vrijwel zeker dat we met dit soort (object-georiënteerde) ontwikkelingen op de goede weg zitten, en dat dit de systemen van de (nabije) toekomst zullen zijn. Op de korte termijn heeft de simulatiewereld, als die wil inspelen op de ontwerp-praktijk op de korte tot middellange termijn, daar echter nog niet veel aan. Momenteel is er nog een behoorlijke investering (voornamelijk in tijd) nodig voor het opzetten van een model van gebouw en installaties om de huidige generatie programmatuur te kunnen gebruiken. Organisaties zoals IBPSA (International Building Performance Association) en IEA (International Energy Agency) vormen een forum waarin aspecten zoals kennis- en technologie-overdracht, training, simulatiemethodologie en validatie aan de orde komen om zodoende de toepasbaarheid van de huidige generatie modellen te vergroten.

Voor degenen die proberen om zulke programma's te ontwikkelen of die nieuwe ideeën en faciliteiten willen toevoegen, zijn het moeilijke tijden gezien het huidige economische

klimaat. De resultaten zijn niet prettig maar geven wel inzicht: de ontwerppraktijk heeft tientallen vereenvoudigde methoden en design-aids ter beschikking. Maar de meeste hiervan zijn, wat je zou kunnen noemen, maar voor 80% af. Ze kunnen door de makers zelf worden gebruikt, maar door beperkte tijd en middelen, is het gebied dat ze beschrijven beperkt, zijn de faciliteiten die geboden worden beperkt, en door hun specifieke 'karakter' is het zeer zelden mogelijk om te communiceren of data uit te wisselen met andere systemen.

Als we kijken naar de uitgebreide modellen, dan zien we dat ze meestal beperkt (c.q. gehinderd) worden door hun gebruikers-interface en de monolitische structuur. Er is veel wilskracht voor nodig om dergelijke systemen verder te ontwikkelen en daarom vinden veranderingen veel langzamer plaats dan de simulatiewereld of de ontwerppraktijk graag zou willen.

Het meest gehandicapt is de onderzoeker die simulatie in een bepaald gebied vooruit wil brengen en die de resultaten niet alleen wil publiceren maar ook wil implementeren in een systeem. Het laat zich raden dat dit de manier is waarop veel systemen zijn ontstaan. Een onderzoeker schrijft een programma voor een bepaald probleem of proces, en na verloop van tijd groeit dit uit tot een vereenvoudigd of tot een uitgebreid model. Gegeven de huidige economische en onderzoeksfinancierings-omstandigheden is het onwaarschijnlijk (en feitelijk ongewenst) dat dit traditionele proces zo door zal gaan. Hiermee komen we bij de kern van dit artikel, namelijk hoe kan de onderzoeks- of simulatiewereld het beste de huidige generatie modellen verder ontwikkelen. Hierbij gaat het met name om het talent van veel geïsoleerde onderzoekers (die veel hebben te bieden maar geen goede uitgangsbasis hebben) en om teams die onderzoek doen op dit gebied en die verse 'injecties' vanuit andere groepen goed zouden kunnen gebruiken. Dit artikel beschrijft een basisbijdrage op het gebied van internationale samenwerking.

Derde generatie simulatiemodellen

De eerste generatie simulatiemodellen waren nog zeer handboekachtig, werden gekenmerkt door een zeer broksgewijze aanpak en gingen meestal uit van stationaire condities. Bij de tweede generatie modellen werd dynamica al belangrijker, maar bleef dit dikwijls beperkt tot een eerste orde benadering. Tweede generatie modellen kenden nog steeds de zeer broksgewijze aanpak.

De vroegste derde generatie benaderingen bij simulatie van warmtehuishouding beperkten zich meestal tot of de bouwkundige kant van het probleemdomein (vooral in het geval van onderzoekers met een bouwkundige achtergrond), of tot de installatietechnische kant (dikwijls bij werktuigbouwkundige achtergrond). Bij de eerst genoemde benadering werd de invloed van de installatie meer of minder verwaarloosd door oververeenvoudiging van de installatie. Het was (is) normaal om bijvoorbeeld schatting van het energiegebruik te baseren op een aangenomen, opgelegd verloop van de binnenluchttemperatuur. Bij de laatst genoemde benadering worden dikwijls de gecompliceerde massa- en warmtetransportmechanismen in een gebouw zeer sterk vereenvoudigd. Het gebouw (of een zone van het gebouw) wordt daarbij min of meer als een installatie-onderdeel gezien, dat niet meer is dan een thermische belasting voor de rest van de installatie.

We zijn thans echter van mening dat geen van beide benaderingen te prefereren is voor het grootste gedeelte van de problemen die te maken hebben met de thermische interactie van gebouw en installatie (en in de praktijk is dit juist het gebied waar de meeste problemen voorkomen).

Wij willen als uitgangspunt hanteren dat gebouw en installatie op gelijk nivo qua complexiteit en detail moeten worden benaderd waarbij alle belangrijke massa- en warmtetransportkoppelingen moeten worden meegenomen. Als men toegang zou hebben tot een dergelijk systeem, dan zou dat een uitermate geschikt uitgangspunt zijn om nieuwe faciliteiten te introduceren. In dergelijke systemen heeft men (als onderzoeker) ook de mogelijkheid om bestaande faciliteiten, die misschien op pragmatische benaderingen berusten, te verbeteren door ze baseren op thermodynamische grondbeginselen.

