OpenEarth-Inter-Company Management of: Data, Models, Tools & Knowledge

Mark van Koningsveld
Van Oord BV
68 PUBLICATIONS **1,249** CITATIONS
SEE PROFILE

Gerben J. de Boer
Van Oord BV
28 PUBLICATIONS **483** CITATIONS
SEE PROFILE

Fedor Baart
Deltares
64 PUBLICATIONS **550** CITATIONS
SEE PROFILE

C. (Kees) den Heijer
Deltares
27 PUBLICATIONS **199** CITATIONS
SEE PROFILE

OPENEARTH - INTER-COMPANY MANAGEMENT OF: DATA, MODELS, TOOLS & KNOWLEDGE

M. van Koningsveld[1, 2, 3], G.J. de Boer[2, 3, 4], F. Baart[5], T. Damsma[3, 6], C. den Heijer[7,8], P. van Geer[9], B. de Sonneville[10]

ABSTRACT

Research, consultancy as well as construction projects commonly 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 practice at a truly significant scale. It is implemented effectively not only at its originally founding organizations Delft University of Technology and Deltares but also in a number of sizeable research programs with multiple partners (such as the 28 million euro 4-year research program Building with Nature – 19 partners from 1 country) from multiple countries (such as the 4.6 million euro 3-year EU FP7 research program MICORE – 15 partners from 9 countries). For data, models and tools that are truly strategic and really cannot be shared, OpenEarth stimulates the setup of internal OpenEarth clones. This way the OpenEarth workflow can still be adopted, promoting collaboration within the organization, while taking care of security considerations at the same time.

This paper describes the OpenEarth philosophy, its infrastructure and main workflow protocols. To illustrate the systems practical effectiveness the paper finishes with an example of OpenEarth as it is applied in Building with Nature.

Keywords: workflow management, protocols, web services, version control, OPeNDAP, netCDF

INTRODUCTION

The sustainable interaction between mankind and planet earth poses huge hydraulic and environmental engineering challenges. Confronting these challenges one-project-at-a-time, while apparently 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. Hardly any project is by itself of sufficient scale to develop easy-accessible and high-quality data archives, state-of-the-art modeling systems and well-tested analysis tools under version control. Research, consultancy as well as major construction projects commonly spend a significant part of their budget 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, due to increasing complexity, large projects nowadays are regularly executed by consortia. The internal services of individual institutions do not

[1] Senior Engineer, Van Oord Dredging and Marine Contractors bv, PO Box 8574, 3009 AN Rotterdam, The Netherlands. Email: mrv@vanoord.com

[2] Also at: Delft University of Technology, PO Box 5048, 2600 GA Delft, The Netherlands

[3] Also at: EcoShape|Building with Nature, Burgemeester de Raadtsingel 69, 3311 JG Dordrecht, The Netherlands

[4] Researcher, Deltares, PO Box 177, 2600 MH Delft, The Netherlands. Email: gerben.deboer@deltares.nl

[5] PhD Candidate, Delft University of Technology, PO Box 5048, 2600 GA Delft, The Netherlands. Email: f.baart@tudelft.nl

[6] Engineer, Van Oord Dredging and Marine Contractors bv, PO Box 8574, 3009 AN Rotterdam, The Netherlands. Email: tda@vanoord.com

[7] PhD Candidate, Delft University of Technology, PO Box 5048, 2600 GA Delft, The Netherlands. Email: c.denheijer@tudelft.nl

[8] Also at: Deltares, PO Box 177, 2600 MH Delft, The Netherlands

[9] Researcher, Deltares, PO Box 177, 2600 MH Delft, The Netherlands. Email: pieter.vangeer@deltares.nl

[10] Researcher, Deltares, PO Box 177, 2600 MH Delft, The Netherlands. Email: ben.desonneville@deltares.nl

allow for collaboration due to technical limitations or simply denial of permission. As a result the way data, models and tools are currently managed, while presumably aimed at protecting the knowledge capital of organizations, in fact also inhibits (individual as well as collective) progress.

Over many years Delft University of Technology and Deltares, together 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). Rather 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 organizations (or at least departments). Put in practice OpenEarth exists only because there is 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 practice at a truly significant scale. It is implemented effectively not only at its originally founding organizations Delft University of Technology and Deltares but also in a number of sizeable research programs with multiple partners (such as the 28 million euro 4-year research program Building with Nature – 19 partners from 1 country) from multiple countries (such as the 4.6 million euro 3-year EU FP7 research program MICORE – 15 partners from 9 countries). As a result OpenEarth is now carried by a rapidly growing user community that currently (March 2010) consists of well over 100 users, approximately 60 active developers, originating from tens of organizations and multiple countries. Together they share and co-develop thousands of tools, giga-bytes of data and numerous models (source code as well as model schematizations).

This paper describes the OpenEarth philosophy, its infrastructure and main workflow protocols. To illustrate the systems practical effectiveness the paper finishes with an example of OpenEarth as it is applied in Building with Nature.

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 etc. Past experience from sizeable consultancy projects as well as numerous research programmes (e.g. Capobianco, 1999; Wilson, 2002) has 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 etc. As a result the workflow in projects is now highly traceable and reproducible with the main aim to achieve the best possible grip 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. According to the British Standards Institution ISO 9001 is the world’s most established quality framework, being used by close to 900,000 organizations in 170 countries worldwide.

