Wat is OpenMI en wat kan het betekenen?

P.W. Dirksen en M.W. Blind

Inleiding en achtergrond

Hydrologisch onderzoek en advies wordt veelal ondersteund door rekenmodellen van diep grondwater, onverzadigde zone en oppervlaktewater die aan elkaar gekoppeld worden. In Nederland is in de negentiger jaren geconstateerd dat modellen vaak slecht koppelbaar zijn en hierdoor onder meer samenwerking tussen de verschillende kennisinstituten en specialismen gehinderd wordt (STOWA, 1997; Zanting e.a., 1997). Andere punten van zorg zijn de ontwikkeling van ad-hoc koppelingen waardoor 'specials' ontstaan, die vervolgens moeten worden onderhouden. Om de situatie te verbeteren is onder leiding van STOWA, RIVM, RIZA, Alterra, WL|Delft Hydraulics en NITG-TNO het project 'standaard raamwerk applicatie' uitgevoerd (Van der Wal, 1999). Dit Nederlandse project heeft een vervolg gekregen in het project 'HarmonIT - IT Frameworks' (www.harmonit.org). Dit project is medegefinancierd door de Europese Unie in het 5e Kaderprogramma Onderzoek. Het product van dit project heet OpenMI: 'Open Modelling Interface' (www.openmi.org). Dit is een set afspraken waaraan modellen moeten voldoen om met elkaar te kunnen communiceren. In tegenstelling tot ad-hoc koppelingen biedt OpenMI generieke oplossingen voor de technische aspecten van dergelijke koppelingen.

OpenMI wordt langzamerhand door een groot aantal instituten en bedrijven in Nederland en Europa omarmd. Omdat de term OpenMI daarom steeds vaker valt volgt in dit artikel een korte kennismaking met OpenMI, in de vorm van een toelichting op de belangrijkste concepten.

Wat is OpenMI?

OpenMI bestaat uit interface definities, dat wil zeggen een set afspraken waaraan software componenten moeten voldoen om gegevens tijdens de uitvoering van modellen (in 'run-time') met elkaar uit te wisselen. Software componenten zijn bijvoorbeeld modellen, maar OpenMI betreft ook andere software functionaliteit, bijvoorbeeld databases, visualisatie-software, etc. Daarnaast bestaat OpenMI uit een software-bibliotheek waarmee bestaande en nieuwe componenten op een relatief eenvoudige wijze aan de interface-definities kunnen worden aangepast.

OpenMI kent geen centrale aansturingsoftware ('controller'); componenten worden geactiveerd zodra zij een vraag om gegevens van een andere component ontvangen. Dit is

P.W. Dirksen en **M.W. Blind** werken bij Rijksinstituut voor Integraal Zoetwaterbeheer en Afvalwaterbehandeling (RIZA), p.w.dirksen@riza.rws.minvenw.nl m.blind@riza.rws.minvenw.nl.

een belangrijke wijziging ten opzichte van de huidige situatie waarin de sturing veelal door een centrale controller wordt geregeld. In vaktermen heet de OpenMI insteek 'pull-driven'. Om in een OpenMI gekoppeld systeem te kunnen werken moeten componenten zogenaamd 'OpenMI-compliant' gemaakt worden.

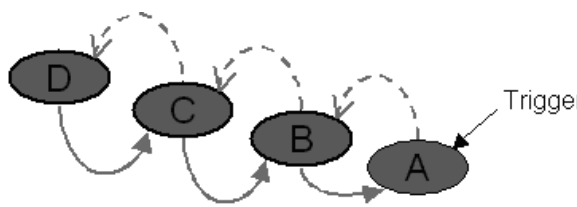
In dit hoofdstuk worden de belangrijkste concepten van OpenMI toegelicht zonder daarbij te veel in te gaan op de details.

Concept 1: scheiding van functionaliteit

OpenMI vereist een gestructureerde opzet van een component. Binnen een rekenmodel moeten processen zoals 'initialisatie', 'inlezen invoergegevens', 'uitvoeren van een enkele rekenstap', 'bewaren van uitvoergegevens' en het 'afronden van een simulatie' apart kunnen worden aangestuurd. Deze functies mogen ook geen onderlinge afhankelijkheid hebben. Rechtstreekse interactie tussen rekenhart en gebruiker tijdens een simulatie is niet gewenst.

