

Projectplan migratie DUFLOW functionaliteit naar SOBEK

jagers

Voorlopig

Opdrachtgever:
STOWA, Deltares

**Projectplan migratie DUFLOW
functionaliteit naar SOBEK**

jagers

Rapport

september 2008

| | | | | | | | | |
|---|--|-----------|--|---------------------|--------|--|-------------|--|
| Opdrachtgever | STOWA, Deltares | | | | | | | |
| Titel | Projectplan migratie DUFLOW functionaliteit naar SOBEK | | | | | | | |
| Samenvatting | | | | | | | | |
| <p>Dit projectplan omvat een beschrijving van doel, uitgangspunten en randvoorwaarden voor de migratie van DUFLOW functionaliteit naar SOBEK om tot een uniform nationaal 1D instrumentarium voor oppervlaktewateren te komen. Er wordt een beknopt overzicht van reeds geïnventariseerde gebruiksaspecten gegeven alsmede een korte beschrijving van de nieuwe werkomgeving voor 1D (en later meerdimensionale) modellering. Vervolgens worden een viertal aandachtspunten geïdentificeerd die (buiten de user interface functionaliteit) in het migratietraject van belang (kunnen) zijn voor de huidige DUFLOW gebruikers. Tenslotte wordt kort ingegaan op organisatie, fasering en financiering van de ontwikkelingen.</p> | | | | | | | | |
| Referenties | | | SOBEK-DUFLOW overleg sinds 2005 tussen DUFLOW gebruikers, STOWA en WL Delft Hydraulics | | | | | |
| Ver | Auteur | | Datum | Opmerk. | Review | | Goedkeuring | |
| 1 | jagers, noort | | februari 2008 | eerste versie | | | | |
| 2 | jagers | | september 2008 | planning+finance | | | | |
| 2.1 | jagers | | september 2008 | herformulering fin. | | | | |
| | | | | | | | | |
| Projectnummer | | | | | | | | |
| Trefwoorden | | | | | | | | |
| Aantal bladzijden | | 17 | | | | | | |
| Classificatie | | Geen | | | | | | |
| Status | | Voorlopig | | | | | | |
| <p>Dit document is een voorlopig rapport, niet een definitief rapport, en uitsluitend bedoeld voor discussiedoeleinden. Aan de inhoud van dit rapport kunnen noch door de opdrachtgever, noch door derden rechten worden ontleend.</p> | | | | | | | | |

Inhoud

| | | |
|----------|--|-----------|
| 1 | Inleiding | 1 |
| 2 | Doel, uitgangspunten en randvoorwaarden | 2 |
| | 2.1 Doel..... | 2 |
| | 2.2 Uitgangspunten..... | 2 |
| | 2.3 Randvoorwaarden | 3 |
| 3 | Gebruiksaspecten | 4 |
| | 3.1 DUFLOW voordeel punten | 4 |
| | 3.2 Algemene verbeterpunten | 5 |
| 4 | Algemene beschrijving nieuwe werkomgeving | 7 |
| | 4.1 Keuze voor DelftShell | 7 |
| | 4.2 Algemene beschrijving GUI | 8 |
| | 4.3 Algemene architectuur en ontwerp..... | 11 |
| | 4.3.1 Uitgangspunten nieuwe ontwikkelingen User Interface..... | 12 |
| | 4.4 Kernbegrippen | 12 |
| 5 | Specifieke componenten uit DUFLOW omgeving | 14 |
| | 5.1 TEWOR functionaliteit | 14 |
| | 5.2 Waterkwaliteitsprocesencompiler..... | 14 |
| | 5.3 Meerdere landgebruiktypes in één RR / RAM knoop..... | 14 |
| | 5.4 Importeren van DUFLOW schematisaties..... | 14 |
| 6 | Organisatie | 15 |
| | 6.1 Projectbeheersing | 15 |
| | 6.2 Kwaliteitsborging..... | 15 |
| | 6.3 Gebruik van projectresultaten..... | 16 |
| 7 | Planning en financiering | 17 |

1 Inleiding

Sinds 2005 hebben STOWA en WL | Delft Hydraulics op verschillende niveaus in de organisaties gesprekken gevoerd over de mogelijkheid om te komen tot integratie van DUFLOW en SOBEK. De wens om tot integratie over te gaan werd mede ingegeven door de algemene wens om te komen tot één nationaal modelinstrumentarium zoals besproken in het onderwateroverleg; daarnaast spelen kennisbundeling en kostenbesparing een rol.

De globale verkenningen hebben geleid tot de conclusie dat aanpassing van de SOBEK werkomgeving met een aantal specifieke gebruikerswensen vanuit de DUFLOW gebruikers de voorkeur heeft. Op basis van dit idee is in 2006 een verkenning uitgevoerd naar de mogelijkheden om te komen tot een nieuwe werkomgeving, waarbij op software technische gronden de voorkeur werd uitgesproken voor nieuwbouw waarbij – waar mogelijk – gebruik zal worden gemaakt van bestaande componenten. Parallel aan deze discussie liep binnen WL | Delft Hydraulics een verkenning om te komen tot een geïntegreerd 1D-2D-3D instrumentarium op basis van SOBEK en Delft3D; dit heeft in 2007 geleid tot een eerste aanzet voor een nieuwe modulaire werkomgeving genaamd DelftShell waarbij rekening is gehouden met de globale wensen van een mogelijk integratietraject van SOBEK en DUFLOW.