Uitgaan van een bestaand en internationaal geaccepteerd simulatiesysteem biedt grote voordelen voor een individuele onderzoeksgroep. De belangrijkste zijn:

- economisch; door de complexiteit en alleen al de omvang van de ermee samenhangende software, is het praktisch onmogelijk voor een (klein) onderzoeksteam om een dergelijk systeem te ontwikkelen en te onderhouden als een zelfstandig produkt,
- academisch:
 - als individuele groep is het niet nodig om in alle gebieden expertise te hebben;
 - aspecten die binnen een bepaald onderzoeksproject buiten beschouwing blijven zullen toch nog 'state-of-the-art' blijven;
 - kennis- en resultatenoverdracht naar de internationale onderzoekswereld gebeurt impliciet en daardoor zeer efficiënt.
- praktisch; hoe meer mensen het systeem gebruiken, hoe groter de kans is dat fouten en 'bugs' zich manifesteren en kunnen worden opgelost.

De ESP-r simulatie-omgeving

Op dit punt aangekomen is het zinvol om meer specifiek te worden. Na de oorspronkelijke eigen ontwikkelingen in de vorm van het computerprogramma KLI (Bruggen 1978, Lammers 1978, Hoen 1987) wordt bij FAGO sinds een aantal (circa 7) jaren gewerkt met de ESP-r simulatie-omgeving voor energiehuishouding in gebouw- en/of installatieconfiguraties. Dit programma is van bouwkundige origine, maar is (met name binnen FAGO) uitgebreid met faciliteiten voor de simulatie van stroming (o.a. ventilatie) en installaties. Voor wat betreft onderzoeksmogelijkheden, wordt ESP-r gekarakteriseerd door:

- het is echt op onderzoek gericht, met als doelstelling de werkelijkheid zo nauwkeurig mogelijk te benaderen op een niveau dat wordt bepaald door internationale onderzoeksinspanningen, c.q. -resultaten, op het betreffende aspect;
- het uitgangspunt is om alle warmte- en massastromen (en de onderlinge koppelingen) in gebouw en installatie mee te nemen. Het is ook mogelijk om de prestatie van gebouw en installatie te beoordelen in termen van thermische behaaglijkheid. Zodoende is het systeem uitermate geschikt om onderzoek te doen naar aspecten waarbij de onderlinge ver-

wevenheid van warmte- en massastromen een belangrijke rol speelt;

- de programmatekst is niet alleen beschikbaar maar ook handelbaar, omdat het systeem sterk modulair is opgezet en (voor onderzoek) belangrijke opties zoals ingebouwde tracé-faciliteiten biedt;
- het systeem is goed gedocumenteerd; in de programmatekst zelf is zeer veel commentaar opgenomen; trainingmateriaal en tutorials zijn ingebouwd; er is een uitgebreide hand-leiding die regelmatig wordt aangepast (Aasem et al. 1993) en er is ook uitgebreide achtergrond documentatie beschikbaar (bijvoorbeeld Clarke 1985, Hensen 1991);
- het systeem wordt thans over de gehele wereld gebruikt door onderzoeksteams;
- het systeem werd (en wordt nog steeds) onderworpen aan diverse internationale validatie-onderzoeken (zie bijvoorbeeld CEC 1989);
- het systeem biedt uitgebreide grafische mogelijkheden;
- omdat het systeem is geïmplementeerd op UNIX werkstations, zijn alle andere programma's (voor software engineering, numerieke technieken, documentatie en rapportage, data-reductie, data-analyse, etc) die bij UNIX horen (of 'public domain' zijn), automatisch ook beschikbaar voor iedereen die het systeem gebruikt.

Er moet overigens ook worden opgemerkt (en dit is zeker niet op een negatieve manier bedoeld) dat door het op onderzoek en evolutie georiënteerde karakter, ESP-r niet zo gladjes is zoals men bijvoorbeeld zou verwachten van een commercieel pakket. Het systeem verwacht (en verdiend) een pro-actieve benadering van de gebruiker.

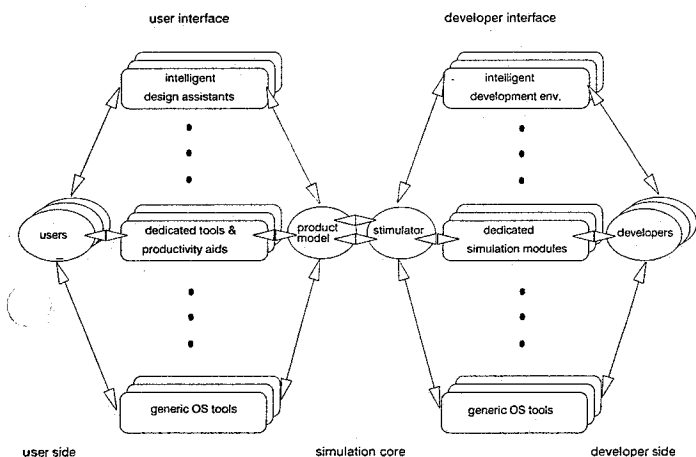


Fig. 1. Diagram van de ESP-r simulatie-omgeving.

