ABSTRACT

Research and consultancy as well as construction projects often spend a significant part of their budget to set up some basic infrastructure for data and knowledge management, most of which dissipates again once the project is finished. Standing initiatives so far have not been successful in providing a proper data and knowledge management system for data, models and tools. OpenEarth (www.openearth.eu) was developed as a free and open source alternative to the current often ad-hoc approaches to deal with data, models and tools. OpenEarth as a whole (philosophy, user community, infrastructure and workflow) is the first comprehensive approach to handling data, models and tools that actually works in hydraulic engineering practice at a truly significant scale. It is implemented effectively not only at its original founding organisations, Delft University of Technology and Deltares, but also in a number of sizeable research programmes with multiple partners (such as research programme “Building with Nature” with 19 partners from one country) and from multiple countries (such as the 3-year European Union FP7 research programme MICORE with 15 partners from 9 countries). It has been adopted as the main data management workflow for all research programmes around the Sand Engine Delfland and was awarded the Dutch Data Prize 2012 for technical sciences by 3TU.datacentrum, the data archiving institute of the Dutch technical universities, and DANS, the data archiving institute of the Dutch National Science Foundation (NWO) and the Royal Dutch Academy of Sciences (KNAW).

For data, models and tools that are truly strategic and really cannot be shared, OpenEarth stimulates the set-up of internal OpenEarth clones. This way the OpenEarth workflow can still be adopted, promoting collaboration within an organisation, while taking care of security considerations at the same time.

This article is based on and updates the OpenEarth philosophy, infrastructure and main workflow protocols as presented at WODCON XIX in Beijing, China (Van Koningsveld et al., 2010). A number of practical example applications that have been realised to date are given to illustrate OpenEarth’s potential for the dredging industry.

INTRODUCTION

The sustainable interaction between humankind and planet Earth poses huge hydraulic and environmental engineering challenges. Confronting these challenges one-project-at-a-time, while seemingly attractive from a budget management perspective, results in grave inefficiencies in developing and archiving the basic elements that are invariably involved: data, models and tools. Both research, consultancy as well as major construction projects commonly spend a significant part of their budgets to set up some basic data and knowledge management infrastructure, most of which dissipates again once the project is finished.

Internally institutions generally employ intranet services and internal networks to collaborate and exchange information. However, owing to increasing complexity, large projects nowadays are regularly executed by consortia.
The internal services of individual institutions do not allow for collaboration because of technical limitations or simply denial of permission for exchanges. As a result the way data, models and tools are currently managed, while presumably aimed at protecting the knowledge capital of organisations, in fact also inhibits (individual as well as collective) progress.

Over many years Delft University of Technology and Deltares, together with many partners from the hydraulic engineering industry, developed OpenEarth (www.openearth.eu) as a clonable, free and open source alternative to the project-by-project and institution-by-institution approaches to deal with data, models and tools (e.g., Van Koningsveld et al., 2004; Van Koningsveld et al., 2010; Baart et al., 2012; De Boer et al., 2012). OpenEarth transcends the scale of single projects facilitating that each project builds on the heritage of previous projects.

OpenEarth at its most abstract level represents the philosophy that data, models and tools should flow as freely and openly as possible across the artificial boundaries of projects and organisations (or at least departments). Put in practice OpenEarth exists only because of a robust user community that works according to this philosophy (a bottom-up approach).

In its most concrete and operational form, OpenEarth facilitates collaboration within its user community by providing an open ICT infrastructure, built from the best available open source components, in combination with a well-defined workflow, described in open protocols based as much as possible on widely accepted international standards.

OpenEarth as a whole (philosophy, user community, infrastructure and workflow) is the first comprehensive approach to handling data, models and tools that actually works in hydraulic engineering practice at a truly significant scale. It is implemented effectively not only at its original founding organisations, Delft University of Technology and Deltares, but also in a number of sizeable research programmes with multiple partners (such as the €28 million 4-year research programme Building with Nature with 19 partners from one country) and from multiple countries (such as the €4.6 million 3-year European Union FP7 research programme MICORE with 15 partners from 9 countries).

It has been adopted as the main data management workflow for all research programmes around the Sand Engine Delfland and was awarded the Dutch Data Prize 2012 for technical sciences by 3TU.datacentrum, the data archiving institute of the Dutch technical universities, and DANS, the data archiving institute of the Dutch National Science Foundation (NOW) and the Royal Dutch Academy of Sciences (KNAW). As a result OpenEarth is now carried by a rapidly growing user community that as of April 2013 comprises some 1000 users, over 280 LinkedIn group members, more than 150 active developers, creating upwards of 6500 contributions, originating from a multitude of organisations and countries. Together they share and co-develop thousands of tools, tera-bytes of data and numerous models (source code, raw data and data products, model schematisations and pre- and post-processing tools).

THE OPENEarth PHILOSOPHY

As outlined above, the availability and accessibility of high-quality data, models and tools is crucial in successfully handling hydraulic engineering problems. The three in some shape or form are involved in any project design, risk analysis, cost estimation, impact assessment, production optimisation and so on. Past experience from sizeable consultancy projects as well as numerous research programmes (e.g., Capobianco, 1999; Wilson, 2002; Stodden, 2010) have shown that while everybody acknowledges its importance, nobody as yet has been able to establish a sustainable, functioning knowledge management system for data, models and tools.