Met de vorming Deltares is begin dit jaar een principeafspraken gemaakt tussen STOWA en Deltares over de uitfasering van DUFLOW per 31 december 2009; ervan uitgaande dat voor die datum de relevante DUFLOW functionaliteit gemigreerd is naar de nieuwe SOBEK omgeving en als zodanig voor alle SOBEK en DUFLOW gebruikers onder maintenance beschikbaar komt. De STOWA levert een financiële bijdrage aan de kosten voor de hiervoor benodigde ontwikkelingen; het voorliggend document dient om de globale lijnen van deze ontwikkelingen aan te geven op basis waarvan een definitief contract ten aanzien van de migratie van DUFLOW functionaliteit tussen de beide organisaties kan worden opgesteld. Hoofdstuk 7 omvat een update van de planning en een globale indicatie van de financiering.

2 Doel, uitgangspunten en randvoorwaarden

2.1 Doel

Het primaire doel van het DUFLOW migratie project is het zorgdragen voor de aanpassing van de SOBEK omgeving zodanig dat deze de inhoudelijke functionaliteit van DUFLOW en de functionaliteit van de DUFLOW werkomgeving afdekt. Wat betreft de aanpassing van de werkomgeving vormt deze ontwikkeling echter een integraal onderdeel van de algehele vernieuwing van de werkomgeving van de Deltares softwaresystemen voor watergerelateerde vraagstukken. De volgende paragrafen en de hoofdstukken 3 en 4 zijn geformuleerd vanuit dit breder perspectief. Hoofdstuk 5 gaat in op de voorziene activiteiten op het gebied van de inhoudelijke functionaliteit.

Het primaire doel van de interface activiteiten is het ontwikkelen van een generiek raamwerk voor de user interfaces. Binnen het nieuwe raamwerk zullen de komende jaren de software modules van Delft3D, SOBEK (inclusief DUFLOW functionaliteit), Habitat en de verschillende verkenners beschikbaar komen. De nieuwe geïntegreerde werkomgeving zal ten minste hetzelfde gebruiksgemak gaan bieden als de huidige afzonderlijke interfaces. Het ontwikkelen van een dergelijk raamwerk vergt een iteratieve procedure met nauwe betrokkenheid van de eindgebruikers. Raamwerk en interface componenten voor de afzonderlijke modules zullen in parallel ontwikkeld worden; hierbij zal de interface voor SOBEK (lees 1D instrumentarium) in eerste instantie het verst worden ontwikkeld zonder generalisatie naar 2D en 3D toepassingen uit het oog te verliezen.

2.2 Uitgangspunten

Als uitgangspunten voor de ontwikkeling van het raamwerk en de daarin werkzame specifieke user interface plugins dienen de volgende punten:

- 1 De opzet moet zoveel mogelijk aansluiten bij de huidige stand van de techniek.
- 2 Bij de ontwikkeling van de nieuwe user interface wordt gebruik gemaakt van Extreme Programming. Kenmerken van deze ontwikkeltechniek zijn de intensieve betrokkenheid van de gebruikers en testgedreven ontwikkelen. Een beschrijving van deze techniek staat in paragraaf 6.1.
- 3 DelftShell wordt als uitgangspunt genomen voor de ontwikkeling van de nieuwe User Interface.
- 4 Waar mogelijk wordt gebruik gemaakt van good software development practices.
- 5 Waar mogelijk wordt gebruik gemaakt van open-source bibliotheken en tools.

Zoals hierboven aangeven zal DelftShell als basis dienen voor de nieuwe ontwikkelingen. De basis voor DelftShell is gelegd in 2007 door voort te bouwen op de recent ontwikkelde Habitat interface. Habitat is nu beschikbaar als een plug-in die binnen de eerste versie van DelftShell werkt. DelftShell is nog niet voldoende uitgewerkt om ook modellen als SOBEK en Delft3D in op te kunnen nemen. Zo ontbreken er bijvoorbeeld editors voor verschillende typen data en bepaalde presentatiemogelijkheden; in 2008 zal op dit vlak een belangrijke stap voorwaarts gemaakt worden.

2.3 Randvoorwaarden

Aan het nieuwe raamwerk worden de volgende eisen gesteld:

- 1 Het raamwerk moet ook werken onder Linux (vergelijkbaar met de huidige Delft3D user interfaces).
- 2 De onderhoudbaarheid van raamwerk en interface componenten moet verbeteren ten opzichte van de huidige systemen.
- 3 De performance van de user interface moet zodanig zijn, dat ook bij grote modellen het systeem werkbaar is. Hiervoor moet in samenspraak met gebruikers responstijden voor bepaalde acties worden afgesproken.
- 4 Integratie met GIS moet mogelijk zijn. Hierbij moet de user interface als plug-in in een systeem als ArcGIS kunnen worden geladen.
- 5 De reeds ontwikkelde GIS plug-ins (Delft3D preprocessing, online-visualisatie, emissiemodule en TRITON interfaces) moeten binnen het raamwerk gebruikt kunnen worden.
- 6 De user interface moet multi-document ondersteunen. Dit betekent dat verschillende cases of scenario's tegelijkertijd geopend moeten kunnen zijn. In- en uitvoer zijn ook tegelijk benaderbaar.
- 7 Van verschillende cases of scenario's moeten de simulatieresultaten in één grafiek getoond kunnen worden (case analysis functionaliteit).
- 8 Internationale user interface (flexibele taal en eenheden).
- 9 Open voor OpenMI-compatibele simulatiecomponenten van derden.

3 Gebruiksaspecten

Specifiek voor de migratie van DUFLOW functionaliteit naar SOBEK dient er aandacht te zijn voor de huidige DUFLOW gebruikers. Op basis van een gebruikersbijeenkomst van juni 2006 is in de volgende paragraaf een lijst met punten opgenomen waarin de huidige DUFLOW interface voordelen biedt ten opzichte van de huidige SOBEK interface. Voor de overzichtelijkheid zijn de punten waarop de SOBEK omgeving voordelen biedt niet in dit plan opgenomen (zie daarvoor de oorspronkelijke memo).