Figuur 1 is een diagram van de verschillende individuen, componenten en interacties binnen de huidige opzet.* Voor een uitgebreide beschrijving kan worden verwezen naar (Clarke 1985, Aasem e.a. 1993, Hensen 1991) en naar een groot aantal andere publicaties.

De kern van het systeem wordt gevormd door de 'simulator'. Dit gedeelte voert de feitelijke simulaties uit op basis van een 'product model'. Dit laatste is de volledige verzameling van data die het model beschrijft; d.w.z. in de huidige con-

* Zoals aangeduid door de puntjes in figuur 1 kunnen er ook hulpmiddelen zijn die niet echt in een van de categorieën vallen.

text: gebouw, installaties, stromingsnetwerk, bewoningsparameters, bouwterrein en omgeving, buitenklimaat, etcetera.

De 'users' (of gebruikers) definiëren en werken aan het 'product model' met een range van hulpmiddelen. De hulpmiddelen die samen het gebruikers-interface vormen, kunnen in drie categorieën worden ingedeeld:

- 'intelligent design assistants'; dit zijn hoog-nivo gebruikers-interfaces gebaseerd op geavanceerde Informatie Technologie technieken (kennissystemen, Human Computer Interface technieken, etc.), zoals een 'intelligent front end' als het bij ESRU ontwikkelde 'IFe'; Dit is een belangrijk onderzoeks- en ontwikkelingsgebied;
- 'dedicated project tools' en 'productivity aids' die specifiek voor ESP-r zijn gemaakt, zoals bijvoorbeeld de 'general project manager' 'ESP-r' die (onzichtbaar voor de gebruikers) indien nodig of gewenst allerlei programmas opstart: database managers, specifieke pre-simulatie data-analyse modules (voor bijvoorbeeld het bepalen van zichtfactoren, verdeling van binnenkomende zonnestraling over de verschillende wanden van een vertrek, beschaduwning van gevel- en dakgedeelten) en het programma waarmee de simulatieresultaten kunnen worden geanalyseerd;
- 'generic tools'; dit zijn de programmas die bij UNIX horen (tekstverwerkers, file system managers, etc), die uit verschillende UNIX-tools zijn samengesteld (zogenaaemde shell scripts), of die door derden worden aangeleverd (hetzij 'public domain' zoals 'grtool', 'xfig' en 'psraster', of (semi-)commercieel zoals 'ww', 'ten' en 'ralbrowse').

Aan de ontwikkelkant van het systeem bevinden zich de 'developers' die, via een 'developers interface' nieuwe modules kunnen toevoegen (of wijzigen) aan de 'simulator'. De hulpmiddelen in het 'developers interface' kunnen ook in drie categorieën worden ingedeeld:

- 'intelligent development environments'; hoog-nivo ontwikkelaars-interfaces die gebruik maken van geavanceerde Informatie Technologie technieken, d.w.z. een object-georiënteerd 'energy kernel system' zoals het 'eks'; dit is ook een belangrijk onderzoeks- en ontwikkelingsgebied;
- 'dedicated simulation modules'; die specifiek gericht zijn op het simuleren van een bepaald aspect van het gehele probleem-domein, installaties, 'mfs' voor stromingen, etc.;
- 'generic development tools'; zoals de tools die bij UNIX horen (tekstverwerkers, debuggers, verifiers, etc.), daarvan afgeleid zijn (shell scripts), of 'software engineering tools' die door derden worden aangeleverd (hetzij 'public domain' zoals 'toolpack', 'floppy' en 'f2c', of (semi-)commercieel zoals 'forchk').

Het bovenstaande geeft een globaal beeld van de context waarin ESP-r 'groeit'. Het geeft ook een indicatie waarom meer en meer mensen te maken hebben met (de software ontwikkeling van) het systeem. Een onderzoeker kan als gebruiker en/of als ontwikkelaar optreden. Het zal duidelijk zijn dat het totale systeem en de organisatie ervan steeds uitgebreider en complexer wordt.

Management van samenwerking

Door de toenemende complexiteit en het aantal mensen dat met het systeem te maken heeft, neemt ook de kans toe dat er

duplicaties optreden, dat bepaalde gedeelten van het systeem elkaar negatief beïnvloeden, dat er niet-gedocumenteerde faciliteiten zijn, dat er niet-ondersteunde gedeeltelijke 'releases' plaats vinden, dat er 'resistente' fouten zijn, dat de stijl van programmeren sterk varieert, etc.

Het idee is nu, dat naleving van bepaalde (voor iedereen acceptabele) regels (of geaccepteerde gedragscodes zo men wil) het risico verkleint dat bovenvermelde zaken voorkomen. Dit zal vooral voordelen opleveren voor de middellange- en lange-termijn. Op de korte-termijn echter, moet men er echter aan wennen en energie in steken, en er wordt ook een zekere mate van zelfdiscipline van alle betrokken personen verwacht.