The widely used and extensively standardised project-based approach effectively handles document control at the start-up, execution and closure phases of projects. Numerous archive systems are available to safely store important project-related information such as tender documents, bids, method statements, reports, official correspondence, contracts, presentations, financial information and more. As a result the workflow in projects is now highly traceable and reproducible with the main aim to achieve the best possible hold on the project realisation process in order to avoid unnecessary errors and mistakes and associated financial penalties, losses and claims.

From this perspective the project approach is clearly effective, explaining its world-wide popularity and implementation. In 2010 the International Organization for Standardization (ISO) announced that ISO 9001 certifications, which are widely regarded as a global benchmark for quality management, topped the one million mark (see also: http://www.iso.org/iso/news.htm?refid=Ref1363, last checked 18-04-2013).

The main strength of the project-based approach, viz., effectively managing a project given its available means in terms of time, money, people and equipment, can become a weakness or even a threat if applied too rigidly. As is commonly acknowledged, certification to, for example, an ISO 9001 standard does not guarantee any quality of end products and services. It only certifies that formalised business processes are being applied. This potential weakness is most visible for those elements that transcend the scope of a single project, notably the quality, availability and accessibility of data, models and tools (in combination with knowledge and practical experience of course).

Project management systems deal with an estimated 20% management, overhead and reporting share of the project budget only. For the actual knowledge and information generated with the other 80% of the budget, no effective, integrated and widely applied quality management system exists to our knowledge (NB: percentages estimated by the authors). An effective approach would be particularly useful for the explicit knowledge as knowledge management founders like Polanyi (1966) and Nonaka and Takeuchi (1995) called it.

It is in fact this “knowledge capital” that can most easily be reused and further developed in subsequent projects, whereas project specific management documents are generally of no further practical use, except perhaps in legal cases.

Many attempts have been undertaken to deal with these issues. Numerous EU and Dutch national research programmes have promised
to deliver and disseminate data gathered throughout the project. This has resulted in many databases, web-portals, CD Roms, DVDs and ftp-sites that have gradually gone rogue. As soon as a project has ended and the incentives to maintain the databases and web links are gone, slowly but surely they are forgotten. When a new project comes along, setting up a new database rather than reviving the old one seems more attractive.

Something similar can be observed for managing models and tools. Many research and consultancy projects have dealt with data analysis. Invariably routines have been developed to import data, structure it in some form suitable for analysis, analyse it and report the analysis results. Clearly when such routines are developed by each project from scratch a great deal of time, money and effort is wasted.

A common reason that numerous existing data, models and tools management initiatives fail is that they are either imposed top-down or they emerge bottom-up without proper consideration for the bigger picture. Both methods are highly unlikely to be successful in the long run. Instead, OpenEarth emerged as a bottom-up approach with a long-term perspective on knowledge sharing and use, rather than focussing on just the technology (even though the use of proper technology is obviously important).

OpenEarth effectively provides and maintains all required technological infrastructure in-support-of, yet independent-from, any individual project. At the same time OpenEarth offers the essential training to allow project members themselves to make use of and contribute to the centralised repositories already available during the course of a project, rather than at its end. A small team invests some editorial and reviewing effort to prevent divergence of the proposed standards and ensure quality of the OpenEarth products.

In summary after several years of successful application, the OpenEarth philosophy consists of:
- A robust community of users …
- collaborating from the philosophy …
- that data, models, tools and information …
- should be exchanged as freely and openly as possible …
- across the artificial boundaries of projects and organisations …
- with an approach that fosters continuous and cumulative quality improvement.

The OpenEarth philosophy, while addressing a crucial gap in common quality management systems, also poses a challenging problem. The aim for maximum efficiency ideally means that all results should be shared – minimally, amongst the legal project contributors and, preferably, with the whole world. Co-operation results in greater overall progress aligning individual with total progress so that they reinforce rather than impede each other.

At the same time, for any individual research project or organisation, full openness quite likely benefits competing consortia or organisations. This presents a typical problem known as the “Social trap” (Platt, 1973).

The term “Social trap” is used for situations where a group of people act to obtain short-term individual gains, which in the long run leads to a loss for the group as a whole (Rapoport, 1988). The “Prisoner’s dilemma” (proposed by Flood and Dresher working at RAND in 1950) and the “Tragedy of the commons” (Hardin, 1968) are some well-known examples.

A similar dilemma exists in the hydraulic engineering sector. Strategic data, models and tools are kept secret for competitive reasons. As a direct consequence, the sector as a whole advances less rapidly and is less advanced than it could be. Despite the downsides of knowledge protectionism, it is clearly an intrinsic characteristic of the dredging industry that cannot be ignored. To accommodate this, OpenEarth stimulates the set-up of internal OpenEarth clones, for data, models and tools that are truly strategic and really cannot be shared with other, outside organisations.