Concept 2: Het pull mechanisme

Het belangrijkste vernieuwende aspect is de vraaggestuurde aansturing, het zogenaamde 'pull' mechanisme (figuur 1). Dit betekent dat een model aan een ander model vraagt gegevens te leveren, en er geen overkoepelende aansturingsoftware noodzakelijk is.



Figuur 1: Het 'Pull'-mechanisme, A vraagt B, B vraagt C, C vraagt D. D doet een berekening en levert data aan C. C kan vervolgens rekenen en levert data aan B. B rekent en levert data aan A die dan tenslotte ook kan rekenen.

Na initialisatie (zie hieronder) blijven verschillende modelcomponenten wachten totdat er door een andere component gegevens gevraagd worden. Als alle componenten op elkaar zouden blijven wachten gebeurt er natuurlijk niets, vandaar dat er een 'trigger' voorzien is die aan de laatste component in een keten gegevens vraagt en daarmee het proces op gang brengt. Het vragen om gegevens vindt plaats via de zogenaamde 'GetValues' opdracht waarbij als argumenten de parameter, locatie en periode meegegeven worden.

Concept 3: Verantwoordelijkheden

Door het ontbreken van overkoepelende aansturingsoftware wordt er veel verantwoordelijkheid gelegd bij de betreffende modelcomponenten, bijvoorbeeld rondom verschalingen in

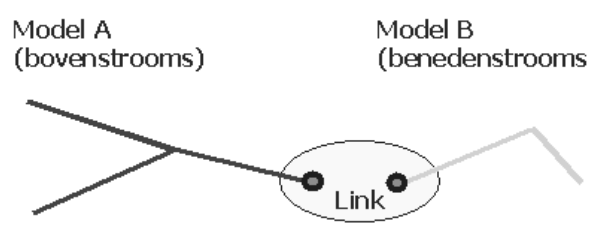
ruimte en tijd. Bij OpenMI is de verantwoordelijkheid voor verschaling in ruimte en tijd gelegd bij de componenten die gegevens leveren: de beheerder van een model weet immers zelf het best hoe bijvoorbeeld interpolaties in de ruimte uitgevoerd moeten worden.

De verantwoordelijkheden voor de verschillende processen worden hieronder besproken.

Initialisatie:
Voordat een simulatie wordt uitgevoerd, moeten de individuele componenten worden geïnitialiseerd. Na de initialisatie moet een component kunnen aangeven welke parameters en op welke locaties het kan leveren of ontvangen. De combinatie van parameter en locatie wordt in OpenMI aangeduid als een 'ExchangeItem'. Verder moet een component kunnen aangeven over welke periode berekeningen kunnen worden uitgevoerd. De trigger kan op basis van deze perioden de gemeenschappelijke simulatieperiode van alle componenten bepalen.

Samenstellen gekoppeld systeem:
Het daadwerkelijk koppelen van componenten kan via de beschikbare gebruikersschil (User Interface), middels andere daarvoor ontwikkelde software of gewoon met een tekstverwerker. In alle gevallen is het uitgangspunt van de koppeling de door de componenten gepubliceerde ExchangeItems. Koppeling van componenten komt neer op het definiëren van een link tussen ExchangeItems van twee componenten. Het leggen van een valide link is de verantwoordelijkheid van de gebruiker. Wel controleert de OpenMI software of het domein (d.w.z. lengte, gewicht, inhoud, etc.) van de beide bij de link betrokken parameters overeenstemmen. In OpenMI termen heet een gekoppeld systeem een 'configuratie'.

Wederzijdse afhankelijkheid:
Bij het koppelen van modellen komt het geregeld voor dat deze modellen van elkaars berekeningsresultaten afhankelijk zijn en dus tijdens de berekeningen, op dezelfde tijdstap, gegevens moeten uitwisselen. Een voorbeeld is weergegeven in figuur 2: Indien een 1-dimensionaal waterbewegingsmodel gekoppeld wordt aan een ander 1-dimensionaal waterbewegingsmodel en zij daarbij op één locatie gekoppeld worden, heeft model A de waterstand van model B nodig om het debiet uit te kunnen rekenen. Tegelijkertijd heeft model B het debiet van model A nodig om deze waterstand te kunnen bepalen. De OpenMI software voorziet in deze mogelijke patstelling door te registreren of een vraag om gegevens het gevolg is van een eerder (zelf) gestelde vraag. In dat geval is deze component verplicht een zo goed mogelijke schatting te leveren. Hierover meer in paragraaf 5, sectie 'Wederzijdse afhankelijkheid en Iteratie'.