Organisatiestructuur

We hebben dus duidelijk te maken met een op onderzoek gerichte simulatie-omgeving waarin de veranderingen met een tamelijk hoog tempo (en vanuit verschillende invalshoeken) plaats vinden. Dit komt tot uitdrukking in de structuur van de organisatie rondom het systeem, zoals schematisch is aangegeven in Figuur 2.

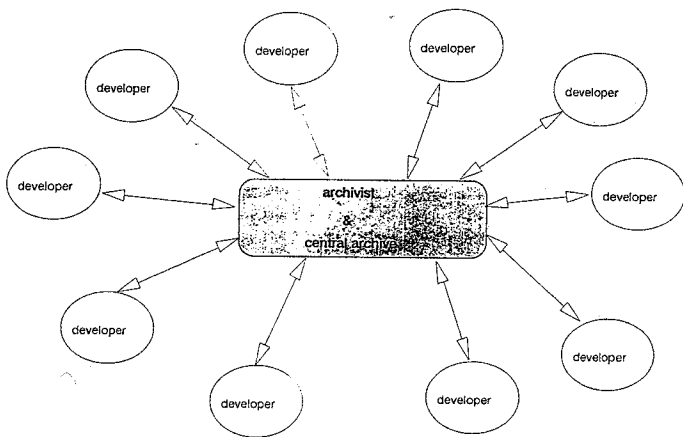


Fig. 2. Schematische weergave van de organisatie structuur rondom ESP-r.

De organisatie is stervormig gestructureerd. In het midden bevindt zich het centrale archief met daarin de meest recente versie van het systeem. De archivering vindt plaats in een centrale locatie (de Energy Simulation Research Unit van de University of Strathclyde in Glasgow, waar ESP-r oorspronkelijk ook vandaan komt). Het archief wordt beheerd door slechts één persoon: de archivaris.

Rond het archief bevinden zich verschillende ontwikkelaars en gebruikers. Deze laatste blijven hier verder buiten beschouwing omdat hun contact met het systeem-archief slechts eenrichtingverkeer betreft. Ontwikkelaars en gebruikers bevinden zich op een groot aantal locaties overal ter wereld. Iedere ontwikkelaar mag (met toestemming van de archivaris uiteraard) een gedeelte van het systeem (of compleet) ophalen, om de voorziene ontwikkelingen of wijzigingen aan te brengen. Als de betreffende onderdelen van het systeem 'klaar zijn', worden ze teruggestuurd naar de archivaris die ze vervolgens toevoegt aan het archief.

Het is de verantwoordelijkheid van de archivaris om een of andere vorm van 'source code control' uit te voeren om

zodoe de ontwikkeling van het systeem te registreren. In de toekomst zouden we ertoe kunnen overgaan, om hiervoor een CASE-tool (Computer Aided Software Engineering) in te zetten.

De archivaris is er ook verantwoordelijk voor dat nieuwe programma-onderdelen geen negatieve invloed hebben op andere gedeelten in termen van structuur en data-uitwisseling; d.w.z. hij/zij moet ervoor zorgen dat het systeem coherent blijft. De ontwikkelaar zelf is en blijft echter verantwoordelijk voor de validiteit in termen van inhoud en functie. De gearchiveerde versie is in principe ook de meest recente 'release' van het totale systeem. Deze versie bevat namelijk alle bekende 'bug-fixes' en de laatste toevoegingen zoals die eerder ergens werden ontwikkeld en getest, en vervolgens geaccepteerd door de archivaris. Uiteraard moeten alle nieuwe toevoegingen aan het systeem zijn voorzien van voldoende documentatie met betrekking tot doelstelling, hoe het werkt, hoe je het kunt gebruiken, voorbeelden, etc.

Na verloop van tijd, kan de archivaris ertoe overgaan om een 'officiële' aankondiging te doen van de release van een nieuwe versie van het systeem. Deze laatste versie zal in principe gewoon bestaan uit de gearchiveerde versie op dat moment. Tussen de verschillende officiële releases, evolueert het systeem gewoon door. Als een onderzoeker van plan is om iets te wijzigen of toe te voegen, dan moet in principe altijd uit worden gegaan van de gearchiveerde versie op dat moment. Het is ook belangrijk om op te merken dat het systeem nooit verder wordt ontwikkeld dan een zogenaamd beta-nivo. De reden hiervoor is dat alle betrokken personen zich met onderzoek bezig houden en niet met bijvoorbeeld commercialisering van het programma. In het algemeen is er geen onderzoeksbudget voor het maken van een alpha-release van het systeem, alhoewel we dat soms wel zouden willen, bijvoorbeeld voor het ontwikkelen van een (extra) robuust systeem voor onderwijsdoeleinden.

Electronische infrastructuur

Aangezien we werken met UNIX op werkstations die via een wereldwijd netwerk met elkaar zijn verbonden, zijn allerlei hulpmiddelen die met deze technologie te maken hebben voor iedereen beschikbaar. Hierbij kan dan gedacht worden aan het via het netwerk transporteren van programma's, source code, data, etc. (met 'ftp'), het op afstand ondersteunen van bijvoorbeeld implementatie van het systeem (met 'telnet' waarmee vanaf een workstation in bijvoorbeeld Eindhoven rechtstreeks kan worden gewerkt op een andere machine ergens ter wereld), maar ook aan zaken zoals automatisering van implementatie en compilatie procedures (met het 'make' programma), en aan het elektronisch discussie platform.