In this way the OpenEarth workflow can still be adopted, promoting collaboration within an organisation, while meeting external security considerations at the same time.

**OPENEARTH INFRASTRUCTURE**

Improper management of data, models and tools can easily result in a wide range of very recognisable frustrations:
- Making the same mistake twice owing to the lack of version control
- Losing important datasets that are extremely hard to replace
- Being unable to reproduce what quantities have been measured and which units apply
- Missing information on measurement position and/or the geographical projection used
- Uncertainty as to the time and time zone the measurements were taken in
- Myriad formats of incoming (raw) data
- A multitude of (slightly different) tools for the same thing
- Difficulty combining numerous databases, each in its own language and style, and so on
- Finding out too late that a colleague has already solved a problem similar to yours
- Facing an impenetrable network of ad-hoc scripts and tools hindering QA/QC and reuse.

Although the above-described frustrations are very common throughout the hydraulic engineering industry, no practical and widely accepted remedy seems to be available.

Many initiatives have been developed though, usually targeting only data, models or tools rather than all three at once (although this certainly needs not be an issue). Some of such initiatives have been granted sizeable budgets to develop a state-of-the-art infrastructure, often outsourced to some ICT company. To promote potential partners to upload their information enormous effort is spent on system security, generally restricting access. As a result of lacking end-user involvement and a focus on access restriction, many systems have been developed at high cost but with low success in terms of active users.

Conversely, repelled by large ineffective yet expensive initiatives, many projects have gone for quick solutions such as simply sharing data on an ftp server or setting up a basic database using any available software. When all project members have write access to a shared project directory or ftp server, data archives can quickly become messy. Without such a shared directory or ftp server, people start to collect their own datasets for their dedicated purpose, resulting in greater claims on limited storage and backup resources, difficulties with data quality control and inefficiencies in terms of mutual learning.

What works better in practice is a Wikipedia-like approach: Set up and maintain an easy to use central system, give write access to anyone while employing a system that logs everything to enable quality control. All data, models and tools that are committed are free for use by anybody. Given a username and password developers can fix bugs, change, delete and add data, models and tools. The use of a version control system ensures that every change is logged.

In fact the version control system can identify for each bit of data and every single line of code who changed it and when. Since 2003 OpenEarth has experimented with numerous approaches available, in the end devising a most convenient infrastructure to support a bottom-up approach to long-term project-transcending collaboration.

The resulting best practice adheres to four basic criteria:

1. **Open standards and open source:**
   The infrastructure should be based on open standards, not require non-libre software and be transferable.

2. **Complete transparency:**
   The collected data should be reproducible, unambiguous and self-descriptive. Tools and models should be open source, well documented and tested.

3. **Centralised access:**
   The collection and dissemination procedure for data, models and tools should be web-based and centralised to promote collaboration and impede divergence.

4. **Clear ownership and responsibility:**
   Although data, models and tools are collected and disseminated via the OpenEarth infrastructure, the responsibility for the quality of each product remains at its original owner.

**Open standards and open source**

The first criterion, *open standards and open source*, is adopted to maximise the number of participants. Known bottlenecks for implementing a new data and knowledge management system are high set-up costs and a fear of changing standards. The first bottleneck is resolved by applying the best available free and open source system components only.

The second bottleneck is addressed by adopting a modular approach that allows for elements of the system to be replaced by other better ones at minimal effort and cost. Fortunately the web provides a large open source community. International groups such as the Open Geospatial Consortium (OGC) and numerous meteorological, oceanographic and remote-sensing collectives have created high-quality software suitable for the OpenEarth purposes. In addition, the requirement of the United States government that all data, models and tools funded by US taxpayers should be available openly has supplied a vast range of free software – an approach that clearly deserves a wider following.

**Complete transparency**

The second criterion, *complete transparency*, is achieved by demanding that collected data are reproducible, unambiguous and self-descriptive. An important distinction made here is between the archival and the dissemination function of a database. To eliminate ambiguity and enhance self-descriptiveness, OpenEarth recommends storing the generally pluriform raw data files in a version-controlled repository with, alongside them, a routine to transform each data format into data products with one single commonly accepted data format.

The raw data and associated conversion scripts are stored to ensure reproducibility, whereas the common data format promotes unambiguity and self-descriptiveness (see Figure 1). For standardised gridded data, the netCDF format (NASA and OGC standard) in combination with the CF standard name vocabularies and EPSG codes for geospatial referencing are recommended. For ecological data, which commonly have a large amount of meta-data, the use of a Relational DataBase Management System (RDBMS) is recommended (the relational database approach is not discussed further in this article).

The distinction between raw data and data products has proven effective in other fields of application as well. In the remote-sensing community, for instance, NASA stores raw data from ocean colour sensing satellites (e.g., NASA’s satellites *Terra and Aqua* are equipped with the MODIS – Moderate Imaging Spectroradiometer – system) as so-called L0 files. These files generally are not available in an easy accessible format as they are optimised to maximise data transmission.
from the satellite to a ground station. The L0 files are stored as raw data files and archived permanently.

