Beachplan as an Open-MI composition

James Sutherland[1], Marion Bolster[1], Adrian Harper[2]

1 HR Wallingford, Howbery Park, Wallingford, Oxfordshire, OX10 8BA, UK
2 Innovyze, Howbery Park, Wallingford, Oxfordshire, OX10 8BA, UK.

Proceedings of 2013 IAHR World Congress

Abstract

ABSTRACT: FluidEarth 2.0 is an open source implementation of the OpenMI 2.0 interface standard for data exchange between numerical models during run-time. It consists of the FluidEarth Software Development Kit (SDK) to assist with making models and components OpenMI 2.0 compliant, the Pipistrelle GUI for linking and running OpenMI compliant models and components using one-way or two-way links for the exchange of data and the FluidEarth portal with news, a discussion forum, case studies, community contact details as well as links to the FluidEarth catalogue, the training website and source code repository. The power of FluidEarth to link together separate models to run as a composition has been demonstrated using the re-written Beachplan model of the long-term development of the coastline of sand or gravel beaches. This is a practical model for coastal management that requires the two-way exchange of data between component models at each time-step. Poole Bay in southern England (UK) has been used as test case and has been shown to reproduce the effect of groynes on beaches over a period of almost two years at a moderate computational cost.

Keywords

OpenMI, FluidEarth, Beachplan, 1-line model, composition

1. Introduction

There is a growing awareness of the need to model environmental systems as a whole, as their components interact, so cannot be modelled in isolation (Sutherland et al., 2013). For example, 'whole catchment modelling' is required to deliver the objectives of the Water Framework Directive (European Commission, 2000) and this requires the linking of a wide variety of models. One approach to this is to integrate separate component models to simulate an interacting system. The FluidEarth 2.0 implementation of the Open Modelling Interface (OpenMI) standard provides software to assist with linking different models during run-time, in order to achieve this integration. This approach is illustrated using Beachplan, a one-line model of the long-term planshape evolution of non-cohesive (sand or gravel) beaches.

2. Open MI and Fluidearth

OpenMI (www.openmi.org) is a software component interface standard for numerical models (Gregersen et al., 2007) that allows two-way exchange of data between compliant components as they run. When OpenMI compliant components are linked and run together, exchanging data as they run, this is known as a composition. Although OpenMI was originally conceived to facilitate whole-catchment modelling, what was developed is a generic solution to the problem of data exchange between models or software components (e.g. Bastin et al., 2012, Becker and Schüttrumpf, 2011, Bulatewicz et al., 2010, Bulatewicz et al., 2012, Castronova and Goodall, 2012, Safiolea et al., 2011, Shrestra et al., 2012). The generality was extended with the release of version 2.0 of the OpenMI Standard (OpenMI Association, 2010a, b) which added:

- Base interfaces and extensions; and
- Adaptors (between the output of one model and the input of another).

All implementations of the OpenMI standard are expected to contain

- a software development kit (SDK) to assist model developers with making model engines OpenMI compliant; and
- a graphical user interface, to allow model users to build and run compositions of OpenMI compliant components, so that environmental systems can be modelled.

HR Wallingford's FluidEarth 2.0 is the Windows (.Net 4) reference implementation of the OpenMI 2.0 standard (Harpham et al., 2013). Multi-threading is used to prevent corruption between multiple runs of the same model or the same adaptor. The FluidEarth SDK has been designed to make it relatively easy to make models (and other components) OpenMI compliant, while the GUI (known as Pipistrelle) provides a user-friendly interface to allow users to assemble and run compositions of models. Pipistrelle and the FluidEarth 2.0 SDK are open source and can be obtained from http://sourceforge.net/projects/fluidearth while a set of examples that use models and adaptors written in C# and FORTRAN are provided on a training website (http://eLearning.fluidearth.net). FluidEarth also provides a portal at http://fluidearth.net with news, links, a discussion forum, case studies and community contact details as well as a catalogue (http://catalogue.fluidearth.net) which lists model engines and their instances.