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 applied too rigidly. It is commonly acknowledged that certification to, for example, an ISO 9001 standard does not guarantee any quality of end products and services. It only certifies that formalized 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 management documents are generally of no further practical use, except perhaps in legal cases.

Many attempts have been undertaken to deal with the above addressed 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 there are no more incentives to maintain the databases and web links, slowly but surely they are forgotten. When a new project comes along it seems more attractive to set up a new database rather than revive the old one. 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. It is easy to see that when such routines are developed by each project from scratch a lot of time, money and effort is wasted (NB: some wheel-reinventing can certainly serve an educational purpose, but it should not become the general practice).

A common cause for numerous existing data, models and tools management initiatives to 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 focusing on just the technology (even though the use of proper technology is obviously important). It effectively provides and maintains all required technological infrastructure *in-support-of* yet *independently-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 centralized repositories already during the course of a project, rather than at its end. A small team invest 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 open as possible …
- across the artificial boundaries of projects and organizations …
- 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 involves that all results should be shared between the legal project contributors minimally and with the whole world preferably. For any individual research project or organization, full openness quite likely benefits competing consortia or organizations. At the same time cooperation results in greater overall progress aligning individual with total progress so that they reinforce rather than impede each other. 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.

The difficulty to come to optimal decisions in such non-zero-sum games is among others associated with difficulties in assigning value needed to estimate relevant payoffs (cf. Flood, 1958) and issues related to trust (Macy and Skvoretz, 1998). Assigning value to particular datasets, models or tools for the purpose of a rational cost-benefit analysis is obviously very difficult and subjective. Nonetheless the general importance of data, models and tools for hydraulic engineering problems and the potential added value of sharing is widely acknowledged. Maintaining productive collaboration in noisy suspicious settings is also a well-known problem. Coleman (1990), for example, already discussed the potential detrimental effect of the sometimes unavoidable (and unintentional) time lag between promise and delivery in collaborative projects. Klapwijk (2009) and Klapwijk and Van Lange (2009) describe the power of generosity, combined with reciprocity, in such settings. Generosity involves investing slightly more than one has received from the other, reciprocity involves responding immediately and in kind to a partners behaviour (Kollock, 1993). For open source initiatives like OpenEarth to work in the long run accepting some (or even great) asymmetry in reciprocity in combination with the continued presence of at least one generous and forgiving partner is very important. The OpenEarth initiators are dedicated to playing that role. For data, models and tools that are truly strategic and really cannot be shared, OpenEarth stimulates the setup of internal OpenEarth clones. This way the OpenEarth workflow can still be adopted, promoting collaboration within the organization, while taking care of 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:

- Accidentally using older versions of data, models or tools
- Not knowing where the most recent version is and what its status is
- Making the same mistake twice due to lack of control over versions
- Losing important datasets that are extremely hard to replace
- Uncertainty as to what quantities have been measured and which units apply
- Uncertainty as to the geographical location of measurements
- Uncertainty as to the time and time zone the measurements where taken in
- Lack of insight in the approach taken and the methods used
- Myriad formats of incoming (raw) data
- Getting the feeling that a certain issue must have been addressed before by another analyst
- Running into a multitude of tools for the same thing
- Running into a multitude of databases each in its own language and style
- Etc.

Although the above-described frustrations are very common throughout the hydraulic engineering industry it seems that no practical and widely accepted remedy is 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 a lot of 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 an ftp server, data archives can quickly become messy. When a typical SQL type relational database is employed, the number of people that can actually use and maintain the data archives generally becomes very small. Both cases are not very desirable as system maintenance inevitably becomes problematic.

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 or code. 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 followed this approach devising an infrastructure to support a bottom-up approach for long-term project-transcending collaboration adhering to four basic criteria:

1. **open standards & 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. **centralized access** - The collection and dissemination procedure for data, models and tools should be web-based and centralized 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.

The first criterion, open standards & open source, is adopted to maximize the number of participants. Known bottlenecks for implementing a new data and knowledge management system are high setup costs and a fear for 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 there is a large open source community on the web. International groups such as the Open Geospatial Consortium (OGS) and numerous meteorological, oceanographic and remote sensing collectives have created high-quality software suitable for our purposes. In addition, the requirement of the United States government that all data, models and tools funded by US tax-payers should be available openly has supplied a vast range of free software. An approach that clearly deserves a wider following

The second criterion, complete transparency, is achieved by demanding that collected data is reproducible, unambiguous and self descriptive. An important distinction we make here is between the archival and the dissemination function of a database. To eliminate ambiguity and enhance self descriptiveness, OpenEarth decided to store the generally pluriform raw data files in a version controlled repository with along side them a routine to transform each data format into one single commonly accepted data format, i.c. netCDF. The raw data and associated conversion scripts are stored to ensure reproducibility, whereas the common data format promotes un-ambiguity and self descriptiveness (see Figure 1).

This distinction has proven effective in other fields of application as well. In the remote sensing community, for instance, NASA stores **raw data** from ocean color sensing satellites as so-called L0 files. These files generally are not available in an easy accessible format as they are optimized to maximize 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 minimally the satellite locations, and stored as L1 data. NASA also adopts a standard exchange format (i.c. HDF). Further processing is carried out to translate sensor readings to geophysical quantities (L2), and generate data for climatologies on standard grids (L3 and L4). The levels L1 and higher are considered **data products** and are primarily meant for dissemination. You can recreate each level (except L0) with different processing steps, using the same open source software (SeaDAS[11]). 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. Due to human errors and progressing knowledge, data products are ephemeral entities in reality 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. Data without a version number should not be considered as data at all. In line with NASA, OpenEarth prescribes that all data processing scripts needed to transform and enrich the raw data should be stored along side the raw data. This enables automated data processing.