Figuur 2

Iteratie:
Indien het uitvoeren van een rekenstap in een keten van componenten iteratief dient plaats te vinden, wordt bij OpenMI de verantwoordelijkheid voor het besturen van deze iteratie bij een aparte component gelegd; de 'IterationController'.

Terug in de tijd:
In geval van iteraties wordt een tijdstap meerdere malen doorlopen. Een component moet daarom in staat zijn een tijdstap opnieuw te berekenen en dus terug te gaan naar de uitgangssituatie van die eerdere tijdstap. Bij OpenMI zijn hiervoor twee interfaces gedefinieerd, te weten 'SaveState' en 'RestoreState'. Implementatie van deze methoden in een component kan een probleem zijn, omdat daarvoor de toestand van alle variabelen op een bepaald tijdstip bewaard en weer terug gelezen moeten kunnen worden. OpenMI heeft de interfaces daarom wel gedefinieerd, maar gebruik van deze interface is niet noodzakelijk om een model 'OpenMI-compliant' te maken. In dat geval is een iteratieve berekening natuurlijk niet mogelijk.

Hoe maak je een component OpenMI compliant?

'OpenMI compliant' maken houdt in dat een rekenkern wordt aangepast aan de OpenMI standaard. Hierbij moet een bestaand rekenhart zo worden aangepast dat deze de door OpenMI gedefinieerde interfaces ondersteunt. Het implementeren van deze interfaces kan door deze rechtstreeks in het rekenhart te programmeren, of door een schil rond het rekenhart te ontwikkelen, de zogenaamde wrapper. De wrapper communiceert met de rekenkern via functieaanroepen.

Het realiseren van de wrapper bestaat uit twee soorten taken:
1 Werkzaamheden aan het rekenhart
2 Werkzaamheden aan de wrapper en de wrapper-rekenhart communicatie

De werkzaamheden aan het rekenhart betreffen met name het scheiden van functionaliteiten (figuur 3). De gebruikersschil, initialisatie, rekenen en afronden moeten strikt gescheiden worden. Het rekenhart moet tevens per tijdstap aangeroepen kunnen worden, dus pas rekenen als daar expliciet om wordt gevraagd. Het rekenhart mag geen eigen resultaatschermen of interactieve berichten naar het scherm versturen. Alle relevante gegevens moeten toegankelijk zijn, hetzij om ze te lezen, te wijzigen of beide.

Figuur 3

Bij de initialisatie moet worden aangegeven welke ExchangeItems beschikbaar worden gesteld aan andere componenten en wat van andere componenten kan worden ontvangen. Tot slot dienen al deze taken via functieaanroepen toegankelijk te zijn.

Voor het daadwerkelijk maken van een modeleigen wrapper kan geput worden uit een aantal functies van de OpenMI functiebibliotheek; er is een complete wrapper in deze bibliotheek opgenomen, de 'SmartWrapper'. Deze wrapper biedt al standaard functies voor het verschalen in ruimte en tijd, vandaar de toevoeging 'Smart'. Alleen indien deze verschalingen niet voldoen, is het noodzakelijk daarvoor zelf een oplossing te creëren.

Het OpenMI compliant maken van modellen valt in de praktijk mee, doordat veel gebruik gemaakt kan worden van beschikbaar gestelde software en het proces van de

migratie goed beschreven is in 'OpenMI Document Series: Part B - Guidelines for the OpenMI (version 1.0); 2005'.

Overige zaken

Opzetten van gekoppelde systemen

Het leggen van koppelingen tussen componenten, de 'configuratie', wordt binnen OpenMI ondersteund door een grafische user interface. Het is echter niet noodzakelijk om deze te gebruiken om een gekoppelde modelberekening te definiëren of te starten. De uiteindelijke systeemdefinitie is niet meer dan een plat bestand met een XML structuur. XML staat voor eXtended Markup Language.