Hierbij dient opgemerkt te worden dat in het migratietraject slechts bestaande functionaliteit overgezet hoeft te worden; waar mogelijk zullen algemene verbeterwensen worden meegenomen. Ook is het mogelijk dat bestaande functionaliteit wordt vervangen door een andere functionaliteit die de gebruikerswensen beter dekt.

3.1 DUFLOW voordeel punten

Tijdens een gebruikersbijeenkomst in juni 2006 (waarbij alleen deelnemers van de ingenieursbureaus aanwezig waren) zijn de volgende punten genoemd waarop de huidige DUFLOW omgeving voordelen biedt ten opzichte van de huidige SOBEK omgeving.

- 1 Op basis van gebruiksgemak heeft de implementatie van de TEWOR toets in DUFLOW heeft de voorkeur boven die in SOBEK
- 2 Werken vanuit één scherm met een niet lineaire werkwijze (ie gelijktijdig invoer en uitvoer kunnen bekijken en editen) leidt tot een snellere calibratie en/of ontwerp cyclus.
- 3 Snel kunnen selecteren van waterkwaliteitsprocessen (overzichtelijkheid van selectieprocedure / predefined sets) en de mogelijkheid om de gebruikte formuleringen te kunnen editen
- 4 De multiple data-editor is toepasbaar op *alle* objecten.
Extra wens: data kunnen editen per deelgebied, nu kan dat alleen per type.
- 5 De implementatie van de export naar GIS voor allerlei (ruimtelijke) analyses (zoals het bepalen van drooglegging) werkt in DUFLOW prettiger.
Extra wens: Export van geometrie en (bulk) modeldata. In het algemeen wordt de voorkeur gegeven aan goede mogelijkheden voor *export naar* standaard analyse tools (zoals GIS en Excel)
- 6 Het vergelijken van een hele rits scenario's / cases.
Gewenst is de mogelijkheid om binnen een case resultaten van bijvoorbeeld een node te vergelijken met resultaten uit een andere case (variant/ontwerp oid). In de sideview kun je al wel een lijn van een andere case inlezen. Case analysis tool (CAT) kan hierbij helpen, maar een goede export naar standaard tools heeft hogere prioriteit (# 5).
- 7 Goede controle over de visualisatie (bijv. zoals presentatie scripts van DUFLOW).
- 8 Uitvoer:
- breedte op waterlijn weergeven als geografische breedte

- de waterbreedte en het wateroppervlak als modeluitvoer
 - volumeberekening van de resterende berging in oppervlakte- en grondwater in de tijd.
- 9 Het eenvoudig kunnen aan- en uitzetten van verschillende objecten.
 - 10 Functionaliteit van de PEST tool van DUFLOW
 - 11 Watergangen samenvoegen. (Het splitsen van watergangen met een harde knoop kan in beide pakketten)
 - 12 De invoer verdient aandacht t.a.v. de volgende punten:
 - invoer van controllers
 - randvoorwaarde knoop in RR uitbreiden met debiet (nu alleen peil mogelijk)
 - pompknoop in CF niet regelen met dag en nacht ritme en vergelijkbaar maken RR en CF knopen
 - automatisch laten berekenen wat het oppervlak open water is in de ingevoerde waterlopen en gebruiken als oppervlak bij de berekening hoeveel water er valt op de watergangen.
 - 13 Wens: rond structures automatisch rekenpunten genereren op een afstand van x m.
 - 14 Het kopiëren (opslaan als / open as new) vanuit SOBEK is veel beter geregeld dan in DUFLOW. Extra wensen:
 - het overzichtelijk kunnen kopiëren van cases met keuze in het meenemen van bepaalde data.
 - een 'Save as' hetzelfde als in windows waarbij je de oude naam kan wijzigen en niet helemaal opnieuw hoeft te schrijven
 - het liefst zoveel mogelijk de bestanden van een scenario / case in één directory.
 - 15 Modelsysteem zou meer ondersteuning kunnen bieden bij analyses van schematisatie. Bijvoorbeeld: controles op duiker boven maaiveld of onder bodem, mogelijk al tijdens import. De scripting mogelijkheid van DUFLOW biedt deze functionaliteit.
 - 16 Mogelijkheid om taklengte zelf hard in te stellen ongeacht lengte op kaart

Het grootste deel van bovenstaande punten worden in het nieuw te ontwikkelen user interface meegenomen. In deze lijst staan echter twee punten waaraan in de context van de migratie van DUFLOW functionaliteit extra aandacht moet worden besteed. Het gaat hierbij om punten 1 (TEWOR functionaliteit) en 3 (waterkwaliteitsprocessencompiler).

3.2 Algemene verbeterpunten

Tijdens de gebruikersbijeenkomst van juni 2006 kwamen ook nog enkele algemene verbeterpunten aan de orde; deze zijn hieronder samengevat.

- 1 Reaches worden nu weergegeven door rechte lijnen terwijl in vectormode de werkelijke loop zichtbaar is. De werkelijke loop zou altijd zichtbaar moeten zijn: afschaffen rechte verbindingen.