To ensure reproducibility OpenEarth currently uses the open source version control system Apache **Subversion** (next called subversion in 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 setup of separate repositories for data, models and tools.

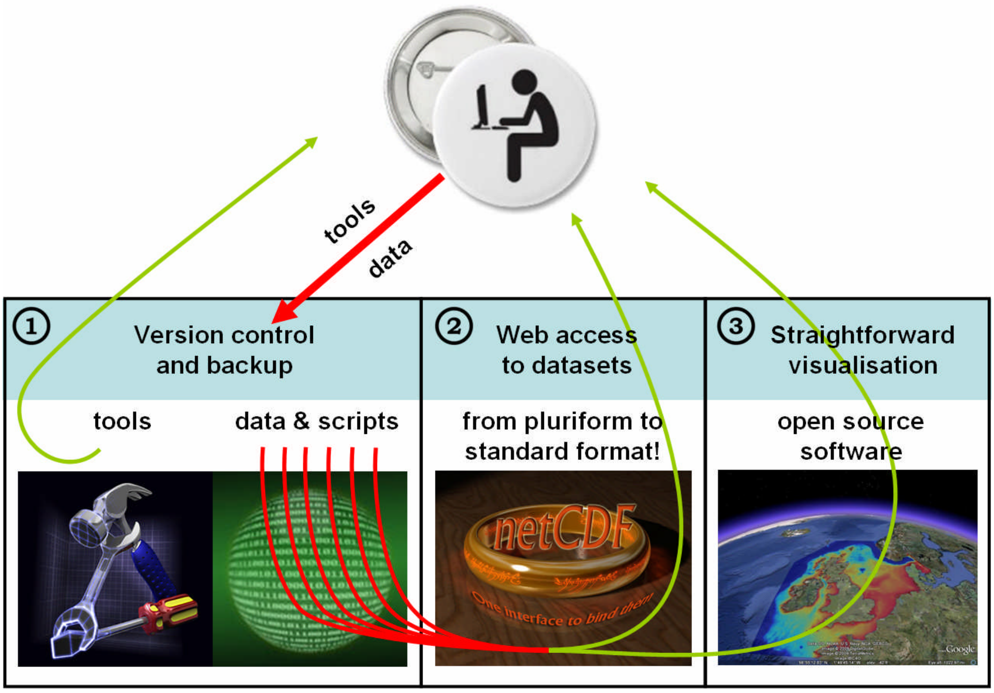
tools
data
① Version control and backup
tools
data & scripts
② Web access to datasets
from pluriform to standard format!
netCDF
One interface to bind them
③ Straightforward visualisation
open source software

Figure 1. Overview of the OpenEarth infrastructure and workflow.

[11] tool: http://oceancolor.gsfc.nasa.gov/seadas; reprocessing: http://oceancolor.gsfc.nasa.gov/REPROCESSING/

The third criterion, centralized 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, we are fully dependant on 'the cloud'. Computing and storage in general move towards common commodities that can best be provided on centralized large servers (Carr, 2008). Strangely for data, models and tools old-fashioned approaches like storing data decentralized on local PCs are still widely spread. 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 system 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 recognized by the open source GIS community that is developing various standards. There are basically two kinds of web services:

- 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), Web Feature Service (WFS) 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 ongoing. 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. First of all OpenEarth proposes to use the **OPeNDAP** protocol for accessing data numbers, and the OGC approved **Google Earth KML** standard for accessing data graphics.

The fourth criterion is clear ownership and responsibility; each dataset, model and/or tool has a clear owner and license for use. OpenEarth facilitates storage, quality control and dissemination as good a 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: netCDF, subversion, KML etc. However, *using* these standards, still a number of important choices remain: how to deal with units, how to deal with variable names, how to deal with coordinate projection information etc. Based on experiences from numerous applications and lessons learned from other initiatives OpenEarth suggest a workflow protocol for the most important ones. The next subsections briefly outline the OpenEarth protocols for handling data, models and tools (more detailed versions are available from www.openearth.eu).

Data protocol

A well developed protocol for data collection is made available by OpenEarth (via www.openearth.eu). This protocol has been developed within numerous projects, notably the EU FP 7 project MICORE and the Building with Nature programme. The data collection protocol is kept up to date at OpenEarth.eu. The OpenEarth data collection procedure is modeled 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 we 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 optimized for only one of the ETL steps, and usually the focus lies at 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 in integral part of the definition.