Visualisatie en opslag van resultaten

Visualisatie en opslag wordt via een 'luister' mechanisme gerealiseerd. Een visualisatiecomponent (bijvoorbeeld de OpenMI DataMonitor) wordt gekoppeld aan een 'ExchangeItem' en luistert of er een verandering in waarde is, waarna het de waarde weergeeft.

Modeleigen user interfaces en user interfaces voor gekoppelde systemen

In principe kan om een bestaande user interface direct een wrapper worden gemaakt, zodat ook het User Interface OpenMI compliant wordt en dus gekoppeld kan worden aan andere componenten. Te denken valt hierbij aan databases en voor- en nabewerkingsprogrammatuur.

Een Beslissings Ondersteunend Systeem (BOS), bestaande uit bijvoorbeeld een gebruikersschil, database met invoergegevens en modellen kan met de OpenMI interfaces gekoppeld worden tot een werkende, flexibele configuratie (Dirksen e.a., 2005).

Performance

Het HarmonIT project heeft tot nu toe vooral gekeken naar de werking van de concepten. Er is nog geen uitgebreide performancetest uitgevoerd op complexe modelsystemen, hier zal in het vervolg van het project meer aandacht aan worden gegeven.

Vast staat dat er meer 'overhead' is. Bij de gewrapte modellen zijn twee koppelingen nodig: van de wrapper naar buiten en van de wrapper naar het rekenhart. Daarnaast heeft de wrapper natuurlijk ook één en ander aan functionaliteit en geheugen nodig. Het geheugengebruik neemt dus toe.

Parallel/gedistribueerd rekenen

OpenMI faciliteert nog geen parallel/gedistribueerd rekenen; het HarmonIT project was te beperkt om ook dit aspect nu al mee te nemen. Binnen een component kan men echter wel gedistribueerd en parallel rekenen, aangezien men hier bijvoorbeeld vrij is om vanaf de wrapper naar meerdere computers opdrachten te zenden.

Een voorbeeld: koppeling van een verzadigde zone model aan een model voor de onverzadigde zone

Steeds meer worden onverzadigde zone modellen gekoppeld aan verzadigde zone modellen (voorbeelden: De Lange en Vermulst, 1999; Pastoors en Kovar 2002). Doordat de onverzadigde zone berekeningen vaak meer detail in ruimte en tijd kennen, moeten verschaling in deze dimensies plaats vinden. Een voorbeeld van een dergelijke koppeling tussen een onverzadigde zone model, hierna aangeduid als OZ, en een verzadigde zone model, hierna aangeduid als VZ, wordt in deze paragraaf uitgewerkt volgens de hiervoor genoemde principes van OpenMI.

De stappen die worden doorlopen zijn weergegeven in het onderstaande 'sequence diagram'. Een beknopte toelichting op het 'sequence diagram' is hieronder weergegeven.