- 2 Water- en stoffenbalansen zou je ook voor clusters van segmenten moeten kunnen bepalen.
- 3 Bestaande modellen moeten geïmporteerd kunnen worden in de nieuwe omgeving. De import optie van DUFLOW modellen in SOBEK is nog niet robuust en volledig.
- 4 Optie om “doorlopende” bodem in culverts te krijgen (zoals bij bridges).
- 5 Het ontbreken van een undo-button in het editor gedeelte (of misschien is een undolijst nog mooier) is wellicht een goede aanvulling. Selecteren en deleten van delen van het netwerk moet mogelijk worden met undo functie.
- 6 Foutmeldingen geografisch georiënteerd conform TUFLOW (<http://www.tufLOW.com/>)
- 7 Mogelijkheid tot het simpel opruimen van de uitvoer van een case (vergelijkbaar met SOBEK RE).
- 8 Documentatie loopt soms achter, is onvolledig of fout (bijv. bij een “gele knoop met berging”); deze punten zijn alleen duidelijk na enige ervaring met het systeem opgedaan te hebben.
- 9 Eenvoudige importmogelijkheden vanuit en exportmogelijkheden naar GIS.

4 Algemene beschrijving nieuwe werkomgeving

In 2007 is een nieuwe grafische user interface DelftShell ontwikkeld voor oa. HABITAT en de Delta-Verkenner. Het is opgezet als een open systeem dat kan worden uitgebreid met nieuwe software componenten, nieuwe data typen en graphical user interface views of editors door het gebruik van plug-ins.

4.1 Keuze voor DelftShell

Keuze voor een framework

In het algemeen geldt dat de keuze voor het ontwikkelen van een user interface met behulp van een generiek raamwerk als voordelen heeft dat:

- 1 de look and feel van de user interface voor de verschillende software modules gelijk blijft.
- 2 hergebruik van verschillende generieke elementen zo veel mogelijk gegarandeerd is.
- 3 het beheer en onderhoud vereenvoudigt door de keuze voor een modulaire opzet van de software. Voorwaarde is wel dat de verschillende componenten zoveel mogelijk onafhankelijk van elkaar worden ontwikkeld.

De consequentie van het gebruik van een raamwerk is dat de complexiteit van de software op het eerste gezicht toeneemt, zeker als er maar één module in de beschouwing wordt meegenomen. De winst ontstaat als er meerdere modules worden ingebouwd, waardoor generieke delen hergebruikt worden.

Een ander aandachtspunt bij het gebruik van generieke raamwerken is dat de gebruikte concepten en terminologie dusdanig abstract kunnen worden dat deze niet meer aansluiten bij de belevingswereld van de gebruiker. Het projectteam denkt dit gevaar te kunnen neutraliseren door de intensieve inbreng van gebruikers, waardoor de aansluiting op de belevingswereld van de gebruiker moet worden geborgd.

Keuze voor DelftShell

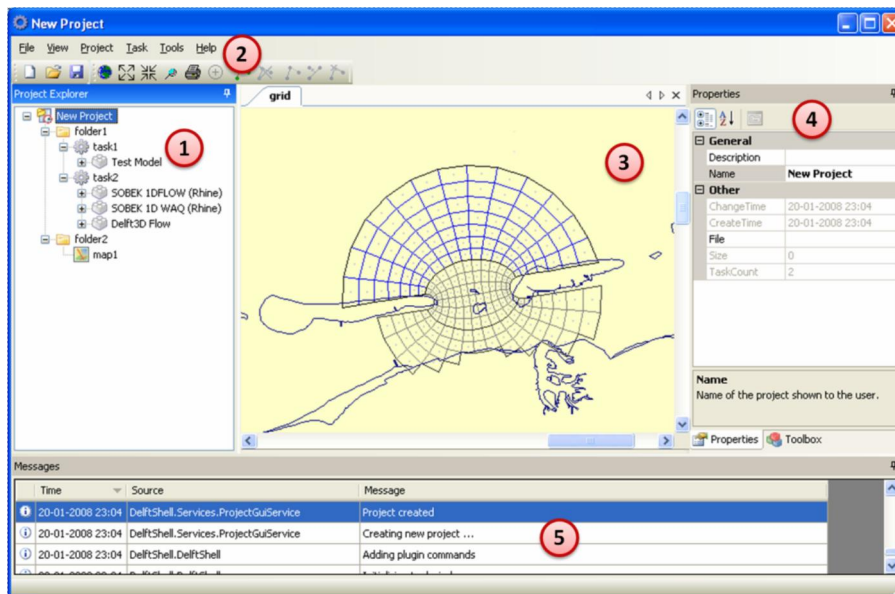
DelftShell wordt als uitgangspunt genomen voor de ontwikkeling van de nieuwe user interface voor de numerieke modellen. Redenen hiervoor zijn:

- 1 DelftShell sluit aan bij de nieuwste ontwikkelingen op IT gebied en voldoet daardoor prima als vernieuwingsslag voor de user interface.
- 2 DelftShell is modulair opgezet waarbij zowel componenten als gebruikersschil vervangen kunnen worden. Hierdoor is zowel stand-alone gebruik als integratie met ArcGIS mogelijk.
- 3 DelftShell heeft een geïntegreerd werkblad conform DUFLOW dat aansluit bij de gebruikerswensen.
- 4 DelftShell is primair opgezet als een generiek raamwerk voor numerieke modellen, de bestaande basisfunctionaliteit zal naar verwachting snel uitgebreid kunnen worden voor numerieke modellen zoals Delft3D, SOBEK en MODFLOW.

- 5 In 2007 is binnen DelftShell de uitbreiding geïmplementeerd om de bestaande Delft3D en emissiemodule plug-ins ook te kunnen laden.
- 6 Met behulp van Mono kan de code ook worden gecompileerd onder LINUX. Aangezien Mono per definitie altijd iets achterloopt op Microsoft Visual Studio, bestaat het gevaar dat nieuwe functionaliteit die in MVS beschikbaar komt niet direct kan worden gebruikt.