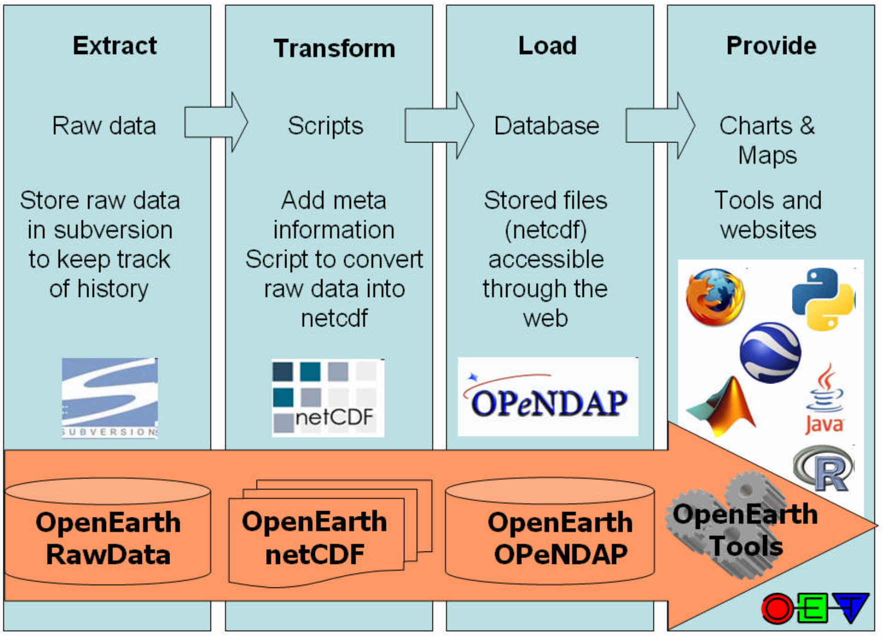

Figure 2. OpenEarth ETL+P protocol for data collection.

ELT+P as used by OpenEarth comprises the following steps (see Figure 2):

1. *Extract*. Take measurements/run models and store the measured raw data files/model input on 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 (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 favorite software.

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

Extract

- Collect your raw data
- Commit your dataset into the OpenEarth raw data repository (using the subversion client TortoiseSVN available from: http://tortoisesvn.tigris.org/) under https://repos.deltares.nl/repos/OpenEarthRawData
- Store your raw data (and associated transformation scripts) in the following folder structure:

```
\_institution
   \_your_dataset_title
     \_dataset_name1 (e.g. elevation_data, cpt_data, vc_data)
       \_raw
         \_data file1
         \_data file2 …
       \_cache (for storing intermediate data products)
       \_scripts (routines)
       \_documentation (information regarding dataset)
     \_inspire_description.xml
     \_internet_link.url
```

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. 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 is 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 in unix epoch as "days since 1970-01-01 00:00 +1:00", always include information on the time zone etc.)
- **Spatial reference** Use spatial reference information as provided by EPSG (http://www.spatialreference.org) (always include in any netCDF file 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** Use for variable names as much as possible 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/)
- **Custom standard names** Where no ready to use standard names are available (as is the case for example for various vessel log files) develop a custom standard name convention and share it via the OpenEarth website to propose the customized 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. All data conversion scripts can be run automatically, for instance triggered by subversion after an update has been made.

Because a data product can be considered as a release of raw data, sufficient checks should be performed that the netCDF is correct. It is recommended to store a test script (along with the data) that can be executed automatically as well before the data are uploaded (see test protocol below). As long as there is no test script, the OpenEarth editor should apply some elementary test before uploading. To make ownership transparent, it is recommended to use exactly 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 etc. The OpenEarth tools contain routines facilitating working with netCDF files and communicating with OPeNDAP servers. The address of the OPeNDAP server can be found via OpenEarth.eu. Preferably each institution has its own OPeNDAP server, e.g

http://opendap.deltares.nl

To further facilitate easy inspection of the data it is recommended to generate a KML file for each netCDF file to enable easy visualization on Google Earth. The address of the 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 schematizations and especially the lessons learned in developing them are not easily transferred beyond the boundaries of an individual project. By scripting the model setup process as much as possible and putting the model setup scripts and the resulting model schematizations under version control the efficiency by profiting/learning from past experiences increases. Note that OpenEarth considers only the model schematizations 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).

- Prepare your model schematization
- Commit your models into the OpenEarth model repository (using the subversion client TortoiseSVN available from: http://tortoisesvn.tigris.org/) under https://repos.deltares.nl/repos/OpenEarthModels
- Store each model (and associated generation scripts) in the following structure:

```
\_modeltype (choose: delft3d, mike21, tass, worldwaves etc.)
   \_your_model_title (e.g. prnumber_prname)
     \_model setup1
       \_scenario1
       \_scenario2
     \_scripts
     \_documentation (information regarding the model)
     \_internet_link.url
```

Tools protocol

Core of the OpenEarth philosophy on tools is that by systematically storing, maintaining and disseminating data I/O routines, engines, applications and programs 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 organizations) which on the long run will be beneficial to any organization that uses customized analysis tools on a regular basis.

OpenEarth prescribes 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):
  - Use a name space convention for files that belong to the same group
- Adhere to some basic conventions for well documented tools (use oetnewfun.m for an easy start in Matlab)
  - Start each routine with a one line description summarizing the routines main purpose
  - Provide a proper help block and adequate comments to make the code understandable
  - Use keyword value pairs for input variables
  - 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 into the OpenEarth tools repository (using the subversion client TortoiseSVN available from: http://tortoisesvn.tigris.org/) under https://repos.deltares.nl/repos/OpenEarthTools
- Store your tools and scripts in the following structure:

```
\_io (contains general input output routines
\_general (general routines useful to multiple applications)
\_applications (combine general routines to perform specific task)
  \_application1 (e.g. mike21, delft3d, tass, etc.)
  \_application2
```

In summary the following repositories and web links are available from OpenEarth:

- http://www.openearth.eu (OpenEarth homepage with information on standards, tutorials etc. and up-to-date links to the following dedicated services:
  - https://repos.deltares.nl/repos/OpenEarthRawData (better not checkout entirely – large!!)
  - https://repos.deltares.nl/repos/OpenEarthModels
  - https://repos.deltares.nl/repos/OpenEarthTools
  - http://opendap.deltares.nl (location of all netCDF data products)
  - http://kml.deltares.nl/ (location of KML files for easy inspection of data via Google Earth)

Test protocol

To ensure its quality all OpenEarth content should be rigorously tested. As all content in OpenEarth is open it can be modified by all OpenEarth users. This enables 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 analyze and convert data are a serious impediment to this process. Peers cannot be expected to go though lengthy tool codes and conversion scripts in detail to judge its quality.

