

Framework voor simulatiesoftware

Tonny Otjens

Q-Ray / Alterra – Group Software Engineering* - Telefoon 0317 – 474773 - otjens@q-ray.nl en otjens@sc.dlo.nl

Gebiedsstudies die door Alterra worden uitgevoerd worden steeds meer gekenmerkt door een integrale benadering. Onderzoeksvragen op het gebied van bijvoorbeeld waterbeheer of landgebruik moeten snel en betrouwbaar beantwoord kunnen worden. Dit vereist een flexibele en efficiënte inzet van modelapplicaties. De ontwikkelingen op het gebied van gebiedsstudies hebben geleid tot de ontwikkeling van een framework voor simulatieapplicaties. Integraal waterbeheer is daarbij als werkgebied genomen. Bij het onderzoek naar inrichting, beheer en afstemming van functies in de Groene Ruimte is water immers een cruciale factor. Het framework faciliteert het ontwikkelen en koppelen van simulatiemodellen, waarbij ook bestaande modelapplicaties kunnen worden opgenomen. Voor ontwikkeling van het framework is een objectgeoriënteerde aanpak gekozen. Implementaties van applicaties op basis van dit framework zijn uitgevoerd in Borland Delphi. Gebruik van het framework heeft geleid tot snellere ontwikkeling van simulatieapplicaties dankzij eenvoudige koppeling van bestaande simulatiemodellen en het gebruik van herbruikbare softwarecomponenten.

Ontwikkelingen op het gebied van gebiedsstudies

Bij het onderzoek dat binnen Alterra wordt uitgevoerd, is steeds vaker de integratie van meerdere simulatiemodellen vereist om de gestelde onderzoeksvragen te kunnen beantwoorden. Dit betekent dat een simulatieapplicatie dient te rekenen met invoergegevens die berekend worden door een andere simulatieapplicatie. Op deze manier kunnen complete modelketens worden gevormd. Het vormen van deze ke-

tens, dus het op elkaar aan laten sluiten van simulatieapplicaties, is meestal niet eenvoudig. Elk model verwacht vaak invoergegevens volgens een eigen bestandsformaat of database en elk model maakt gebruik van een eigen gebiedsschematisatie.

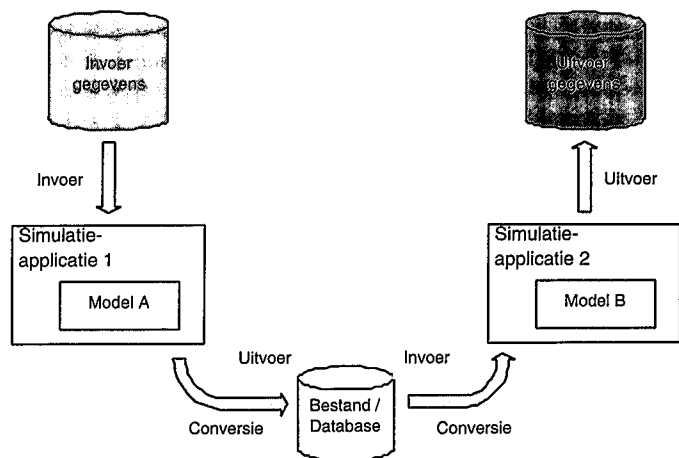
De ontwikkeling van gestandaardiseerde datamodellen heeft de complexiteit van het koppelen van simulatieapplicaties nauwelijks kunnen verminderen. Omdat elk model nog steeds uitgaat van een eigen gebiedsschematisatie zijn vertaalslagen nodig om modellen op elkaar aan te laten sluiten, ondanks een gemeenschappelijk bestandsformaat of datamodel. Daarnaast zijn in bestaande simulatieapplicaties de modellen (rekenkernen) en generieke functionaliteiten als datamanagement en visualisatie vaak te veel met elkaar verweven om eenvoudig te kunnen delegeren aan andere modules in een procesketen.

Om sneller een modelinstrumentarium te kunnen ontwikkelen voor gebiedsstudies is een framework ontwikkeld waarmee het mogelijk is om simulatiemodellen eenvoudiger te laten samenwerken. Door gebruik te maken van een framework kan applicatieontwikkeling sneller worden uitgevoerd. De generieke tools vormen hierbij herbruikbare softwarecomponenten, wat extra tijdswinst in het ontwikkeltraject oplevert.