4.2 Algemene beschrijving GUI

DelftShell is ontwikkeld als een top-level grafische user interface waarin meerdere applicaties op een eenduidige manier kunnen worden geïntegreerd. Het concept van de DelftShell GUI is geschetst in Figuur 1. De hoofdonderdelen van de User Interface zijn:



Figuur 1 Look & feel van grafische user interface DelftShell

1 Project Explorer

Projecten en de onderdelen van projecten zoals de taken en modellen worden gepresenteerd in de project explorer. De project explorer toont een hiërarchische presentatie van een project. Een gebruiker kan objecten in de project explorer selecteren en vervolgens een actie op uitvoeren.

2 Menu bar and Tool bar

De shell implementeert een aantal basiscommando's en stelt die beschikbaar via menu's en/of knoppenbalk. Daarnaast biedt het framework de mogelijkheid aan plug-ins om menu's en knoppenbalk uit te breiden. Welke commando's op een moment beschikbaar zijn voor de gebruiker, wordt bepaald door de plug-in en werkblad die op dat moment actief zijn.

3 Werkblad

Primaire weergave van het document. Het werkblad kan meerdere views of editors bevatten die eventueel ook naast elkaar getoond kunnen worden. Het werkblad is het onderdeel van de GUI dat typisch niet ingevuld zal worden door

het framework zelf maar door de plug-ins. Het werkblad is het onderdeel waar de gebruiker primair data zal editen en waar resultaten getoond kunnen worden. Het is mogelijk meerdere werkbladen tegelijk te openen (naast elkaar of via tab achter elkaar). Er is er altijd maar één tegelijk actief (focus muis en toetsenbord, beschikbare commando's in menu's). Bij een werkblad kunnen ook meerdere additionele vensters horen; voorbeeld een lagenlijst bij een mapcontrol.

4 Property Editor

Elk project bestaat uit een verzameling van objecten, meestal dataitems. De properties van deze objecten zijn via reflectie uit te vragen en hiervan gebruik makend is het mogelijk om elk type object te editen in een generieke property editor.

5 Log venster

Shell en plug-ins kunnen voortdurend berichten genereren. Deze berichten worden verwerkt door de logger. De logger kan de berichten bijvoorbeeld opslaan in een bestand of op het scherm tonen in het logvenster. De logger kan onderscheid maken tussen verschillende soorten berichten; bijvoorbeeld error, warning, etc.

De beschreven onderdelen van de shell zijn sterk aan elkaar gekoppeld. Voorbeelden:

- Via een keuze in de project explorer wordt een nieuw werkblad geopend.
- Selecteren van een object in de project explorer of op het werkblad leidt tot het tonen van de properties van dat object in de property editor.
- De inhoud van de toolbox is afhankelijk van het actieve werkblad.
- De getoonde en beschikbare menu's en knoppenbalken zijn afhankelijk van het actieve werkblad.
- Functies die worden opgestart in de project explorer of het werkblad hebben uitvoer tot gevolg in het logvenster.

Hierna volgen enkele schermafdrucken die de functionaliteit van bestaande applicaties op basis van DelftShell tonen.