The raw data are subsequently enriched with meta-information, such as the satellite locations, and stored as L1 data. NASA also adopts a standard exchange format (i.e., HDF). Further processing is carried out to translate sensor readings to geophysical quantities (L2), and generate data for climatologies and other applications on standard grids (L3 and L4). The levels L1 and higher are considered data products and are primarily meant for dissemination. Each level (except L0) can be recreated with different processing steps, using the same open source software (SeaDAS – tool: http://oceancolor.gsfc.nasa.gov/seadas; reprocessing: http://oceancolor.gsfc.nasa.gov/REPROCESSING/).

The data products are frequently deleted and replaced by improved versions (bug fixes, better calibration, incorporating deterioration of the equipment). All data products carry a version number, i.e., the version of the SeaDAS version that created it.

Unfortunately, outside the remote-sensing community, data products are often considered to be a permanent entity. As a result of human errors and progressing knowledge, in reality, data products are ephemeral entities, and only the raw data should be considered permanent. OpenEarth adopts NASA’s philosophy that data products are ephemeral entities, the sole purpose of which is to facilitate dissemination. Data products should always carry a version number. In line with NASA, OpenEarth prescribes that all data processing scripts needed to transform and enrich the raw data should be stored alongside the raw data. This enables automated data processing.

To ensure reproducibility OpenEarth currently uses the open source version control system Apache Subversion (to be called here Subversion for short) for version control, backup, and access control. If the raw data are really raw, Subversion in practice does not have to do any versioning of these files and storage is mainly for backup purposes and to determine ownership. Although for version control ASCII formats are preferred, binary files can be added to the raw data repository also. The database behind Subversion scales well for large repositories. User friendliness considerations triggered the set-up of separate repositories for data, models and tools.

Centralised access

The third criterion, centralised access, is incorporated in the OpenEarth infrastructure through use of the state-of-the-art regarding the management of data, models and tools: web-services. A myriad of functionalities is already available via web services, and for some, such as webmail, OpenEarth is fully dependant on “the cloud”. Computing and storage in general move towards common commodities that can best be provided on centralised large servers (Carr, 2008).

Strangely for data, models and tools old-fashioned approaches like storing data decentralised on local PCs are still widespread. Offering the OpenEarth infrastructure as web-services allows users to participate with “normal” laptops requiring some form of web access only. The bulk data is stored on the central database and users only extract what data they need. Though the OpenEarth data, models and tools are disseminated via web services the netCDF files can be used off-line too (albeit not updated). A user can for example choose to download a certain (part of a) data file once and store it on the local hard drive if for a certain type of use this is more convenient (e.g., use in a remote location where internet access is not available or slow).

A big advantage of employing web-services is that any dataset, any model and any tool will be accessible via the web, via a known URL; preferably even a permanent URL (PURL). The importance of web-services has been recognised by the open source GIS community that is developing various standards. Basically two kinds of web services are available:
- URLs for data numbers, and
- URLs for data graphics.

For both kinds the Open Geospatial Consortium (OGC) web-services are a promising solution: Web Map Service (WMS) for maps (images), for features and Web Coverage Service (WCS) for coverages (data). The actual implementation of the OGC spatial web services, however, is still in its early stages.

The definition stage of OGC temporal services is still on-going. OpenEarth proposes to adhere to W’S* services (WMS, WFS and WCS) as soon as easy implementations become operational. Meanwhile, OpenEarth adopted two existing web services that are already fully operational and have a large community of users: OpenEarth proposes to use the OPeNDAP protocol for accessing data numbers, and the OGC approved the Google Earth KML standard for accessing data graphics.
**Clear ownership and responsibility**
The fourth criterion is clear ownership and responsibility. Each dataset, model and/or tool has a clear owner and licence for use. OpenEarth facilitates storage, quality control and dissemination as best as possible. At the same time OpenEarth cannot assume any responsibility for the data, models and tools that users put into the system. Each user is thus individually responsible for using each dataset, model and tool with the utmost caution.

Results should always be checked carefully and users are encouraged to feed any resolved data errors and software bugs back into the system. To make ownership transparent, all data are stored under a directory with the name of the copyright holder (see protocols below). In addition, each data product is supposed to have the name of the owner as a global attribute and each raw data file from a European institute is supposed to have an INSPIRE meta-data file stored with it.

**OPENEARTH PROTOCOLS**
The OpenEarth infrastructure outlined above already involves adhering to a number of open source software/standards: Subversion, OPeNDAP, netCDF, KML, for example. However, while using these standards, still a number of important choices remain: what data storage structure to use, how to deal with units, how to deal with variable names, how to deal with coordinate projection information and so on.

Based on experiences from numerous applications and lessons learned from other initiatives, OpenEarth suggests a workflow protocol for the most important choices that invariably come up.

The next subsections briefly outline the OpenEarth protocols for handling data, models and tools (more detailed information is available from www.openearth.eu).

**Data protocol**
A well-developed protocol for data collection is made available by OpenEarth. This protocol has been developed within numerous projects, notably the EU FP 7 project MICORE and the Building with Nature innovation programme.