3. Beachplan

Beachplan is HR Wallingford's long-established one-line model that is used to simulate changes in the planshape of a non-cohesive (sand or gravel) beach over a period of time (Ozasa & Brampton, 1980). The beach is discretised into a number of cross-shore profiles with a constant beach gradient. Offshore wave conditions are transformed to the breaker line and a formula is used to calculate the longshore sediment transport rate at each profile, caused by wave breaking. The effects of structures on longshore transport are represented by a set of rules, based on refraction, diffraction and blocking of incident waves. Plan-shape changes to a beach are driven entirely by changes in the rate of wave-driven longshore sediment transport, using the constant beach profile and assuming continuity of sediment. The model is suited to gently curving beaches with parallel depth contours and only simulates the changing position of one single beach contour. Beachplan has been re-written in C# as an OpenMI composition, which is run through the pyxis workflow manager (San Roman Blanco et al., 2012).

3.1. Beachplan as an OpenMI Composition

As an Open-MI composition, the individual Beachplan modules use data from one module to update the data of another. Figure 1 outlines the four modules which are used in Beachplan, together with the direction of links, indicating a two way exchange whereby the module is set up to both sending and receiving data or a one way link set up to only send or receive data.

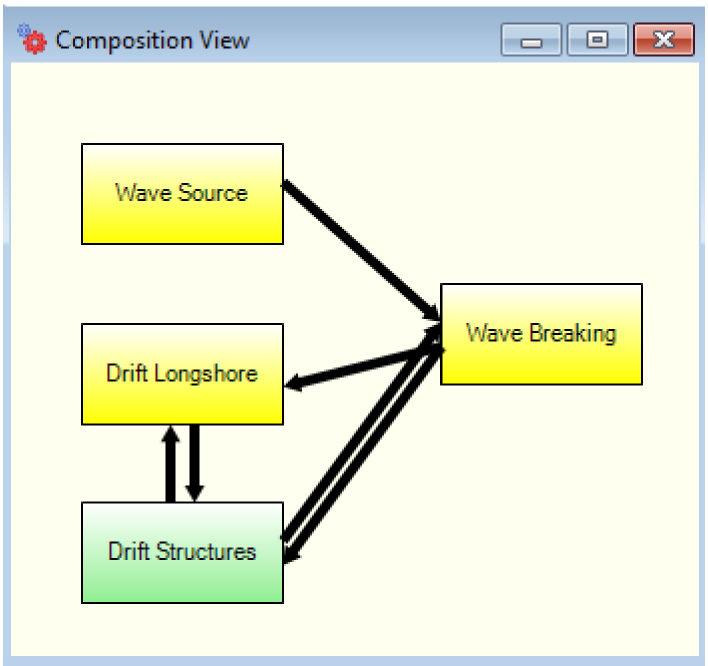

Figure 1: Pipistrelle GUI showing Beachplan as an Open-MI composition

3.2. Basic computational model sequence

The computational loop within the numerical model Beachplan can be simplified into five main steps which are described below and depend on the four modules represented by the boxes in the Open-MI user interface of the Pipistrelle console.

1. wave source is read/interpreted;
2. breaking wave conditions at each beach section are defined (using the wave source) not including the locations of groynes;
3. the longshore drift is calculated across each beach section, not considering sections with groynes or structures;
4. the longshore drift is calculated across those sections with groynes and structures;
5. the longshore drift is calculated for each beach cell, including sediment addition or removal.

The resulting shoreline position is updated and the loop is repeated for as long as the user has defined. Each step is described in more detail below.

3.3. Step 1 - Wave source

The wave conditions required by Beachplan can be entered for one or several locations along the beach, and need to be at an equal depth contour seawards of the breaker zone. The user may choose a constant wave condition or the use of a time-series of wave data at regular intervals, e.g. hourly. By interpolation, the model can then convert the specified wave input information and calculate the required significant wave height, direction and frequency at each beach cross-section.

3.4. Step 2 - Breaking wave conditions

Once the wave conditions for each cross-section have been defined the data is sent to the wave breaking module and the next step proceeds. This is a one way link in which the wave source module sends data to the wave breaking module. The information is then used to calculate the breaking wave conditions at each cross section, except on those cross-sections where a groyne has been defined. To calculate the wave height and direction at the break point, Beachplan considers both wave refraction and diffraction but in a simplified manner - each wave condition is assumed to have a single period and refraction is calculated using Snell's Law between the depth of the wave point and the breaker line. The shoreline orientation is used to obtain breaking wave conditions where waves are refracted.