Electronisch discussie platform

Om elkaar op de hoogte te houden van lopende en voorgestelde ontwikkelingen, nieuwe releases, etc. zou het nuttig kunnen zijn om regelmatig bij elkaar te komen. Dit brengt uiteraard de nodige reizen en kosten met zich mee. In ons geval is er echter een 'natuurlijk' alternatief, namelijk om contact te houden via een soort elektronisch discussie platform voor ESP-r.

De bedoeling van deze faciliteit is om elkaar op de hoogte te houden van lopende en voorgenomen ontwikkelingen, ver-

schijnen van publicaties die met het systeem te maken hebben, nieuwe resultaten c.q. problemen, aankondigingen van nieuwe releases, etc. Een van de belangrijkste voordelen hiervan zou een betere effectiviteit (te weten meer output) van de onderzoekers zijn, maar het is eenvoudig in te zien dat er veel meer voordelen zijn.

Voor wat betreft de vorm van dit communicatie medium zou men kunnen denken aan een gedrukte nieuwsbrief of iets dergelijks. Maar vanwege de beschikbare elektronische infrastructuur is het veel zinvoller om dan ook van die technologie gebruik te maken (dit lijkt zo triviaal, maar het is in de praktijk verbazingwekkend om te zien hoe weinig dit dikwijls nog wordt gedaan). Je zou daarbij kunnen denken aan een of ander centraal 'bulletin board', maar dat moet dan weer worden onderhouden door iemand (die waarschijnlijk wel andere dingen te doen heeft). Als alternatief hiervoor hebben we er vooralsnog voor gekozen om een communicatie platform op te zetten op basis van elektronische post.

De manier waarop dit mechanisme kan worden gebruikt is als volgt: iedereen die zich aan wil sluiten of een of ander bericht naar de ESP-r groep wil zenden, stuurt dit gewoon per elektronische post naar 'esp-r@esu.strathclyde.ac.uk'. Deze virtuele 'server' stuurt vervolgens automatisch een kopie van het inkomende bericht naar alle deelnemende personen. Op deze manier hebben we een automatisch, elektronisch bulletinboard. Momenteel zijn er ruim 50 onderzoekers in circa 20 landen over de gehele wereld aangesloten.

Good practice procedures

Het werd hierboven al gesuggereerd, maar het is belangrijk om nog eens op te merken dat wij ingenieurs en wetenschappers zijn, met meestal weinig training in informatica, voor wie het oplossen van een technisch/wetenschappelijk probleem belangrijker is dan het toepassen van geavanceerde software ontwikkel-methodologieën. We schrijven onze eigen programma's en hebben daarbij meestal alleen elementaire software (zoals editors en compilers) beschikbaar om applicaties te ontwikkelen die zeer uitgebreid en gecompliceerd kunnen zijn. We doen dat omdat de methodologieën die door moderne software-ontwikkelprogrammatuur (onderdeel van commerciële pakketten) ondersteund worden, niet passen binnen onze werkgewoonten en (wat belangrijker is) ook niet binnen ons budget. Door 'trial and error' hebben we een eigen methodologie ontwikkeld die is gebaseerd op stapsgewijze ontwikkeling en vooral gericht is op produktie. In zekere zin, proberen we meestal zo snel mogelijk een eerste prototype te creëren. Deze wordt vervolgens getest waarbij ad hoc de fouten worden opgespoord en opgelost. We willen zo snel mogelijk resultaten, en dit kan wel eens prioriteit krijgen boven het zich zorgen maken over het opstellen van specificaties, ontwikkelfasen of documentatie.

Als we niet oppassen, kan dit makkelijk leiden tot een monolithisch programma of een module die zeer moeilijk te testen, te veranderen, opnieuw te gebruiken of over te dragen is, en in principe alleen nog door degene die het heeft ontwikkeld te begrijpen is. Kort gezegd: het eindproduct is van een slechte kwaliteit, hetgeen meestal pas blijkt als men het opnieuw wil gebruiken in een andere toepassing of als er 'mysterieuze' fouten beginnen op te treden.

Om de kwaliteit van onze software te waarborgen en te verbeteren, hebben we een aantal 'good practice guidelines' aangenomen die zijn gebaseerd op de wijze zoals we nu werken en die redelijke resultaten lijken op te leveren.

De eerste en meest belangrijke regel is dat iedere ontwikkelaar zelf verantwoordelijk is voor zijn of haar werk. Uiteraard houdt dit in dat eigen ontwikkelingen goed worden getest en van documentatie worden voorzien. Als dit niet wordt gedaan heeft dat niet alleen consequenties voor hem of haar zelf maar in feite ook voor alle andere collega onderzoekers. Op korte termijn omdat die dan last hebben van de fouten geïntroduceerd door iemand anders, en op de lange termijn omdat het zeer moeilijk is om een goede naam op te bouwen, maar zeer makkelijk om een slechte naam (en alle daaraan verbonden consequenties) te krijgen.