The data collection protocol is kept up to date at www.openearth.eu. The OpenEarth data collection procedure is modelled after the Extract, Transform and Load (ETL) process that is commonly adhered to in the world of database developers and especially in data warehousing. It involves:
- Extracting data from outside sources;
- Transforming it to fit operational needs (which can include quality levels); and
- Loading it into a database or data warehouse.

Although the actual use of the data is implicitly included, presumably in the transformation process, OpenEarth decided to make this part of the process an explicit element of the data collection procedure by extending the ETL procedure to ETL+P: Providing the data back to the user.

In the end OpenEarth put data in a database primarily so that an end-user may easily get it out again. This may seem like a trivial extension – but practical experience shows that it is not! Regularly databases are optimised for only one of the ETL steps, and the focus lies usually on making the life of data managers easier. OpenEarth aims for a system that makes the life of the end-users easier as well.

OpenEarth thus adheres to the ETL+P approach, where data use and dissemination are an integral part of the definition. ETL+P as used by OpenEarth comprises the following steps (see Figure 2 above):

1. **Extract.** Take measurements/run models and store the measured raw data files/model input in the OpenEarth repository.
2. **Transform.** Enrich the gathered raw data/model results with metadata and transform it from its original arbitrary format to the agreed upon standard format for data products (e.g., netCDF).
3. **Load.** Load the data products into the OPeNDAP database.
4. **Provide.** Provide access to the OPeNDAP database and facilitate easy dissemination to all potential end-users allowing them to easily continue to use their own favourite software.

As still a number of important choices can be made following this approach, the protocols for each of the steps are elaborated slightly below.

**Extract**
- Collect your raw data.
- Commit your raw data to the open OpenEarth raw data repository (using the Subversion client TortoiseSVN available from: http://tortoisesvn.tigris.org/) under https://svn.oss.deltares.nl/repos/openearthrawdata or to an in-house clone.
- Store your raw data (and associated transformation scripts) in the following folder structure (as shown in blue below):

```
\ <institution/copyright holder>
  \ <project number if appropriate> <descriptive name>
    \ <category>
      \ dataset name
        \ data files
        \ metadata
        \ documentation
          \ Inspire description.xml
          \ internet link.url
```

**Figure 2. OpenEarth ETL+P protocol for data collection.**
Transform
OpenEarth adopts the philosophy that raw data files can have any format, but all data products should have the same format. OpenEarth uses netCDF with the CF convention as its standard format for data products that are gridded (either in space or in time or both).

Note that the relational database approach that is recommended for ecological data with large amounts of metadata is not addressed here. Although the technical specifications for the netCDF storage format are set, still a wide range of choices is left to the user.

To enhance transferability of the OpenEarth data products a number of standards are promoted by OpenEarth transforming and enriching raw data:

- **Time**: Use the time convention suggested in the netCDF Climate and Forecast (CF) Metadata Convention (http://cf-pcmdi.llnl.gov) (use the Gregorian calendar, express time as time in <time units> since <epoch>, e.g., “days since 1970-01-01 00:00 +1:00”, always include information on the time zone etc.)
- **Spatial reference**: Include for each coordinate system the parameters that should be used for conversion to lat/lon. A useful spatial reference is provided by EPSG (http://www.epsg.org) (include in any netCDF file conversion parameters for the regional coordinate projection the data was measured in, including its EPSG code, for easy use in the regional context (x and y), as well as longitude and latitude information with a WGS84 datum, to enable easy projection on Google Earth for example)
- **Units**: Use the standards defined by the SI (http://www.bipm.org/en/si/) using the controlled units vocabulary of the UDUNITS package (http://www.unidata.ucar.edu/software/udunits). 
- **Variable names**: For variable names use the standard naming convention as suggested in the netCDF Climate and Forecast (CF) Metadata Convention (http://cf-pcmdi.llnl.gov) and the National Environmental Research Council (NERC) Data Grid Vocabularies (http://www.bodc.ac.uk/products/web_services/) as much as possible.

- **Custom standard names**: Where no ready to use standard names are available (as is the case for example for various signals in vessel log files) develop a custom standard name convention. If possible, share it via the OpenEarth website to propose the customised naming convention as a new standard.
- **Automatically added version information**: Use Subversion version keywords in the script that creates the netCDF file to enable reproducible and unambiguous data products. These keywords should be applied in a code line that writes global attributes of the netCDF file, so that each data file contains the full URL of the script that generated it, as well as its version.

**Load**
Load the newly generated netCDF data product to the OPeNDAP server for easy access. Unlike the raw data repository, the netCDF collection under an OPeNDAP server does not have automated version control.

Since all raw data have their processing scripts stored along with them, an automated procedure can easily be set up. To make ownership transparent, it is recommended to use as much as practicable the same directory structure for the OPeNDAP server as was applied in the raw data repository (see Extract).

**Provide**
The data can now easily be used by users, e.g., employing any web browser or using Matlab, Python, Fortran, R and so forth. The OpenEarth tools repository, hosted freely by Deltares, contains routines facilitating working with netCDF files and communicating with OPeNDAP servers.