When calculating the wave diffraction, the simplified process ignores changes in water depth between the shoreline and the wave point. At a breakwater or diffracting groyne the diffraction is therefore calculated using the direction of the approaching wave and the angle the end of the structure makes to the shoreline for each section. To allow for the combination of wave diffraction and refraction, the diffraction routine changes the height and direction of the input wave conditions and then goes on to calculate the wave refraction. Beachplan calculates wave refraction by assuming the beach and seabed contours are parallel and straight for each section using Snell's Law, assumes that the incoming waves are monochromatic with a single direction and period. The breaker depth at each beach section is evaluated through an iterative procedure for each section. To calculate the breaker depth and breaking wave height, wave direction and group velocity, the iterations are run until the significant wave height at the depth is 0.55 times the water depth. Where a tidal level has been specified, it can be used to define the water depth at the wave input condition.

3.5. Step 3 – Calculating longshore drift rates

To calculate the longshore drift, Beachplan uses a modified form of the well-known CERC equation which takes into account the effects of longshore variations in breaking wave heights which may occur in the lee of a breakwater for example (Ozasa and Brampton, 1980):

$$Q = K_1(\gamma_s)^{-1}E_b(nC)_b\left(Sin2\alpha_b - 2K_2\frac{\partial H_b}{\partial x}cot\beta cos\alpha_b\right)$$

Where:

Q is the bulk longshore drift rate (m^3/s);

K_1, K_2 are calibration coefficients;

α is the angle (wave crests to beach contours);

E is the wave energy density = $\rho g H_b^2/4$

ρ is the water density (kg/m^3);

g is gravitational acceleration (m/s^2);

(nC) is the wave group velocity (m/s);

γ_s is the submerged weight of the sediment;

β is the beach slope;

H_b is the breaking (significant) wave height (m); and

x is the longshore coordinate (m).

The subscript “b” refers to breaking wave conditions. The variables are derived in step 2 or provided by the user. The equation gives the total amount of beach sediment that is transported across the beach cross-section, i.e. the potential bulk longshore drift rate. It assumes that there is sufficient sediment available, and thereby equates the maximum bulk drift rate at each section except those occupied by a groyne. Where a groyne is present no breaking wave height is calculated and the bulk longshore drift has to be calculated differently, see the next step.

The potential transport rate may be reduced where the landward end of a beach profile is restricted by a seawall or where the drift rate along a beach is affected by the lack of sediment beyond the beach toe, i.e. closure level (referred to as rock level in Beachplan). The cross-shore distribution, and thereby the bulk longshore drift, is adjusted to account for the rock level.

3.6. Step 4 - Calculating drift rates at structures

To calculate the effect of structures on the longshore drift, the information from both the ‘drift longshore’ module and ‘waves breaking’ module are fed into the ‘drift structures’ module. The effect on the longshore drift from both seawalls and rock platforms has been considered before the effect of other structures such as groynes and offshore breakwaters, is taken into account.

To calculate the longshore drift occurring at an offshore breakwater, the level landwards of a structure is calculated using the distance between the structure’s tip and the shoreline, and the beach slope. If the rock level is above this level, then the drift calculated for the rock level is used (i.e. no further action is required) but if the rock level is below, the drift needs to be reduced further. This is done using the cross-shore distribution as above and reducing it to the level calculated at the breakwater (and assumes that there is no available sediment seawards of the breakwater).

The amount of sediment that will bypass the groyne end is then calculated by estimating the beach width immediately updrift of the groyne as well as the amount of drift travelling over the crest or through the body of the groyne.

Finally, the routine needs to consider the amount of drift that is predicted to pass seawards of the structures. Where only some of the sediment is lost offshore, the remaining drift is assumed to travel along the beach, roughly parallel along the contours, and then distributed downdrift of the structure for some length of the beach. In some instances all may be assumed lost offshore into the deeper water.

3.7. Step 5 - Calculating the sediment balance

Finally, the information on the effect of structures on longshore drift is fed back into the ‘drift longshore’ module and the longshore drift is calculated for each beach cell, taking into account any sediment addition or removal. The resulting shoreline position is updated (which affects wave breaking if the shoreline angle is changed) and the loop is repeated for as long as the user has specified.