Figure 1 Voorbeeld van DelftShell als Delta-Verkenner

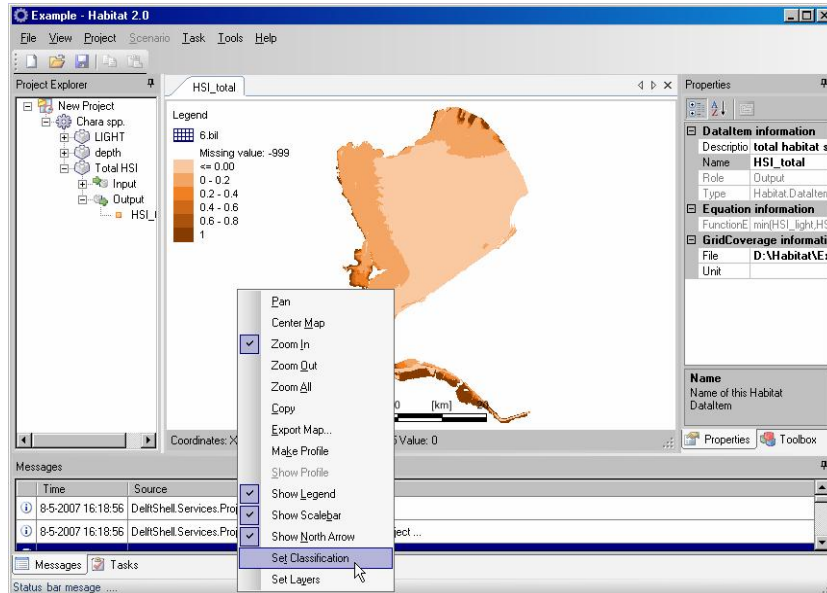


Figure 2 Voorbeeld van DelftShell als HABITAT ruimtelijke analyse tool

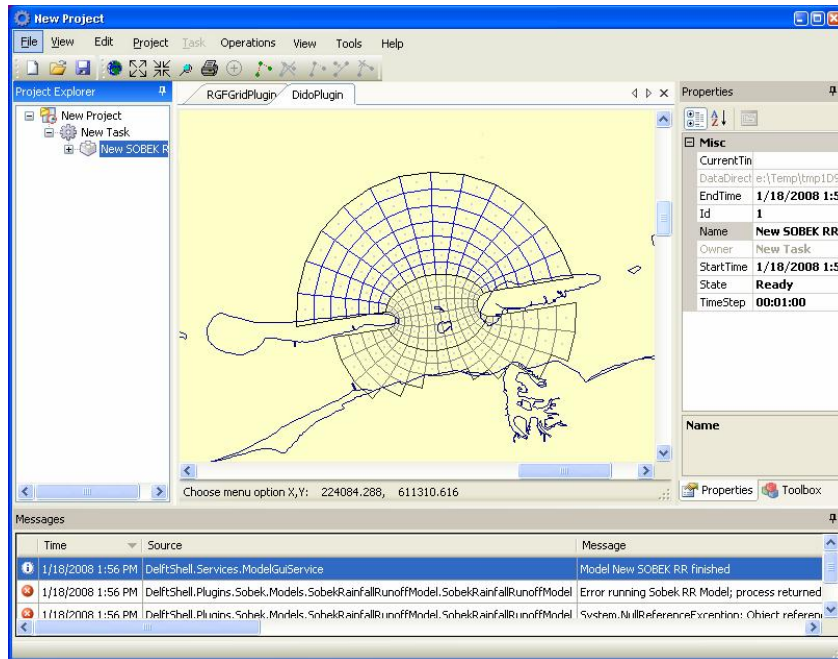
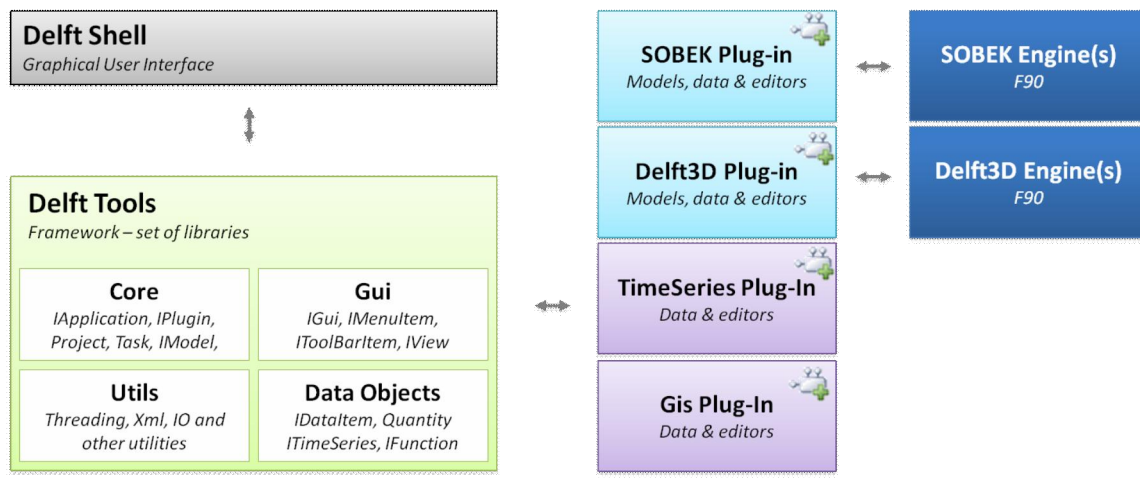


Figure 3 Voorbeeld van DelftShell als RGFRID

4.3 Algemene architectuur en ontwerp

De Grafische User Interface is slechts één van de componenten van het DelftShell raamwerk. Teneinde het systeem flexibel te maken was het ontwerp van een verzameling libraries als software framework voor DelftShell noodzakelijk. Het idee achter dit framework is een set van basis programmeer interfaces voor de verschillende onderdelen van de software te definiëren.

De IPlug-in interface is bijvoorbeeld noodzakelijk om het mogelijk te maken de User Interface uit te breiden met nieuwe functionaliteit. Op een zelfde wijze is een set interfaces gedefinieerd die de integratie van nieuwe modellen, nieuwe data types en nieuwe viewers en editors mogelijk maakt. Al deze class libraries zijn gecombineerd in een software framework genaamd DelftTools en gebruikt bij de ontwikkeling van DelftShell. Figuur 2 vat de huidige architectuur samen. Een aantal van de getoonde model- en data-plugin-ins zijn alleen opgenomen ter illustratie van het concept en nog niet geïmplementeerd. De bouw hiervan is één van de doelen van dit project.



Figuur 2 Hoofd componenten van DelftShell en DelftTools

4.3.1 Uitgangspunten nieuwe ontwikkelingen User Interface.

De in het voorgaande geschetste architectuur met shell en plug-ins is de basis voor verdere ontwikkelingen. Door de omvang van de shell en plug-ins is het gevaar dat het geheel te complex en ondoorzichtig wordt erg groot. Veel aandacht zal dan ook besteed moeten worden aan het ontwikkelen van eenduidige bedieningsscenario's. Plug-ins zullen zich moeten conformeren aan eenzelfde UI standaard.

De filosofie die ten grondslag ligt aan moderne GUI's is de directe manipulatie (DM). Dit betekent dat de gebruiker acties direct op zichtbare objecten uitvoert. Bijvoorbeeld op de eigenschappen van een pomp te wijzigen hoef je niet eerst in een lijst de pomp op te zoeken, maar klik je op de kaart op een symbool, dat de pomp representeert. Vervolgens zijn een aantal context gevoelige functies beschikbaar. Een ander voorbeeld het verplaatsen van een pomp door deze met de muis te verplaatsen. Dit in tegenstelling tot het invoeren van x, y coördinaten. Het één sluit het ander overigens niet uit. Het kan heel goed zijn dat de mogelijkheid voor het invoeren van x en y coördinaten beschikbaar blijft omdat dit meer preciese controle geeft dan de muis.

DM speelt ook een rol in de terugkoppeling naar de gebruiker; algemene voorbeelden daarvan zijn:

- sterretje achter een naam als een object is gewijzigd
- objecten rood in editor als er een fout in dat object zit
- tooltips

DM kan ook worden toegepast op meerdere objecten tegelijkertijd; je kunt in die context denken aan bijvoorbeeld een multi-object editor bij een 1d-schematisatie.