The addresses of the OpenEarth community OPeNDAP servers can be found via OpenEarth.eu. Preferably each institution has its own OPeNDAP server, e.g., http://opendap.deltares.nl; and so on.

To further facilitate easy inspection of geographically (and temporally) specified data it is recommended to generate a KML file for each netCDF file to enable easy visualisation on Google Earth. The address of the OpenEarth community web server for KML files can be found via OpenEarth.eu. Preferably each institution has its own KML server, e.g., http://kml.deltares.nl.

**Models protocol**
Especially for locations where models are regularly used, model schematisations and especially the lessons learnt in developing them are not easily transferred beyond the boundaries of an individual project.

By scripting the model set-up process as much as possible and putting the model set-up scripts and the resulting model schematisations under version control they are not easily transferred beyond the boundaries of an individual project.

Note that OpenEarth considers only the model schematisations as MODELS (e.g., a mesh type grid with initial and boundary conditions). The model codes are considered as TOOLS (e.g. Fortran, Matlab), and the model output as DATA (e.g., netCDF files).

In fact, for MODELS the OpenEarth team proposes to include all components that are not strictly DATA and TOOLS, i.e., all components that are needed for complete mapping (in mathematical sense) of all input components to all output components.

- Prepare your model schematisation.
- Commit your models to the open OpenEarth model repository under https://svn.oss.deltares.nl/repos/openearthmodels or to an in-house clone.
- Store each model (and associated generation scripts) in the following structure (as shown in blue below):

```xml
\<institution/copyright holder>
\<modeltype> (choose: delt3d, mike2, sbeach, wellwaves etc.)
\<project number>/\<descriptive name>
\model set-up
\scenario
\script
\documentation
\internet link url (url to background info on the model if available)
```
Tools protocol
Core of the OpenEarth philosophy on tools is that by systematically storing, maintaining and disseminating data I/O routines, general engines and applications at a central location, slowly but surely a toolbox emerges that acts as a collective memory to which analysts and end-users naturally gravitate regarding their basic information needs. The long-term focus of the approach promotes collaboration and the exchange of ideas (across the artificial boundaries of projects, departments and organisations) which in the long run will be beneficial to any organisation that uses customised analysis tools on a regular basis.

OpenEarth suggests several ways to generate tools that are easily used and improved by others:
- Conform to the standards of the programming language of your choice (e.g., Matlab).
- Adhere to some basic conventions for well-documented tools.
  - Provide a proper help block and adequate comments to make the code understandable.
  - Include a copyright block indicating terms for use (LGPL recommended).
  - Add an example, either in the documentation or as a "_test" script (see below).
- Commit your tools to the open OpenEarth tools repository under https://svn.oss.deltares.nl/repos/openearthtools or to an in-house clone.
- Store your tools and scripts in the following structure (as shown in blue below):

```
<language>
  \_dir
    \_general
      (contains general input/output routines)
    \_applications
      (combine routines to perform specific tasks)
```

In summary, the following community repositories and web links are available openly from OpenEarth: http://www.openearth.eu – OpenEarth homepage with information on standards, tutorials and so on, and up-to-date links to the following dedicated services:
- https://svn.oss.deltares.nl/repos/OpenEarthRawData – better not checkout entirely: large!!
- https://svn.oss.deltares.nl/repos/OpenEarthModels
- https://svn.oss.deltares.nl/repos/OpenEarthTools – checkout and update regularly!
- http://opendap.deltares.nl – location of all netCDF data products

Companies that also host an internal clone of the OpenEarth infrastructure for information security reasons are recommended to follow a similar structure to facilitate exchange.

Test protocol
To ensure its quality all OpenEarth content should be rigorously tested. As all content in a specific OpenEarth instance can be freely exchanged between its users, it can be modified by all users that have access to that instance.

This urges adopting the Wikipedia-like approach to quality control: Immediate and continuous peer review rather than the one-time peer review commonly implemented at scientific journals. However, the increasingly complex computer tools that are used to analyse and convert data are a serious impediment to this process (Merali, 2010). Peers cannot be expected to go through lengthy tool codes and conversion scripts in detail to judge their quality.

To solve this issue the authors propose to adopt the scientific method in combination with component level unit tests as proposed by Kleb and Wood (2005). For each tool or data product the quintessential properties need to be tested independently using well-defined test cases. Reviewers can now simply assess the quality of a tool or data product by assessing the completeness of the set of test cases rather than having to examine the tool itself. Test cases may be added or modified if adapting the tool itself is too cumbersome.

OPENEARTH DREDGING EXAMPLES
The previous sections outlined the OpenEarth philosophy, its enabling ICT infrastructure and practice based workflow protocols. To demonstrate the potential of the OpenEarth approach for the dredging industry a number of applications as they have been implemented on dredging projects are discussed.

Facilitating eco-morphological research around dredging projects
An interesting dredging project where the use of OpenEarth as a standard for data sharing is applied is the Pilot Sand Engine Delfland project (www.dezandmotor.nl). The project itself, executed in 2010-2011 in a joint venture between Van Oord and Boskalis, is a large-scale experiment with an alternative method for coastal maintenance.