To solve this issue we propose to adopt the scientific method in combination with component level units 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. These tests should also be coded in a script such that the tests can be performed automatically. For each test a strict result in terms of either true or false should be defined. A tool or data product passes quality control if and only if when all unit tests are true. Most of the unit tests can be very simple as only quintessential properties are tested separately. To test behavior for large datasets, regression type tests can be defined that simply require identical results as the previous time. For graphical tools a user can be prompted to respond y/n, although this does not allow for automation. The 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. Reviewers can now simply add or change a test case if adapting the tool it self is too cumbersome.

To maximize the quality of OpenEarth content, preferably not only end products should be tested, but also each component. For tools this means separate test for subroutines/functions, for data products this means testing also intermediate results. This is especially an issue for Matlab code in OpenEarth because Matlab lacks a well tested testing framework. In OpenEarth the convention is that each function has a function with the same name with extension '_test' stored alongside it. Using this apparent name, the test function can also serve as an example on how to call the main function. For all Matlab tools a testing framework has been designed that runs all '_test' functions, and publishes the results to the OpenEarth wiki.

OPENEARTH EXAMPLES

To demonstrate the potential of the OpenEarth approach we discuss the Holland Coast case, one of the four case studies of the Building with Nature research program (http://www.ecoshape.nl). This case develops alternative strategies for the sustainable development of the Dutch coast from the Hook of Holland up to Den Helder (see Figure 3), over a timescale of 50 to 100 years. It deals with a range of possible measures, both for sand mining and coastal interventions, addressing (1) the wide range of possible approaches to enhancing the natural potential of a site or design and (2) alternative methods to work with natural processes rather than against them.

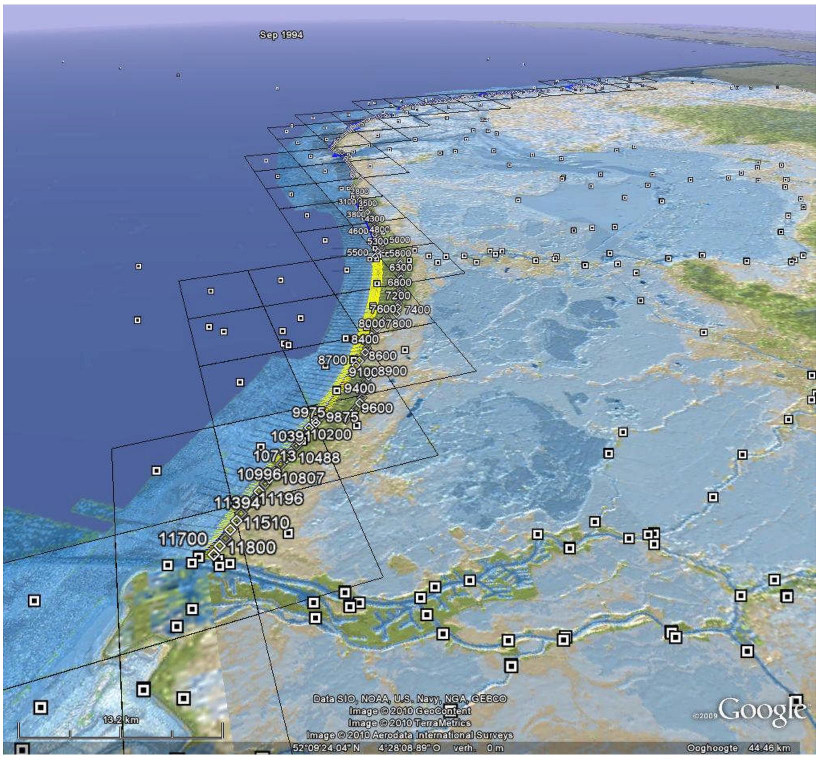
Sep 1994
9975
9875
9600
10200
10488
10807
11196
11510
11700
11800
13.2 km
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Image © 2010 GeoContent
Image © 2010 TerraMetrics
Image © 2010 Aerodata International Surveys
©2009 Google
52°09'24.04" N 4°28'08.89" O verh. 0 m
Ooghoogte 44.46 km

Figure 3. Overview of data already available in OpenEarth to the Holland Coast case. Visible are the JarKus transect data near the coast (yellow), the JarKus grids overview (black mesh), the general elevation data (AHN) for the dry part of the Netherlands, the VakLodingen data (bathymetric grids) for the near shore coastal bathymetry and the Waterbase waterlevel data conveniently visualized on Google Earth (square dots).

Availability of important datasets

The sandy Holland coast offers many important services such as providing a steady high quality drinking water supply, attractive space for recreational, residential and industrial use and an extensive and diverse habitat for a great number of often rare and endangered species. Although the sandy dunes constitute only an approximate 1 % of the Netherlands' surface area, more than 60 % of the Dutch flora occurs there (TAW, 2003). Besides these important functions, the main function provided by the sandy coast is to protect the low-lying hinterland from flooding. The Netherlands is well-known for its advanced coastal protection policies and in association a lot of projects have dealt with this particular stretch of coast before. Invariably each of these projects always dealt with (subsets of) Rijkswaterstaat's datasets on bathymetry (transects as well as grids), hydrodynamics (waves, currents and waterlevels) and water quality (TSS, etc), TNO's datasets on topography (the general elevation data of the Netherlands), bathymetry (Dutch continental shelf bathymetry) and soil composition (soil types and grain size distributions), and KNMI's datasets on meteorology (wind and pressure fields).