4. Poole Bay case study

Although Beachplan has been proven to work best for gently curving beaches, beaches with a substantial curvature and a large change in orientation have proven successful. One of these examples is Poole Bay which stretches over 17 km from Sandbanks in the west to Hengistbury Head in the east. On this stretch of coastline the orientation changes by about 90º. A Beachplan model should be calibrated for each stretch of coastline that it simulates before going on to predict future beach changes. This ideally involves comparing beach surveys with model results from a Beachplan run over the same period, ideally using measurements of wave (and sometimes tidal) conditions.

4.1. Setting up the model

A new set of rock groynes were installed at the western end of Poole Bay in early 2009. A post installation beach survey conducted by Poole Borough Council in May 2009 was used to provide the starting shoreline for Beachplan, which was then calibrated using surveys from spring 2010 and 2011. The location and length of the old and new groynes (which were the only active engineering structures) were obtained from aerial photographs and data supplied by the Channel Coastal Observatory (www.channelcoast.org).

Offshore wave conditions were measured in Poole Bay by CEFAS wave rider buoy. The wave data were transformed from the wave buoy to several specified nearshore locations along the Poole frontage using a back-tracking ray model. Hindcast nearshore wave climates were derived from the model output. The resulting time-series for the nearshore wave data are given at six locations between Bournemouth pier and Sandbanks (Figure 2) on the -5 m OD contour. Normally, the -5 or -10 m OD contour is used for nearshore wave input in the Beachplan model as the Beachplan model itself only uses simple transformations of the waves into breaking point, hence much of the effect of a more complex seabed would be lost if the time series from the offshore buoy was used. Transformations from the -5 m contour into break point is then completed within the Beachplan model. The seabed around Hook sands is particularly complex and would be lost if the -10 m contour was used.

4.2. Beachplan example

The model was calibrated between May 2009 and March 2011 by varying standard input parameters within their accepted ranges. The modelled and measured shorelines along a section of the modelled shoreline are shown in Figure 3. This demonstrates that Beachplan is capable of representing the effects of groynes on the beach position with sufficient accuracy to be used on coastal management. The models takes about 10 minutes to run a 2-year simulation on a single core. This Beachplan model has now been linked to a generic model for the forward propagation of uncertainty (which also runs under the pyxis workflow manager) so that the calibration and sensitivity analysis can be automated. This will form the next stage of the study.

4.3. Beachplan developments

One of the future developments planned for Beachplan is to couple the model with HR Wallingford's coastal profile model COSMOS (Nairn and Southgate, 1993). COSMOS calculates cross-shore sediment transport distribution rates using profile data and wave information which is already needed for a Beachplan setup. Currently Beachplan uses a standardized cross-shore sediment transport distribution for either sand or shingle, but can also use the output from COSMOS for a more site specific distribution, whereby the user manually has to enter the data into Beachplan. This requires the user to run the two models separately, so by joining the two models using FluidEarth 2.0 the models can feed into each other to allow for quicker and easier modeling, which will in turn provide the user with more accurate and site specific results.