Rather than nourishing a given coastal stretch with small amounts of sediment, the Sand Engine Delfland project experiments with applying the amount for 20 years’ worth of maintenance all at once: 21.5 million m$^3$ of sand in the form of a hook-shaped peninsula between Ter Heijde and Kijkduin, on the North Sea coast of the Netherlands (Figure 3). This innovative approach to coastal maintenance is hypothesised to be more efficient and environmentally friendly as the concentrated nourishment uses a relatively small footprint and makes use of natural processes to gradually redistribute the sediment over the coastal system.

Local flora and fauna is quite capable of dealing with this gradual redistribution of sediment. Furthermore, the hook-shaped peninsula stimulates other functions besides coastal safety such as recreation and nature development. This latter aspect is a clear added value compared with the traditional approach.

A great quantity of survey data was already gathered during the execution of the project. Additionally a number of research and monitoring programmes, each with sizeable budgets and a variety of involved partners, have been initiated to study the behaviour of the innovative Sand Engine design:
- 3.5 M€ Building with Nature programme, Holland Coast Case
- 2.5 M€ EIA Measurements Rijkswaterstaat
- 3 M€ EU EFRO Grant for additional measurements Sand Engine
- 4 M€ STW Perspective Grant: NatureCoast
- 3 M€ EU ERC Grant: NEMO
An integral part of the monitoring plan was the establishment of the Sand Engine Data Lab: A central data management facility for all monitoring programmes for the benefit of current as well as future research.

The Sand Engine Data Lab is developed in close collaboration with Delft University Library and the 3TU.datacenter, and adopts the OpenEarth workflow and protocols to establish good quality data archives that can easily be shared amongst the researchers involved in the various programmes.

A data use and publication protocol has been developed to create a clear set of ‘house rules’ for the use of each other’s data within Data Lab. The involvement of Delft University Library and the 3TU.datacenter is crucial to guarantee durable data storage (also beyond the lifetime of the current research programmes) and professional data dissemination using digital object identifiers (DOI’s) per dataset.

Using system knowledge to enhance efficiency in long-term maintenance projects
The Sand Engine Data Lab is a nice example of dredging companies working together with a large number of research institutes and universities to develop new knowledge on the behaviour of an innovative design. In practice such collaborations are not abundant. A much more common setting, where system knowledge can make the difference, is for long-term maintenance contracts of navigation channels or ports.

When awarded a long-term maintenance contract it is worthwhile to develop a good understanding of how the navigation channel or harbour responds to natural variations and human-regulated events such as fluctuations in river discharge, storms, dredging and dredge material placement. Especially when a project runs for several years, storing and processing project information consistently for later analysis is extremely helpful.

Various sources of data need to be combined to develop an understanding of the system’s behaviour, viz., consecutive surveys to enable the analysis of volume changes, design levels to assess the need for dredging, production information to understand where volume changes may have been caused by natural processes or dredging or placement of material, hydrodynamic data to understand how volume changes in the project area correlate with discharge events, sediment samples to identify what kind of material is found where, and so on.

Depending on the nature of the phenomena observed, models may be used to predict/understand some parts of the systems behaviour. Think of natural sedimentation rates and hotspots during normal conditions or the advent and decay of sand dunes during high water discharge events. Such knowledge may be used to optimise dredging and placement strategies. In any case, tools are needed to analyse, visualise and report the observed and predicted system behaviour.

Figure 3. Left panel: Aerial photograph of Sand Engine Delfland. Right panel: Visualisation of measured bathymetry and topography for the same date.

Figure 4. Left panel: Measured bathymetries for the long-term maintenance of the Dortsche Kil, the Netherlands. Right panel: Visualisation of calculated volume changes (outside) and dredging and placement records (inside) for the same area. Notice that the both visualised datasets can be animated over time.
is highly dynamic. To promote economically viable dredging strategies the concept of the "nautical depth" was introduced. Nautical depth is defined as a calculated plane below which ships encounter so much resistance from the mud layer that it inhibits manoeuvrability.

Conveniently deriving this nautical depth from various sources of data requires the development and smooth applicability of tools and the availability of a large and consistent dataset. The OpenEarth workflow was used to store various available datasets in a consistent manner. As a result, five years of monthly surveys and dredging/placement info is made available for easy analysis.

Figure 4 shows a long-term maintenance project of the Dordtsche Kil and Merwede. All available survey data has been stored and processed according to the OpenEarth workflow. Figure 4 (left panel) shows a visualisation of the measured bathymetries per dredge box. Also all available dredging and placement records, again specified per dredge box, were stored and processed.

Post-processing of the available data enabled engineers to produce the visualisation in the right panel of Figure 4, which shows calculated volume changes (outside) and dredging and placement records (inside) for each dredge box. Notice that the both visualised datasets can be animated over time. Combined with information on river discharge one can develop an understanding of the system behaviour. With this understanding knowledge-based dredging strategies may be developed and put into practice.