All these datasets are now available through OpenEarth in readily accessible uniform netCDF files (see Figure 3). The datasets are stored in raw form in the OpenEarthRawdata repository (Box 1 in Figure 1). The same repository also contains the Matlab and Python scripts that can process these data into netCDF files. The raw data have been enriched with meta information and transformed to netCDF. These files are stored on an OPeNDAP server (Box 2 in Figure 1). This server allows everyone to make calculations with these data without the need to download the full data collection that covers over 10 GB. Finally, for non-specialists that do not need to perform calculations, all data are also presented as readily processed KML feeds for straightforward visualization on Google Earth (Box 3 in Figure 1). Because all this data is already available, the Holland Coast case needs to reserve less of its budget for gathering and processing of historic external datasets. As a result more budget remains for additional data acquisition and/or more detailed (data) analysis and reporting. Just as the Holland Coast case benefits from all this previously gathered data, any future project will benefit from the additional data gathered specifically by this project and new or improved tools for data analysis.

Availability of important basic analysis tools

Besides the use of various datasets a number of important coastal state indicators is associated with the coastal policies that apply to this coastal region. The Dynamic Preservation policy (Min V&W, 1990), for example, guarantees sustainable preservation of safety and of values and functions in the dune area. The term 'sustainable' is interpreted as 'for a period of several years' (~ 10). The main policy tactic is to maintain the coastline at a position not landward of the 1990 reference by a process of regular nourishment. The decision recipe for this policy is based on a volume trend approach applying the Momentane Coastline (MCL) concept. This concept is used in a maintenance indicator comparing the extrapolated expected coastline position (TCL) with a reference value: the 1990 coastline position (BCL) (see Figure 4) (TAW, 1995; Van Koningsveld and Mulder, 2004).

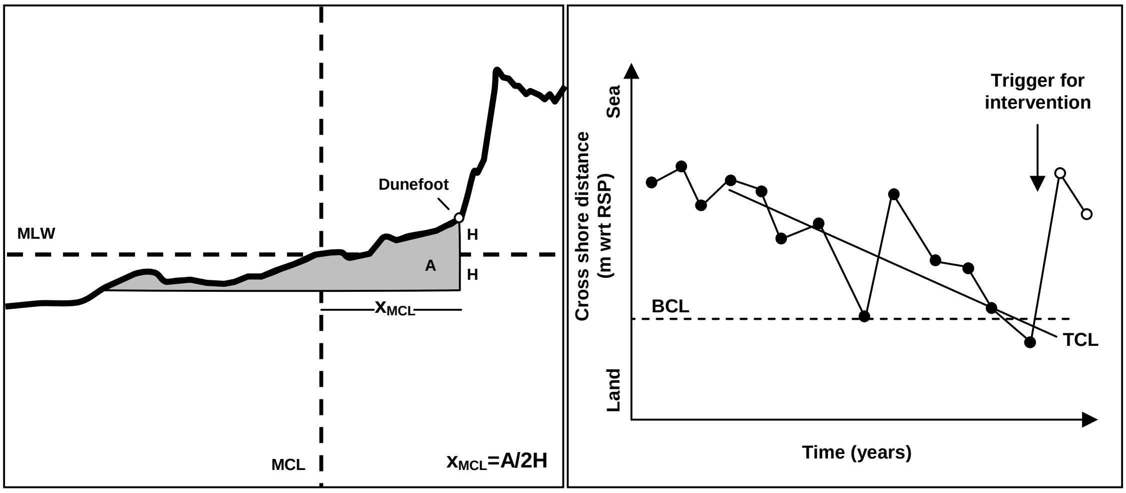

Figure 4. Left panel: definition sketch of the quantitative state concept MCL. Right panel: illustration of how for a given transect a number of MCL positions is used to compare the TCL (10 year trend) with the BCL. Intervention is considered if the TCL moves landward of the BCL.

Another example is the Flood Defence Act (TK, 1996) and the dune strength indicator it uses to guarantee the safety of the hinterland against flooding. The policy tactic in this case is to ensure that the dune strength satisfies some predetermined safety limits. The Holland Coast dunes, protecting the most densely populated part of the country representing the highest economic value, have to resist a storm flood with a probability of occurrence of

1:10,000 a year. The associated probability of dune failure (breach) is a factor 10 smaller, viz. 1:100,000 a year. A first deterministic step in any coastal safety assessment is the calculation of dune erosion for a given cross shore profile and predetermined design water level, wave height, wave period and sediment fall velocity. With these parameters, an equilibrium post storm erosion profile can be determined (see Figure 5). This equilibrium profile should then be 'fit' into coastal profile at hand. The position of the erosion profile in the vertical sense is determined by the computational water level (In Dutch: Rekenpeil). The position in the horizontal sense is determined by iteratively positioning the erosion profile over the coastal profile in such a way that an erosion-sedimentation balance is obtained in the direction perpendicular to the coast. The resulting profile is assumed to be the equilibrium profile after the storm. For more information see TAW (1984) and TRDA (2006).