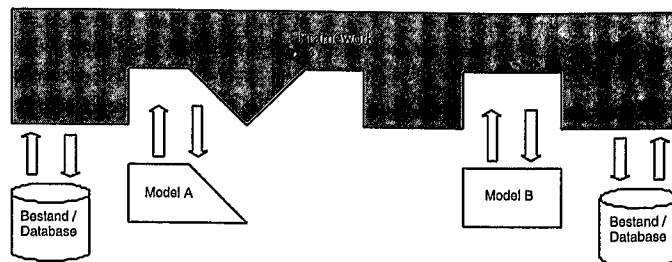
Bij de koppeling van modellen en datasets met behulp van een framework verloopt de communicatie niet direct tussen de modellen en/of datasets, maar via het framework. Door te definiëren aan welke interfaces modellen, datasets, visualisatiecomponenten, etc. moeten voldoen hoeft slechts eenmalig een communicatiestructuur ontwikkeld te worden. Het ontwikkelen van simulatieapplicaties komt vervolgens neer op het ontwikkelen en hergebruiken van bouwstenen die de functionaliteit van modellen en generieke tools bezitten.

Ontwikkeltraject

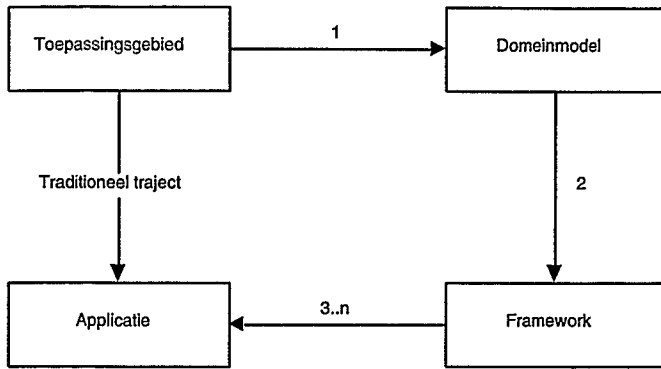
Het proces dat is gevolgd bij de ontwikkeling van het framework is weergegeven in figuur 2. Het toepassingsgebied is als uitgangspunt genomen. Het proces verloopt vervolgens met de klok mee tot uiteindelijke simulatieapplicaties. Bij de ontwikkeling van het framework is allereerst een domeinmodel ge-



Figuur 1a – Traditionele koppeling van simulatieapplicaties tot een modelketen



Figuur 1b – Koppeling van simulatiemodellen en datasets met behulp van een framework



Figuur 2 - Procescyclus

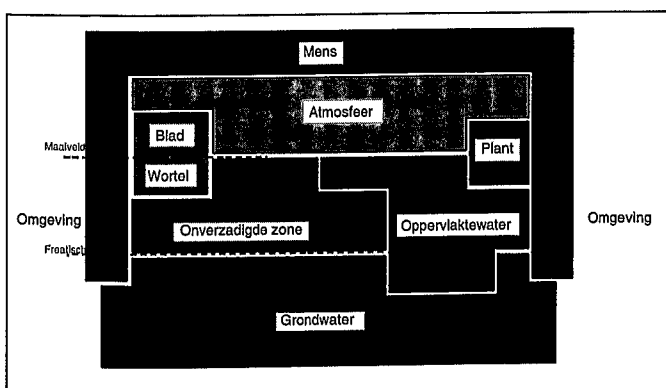
maakt van het toepassingsgebied Integraal Waterbeheer. Vervolgens is het domeinmodel uitgewerkt tot een ontwerp voor het framework. Uiteindelijke applicatieontwikkeling vindt vervolgens plaats op basis van het framework. De directe relatie tussen toepassingsgebied en applicatie geeft het 'traditionele' traject weer van de ontwikkeling van simulatieapplicaties. Het is ook denkbaar dat vanuit bestaande applicaties wordt teruggewerkt naar een framework en domeinmodel. Een dergelijk traject leidt echter zelden tot een voldoende generiek framework.