4.4 Kernbegrippen

Tijdens het ontwerp bleek het noodzakelijk een set van entiteiten te ontwerpen voor gebruik door het systeem. Onderstaande tabel geeft een korte toelichting van de belangrijkste entiteiten.

| Term | Description |
|-------------|--|
| Application | Het systeem, dat toegang biedt tot de plug-ins waaruit de applicatie bestaat. Application kan gebruikt worden voor verschillende typen applicaties: grafische user interface (rich client), web applicatie, console programma of ArcGIS plug-in. |
| Plug-in | Biedt de mogelijkheid het systeem met nieuwe functionaliteit uit breiden zonder dat enige wijzigingen nodig zijn in het basis systeem zelf. Alle geregistreerde plug-ins worden automatisch geladen bij het starten van de applicatie. Plug-ins worden gebruikt voor de volgende functionaliteit: <ul style="list-style-type: none">• Integratie modellen• Toevoegen nieuwe data typen• Uitbreiden grafische User Interface Plug-ins maken het ook eenvoudig mogelijk om customized versies van de tools uit te leveren. |
| Project | Is het primaire werkdocument van de gebruiker. Het is ontworpen als de hoofdcontainer van de data en de in taken onderverdeelde modellen. Het is gebruikelijk dat er meerdere modellen zijn opgenomen die onderlinge relaties hebben. |
| Task | Is gedefinieerd als een container van een verzameling modellen die via data item aan elkaar gekoppeld zijn. Een gebruiker kan taken opstarten voor verschillende berekeningen. Binnen het FEWS systeem heeft taak ook deze betekenis. |
| Model | Wordt meestal gebruikt als een wrapper van een enkelvoudige reken engine, bijvoorbeeld SOBEK 1DFLOW, Delft3D FLOW, etc. De reken engines zelf zijn geen onderdeel van de UI en kunnen ook blijvend buiten de UI om aangestuurd worden. |
| Data Item | Definieert de basis interface, die voor elk datatype in het systeem geïmplementeerd moet worden. Voorbeelden zijn model parameter, tijdreeks, model schematisaties zoals 1d netwerken en ongestructureerde grid, etc. |

Tabel 1 Lijst van de meest gebruikte Kernbegrippen

5 Specifieke componenten uit DUFLOW omgeving

In het vorige hoofdstuk zijn de algemene aspecten van de nieuwe user interface omgeving uiteen gezet. Bij de ontwikkeling van de nieuwe SOBEK user interface worden reeds vele gebruikerswensen ten aanzien van flexibiliteit en werkwijze meegenomen. Er is echter ook nog een aantal onderdelen dat meer specifiek of inhoudelijk van aard is. Dergelijke punten worden niet automatisch meegenomen in het migreren van de algemene user interface.

In dit kader zijn in hoofdstuk 3 de onderwerpen TEWOR functionaliteit en waterkwaliteitsprocessencompiler genoemd. Deze en andere verschilpunten die nadere uitwerking behoeven worden in de onderstaande paragrafen toegelicht. Per punt zal worden gekeken of aanpassing van de werkwijze, formuleringen of code gewenst is.

5.1 TEWOR functionaliteit

De TEWOR functionaliteit betreft een binnen Nederland gestandaardiseerde gevoeligheidsanalyse. Hoewel de benodigde simulaties door middel van een algemene batch functionaliteit kunnen worden uitgevoerd is vanwege de frequentie waarmee dergelijke analyses worden uitgevoerd het wenselijk meer specifieke functionaliteit op dit vlak beschikbaar te hebben binnen het instrumentarium (bijv. als dedicated schil rondom generieke functionaliteit).

5.2 Waterkwaliteitsprocessencompiler

De waterkwaliteitsmodule van DUFLOW maakt gebruik van een simpele eigen ontwikkelde DUPROL taal voor het beschrijven van de gesimuleerde waterkwaliteitsprocessen. Deze werkwijze wijkt af van de processenbibliotheek van SOBEK waarbij de procesformuleringen niet als code beschikbaar zijn (wel in de manual) of via de open processen bibliotheek zelf in FORTRAN geprogrammeerd dienen te worden. Of en hoe de processencompiler in SOBEK kan worden geïntegreerd is op dit moment niet duidelijk. Voorgesteld wordt om gedurende het traject te onderzoeken hoe de gewenste functionaliteit in het systeem kan worden ingebouwd.

5.3 Meerdere landgebruiktypes in één RR / RAM knoop

De wijze van neerslag-afvoermodellering verschilt tussen SOBEK en DUFLOW. Binnen SOBEK-RR wordt per knoop slechts één landgebruikstype toegestaan terwijl binnen DUFLOW-RAM per knoop meerdere landgebruikstypes gecombineerd kunnen worden. Hoewel op zich dezelfde functionaliteit beschikbaar is (door andere wijze van schematisatie te kiezen), dient een beslissing genomen te worden of beide aanpakken naast elkaar gewenst zijn of dat de SOBEK aanpak wordt gehandhaafd. Dit punt speelt ook bij de koppeling met de emissiemodule.

5.4 Importeren van DUFLOW schematisaties

De nieuwe user interface zal oude SOBEK projecten kunnen importeren (de nieuwe user interface gebruikt een andere filestructuur). SOBEK kent nu al een optie om een DUFLOW schematisatie te importeren; om het overstappen voor DUFLOW te vereenvoudigen is het wenselijk om deze import optie uit te breiden en zo volledig en correct mogelijk te maken (in hoeverre dit voor DUPROL processen ook automatisch lukt of nodig is, is nog niet geheel duidelijk). Omdat voor DUFLOW gebruikers het rekenhart verandert bij overstap op SOBEK is hercalibratie waarschijnlijk niet te voorkomen.

