

Regelstrategieën in software gebouw- prestatiesimulatie

De huidige generatie software voor gebouwsimulatie is in staat een gebouw heel gedetailleerd te modelleren, inclusief de installaties. Waar dergelijke programma's tekortschieten is op het gebied van de regeling van de installatie. Meestal kunnen alleen eenvoudige regelstrategieën worden onderzocht, volgens een stramien dat al van tevoren door de programmeur is bedacht. Moderne geavanceerde regelstrategieën, zoals het geïntegreerd regelen van temperatuur en verlichting of zelflerende systemen, kunnen niet worden gesimuleerd.

Een manier om regelingen flexibeler te integreren in gebouwsimulatie is door een koppeling te realiseren tussen software voor gebouw-simulatie en software voor modellering van regelstrategieën. Dit artikel beschrijft de technische mogelijkheden voor een dergelijke koppeling.

-door A. Yahiaoui, prof.dr.ir. J. Hensen* en dr. L.L. Soethout***

Het ontwerp van de installaties in een gebouw, inclusief de regeling van deze installaties, vereist een vergelijkende studie van diverse mogelijke configuraties om te komen tot een optimaal ontwerp. Te meer omdat een groter comfort van de bewoners of gebruikers, en hieraan gerelateerd hun gezondheid en productiviteit, van een gebouw steeds belangrijker wordt gevonden terwijl het energiegebruik zo laag mogelijk dient te blijven. Voor dit ontwerp zijn gedetailleerde simulaties van groot belang, het liefst al zo vroeg mogelijk in het ontwerpproces.

De huidige software voor gebouwsimulatie (denk bijvoorbeeld aan ESP-r [1] en VA114 [2]) modelleert gebouwen en hun installaties zeer gedetailleerd en flexibel. Wat betreft de bijbehorende regelingen voor de installaties schieten ze echter tekort. Alleen conventionele

regelstrategieën zijn doorgaans beschikbaar. De complexiteit van moderne gebouwen (grote glasvlakken, grote open ruimten, passieve koeling), het verwachtingspatroon van gebruikers (eigen invloed op het persoonlijke comfort) en de onderlinge interactie tussen regelingen (zie figuur 1) stellen echter andere eisen aan de regeling van de installaties. Geavanceerde regelingen die hierop inspelen kunnen momenteel niet afdoende worden getest.

Een manier om toch simulaties te kunnen doen aan dergelijke regelingen is om deze regelingen in te bouwen in de bestaande software voor gebouwsimulatie. Gezien de huidige trend dergelijke software modulair op te bouwen, om de complexiteit van de software het hoofd te bieden, lijkt dit geen goede oplossing. Ook blijft de gebruiker van de software gebonden

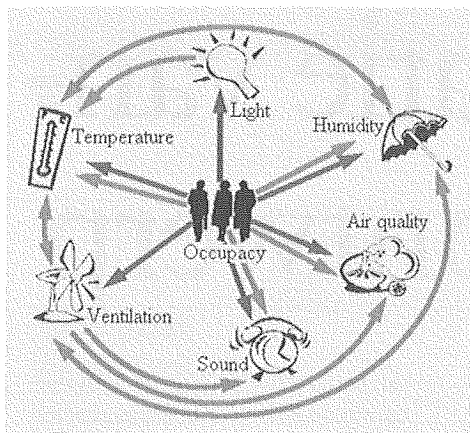
aan de toevallig door de programmeur geïmplementeerde regelingen.

Een andere oplossing is de simulaties compleet te implementeren in een ontwikkelomgeving bedoeld voor regeltechniek. Een voorbeeld van een dergelijke omgeving is Simulink, onderdeel van de Matlab-omgeving [3]. Het nadeel hiervan is dat deze omgevingen minder geschikt zijn voor een gedetailleerde modellering van het gebouw en simulaties daardoor leiden tot lange rekentijden.

In onze ogen is de beste oplossing een koppeling tussen aan de ene kant een omgeving voor gebouwsimulatie en aan de andere kant een omgeving voor simulatie van regelsystemen. De voordelen qua modellering van beide omgevingen blijven gehandhaafd, terwijl de combinatie leidt tot een toename van de simulatiemogelijkheden. Deze oplossing is ook flexibel, daar een wijziging van één van de twee onderdelen geen invloed heeft op de werking van de andere. Dit vereist wel goede afspraken over de onderlinge communicatie, zowel voor het technische mechanisme als wat betreft de informatie die over en weer gaat. Ook stimuleert het gebruik van gescheiden omgevingen het hergebruik van reeds ontwikkelde modules, ongeacht het gebruikte ontwikkelplatform. De enige eis is dat dergelijke modules worden voorzien van een interface dat het communicatieprotocol volgt. Op deze manier vervallen de grenzen tussen applicaties van verschillende producenten en worden dubbelures in de ontwikkeling vermeden, wat weer leidt tot lagere kosten.