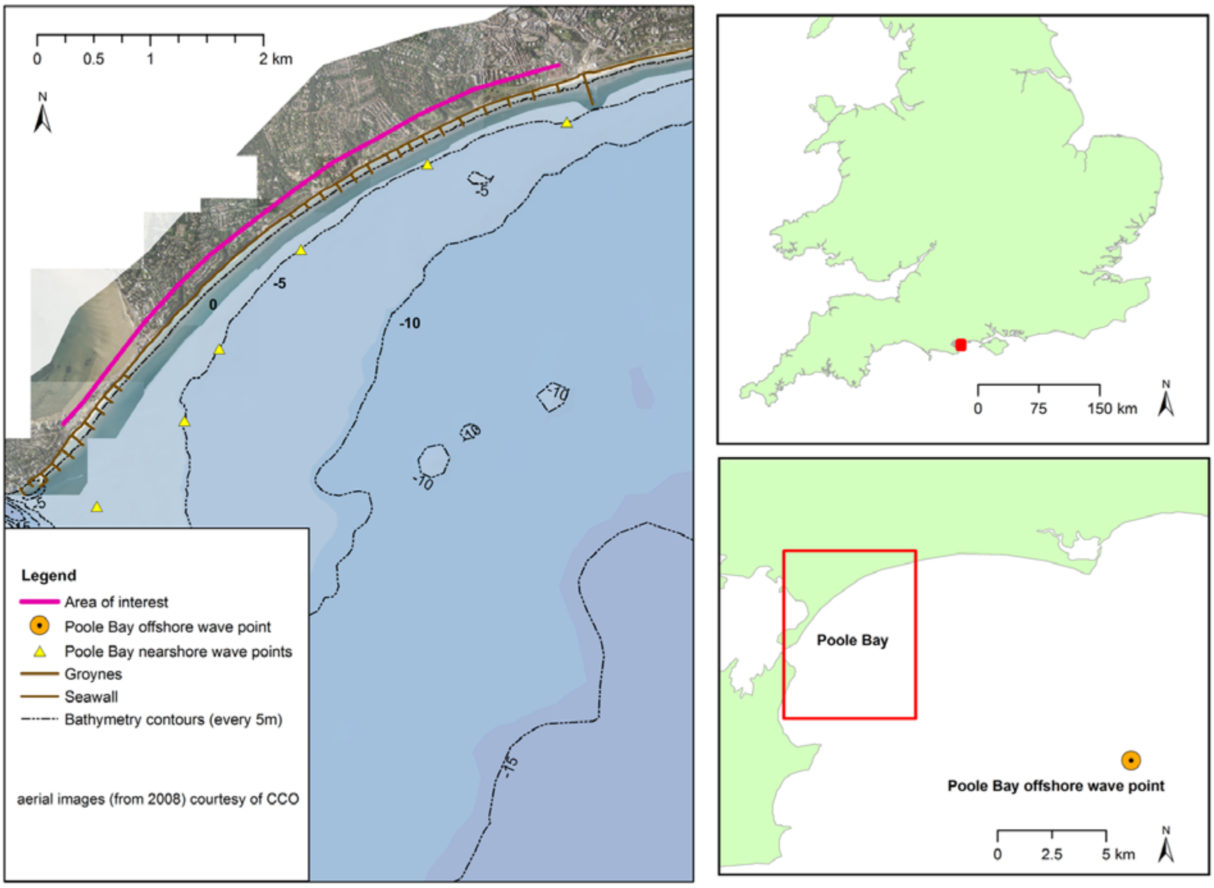

Figure 2: Wave buoy and inshore wave points along the western end of the model

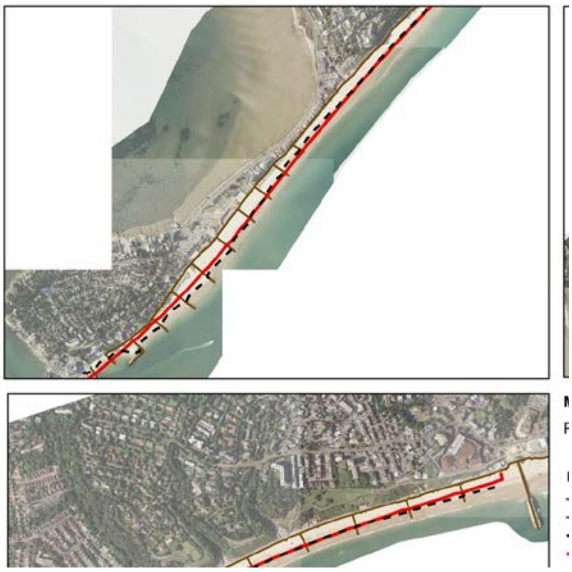

Figure 3: Modelled and measured shoreline for March 2011

5. Conclusions

FluidEarth 2.0 is an open source implementation of the OpenMI 2.0 interface standard for data exchange between numerical models during run-time.

It consists of:

- FluidEarth Software Development Kit (SDK) to assist with making models and components compliant with the OpenMI 2.0 standard;
- Pipistrelle GUI for building and running compositions of OpenMI compliant components, using one-way or two-way links for the exchange of data;
- FluidEarth portal, http://fluidearth.net, with news, a discussion forum, case studies, community contact details as well as links to the
  - FluidEarth catalogue (http://catalogue.fluidearth.net)
  - the training website (http://eLearning.fluidearth.net) and
  - Source code repository (http://sourceforge.net/projects/fluidearth).