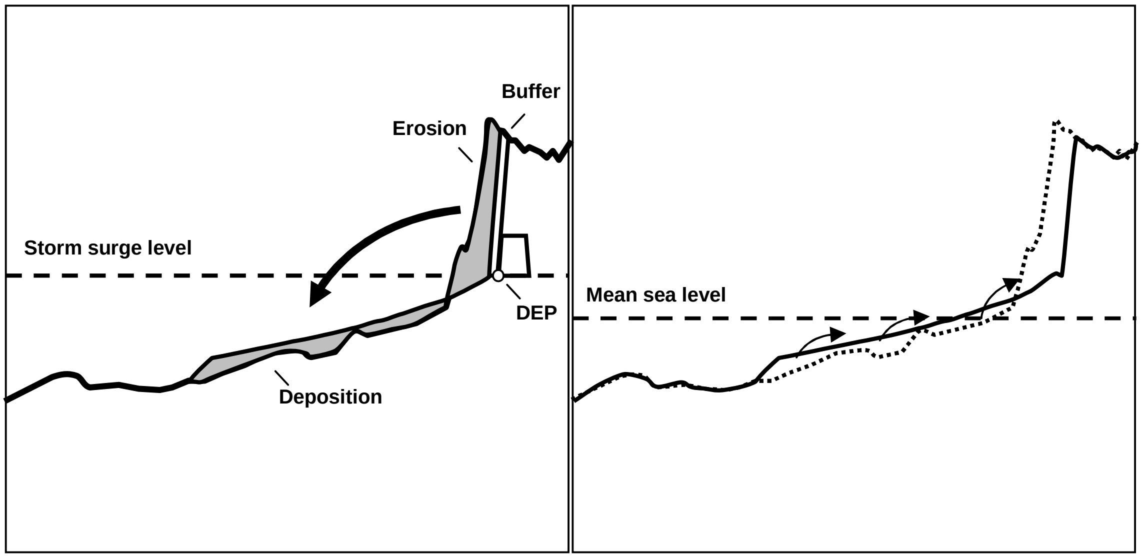

Figure 5. Left Panel: Dune Erosion Point (DEP) definition sketch (time scale hours). Right Panel: Coastal profile regeneration under calm conditions (time scale months)

Whatever innovative sustainable development strategies the Holland Coast case manages to develop, the impact will always (at least) have to be expressed in terms of these indicators: what is the positive/negative impact on coastline maintenance? What is the positive/negative impact on coastal safety? Since well tested tools associated with both indictors are already available from the OpenEarthTools repository, the Holland Coast case needs to reserve less of its budget for tool development. As a result, more budget remains for thorough application of these tools to the also available datasets. The availability and accessibility of data and tools in this case challenges the analyst to apply his/her analysis routines not to just one transect or a couple of transects near the area of interest, but simply to all transects available in the database. This is an excellent way to (1) see whether or not the routine is robust enough to deal with variations/flaws in the data, and (2) to see whether an observed phenomenon can be detected throughout the dataset or just in a small sub selection of the data.

Currently the Building with Nature consortium is executing a field campaign at Vluchtenburg, located in the southern region of the Holland Coast. Data that is collected ranges from Argus video imagery, Aeolian sediment transport, grain size distributions and 3D laser mapping bathymetries to jetski based near shore bathymetries. This data, gathered by various partners, is already routinely stored in the OpenEarthRawData repository and will in the near future become available through OpenEarth for other projects as well.

FUTURE DEVELOPMENTS

The OpenEarth initiative has gradually been developed over many years. The infrastructure and protocols as described above are the result of numerous concepts that have been probed in the past. Some experiments were successful, others were not. For instance, employing a relational (SQL) database for 2D data matrices did not prove viable concept, whereas disseminating a netCDF collection via an OPeNDAP server did. Also an early protocol that each tool name should start with either 'get' or 'put' was not workable. The name space convention where each tool name starts with the group name to which it belongs is more useful. In the future OpenEarth will continue to experiment with new (variations on) technologies and new protocols as they come along. Because all OpenEarth modules serve a single purpose and are open source, there is no critical dependency on any of the components, and any one can be replaced. The infrastructure and protocols as described above are therefore bound to evolve in the future.

Before OpenEarth, common complaints regarding data set access were that (1) a dataset's existence was not known, (2) when it was known you often did not have the privileges to work with it, and (3) when you did, you had to waste a lot of time parsing the wide variety of formats. With the OpenEarth system operational, the complaints become totally different: how can one find the correct data in the overwhelming collection. To resolve this issue OpenEarth is currently experimenting with meta-data crawlers/harvesters. The uniformity of the netCDF collection on an OPeNDAP server opens up the possibility to extract the meta-information from all files automatically (e.g. begin/end time, geographical bounding box, available parameters). This meta-information can subsequently be stored in a small cache (relational) database to allow for a quick meta-search through all the data. Prototype harvesters have been implemented in Python and Matlab. Any OPeNDAP server can be crawled, and the meta-data data can be cached locally. This means that even remote servers, such as the USGS and NOAA OPeNDAP servers, can be searched effectively.

With the OpenEarth approach all data, models and tools have a fixed web address. In collaboration with the Delft University of Technology library OpenEarth is currently experimenting with assigning doi's to them. A doi (digital object identifier) is a permanent url for which the world-wide doi organization maintains a resolving table that contains an up-to-date link to the object. Currently these objects are scientific manuscripts, but extension to repository addresses for raw data and tools and OPeNDAP addresses for published data is possible. This would allow for direct citation of datasets and tools.