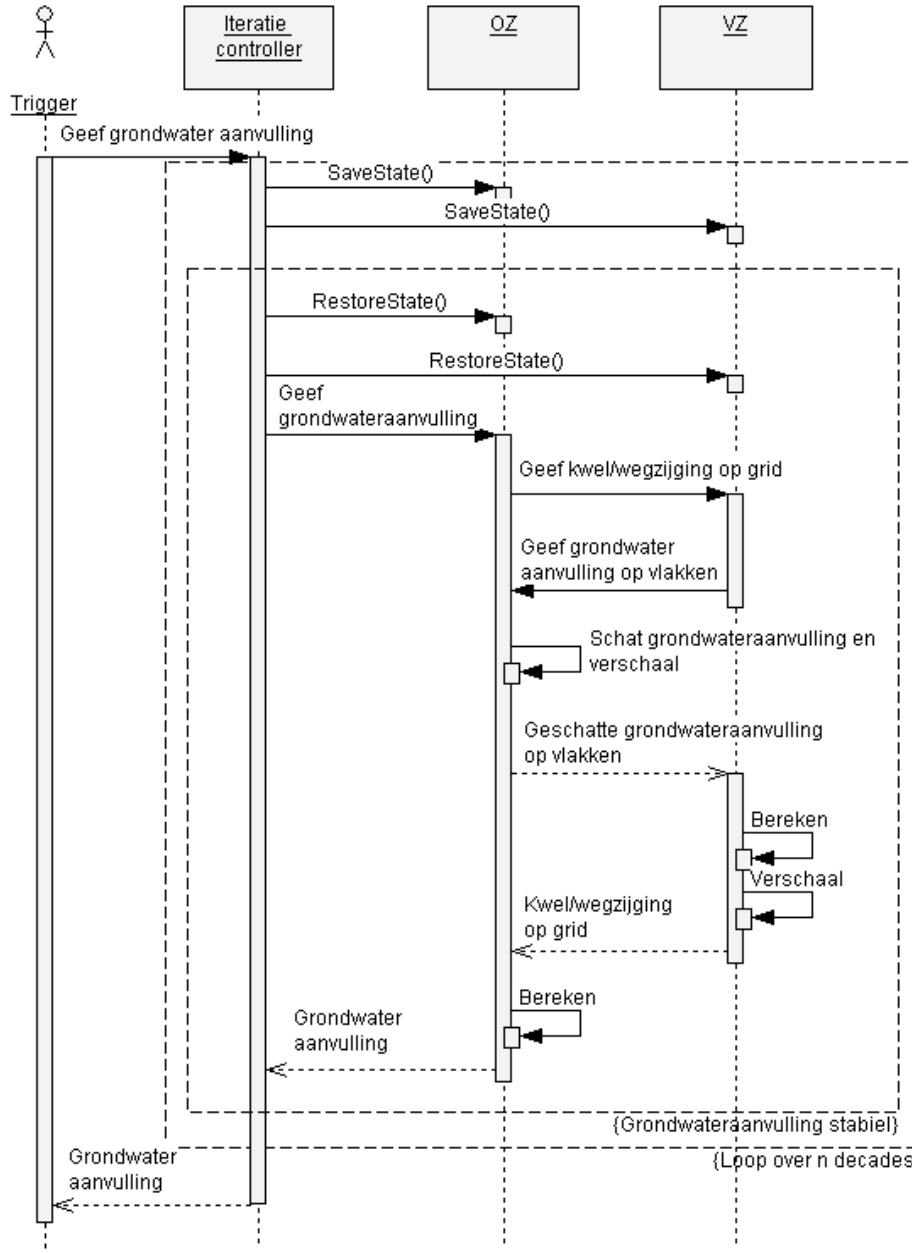
> 'Sequence diagrammen' worden gebruikt om processen in de tijd in samenhang weer te geven. De componenten staan bovenaan; de blokjes op de verticale lijnen hieronder geven activiteiten weer. T = 0 staat bovenaan. Blokken geven iteraties weer, waarbij het bijschrift rechts onder een blok informatie weergeeft over de iteratie. Peilen tussen modellen geven interactie weer; een vraag (`call') wordt in de peilrichting gesteld.

Het meest interessante aan de koppeling is de verschaling in ruimte en tijd. In het voorbeeld veronderstellen wij een tijdstap van 1 decade voor het OZ-model en een jaar voor het VZ-model. De volgende verschalingen in ruimte en tijd worden uitgevoerd (Tabel 1):

	OZ –> VZ	VZ -> OZ
Tijd	Per plot wordt de jaarlijkse grondwateraanvulling bepaald op basis van de 36 decades.	De berekende jaarwaarde van kwel/wegzijging wordt over decades verschaald met behulp van een seizoensfunctie.
Ruimte	De aanvullingen per plot binnen een vlak worden opgeteld. Stijghoogte en weerstanden worden gebruikt om de kwel/wegzijging te te verschalen.	

Tabel 1: Eenvoudige voorbeelden voor verschalingen tussen OZ en VZ. In complexere situaties kan men ook weerstanden en peilen in tijd en ruimte opschalen en uitwisselen.

Bovendien is te zien dat de koppeling een iteratie-controller nodig heeft om tot een oplossing te komen.



Figuur 4: Sequence diagram voor een OpenMI-koppeling.