6 Organisatie

De ontwikkeling van de user interface wordt uitgevoerd door Deltares.
Contactpersoon/projectleider voor dit project is **...<nog nader in te vullen>**

Voor de technische aansturing en de invulling van de gebruikerswensen wordt een gebruikersgroep opgericht. De rol van de groep staat beschreven in par. 6.2 Kwaliteitsborging.

6.1 Projectbeheersing

Bij de ontwikkeling van de SOBEK User Interface wordt gebruik gemaakt van de ontwikkelmethode Extreme Programming. Bij Extreme programming is een aantal ontwikkelprincipes tot in het extreme doorgevoerd.

Voorbeelden hiervan zijn¹:

- Als het goed is dat ontwikkelaars code bij elkaar reviewen, doe het dan voortdurend: ontwikkel alle software in koppels. Met andere woorden, twee mensen achter één computer.
- Als testen goed is, schrijf dan unit tests en voer ze iedere keer uit als er ook maar een regel code is veranderd.
- Als ontwerpen goed is, maak het dan onderdeel van ieders dagelijks werk: verbeter het ontwerp stapsgewijs, zodra de noodzaak zich voordoet.
- Als eenvoud goed is, houd dan het ontwerp zo simpel mogelijk. XP werkt veel met het KISS (Keep It Simple Stupid!) principe.
- Als architectuur zo belangrijk is, laat dan iedereen werken aan het ontwikkelen van de architectuur.
- Als integratietesten belangrijk is, integreer de code dan zo vaak mogelijk, liefst meerdere keren per dag.
- Als korte iteratieslagen goed zijn, maak ze dan werkelijk heel erg kort: seconden, minuten, uren, in plaats van weken, maanden, jaren.

Bij Extreme Programming (XP) worden de testcases door de klant gespecificeerd er worden geen functionele specificaties opgesteld. Bij voorkeur moet een toekomstige gebruiker te allen tijde aanwezig zijn, zodat er voor gezorgd kan worden dat de applicatie exact dat wordt wat de klant wenst. Om dit proces zo goed mogelijk te laten verlopen, werkt XP met korte ontwikkelcycli, waarin steeds een aantal geselecteerde testcases wordt geïmplementeerd tot een werkend systeem.

6.2 Kwaliteitsborging

Zoals in de vorige paragraaf is gesteld, zou het wenselijk zijn als er continu een gebruiker bij de ontwikkeling aanwezig is. In de praktijk zal dit om verschillende redenen niet haalbaar zijn. Vooral ook, omdat de te ontwikkelen user interface niet voor

¹ Bron: Wikipedia – Extreme Programming

één specifieke gebruiker wordt ontwikkeld, maar voor verschillende typen gebruikers. Daarom wordt voorgesteld om een gebruikersgroep op te richten. De opdrachtgever dient een coördinator aan te wijzen om de contacten te realiseren.

Leden van deze gebruikersgroep worden regelmatig uitgenodigd voor het bijwonen van workshops. Tijdens deze workshops worden de gebruikers ingelicht over de stand van zaken van het project en ook zullen zij gevraagd worden mee te denken over de uiteindelijke user interface. De workshops kunnen georganiseerd worden voor de gehele groep tot aan één specifiek lid van de groep die op de werkvloer komt meedenken over de user interface, door bijvoorbeeld samen met een ontwikkelaar een aantal user stories op te stellen .

6.3 Gebruik van projectresultaten

Gedurende de projectperiode zullen tussenversies van de gerealiseerde software beschikbaar komen voor testen door de gebruikers. De nieuwe ontwikkelingen zullen geleiding overgenomen worden in het reguliere B&O traject zodra deze voldoende stabiel en dekkend zijn qua functionaliteit.

7 Planning en financiering

Op basis van de huidige voortgang met de bouw van de nieuwe SOBEK user interface wordt gestreefd om rond de jaarwisseling 2008/2009 een eerste beta versie van de user interface voor de 1D stromingsmodule beschikbaar te hebben. Deze user interface zal in 2009 worden uitgebreid met 1D functionaliteit dekkende neerslag-afvoer, waterkwaliteit, morfologie en regeling van structures. Gedurende 2009 zal deze functionaliteit geleidelijk als beta versie beschikbaar komen, hetgeen moet leiden tot de beschikbaarheid van een beta versie die de volledige 1D functionaliteit omvat. Gedurende 2009 en eerste helft van 2010 zal deze versie zich in de praktijk moeten bewijzen alvorens deze naar verwachting medio 2010 als algemene release versie beschikbaar kan komen. Vrijwel gelijktijdig met de afronding van het migratietraject van de DUFLOW functionaliteit zal dan ook het migratietraject van SOBEK-RE naar de geïntegreerde SOBEK-Rural/Urban/River lijn afgerond worden.

De financiering van de bouw van de nieuwe SOBEK user interface wordt vrijwel volledig gefinancierd door Deltares zelf (in 2008 wordt zo'n 700 k€ in deze ontwikkeling geïnvesteerd, voor 2009 wordt een bedrag in dezelfde orde van grootte voorzien). Vanuit de RWS financiering zal in 2009 ook bijgedragen worden aan de implementatie van de laatste SOBEK-RE functionaliteit (o.a. morfologie) in de nieuwe werkomgeving.

Bij de ontwikkeling van de nieuwe user interface worden nu al belangrijke aspecten van de DUFLOW interface meegenomen; toch wordt voor de migratie van DUFLOW functionaliteit een financiële bijdrage aan STOWA cq. DUFLOW gebruikers gevraagd. Deze bijdrage zou dienen ter dekking van meer specifieke user interface wensen van de huidige DUFLOW gebruikers en voor de uitwerking/implementatie van de punten genoemd in hoofdstuk 5.