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.
  - It is implemented effectively not only at its originally founding organizations Delft University of Technology and Deltares but also in a number of sizeable research programs with multiple partners from multiple countries.
  - The infrastructure is free and clonable, built from the best open source components available, and the associated workflow is based on widely accepted and open international standards as much as possible.
- Practical applications in various research programs and projects show that inter-company management of data, models, tools and knowledge *can* actually work in practice. Otherwise competitive organizations now work together exchanging information via the OpenEarth repositories.
  - 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.
  - For data, models and tools that are truly strategic and really cannot be shared, OpenEarth stimulates the setup of internal OpenEarth clones. This way the OpenEarth workflow can still be adopted, promoting collaboration within the organization, 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 *can* guarantee the complete transparency and durable accessibility of the data products, models and tools used in the process.
- OpenEarth demonstrates the power of collaboration and welcomes interested individuals to join, use and further contribute to the data, models and tools already available.

REFERENCES

Carr, N. (2008). *The Big Switch: Rewiring the World, from Edison to Google*. ISBN 978-039306228

Capobianco, M. (1999). *The role and use of technologies in relation to ICZM*. Final report, Venice, 26-03-1999. Document prepared for the EU demonstration programme on integrated management in coastal zones (1997-1999). Contract B4-3040/96/000599/MAR/D2

Flood, M.M. (1958). "Some experimental games." *Management Science*, **5(1)**, pp. 5–26.

Hardin, G. (1968). "The Tragedy of the Commons." *Science*, **162**, pp.1243-1248

Klapwijk, A. (2009). *The Power of Interpersonal Generosity*. PhD Thesis, VU University, Amsterdam, the Netherlands.

Klapwijk, A. and Van Lange, P.A.M. (2009). “Promoting Cooperation and Trust in “Noisy” Situations: The Power of Generosity.” *Journal of Personality and Social Psychology*, **96(1)**, pp. 83–103. doi: 10.1037/a0012823.

Kleb, B. and Wood, B. (2005). “Computational Simulations and the Scientific Method.” *17th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2005-4873.

Kollock, P. (1993). “An Eye for an Eye Leaves Everyone Blind: Cooperation and Accounting Systems.” *American Sociological Review*, **58**, pp. 768-786

Macy, M.W. and J. Skvoretz (1998). “The Evolution Of Trust And Cooperation Between Strangers: A Computational Model.” *American Sociological Review*, **63**, pp 638-660

Min V&W (1990). *Coastal defence after 1990, a policy choice for coastal protection. 1st Coastal Policy Document*. The Hague: Ministry of Transport, Public Works and Watermanagement.

Mulder, J.P.M., G. Nederbragt, H.J. Steetzel , M. van Koningsveld and Z.B. Wang (2006). “Different Scenarios for Implementation of the Netherlands Large Scale Coastal Policy.” *Proceedings of the 30th Int. Conf. of Coast. Eng*. San Diego, USA, 2006.

Nonaka, I. and H. Takeuchi (1995). *The knowledge-creating company; how Japanese companies create the dynamics of innovation*. Oxford University Press

Platt, J. (1973). "Social Trap." *American Psychologist* **28** pp.641-651.

Polanyi, M. (1966). *The Tacit Dimension*. Routledge and Kegan Paul.

Rapoport, A. (1988). “Experiments with N-Person Social Traps I. Prisoner's Dilemma, Weak Prisoner's Dilemma, Volunteer's Dilemma, and Largest Number.” *Journal of Conflict Resolution*, **32(3)**, pp. 457-472

TAW (Technische Adviescommissie voor de Waterkeringen) (1984). *Leidraad Duinafslag*. Den Haag: Staatsuitgeverij ’s-Gravenhage. (in Dutch)

TAW (Technische Adviescommissie voor de Waterkeringen) (1995). *Basisrapport Zandige Kust*. Delft: Rijkswaterstaat. (in Dutch)

TRDA (2006). *Product 4: Technisch Rapport Duinafslag. Beoordeling van de veiligheid van duinen als waterkering ten behoeve van Voorschrift Toetsing op Veiligheid 2006*. Published by WL|Delft Hydraulics (H4357) (In Dutch)

TK (Tweede Kamer) (1996). *Wet op de Waterkering*. The Hague: SDU. (in Dutch)

Van Koningsveld, M.; M.J.F. Stive and J.P.M. Mulder (2004). “Balancing research efforts and management needs. A challenge to coastal engineering.” *Proceedings of the 29th Int Conf. of Coast. Eng.* Lisbon, Portugal, 2004. pp. 2985 – 2997.

Wilson, T.D. (2002). “The nonsense of 'knowledge management” *Information Research*, **8**(1), paper no. 144 Available at http://InformationR.net/ir/8-1/paper144.html

ACKNOWLEDGEMENTS

The work reported in this paper was, and further developments will be, carried out in the framework of the innovation program Building with Nature and the EU FP7 research project MICORE.

The Building with Nature program is funded from several sources, including the governmental Subsidieregeling Innovatieketen Water (SIW, Staatscourant nrs 953 and 17009), the Dutch Ministry of Transport, Public Works and Water Management and contributions of the participants to the program. Co-funding is provided by the European Fund for Regional Development EFRO and the Municipality of Dordrecht.

The MICORE project has received funding from the European Community's Seventh Framework Programme under grant agreement n° 202798 (MICORE Project).