Another example is the long-term maintenance contract for the entrance channel of Utmuiden port, the Netherlands, shown in Figure 5: Left panel, calculated ‘nautical depths’; right panel, visualisation of calculated “nautical depths” relative to design levels (green colours indicate “nautical depths” deeper than the design level; red colours indicate high spots to be dredged). The entrance channel of Utmuiden port contains large quantities of fine sediment/mud in a layer that is highly dynamic. To promote economically viable dredging strategies the concept of the “nautical depth” was introduced. Nautical depth is defined as a calculated plane below which ships encounter so much resistance from the mud layer that it inhibits manoeuvrability.

Conveniently deriving this nautical depth from various sources of data requires the development and smooth applicability of tools and the availability of a large and consistent dataset. The OpenEarth workflow was used to store various available datasets in a consistent manner. As a result, five years of monthly surveys and dredging/placement info is made available for easy analysis.
The resulting data set can now easily be visualised (see Figure 5) as well as analysed (OPeNDAP, netCDF, Matlab). Work is on-going to increase understanding of the available datasets. The use of models is considered to help better understanding of the systems behaviour.

**Advanced project reporting and strategic knowledge development**

A third example deals with the analysis, reporting and visualisation of deep compaction activities in the Middle East. Figure 6 shows a reclamation that has been divided into so-called control boxes. For each of these control boxes a complicated analysis must be performed to determine if bearing capacity and resistance against liquefaction are sufficient.

Initially the client requirements were implemented in an Excel sheet. Anywhere between three and eight cone penetration test (CPT) files were manually loaded in this sheet that took some 20 minutes to iterate to the final verdict: Compliant/non-compliant.

Though this approach had worked before on smaller-scale projects, the sheer number of tests and control boxes (thousands) required a more advanced approach. Not only to reduce the workload, but especially to be able to learn lessons from the work already done by, e.g., accessible visualisation of test results.

Following the OpenEarth philosophy, a data management and IT infrastructure was implemented. Key aspects were that:
- data would be checked, stored and structured by personnel on site;
- analysis tools would be developed by experts based in the Netherlands;
- data products (reports, visualisations) would be available on site as well as in the Netherlands; and
- the application of more efficient tools on-site would help site personnel to focus on soil mechanics rather than programming Excel sheets.

Above key aspects may only be achieved when a fixed protocol is available to assess the bearing capacity. First of all per box a fixed minimum number of CPTs has to be performed at predetermined locations.

Analysis of this data and a comparison to the bearing capacity requirements imposed by the client yields either success or failure of the control box. In case of insufficient bearing capacity additional measures have to be taken, e.g., in the form of a re-compaction effort using either the vibro-compaction method, rapid impact compaction or surface rolling. After the additional measures, new CPTs have to be performed to reassess the control box.

One can imagine that depending on the size of the reclamation the number of data files and analysis steps can quickly get out of hand. Next to the fact that soil improvement projects are already logistically challenging in themselves, it is important to realise that subsequently demonstrating the quality per control box is equally challenging.

To get a feeling for scale it is not uncommon to have projects with several thousand control boxes each requiring between the 40–80 vibro poker actions and 3–8 CPTs to demonstrate sufficient quality. Often payment of the project is coupled with the approval of the control boxes. One can imagine that in such projects careful data storage and easy access, combined with effective and efficient data processing, visualisation and reporting scripts is of crucial importance.

To limit the number of (highly trained) project engineers and data secretaries needed to collect, check, process, analyse and visualise the data, the whole process is automated as much as possible. After adopting the OpenEarth workflow for ground improvement works, this process is streamlined such that just one project engineer, supported by a small number of data secretaries, is now
CONCLUSIONS

OpenEarth as a whole (philosophy, user community, infrastructure and workflow) presents a comprehensive approach to handling data, models and tools that actually works in practice at a truly significant scale:

- The infrastructure is built from the best open source components available, and thus free to use and easy to implement, and the associated workflow is based on widely accepted and open international standards as much as possible.
- The workflow is implemented effectively not only at its originally founding organisations Delft University of Technology and Deltares in a research setting, but also at Van Oord in large commercial dredging projects where proper handling of (big) data, models and tools is of direct influence on the project success.

Practical applications in various research programmes and projects show that management of data, models, tools and knowledge can indeed be lifted to a higher level. Otherwise competing organisations now work together exchanging information via the OpenEarth repositories.

But also within companies collaboration between departments is improved:

- Sharing the most generic datasets, models and tools has clear positive spin-off in the sense that many basic analyses can be performed much more efficiently. This facilitates that more work can be done given the same amount of available resources.
- The practical examples illustrate the potential of OpenEarth for dredging projects. For data, models and tools that are truly strategic and really cannot be shared, OpenEarth stimulates the set-up of internal OpenEarth clones. This way the OpenEarth workflow can still be adopted, promoting collaboration within the organisation, while taking care of security considerations at the same time.

Just like other quality systems, OpenEarth cannot guarantee the quality of the analysis but it adds to the complete transparency and durable accessibility of the data products, models and tools used in the process. Various practical applications of the OpenEarth workflow have demonstrated the power of collaboration and the positive effects of the project transcending approach.

REFERENCES