Zoals aangegeven in Figuur 2 is de organisatie gestructureerd als een ster. Het overdragen van programmatekst naar collega's moet alleen plaats vinden via het centrale archief; d.w.z. een ontwikkelaar moet geen programmatekst rechtstreeks van een collega overnemen. Verder geldt dat bij iedere nieuwe ontwikkeling uitgegaan moet worden van de gearchiveerde versie op dat moment. Deze regels moeten ervoor zorgen dat er geen 'mix-up' plaats vindt. Uiteraard kan een ontwikkelaar ervoor kiezen om zijn of haar recente werk te laten gebruiken en testen door een lokale groep van (positief ingestelde) gebruikers. Dat zorgt er alleen maar voor dat de software meer robuust wordt en voorkomt een te vroege algemene release. Voordat programmatekst wordt teruggestuurd naar de archivaris moet het grondig worden getest, bij voorkeur met een of andere 'code checker'. Een dergelijke 'checker' kan alleen semantische of structurele fouten opsporen. Functionele of algoritmische fouten zijn veel moeilijker op te sporen. Dit moet dan ook op een andere manier verwezenlijkt worden; dit is het uitermate belangrijke terrein van verificatie en validatie zoals bijvoorbeeld beschreven in het PASSYS eindrapport (CEC 1989).

Als nieuwe of gewijzigde programmatekst naar het archief wordt gestuurd, moet het zijn voorzien van een globale beschrijving van de wijzigingen. Deze informatie wordt door de archivaris gebruikt om de ontwikkeling van het systeem te kunnen registreren.

Als nieuwe (of gewijzigd met betrekking tot functionaliteit voor de gebruikers) gedeeltes worden ingebracht, dan moeten deze zijn voorzien van de betreffende wijzigingen in de handleiding en bij voorkeur ook van een bijbehorend (uitgewerkt) voorbeeld. Dit soort documentatie is in principe gericht op de gebruikers maar het is ook zeer nuttig voor de andere ontwikkelaars omdat ze een soort verbale beschrijving geven van de veranderingen/nieuwigheden in het systeem.

Het lijkt verder een goede zaak om voor iedere module een referentieprobleem (gedocumenteerd in de handleiding) te hebben. Deze kunnen dan worden gebruikt om bij iedere (voorgestelde) nieuwe versie van het systeem de impact te testen (voor de verschillende problemen) van de nieuwe veranderingen ten opzichte van de voorgaande versie van het systeem.

Good Practice Coding

Het is belangrijk dat de programmatekst steeds min of meer dezelfde stijl heeft. Het is hierbij overigens zeker niet de

bedoeling om bepaalde nieuwe elementen aan de ontwikkelaars op te leggen; het is veel meer een zaak van het beschrijven van wat algemeen geaccepteerde regels/ stijl etcetera is op dit moment. Uiteraard kan dit veranderen in de toekomst. Een identieke stijl heeft een esthetische functie, maar, wat belangrijker is, het verbetert ook de leesbaarheid en maakt een programma meer toegankelijk voor andere (toekomstige) ontwikkelaars.

Gedeeltelijk om dezelfde redenen, is het uitermate belangrijk dat een programmatekst goed gedocumenteerd is.

Stijl

In termen van stijl, is één van de belangrijkste zaken om in de gaten te houden dat een programma op een nette en strakke manier wordt geschreven; men moet zich houden aan de KISS (Keep It Simple, Stupid) benadering. Er moet voor worden gezorgd dat het programma eenvoudig te lezen en (opnieuw) te begrijpen is voor de ontwikkelaar zelf (zelfs na er een paar maanden niet mee bezig geweest te zijn) en voor anderen. Dit houdt bijvoorbeeld in dat als een vergelijking moet worden geëvalueerd, je deze beter kunt programmeren zoals je het op papier zou opschrijven. Dus geen termen combineren (en daarmee de duidelijkheid verminderen) alleen om een beetje CPU rekentijd te besparen. In onze context, zijn we veel meer geïnteresseerd in het besparen van ontwikkeltijd.

Men moet proberen om het programma zo gestructureerd mogelijk te maken. Uiteraard kan dit makkelijker (en op een meer natuurlijke manier) in C dan in FORTRAN. Maar zelfs in het laatste geval zijn er verschillende mogelijkheden (soms alleen visuele) om dat te bereiken. Om te beginnen doet men er goed aan om in FORTRAN de GOTO constructies zoveel mogelijk te vermijden.

Er dient bovendien opgemerkt te worden dat sommige FORTRAN compilers (zoals bijvoorbeeld de Sun FORTRAN compiler) de mogelijkheid bieden om hoger-nivo mechanismen zoals bijvoorbeeld DO WHILE-loops toe te passen. Het is sterk aan te bevelen om dergelijke niet-genormeerde (in ANSI 1978) faciliteiten niet te gebruiken, omdat de kans zeer groot is dat deze problemen opleveren zo gauw het programma naar andere machines moet worden overgebracht of als andere compilers moeten worden gebruikt.