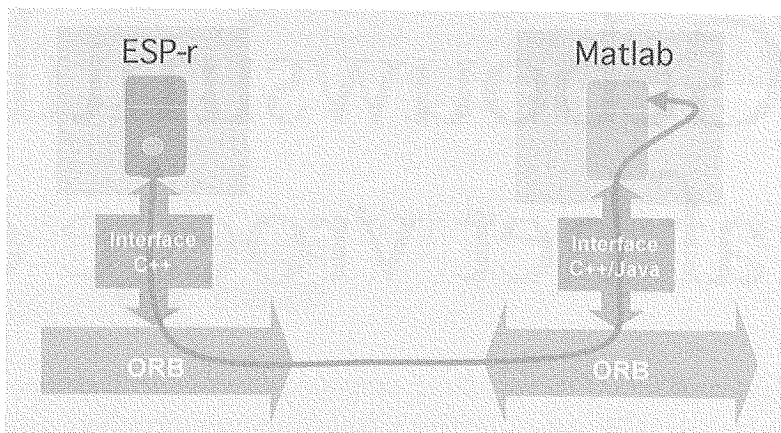
* Center for Building and Systems TNO-TU/e

** TNO Bouw



De onderlinge interactie tussen regelingen.

-FIGUUR 1-



Schematisch overzicht van de communicatie met CORBA tussen simulatie-omgevingen voor gebouwsimulatie en regelingen.

-FIGUUR 2-

COMMUNICATIETECHNIEKEN

De communicatie tussen de software-omgevingen voor gebouwsimulatie en voor regelingen is zeer intensief, niet zozeer wat betreft de hoeveelheid data die wordt uitgewisseld maar wel wat betreft de frequentie. Communicatie vindt minstens één maal per tijdstap (in de simulatie) plaats, en vaker wanneer een iteratie vereist is om tot de gewenste nauwkeurigheid te komen. Bij gebouwsimulatie varieert een typische tijdstap van één minuut tot een uur, terwijl dit bij regelalgoritmen in de orde van seconden kan zijn. Bij een simulatie die een jaar doorrekenen ligt het aantal tijdstappen in de orde van tienduizend en hoger, dus ook het aantal communicatiemomenten. De snelheid van de communicatie is dus van groot belang, zeker als voor een optimaal ontwerp vele simulaties nodig zijn.

Een andere belangrijke eis is platform-onafhankelijkheid. Één van de prototypes die we willen bouwen bestaat uit de combinatie ESP-r, werkend onder Unix, en Simulink, werkend onder Microsoft Windows. Hierdoor vallen communicatietechnieken die zijn ontwikkeld voor één type besturingssysteem af. Hiertoe behoren pure Microsoft Windows oplossingen als distributed component object model (DCOM) en de hierop geënte gedistribueerde technieken van .NET, evenals technieken die weliswaar op verschillende platforms draaien maar niet in een heterogene omgeving van bijvoorbeeld Microsoft Windows en Unix computers, zoals anonymous pipes, shared memory en semaphores.

Voor de snelheid is een directe koppeling met de achterliggende simulatie-omgeving een pre. Hierdoor valt ook

de op Java gebaseerde remote method invocation (RMI) af.

Uiteindelijk blijven maar twee mechanismen voor ons doel over: sockets en de common object request broker architecture (CORBA). Beide technieken zijn algemeen toepasbaar voor het uitwisselen van informatie tussen applicaties, ongeacht het besturingssysteem, en maken gebruik van de internetstandaard TCP/IP. De technieken zijn gebaseerd op het client-servermodel, waarbij één zijde van de communicatie gebruikmaakt van de diensten die de andere zijde biedt. CORBA is een stuk complexer dan sockets, maar biedt een objectgeoriënteerde structuur die communicatie op een hoger conceptueel niveau mogelijk maakt. Bij sockets dienen veel meer afspraken over het communicatieprotocol te worden gemaakt, implementatiedetails die weinig te maken hebben met de eigenlijke run-timekoppeling die ons voor ogen staat. Binnen CORBA zijn dergelijke details al voor ons geregeld. Experimenten laten ook nog eens zien dat CORBA in vergelijking met sockets een factor zestig sneller is, ongeacht de grootte van de overgedragen datapakketjes. Dit maakt CORBA bij uitstek geschikt als koppelingsmechanisme voor simulatiesoftware.


Figuur 2 toont de communicatie met CORBA tussen simulatieomgevingen voor gebouwsimulatie en regelingen.

CONCLUSIES

Om geavanceerde regelstrategieën al tijdens het ontwerpproces te kunnen evalueren is ons inziens een run-timekoppeling noodzakelijk tussen bestaande software voor gebouwsimulatie en

simulatieomgevingen gespecialiseerd in regelsystemen.

Een dergelijke koppeling vereist een robuuste communicatietechniek die snel is, die simulatieomgevingen kan koppelen in een heterogene netwerk-omgeving, en die een beschrijving van de informatiebehoefte op een hoog conceptueel niveau toelaat. Experimenten tonen aan dat de beste kandidaat hiervoor CORBA is.

De volgende stap in ons onderzoek gaat kijken naar de conceptuele kant van de informatie-uitwisseling tussen de simulatieomgevingen. Hierbij spelen vragen een rol als: welke grootheden zijn van belang voor diverse typen regelingen, en hoe verloopt de synchronisatie voor de tijdstappen in beide omgevingen? Na afloop hiervan zijn alle lagen van de communicatie tussen de simulatieomgevingen bekend. Dan zal op basis hiervan een prototype koppeling worden ontwikkeld tussen diverse omgevingen om na te gaan of een dergelijke koppeling ook praktisch haalbaar is. Hiermee zal vervolgens een realistische casus worden gemodelleerd, die een complexe bouwstructuur zal combineren met een geavanceerde regelstrategie. 

REFERENTIES

1. ESP-r, <http://www.esru.strath.ac.uk/Programs/ESP-r.htm>.
2. VA114, <http://www.vabi.nl>.
3. Matlab en Simulink, <http://www.mathworks.com>.
4. CORBA, <http://www.omg.org>.