The power of FluidEarth to link together separate models to run as a composition has been demonstrated using the re-written Beachplan model of the long-term development of the planshape of sand or gravel beaches. This is a practical model for coastal management that requires the two-way exchange of data between component models at each time-step. Poole Bay in southern England (UK) has been used as test case and has been shown to reproduce the effect of groynes of beaches over a period of almost two years at a moderate computational cost.

References

Bastin, L., Cornford, D., Jones, R., Heuvelink, G.B.M., Pebesma, E., Stasch, C., Nativi, S., Mazzetti, P. and Williams, M., 2012. Managing uncertainty in integrated environmental modelling: The UncertWeb framework, Environmental Modelling & Software, http://dx.doi.org/10.1016/j.envsoft.2012.02.008.

Becker, B.P. & Schüttrumpf, H., 2011. An OpenMI module for the groundwater flow simulation programme Feflow. Journal of Hydroinformatics, 13 (1) 1-12.

Bulatewicz, T., Yang, X., Peterson, J.M., Staggenborg, S., Welch, S.M. & Steward, D.R., 2010. Accessible integration of agriculture, groundwater, and economic models using the Open Modeling Interface (OpenMI): methodology and initial results. Hydrology and Earth System Sciences, 14, 521-534. Copernicus Publications

Bulatewicz, T., Allen, A., Peterson, J.M., Staggenborg, S., Welch, S.M. & Steward, D.R. 2012. The Simple Script Wrapper for OpenMI: Enabling interdisciplinary modeling studies, Environmental Modelling & Software, http://dx.doi.org/10.1016/j.envsoft.2012.07.006.

Castronova, A.M. & Goodall, J.L., 2012. Simulating watersheds using loosely integrated model components: Evaluation of computational scaling using OpenMI. Environmental Modelling & Software, http://dx.doi.org/10.1016/j.envsoft.2012.01.020.

Gregersen, J.P., Gijsbers, P.J.A. and Westen S.J.P., 2007. OpenMI: Open modelling Interface. Journal of Hydroinformatics, 9(3), 175-191. http://dx.doi.org/10.2166/hydro.2007.023.

Harpham, Q., Cleverley, P. and Kelly, D., 2013. FluidEarth 2 – the implementation of OpenMI 2.0. Journal of HydroInformatics (in preparation).

Nairn, R.B. and Southgate, H.N., 1993. Deterministic profile modelling of nearshore processes. Part 2. Sediment transport and beach profile development. Coastal Engineering 19:56–96.

Ozasa, H. and Brampton, A.H., 1980. Mathematical modelling of beaches backed by seawalls. Coastal Engineering 4: 47 – 63.

Safiolea, E., Baki, S., Makropoulos, C., Deliege, J.F, Magermans, P., Everbecq, E., Gkesouli, A., Stamou, A. & Mimikou, M., 2011. Integrated modelling for river basin management planning, Proceedings of the ICE, Water Management, 164 (8), 405-419.

San Roman Blanco, B.L., Sutherland, J. and Harper, A., 2012. pyxis: a new workflow manager for integrated modelling. Proceedings of 10th International Conference on Hydroinformatics, HIC 2012, Hamburg, Germany. ISBN 978-3-941492-45-5.

Shrestha N.K., Leta O.T, de Fraine B., Van Griensven A., Garcia-Armisen T., Ouattara N.K., Servais P. and Bauwens W., 2012, Integrated modelling of river Zenne using OpenMI. Proceedings of 10th International Conference on Hydroinformatics,, HIC 2012, Hamburg. Germany. ISBN 978-3-941492-45-5.

Sutherland, J., Townend, I.H., Harpham, Q.K. and Pearce, G.R., 2013. From Integration to Fusion: the challenges ahead. In Model Fusion: integrating environmental models to solve real world problems. The Geological Society (in press).

The OpenMI Association, 2010a, OpenMI Standard 2 Specification for the OpenMI (Version 2.0). Part of the OpenMI Document Series. www.openmi.org

The OpenMI Association, 2010b. OpenMI Standard 2 Reference for the OpenMI (Version 2.0). Part of the OpenMI Document Series. www.openmi.org