Verschaling in tijd en ruimte

Om het totale neerslagoverschot te berekenen heeft het OZ-model 36 decade waarden nodig waarvoor het echter van het VZ-model 36 kwel/wegzijgingen nodig heeft. In dit voorbeeld stelt OZ per decade de vraag aan VZ 'Geef mij de totale kwel/wegzijging voor

tijdspanne t voor elke grid-cel'. VZ rekent echter met een tijdstap en van een jaar op vlakken (polygonen). Door de vraag op deze wijze te stellen moet in VZ de verschaling in tijd en ruimte worden geïmplementeerd. Via de vraag wordt de ruimtelijke schematisatie (grid) van OZ aan VZ bekend gemaakt. Volgens de OpenMI filosofie weten de VZ specialisten het beste hoe de jaarberekeningen in tijd en ruimte verschaald moeten worden.

Om de flexibiliteit van OpenMI te onderstrepen volgen hier twee alternatieven: Ten eerste zou je de verschalingsfuncionaliteit als apart model in het systeem kunnen opnemen. OZ stelt zijn vraag aan dit aparte model, dat zelf weer VZ bevraagt om de waarde per VZ vlak. Nadat VZ de antwoorden heeft gegeven verschaalt het tussenmodel de gegevens en geeft antwoord aan OZ. Op deze wijze werken veel gekoppelde systemen. Het tweede alternatief is dat OZ op bijvoorbeeld een 1m meter resolutie voor elke coördinaat de flux vraagt, en vervolgens zelf de verschaling uitvoert. Deze implementatie kan handig zijn als men wil koppelen aan een model dat bepaalde verschalingsfuncties niet kent, en er geen mogelijkheid is deze toe te voegen.

Wederzijdse afhankelijkheid en Iteratie

Als VZ waarden moet leveren aan OZ, heeft het eerst nog de volgende gegevens van OZ nodig: grondwateraanvulling, weerstand en peil per vlak per jaar. VZ zal deze grootheden via het GetValues mechanisme opvragen. OZ herkent dat een cirkelredenering dreigt, en zal daarom een beste schatting geven. Afhankelijk van de situatie zijn dit waarden van een vorige tijdstap, de startwaarden of de resultaten van een eerdere iteratie: wederom zijn het de experts die verantwoordelijk zijn voor de beste schatter.

In principe kan het systeem vervolgens doorrekenen: VZ krijgt de benodigde (geschatte) invoer, levert het antwoord op de GetValues call van OZ. OZ geeft na berekening het antwoord aan de trigger. In werkelijkheid en in dit voorbeeld zal men vaak een iteratiecontroller toevoegen aan het systeem: Deze controller laat de VZ-OZ koppeling zo vaak herhalen tot de geschatte grondwateraanvulling de berekende grondwateraanvulling benadert. Nota bene: Dit houdt in dat OZ zijn initiële schatting moet kunnen aanpassen, en dat beide modellen terug in de tijd moeten kunnen (SaveState/RestoreState).

Initialisatie

Nog voordat een eerste GetValues call wordt gegeven, worden OZ en VZ geïnitialiseerd. Hierbij maken ze ze kenbaar welke gegevens ze kunnen uitwisselen, en voor welke punten en voor welke tijdspanne berekeningen kunnen worden uitgevoerd. Hiervoor is het noodzakelijk dat de modellen de schematisatie en overige invoer ingelezen hebben. De meest gangbare methode voor het beschikbaar stellen van invoergegevens is via een set invoerfiles. Echter de data kunnen ook opgeslagen worden in OpenMI compliant data componenten, de voorbewerkingsfuncties kunnen ook OpenMI compliant gemaakt worden etc. Deze opties zijn niet in figuur 4 opgenomen. Wederom is de keuze aan de experts: Welke gegevens en schematisaties worden naar buiten toe kenbaar gemaakt? Als modellen dezelfde invoer gebruiken, is het te overwegen om een gezamenlijke OpenMI-compliant datacomponent te ontwikkelen.

Stand van zaken en toekomstplannen

Met ingang van 1 mei 2006 is het HarmonIT project afgerond. De HarmonIT partijen en vele anderen gaan echter door met ontwikkelen. Tabel 2 geeft de huidige stand van zaken rondom OpenMI compliant modellen weer zoals deze bij RIZA momenteel bekend is.