Domeinanalyse

Het onderstaande diagram beschrijft het domein 'Modellering voor Integraal Waterbeheer'. Het conceptuele model (figuur 3) geeft aan uit welke onderdelen (sub-domeinen) het domein integraal waterbeheer is opgebouwd. De raakvlakken tussen de sub-domeinen geven aan waar interactie plaats kan vinden. Voor elk sub-domein is een interface gedefinieerd. Zo biedt de interface van het sub-domein Atmosfeer methodes waarmee bijvoorbeeld neerslaggegevens zijn te verkrijgen. Elk sub-domein kan ingevuld worden door een softwarecomponent dat gebruik maakt van één of meerdere simulatiemodellen. Omdat de softwarecomponenten volgens de standaard interfaces van de sub-domeinen worden ontwikkeld is eenvoudige uitwisseling van deze componenten mogelijk.

Ontwerp van het framework

Het ontwerp van het framework is een verdere uitbreiding van het conceptuele model. Het ontwerp kan grofweg verdeeld



Figuur 3 - Conceptueel model van het domein Integraal Waterbeheer

worden in twee onderdelen: een statische beschrijving en een beschrijving van de dynamiek van het framework. Voor de beschrijvingen is gebruik gemaakt van de Unified Modeling Language (UML), de standaard notatiewijze voor objectgeoriënteerde architecturen. Het statische deel van het ontwerp geeft een beschrijving van alle klassen binnen het framework, de interfaces behorende bij deze klassen en de relaties tussen de klassen. Hiervoor is gebruik gemaakt van *class diagrams*. Het dynamische deel van het ontwerp geeft aan op welke manier de klassen met elkaar samenwerken. Hiervoor is vooral gebruik gemaakt van *sequence diagrams* en *collaboration diagrams*, ook standaard diagrammen binnen UML.

Design patterns

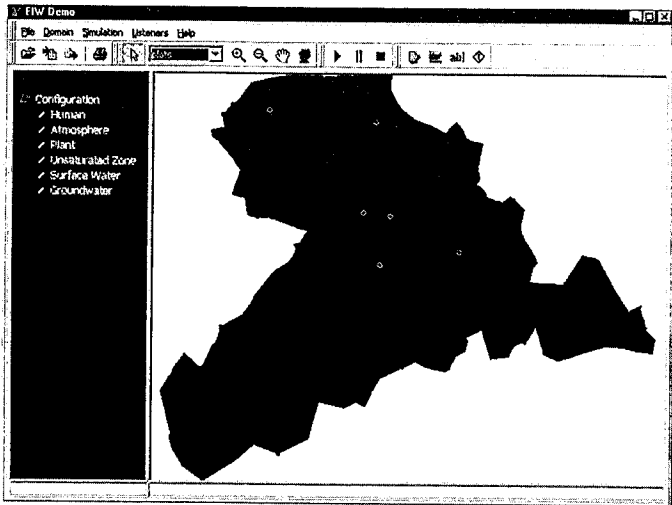
Bij het ontwerp van het framework is gebruik gemaakt van een groot aantal design patterns. Een pattern is een beschrijving van een algemene oplossing voor een veel voorkomend ontwerpprobleem. Patterns zijn meestal domein-, problemen en programmeertaal onafhankelijk. Een design pattern kan voorgesteld worden als een groep klassen met een beschrijving van de diensten per klasse en de manier waarop de klassen samenwerken. Een pattern geeft een op ervaring gebaseerd advies voor het komen tot een goede oplossing voor een bepaald ontwerpprobleem. Uiteraard zijn de toegepaste patterns nader ingevuld voor de specifieke toepassing binnen het framework (een pattern is immers een algemene oplossing). Binnen het framework is onder andere gebruik gemaakt van patterns voor het benaderen van externe gegevensbronnen (databases, bestanden), het werken met verzamelingen, het maken van hiërarchieën en het gebruik van visualisatiecomponenten.