Er moet zoveel mogelijk gebruik worden gemaakt van variable/parameternamen die zelfverklarend zijn. De betekenis van een bepaalde variable/parameter moet niet zomaar ergens worden veranderd; als bijvoorbeeld 'windspeed' wordt gebruikt dat moet de intrinsieke betekenis niet worden veranderd in 'kwadraat van de windsnelheid' alleen omdat het als zodanig op de meeste plaatsen wordt gebruikt. Als men voor het laatste kiest, dan is het veel beter om dat in de naam van de variable/parameter tot uitdrukking te laten komen.

Het komt dikwijls voor dat in een wetenschappelijke applicatie zoals ESP-r een aantal modules voorkomen met dezelfde structuur omdat ze eenzelfde soort taak uitvoeren. Bijvoorbeeld de matrixcoëfficiënten generatoren voor de verschillende installatieonderdelen verschillen meestal alleen voor wat betreft de vergelijkingen die de feitelijke coëfficiënten berekenen. Als het dan zo is dat er modules (in dit voorbeeld nieuwe installatiecomponenten) moeten worden toegevoegd, dan is het aan te bevelen om uit te gaan van de bestaande structuur omdat dat waarschijnlijk veel duidelijker zal zijn voor toekomstige verdere ontwikkelaars.

Documentatie

Het soort documentatie waar het hier over gaat is bedoeld voor ontwikkelaars; dus niet voor de gebruikershandleiding. Het gaat hier ook niet over de globale beschrijvingen zoals bedoeld in de vorige paragraaf.

Geen enkele ontwikkelaar maakt graag documentatie, maar iedereen heeft het nodig en klaagt als het niet goed is gedaan. Documentatie moet volledig en consistent zijn. Daarvoor zijn twee redenen. Ten eerste kunnen ontwikkelaars na hun eigen werk een tijdje niet meer te hebben gezien zich de details ook niet meer herinneren, en ten tweede verdwijnt de persoonlijke context als iemand anders ermee verder moet werken.

Net zoals voor stijl, geldt voor documentatie dat het sterk afhangt van de persoonlijke voorkeur. Ook geldt weer dat dit iets is wat waarschijnlijk evolueert in de tijd. Er zijn echter een aantal algemene uitgangspunten die kunnen worden gehanteerd:

- gebruik van volledige (en correcte) Engelse zinnen;
- iedere subroutine begint met een globale beschrijving van doelstelling en hoe dit wordt verwezenlijkt;
- iedere volgende (belangrijke) taak of activiteit binnen een subroutine wordt voorafgegaan door commentaar.

Als deze simpele regels worden gevolgd, dan zijn subroutines makkelijk te volgen en te begrijpen. Als we alle commentaar zouden verzamelen dan zou dit opzichzelf al leiden tot een 'rapport' van wat er gebeurt.

CASE Tools

Er bestaan een aantal Computer Aided Software Engineering (CASE) tools die bij het voorgaande kunnen helpen. Sommige van dit soort programma's horen bij het UNIX operating system, zoals: 'dbx' en 'lint'. Andere CASE tools zijn zogenaamde '3rd party' producten die (semi-)commercieel zijn of 'public domain'. Een vermeldenswaardig voorbeeld van de laatste categorie zijn de 'toolpack' FORTRAN CASE-tools met o.a. modules voor het controleren van syntaks en semantiek, voor statische analyse, voor run-time analyse, voor het opschonen en polijsten van programmatekst, herstructureren in gestandaardiseerde vorm, etc.

Conclusies

Internationale samenwerking op het gebied van simulatie van de energiehuishouding in gebouwen heeft belangrijke voordelen voor zowel individuele onderzoekers (of onderzoeksteams) als voor de simulatiegemeenschap in zijn geheel. De belangrijkste voordelen zijn een verhoogde effectiviteit, een grotere onderzoeksoutput en snellere ontwikkelingen. Dit kan niet worden bereikt door een of andere 'opgelegde' samenwerking, maar kan alleen op basis van individuen die een gezamenlijke doelstelling of overtuiging hebben. Hetzelfde geldt voor gebruikte infrastructuur en de gevolgde procedures. In plaats van te proberen om bepaalde nieuwe elementen of een strakke structuur op te leggen, is het veel beter om uit te gaan van geaccepteerde praktijk en om een of andere flexibele manier van interactie en management op te zetten. Dit alles kan veranderen in de loop van de tijd, als zich nieuwe mogelijkheden voordoen of indien bepaalde inzichten/opinies ver-

anderen. We hebben de indruk dat de organisatie structuur, het onderlinge communicatie mechanisme en de 'good practice procedures' betreffende ESP-r momenteel goed voldoen, en dat er inderdaad sprake is van een grotere onderzoekseffectiviteit.

Nawoord

Het bundelen van krachten op het terrein van simulatie van de energiehuishouding in de gebouwen, lukt alleen als de betrokken personen dit echt willen en zich positief opstellen. De resultaten die tot nu toe zijn bereikt, zouden niet zijn bereikt zonder de voortdurende inspanningen van een groot aantal personen. Hiervoor is dan ook dank verschuldigd aan alle ESP-r collega's, en daarom moet dit artikel feitelijk als een gezamenlijke activiteit worden gezien.