Eigenaar	Product
Delft Hydraulics Software	Sobek Channel Flow, Delft3D en DelftFEWS
PCRaster	PCRaster
Alterra	Capri, FSSIM, Apes, MetaSwap en Simgro
RIZA	Sobek-RE, Mozart, NwSim, Agricom, DemNat en DistrConnector
Waterloo Hydrologic Inst.	Visual Modflow
WRc Plc	STOAT
DHI Software	Mike11, Mike-She, Mike Basin en Mike Urban
Wallingford Software	Infoworks RS en Infoworks CS
HR Wallingford	Sulis (3D)
CEH Wallingford	CLASSIC

Op dit moment wordt vooral in enkele Europese projecten voortgeborduurd op OpenMI: AquaStress (http://www.aquastress.net/), Seamless (http://www.seamless-ip.org/) en Sensor (http://www.sensor-ip.org/). In deze projecten wordt nagedacht over uitbreidingen naar andere software groepen zoals workflow managers en kennisssytemen, en wordt gewerkt aan een ontologie om de inhoudelijk aspecten van koppelingen verder te vereenvoudigen. Het project 'nature-oriented flood damage prevention' (http://nofdp.bafg.de/) is bezig softwareontwikkeling uit te besteden met OpenMI als randvoorwaarde.

Een OpenMI demonstratieproject voor de kaderrichtlijn water is doorgedrongen tot de laatste fase, maar er wordt nog geen garantie gegeven dat het wordt gesteund. In dit project zal veel aandacht besteed worden aan een internationale ondersteuningsorganisatie voor OpenMI gebruikers.

Er is een zelfstandige organisatie in oprichting die OpenMI gaat beheren. Het lidmaatschap van deze organisatie zal open zijn voor geïnteresseerden. De papieren OpenMI standaard en bijbehorende bibliotheken zijn vrijelijk beschikbaar (http://sourceforge.net) en open source onder de Lesser Gnu Public License (www.gnu.org/licenses/lgpl.html).

Tenslotte: In Nederland wordt op dit moment gewerkt aan verdere inbedding op verschillende niveaus. Er is groot commitment gegeven door WL|Delft Hydraulics, Alterra, en RWS RIZA om OpenMI waar zinvol toe te passen. Op RWS en landelijk niveau zijn discussies gaande voor een breder draagvlak. WL|Delft Hydraulics, Alterra, MX.Systems, HKV|Lijn in Water en RIZA hebben ervaring opgedaan met OpenMI.

Discussie

OpenMI is een open standaard en geen modellen raamwerk. Het kan gebruikt worden voor situaties waarin gegevens moeten worden uitgewisseld, ook voor complexe, wederzijds afhankelijke koppelingen. Alhoewel de meeste ervaring is opgedaan met hydrologische

modellen, is de bruikbaarheid ook getoetst op modellen uit andere disciplines. De ontwikkelaars zijn overtuigd dat door de genericiteit van de concepten het gehele milieu-domein baat kan hebben bij de standaard.

OpenMI wordt gedragen door een aantal grote (semi-) (commerciële) instituten. Dit zou niet zijn gebeurd als er geen gemeenschappelijke overtuiging zou zijn dat softwarekoppelingen vereenvoudigd moeten worden om zo de vragen efficiënter te kunnen beantwoorden. Dit wordt verder gestaafd doordat de meeste partijen slechts 50% van hun budget van de EU krijgen, het overige geld komt uit instituuteigen fondsen.

Wat betekent het OpenMI nu voor de Nederlandse hydrologische modellering en kennisontwikkeling?

Vanuit wetenschappelijk oogpunt levert OpenMI verbeterde mogelijkheden om verschillende modellen te koppelen en te onderzoeken hoe de interacties beter gemodelleerd kunnen worden, en welke modelcombinaties bijzonder succesvol zijn. Het biedt de hydrologische modellering verbeterde mogelijkheden om bijvoorbeeld te de waterkwantiteit te koppelen aan de waterkwaliteit of uit te wisselen met effectmodellen voor natuur, landbouw, recreatie etc. Dit zal ook tot gevolg hebben dat specialisten van verschillende domeinen beter met elkaar kunnen samenwerken, en veel minder door IT technische discussies worden gehinderd.

Doordat verschalingsbibliotheken beschikbaar en uitbreidbaar zijn, kunnen alternatieve verschalingsfuncties in ruimte en tijd eenvoudiger worden onderzocht.