Hergebruik van bestaande simulatiemodellen

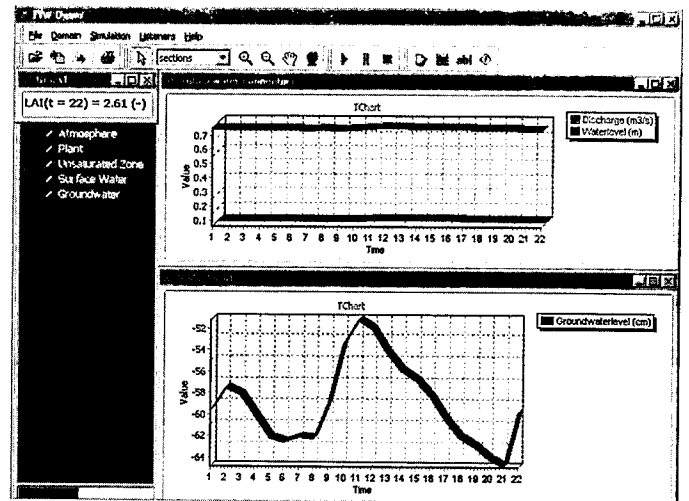
Binnen Alterra wordt op dit moment gebruik gemaakt van een groot aantal intern en extern ontwikkelde simulatieapplicaties. De ontwikkeling van het framework kan uiteraard niet tot gevolg hebben dat binnen korte tijd deze programma's worden herontworpen volgens de frameworkconcepten. De mate van gebruik van het framework binnen Alterra is dus erg afhankelijk van de mogelijkheid om bestaande applicaties op te nemen in een framework applicatie. Er is daarom een techniek ontwikkeld om bestaande applicaties in te pakken in een zogenaamde wrapper. Het framework gaat ervan uit dat een simulatiemodel een bepaald interface bezit. De wrapper zorgt voor de vertaalslag van applicatieinterface naar het interface dat het framework verwacht omdat bestaande applicaties deze interface niet bezitten. Deze techniek is reeds voor meerdere simulatieprogramma's toegepast. Voor het ontwikkelen van de wrappers is een tool gemaakt waardoor circa 90% van de wrapper-code kan worden gegenereerd (op dit moment alleen Delphi code).

Testapplicatie

In 1998 is een testapplicatie ontwikkeld waarbij bestaande modellen voor het sub-domein Onverzadigde Zone en het sub-domein Oppervlaktewater zijn opgenomen in één simulatieapplicatie. Voor het proefgebied Winterswijk (zie kaart in figuur 3) kunnen door de applicatie onder andere grondwaterstan-



Figuur 3 – Framework testapplicatie



Figuur 4 – Presentatie simulatieresultaten tijdens de simulatierun

den en waterafvoer berekend worden, rekening houdend met weersinvloeden, bodem-, gewas- en waterafvoereigenschappen. Karakteristiek voor de testapplicatie is ook de mogelijkheid om tijdens een simulatierun reeds resultaten te presenteren (figuur 4) zodat de gebruiker ook tijdens de simulatie in kan grijpen in bijvoorbeeld instellingen van pompen of stuwten.

Naast de testapplicatie zijn op dit moment een tweetal nieuwe applicaties op basis van het framework in ontwikkeling. In deze applicaties wordt onder andere de embedding van GIS-functionaliteiten verder uitgewerkt.

Impact

De veranderingen door de ontwikkeling en het gebruik van een framework zijn niet te vergelijken met het introduceren van een nieuwe ontwikkel tool of het inrichten van een code-library. Het framework heeft niet alleen invloed op de structuur van de uiteindelijke applicaties en hergebruik van componenten maar op het gehele software ontwikkelproces. Het proces vereist een zeer intensieve samenwerking tussen software engineers en domeindeskundigen, gedurende het gehele

traject. Het framework faciliteert in eerste instantie de applicatieontwikkeling. Het op die juiste manier incorporeren van simulatiemodellen maakt de input van domeindeskundigen echter onmisbaar. Door goede aansluiting bij het hele domein faciliteert het framework eenvoudig uitbreidingen en aanpassingen wat tot beter onderhoudbare applicaties leidt. Onderzoekers kunnen eenvoudig nieuwe applicaties ontwikkelen met verhoogde kwaliteit en inzetbaarheid.

Literatuur

- Florijn, G.H. (1998). Object-oriëntatie – Op weg met SERC. Whitepaper SERC. Utrecht.
- Gamma, E., R. Helm, R. Johnson and J. Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. Reading.
- Wal, T. van der (1998). Kennisintensieve systemen in onderzoek. Agro Informatica jaargang 9, nr. 3. Wageningen.

* Group Software Engineering (GSE) is een samenwerkingsverband tussen Alterra en Q-Ray B.V. en werkt in projectverband aan opdrachten voor DLO en externe opdrachtgevers.