Literatuurverwijzingen

Aasem, E., Clarke, J.A., Hand, J.W., Hensen, J.L.M., Pernot C.E.E., Strachan, P. 1993, ESP-r, A Program for Building Energy Simulation; Version 8 Series, Energy Simulation Research Unit, ESRU Manual U93/1, University of Strathclyde, Glasgow.

ANSI 1978, American National Standard Programming Language FORTRAN, ANSI X3.9-1978, American National Standards Institute Inc.

Bruggen, R.J.A. van der, 1978., Energy Consumption for Heating and Cooling in relation to Building Design, Dissertatie Technische Universiteit Eindhoven (FAGO).

CEC 1978, The PASSYS Project Phase 1. Subgroup Model Validation and Development final report 1986-1989, 033-89-PASSYS-MVD-FP-017, Commission of the European Communities, DG XII of Science, Research and Development, Brussel.

Clarke, J.A. 1985, Energy Simulation in Building Design, Adam Hilger Ltd, Bristol (UK).

Hensen, J.L.M. 1991, On the Thermal Interaction of Building Structure and Heating and Ventilating System, Dissertatie Technische Universiteit Eindhoven (FAGO).

Hoën, P.J.J. 1987, Energy Consumption and Indoor Environment in Residences, Dissertatie Technische Universiteit Eindhoven (FAGO).

Lammers, J.T.H. 1978, Human Factors, Energy Conservation and Design Practice, Dissertatie Technische Universiteit Eindhoven (FAGO).

Muziekcentrum Frits Philips Eindhoven

prof. ir. P.A. de Lange

Emeritus hoogleraar TU Eindhoven

ir. L.C.J. van Luxemburg

Centrum Bouwonderzoek TNO-TUE

Inleiding

De media hebben uitvoerig bericht over het Muziekcentrum Frits Philips te Eindhoven, dat op 2 september 1992 officieel voor het publiek is opengesteld. De schrijvers waren verantwoordelijk voor het akoestisch ontwerp, drie lokale architecten (een gelegenheidscombinatie)* voor het architectonisch ontwerp.

Eindhoven had al jaren een 'echte' concertzaal nodig en in 1987 werd tot stichting van een muziekcentrum met een kleine en een grote zaal besloten. Eerste vereiste: een zaal met 1300 stoelen met 'een perfecte akoestiek voor natuurlijk, niet versterkt geluid'. Daarnaast een kamermuziekzaal met 400 stoelen. Men sprak de wens uit Het Brabants Orkest (HBO) naar Eindhoven te laten verhuizen. Dat is gebeurd.

In maart 1988 werd opdracht gegeven voor de akoestische adviezen. Op 20 april gaven wij ons eerste 'schot voor de boeg' in de vorm van een 'Nota van aanbevelingen inzake akoestische aspecten'. De architecten waren toen nog niet aangewezen! Deze nota was een handreiking voor de architecten en behandelde de criteria in voor hen (zonder ervaring in het ontwerpen van een concertzaal of met goede kennis van dergelijke zalen) begrijpelijke taal. Met plaatjes, criteria die o.i. moesten worden aangehouden en informatie over wat deze voor het ontwerp betekenen.

* Ir. R.J. van Aken, L.L.J. de Bever en dr. ir. C. van der Ven, allen te Eindhoven.

Het ontwerp

Het Muziekcentrum Frits Philips is geïntegreerd in een zogeheten Shopping Mall: de Heuvelgalerie.

De toegang tot het Muziekcentrum bevindt zich in de hal van de Heuvelgalerie die de verbinding vormt van dit winkelcomplex met de Markt. Op de begane grond zijn naast de entrees naar beide zalen de kassa en de garderobe gesitueerd. Op deze bouwlaag bevindt zich ook een restaurant dat oorspronkelijk bedoeld was als onderdeel van het Muziekcentrum maar uiteindelijk los daarvan geëxploiteerd wordt (zie figuur 1).

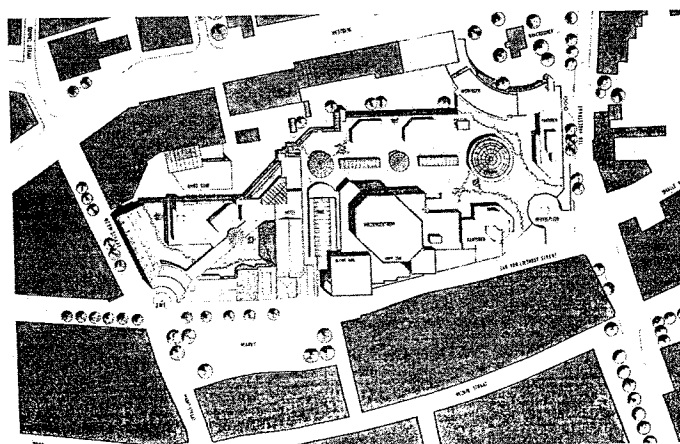


Fig. 1. Plattegrond Heuvelgalerie met Muziekcentrum.