OpenMI kan een belangrijke rol gaan spelen bij bijvoorbeeld de huidige ontwikkelingen naar één landelijk hydrologisch modelinstrumentarium, die nu speelt bij de kennisinstituten [zie elders in deze uitgave van Stromingen (Kroon e.a). Van der Velde et.al (2006) hebben een eerste stap naar een dergelijk instrumentarium met behulp van OpenMI gezet. OpenMI is de standaard die ervoor zorg kan dragen dat een 'landelijk instrument' voldoende flexibel blijft om deelmodellen te vervangen door verbeterde versies, of effecten van alternatieve modellen of modelconcepten te onderzoeken.

Implementatie van de OpenMI houdt meestal ook in dat de modelspecialist nauwer samen moet gaan werken met programmeurs. Veel huidige rekenkernen zijn in het verleden door de inhoudelijke specialist opgezet in Fortran. Tegenwoordig wordt 'Object georiënteerd' geprogrammeerd, een werkwijze waarvoor veel inhoudelijke specialisten bijgeschoold moeten worden, maar ook een werkwijze die door Fortran in mindere mate wordt ondersteund. De OpenMI functiebibliotheken zijn volledig Object georiënteerd opgezet, en geprogrammeerd in C#. Het is de vraag of de inhoudelijke specialisten zich dit eigen moeten maken: de ervaring in HarmonIT leert dat scheiding van taken tussen model en IT specialist gewenst is.

Binnenkort komt naast de .Net/C# implementatie ook een op Java gebaseerde versie beschikbaar.

Tenslotte: De laatste jaren staan de ontwikkeling en toepassing van modellen onder toenemende druk. De sector heeft dit ons inziens in de hand gewerkt door de grote hoeveelheid ontwikkelingen en gebrek aan samenwerking en consensus. De stap naar een nationaal hydrologisch modelinstrument is hier een uitvloeisel van. OpenMI biedt een goede mogelijkheid om zelf te onderzoeken welke modellen welke vragen het beste kunnen beantwoorden. Het zou mooi zijn als deze modellen vervolgens toegankelijk zijn, idealiter via open source.

Dankwoord

De auteurs willen Timo Kroon, Rijkswaterstaat RIZA bedanken voor zijn waardevolle bijdrage bij de totstandkoming van dit artikel.

Literatuur

Dirksen P.W., M.W. Blind, T. Bomhof en N. Srikrisnhudu (2005) 'Proof of Concept of OpenMI for Visual DSS Development', In Zerger, A. and Argent, R.M. (eds) MODSIM 2005 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand, December 2005, pp. 170-176. ISBN: 0-9758400-2-9, p184-189, http://www.mssanz.org.au/modsim05/papers/dirksen.pdf

OpenMI Document Series: Part B - Guidelines for the OpenMI (version 1.0)' http://www.openmi.org/documents/B_Guidelines.pdf.

Stoppelenburg, F., K. Kovar, M.J.H. Pastoors, A. Tiktak, en A. Leijnse (2002) 'Two-way coupling of 1D unsaturated-saturated flow model SWAP with 3-D saturated regional groundwater flow model LGM: Time-average coupling approach'.

STOWA (1997) 'Behoeftenonderzoek Consensusmodellenlijn – Hoofdrapport', STOWA rapport 1997-01, ISBN 90.74476.67.8, 26p + bijlagen.

Velde, Y. van der, H. Hakvoort en J. H. Hoogendoorn (2006) 'Een stap naar één modelinstrumentarium voor grond- en oppervlaktewater'; in: H_2O, jrg 39, nr 2, pag 47–49.

Vermulst, J.A.P.H. en W.J. de Lange (1999) 'An analytic-based approach for coupling models for unsaturated and saturated groundwater flow at different scales'; in: *Journal of Hydrology*, vol 226, pag 262–273, 199.

Wal, T. van der (1999) 'Architectuur voor een 'Standaard Raamwerk Water': toepasbaar bij modelstudies in het waterbeheer'; in: H_2O, jrg 32, pag 46–47.

Zanting, H.A., G. Baarse, P. Kouwenhoven, M.Taal, J.Tacke en S. Ooms (1997) 'Evaluatie PAWN-instrumentarium', Resource Analysis, EDS International BV in opdracht van Rijkswaterstaat RIZA,RA-97/281, 40p + bijlagen.