



Web-based Knowledge Transfer

FINAL REPORT – SECOND EDITION

Date March 2009 (Second Edition)

Report Number T30-09-01

Revision Number 2_1_P01

Deliverable Number: D30.2

Due date for deliverable: August 2008

Actual submission date: March 2009

Task Leader IHE

FLOODsite is co-funded by the European Community

Sixth Framework Programme for European Research and Technological Development (2002-2006)

FLOODsite is an Integrated Project in the Global Change and Eco-systems Sub-Priority

Start date March 2004, duration 5 Years

Document Dissemination Level

PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Co-ordinator: HR Wallingford, UK
Project Contract No: GOCE-CT-2004-505420
Project website: www.floodsite.net

DOCUMENT INFORMATION

Title	Web-based Knowledge Transfer (Task 30 Final Report)
Lead Author	Yunqing Xuan
Contributors	Dimitri Solomatine, Andreja Jonoski
Distribution	Public
Document Reference	T30-09-01

DOCUMENT HISTORY

Date	Revision	Prepared by	Organisation	Approved by	Notes
11/12/08	1_0_P01	Yunqing Xuan	IHE		Need to conform to the standard of the deliverable of final report.
02/03/08	2_0_P01	Yunqing Xuan	IHE		Completely rewritten
20/05/09	2_1_P01	J Rance	HR Wallingford		Formatting for publication

ACKNOWLEDGEMENT

The work described in this publication was supported by the European Community's Sixth Framework Programme through the grant to the budget of the Integrated Project FLOODsite, Contract GOCE-CT-2004-505420.

DISCLAIMER

This document reflects only the authors' views and not those of the European Community. This work may rely on data from sources external to members of the FLOODsite project Consortium. Members of the Consortium do not accept liability for loss or damage suffered by any third party as a result of errors or inaccuracies in such data. The information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and neither the European Community nor any member of the FLOODsite Consortium is liable for any use that may be made of the information.

© **Members of the FLOODsite Consortium**

SUMMARY

The overall objective of Task 30 was to develop the Web-based interactive platform to support the communication of the findings of the whole FLOODsite project and to promote uptake of the FLOODsite framework and methodologies by the three main target groups: public, professional and educational.

Task 30 developed a Web-based platform initially called E-FLOOD on which the project outcome, team expertise, findings are disseminated through a number of components that are defined into two groups: (1) Knowledge Map which provides a Web GIS interface for user to access descriptors of people, organisations, projects, training courses and documents related to FLOODsite; (2) Modelling Facility supplies web access to the tools and modelling systems with suitable web-enabling interfaces developed in Themes 1 to 3 and demonstrated/tested in Theme 4.

The development of Task 30 undergoes three main stages: input analysis, functional design and implementation. In order to take advantage of the Web platform, technologies are carefully chosen to: (1) keep the development technically up-to-date; (2) use deliberate combination of cutting-edge technique to suit the ad-hoc transformation of traditionally desktop modelling tools to the web platform; (3) ensure the requirements on dynamic contents and interactivity be met.

Apart from the final implementation of the Web-based knowledge transfer platform together with feature-rich components as described in DOW, Task 30 also made substantial progress in developing: (1) the multi-step cascading classification method to unify the very-different-in-nature materials in the same framework; (2) the techniques developed to represent existing traditional models and their applications on the Web platform; and (3) the management practice of the task in terms of effective communication and coordinating to support the smooth development in a multi-disciplinary, highly inter-project dependent context.

The achievements made in Task 30 shed the light for further development in Web-based knowledge transfer and dissemination. It is recommend the following points should be accounted for future development in this area:

1. Well-planned inter-project coordination.
2. Awareness of wider audience when presenting research results.
3. Maintenance issues of the developed products.
4. Utilisation of free or public domain software technologies.

CONTENTS

Document Information	ii
Document History	ii
Acknowledgement	ii
Disclaimer	ii
Summary	iii
Contents	v

1.	Introduction	1
1.1	Objectives	1
1.2	Revision of the objectives.....	1
1.3	Formation of the Research Team.....	2
1.4	Scope of the Document.....	2
2.	Requirement Analysis	3
2.1	General Review	3
2.2	Classification of Task Inputs for Modelling Facility	3
2.3	Input Analysis for Knowledge Map	5
2.4	General Guidance for Communication and Dissemination	5
3.	Design and Implementation	6
3.1	General Procedure	6
3.2	Choice of Platform.....	6
3.3	Selection of Technology for implementation	8
3.4	Functional Design.....	9
3.4.1	Design as per task classification.....	9
3.4.2	Common practices in MF functional design	10
3.4.3	Keep consistency in visual effect and underlying functions	10
3.4.4	Functional Design of Knowledge Map	11
3.5	Implementation.....	12
3.5.1	General procedure	12
3.5.2	Technical Aspects of Implementation.....	12
3.5.3	Selected Examples of Implementation.....	12
4.	Acknowledgements	21
5.	References	22

Tables

Table 1 Classification of Inputs for Task 30	4
--	---

Figures

Figure 1 The two-step classification processes of Task 30 development	6
Figure 2 The topology and network structure of Task 30.	7
Figure 3 A typical use of the common template to maintain the consistency in visual effects and structure	10
Figure 4 The Web UI of the Knowledge Map where information can be queried in various ways. Also shown here is the use of Google Maps service as its underlying Web-GIS layer.	11

Figure 5 The implementation of the Knowledge Map (example shown here is the query of professional organisations)	13
Figure 6 The DiCHiTop model page with interactive selection of model parameters.	14
Figure 7 The result page of the DiCHiTop model where the user can track hydrographs with dynamic information. The model result is show by the LightBox.	14
Figure 8 Cascading of interactive selection in Task 4	15
Figure 9 Breach mode selection and the model result changing with additional parameters.	16
Figure 10 The animation presentation of selected model run.	16
Figure 11 Google Maps API provides extra entry for users to acquire geo-referenced information.	17
Figure 12 Dynamic contents with text and images.	17
Figure 13 DSS simulation with an online model implementation.	18
Figure 14 Pages with dynamic contents and a range of interactivities in Task 25/18 implementation to facilitate exploring of the model result.	18

1. Introduction

1.1 Objectives

Task 30 is one of the key work packages of Theme 5 that aims to support the demonstration and transfer of the developed FLOODsite methodologies in the field of integrated flood risk analysis, modelling and management. The task differs from others in that it makes the use of Internet as the main platform to realise its objectives.

The overall objective of Task 30 is to support the communication of the findings of the whole FLOODsite project and to promote uptake of the FLOODsite framework and methodologies by the three main target groups: public, professional and educational.

The specific objective of Task 30 is to enhance uptake through the adoption of a web-enabled knowledge based, modelling and dissemination platform, which is referred to as E-Flood.

The Knowledge Map will provide access to descriptors of people, organisations, projects, training courses and documents related to FLOODsite;

The Modelling Facility will provide web access to the tools and modelling systems with suitable web-enabling interfaces developed in Themes 1 to 3 and demonstrated/tested in Theme 4;

The Knowledge Transfer and Learning Facility will provide structured web access to the web-based training activities (E-Learning) for the FLOODmasters and Continuing Professional Development (CPD) components of Task 31 (Face-to-face Knowledge Transfer).

1.2 Revision of the objectives

Task 30 is highly dependent upon the collaboration with other partners of the project in order to process, re-organise and disseminate their outcomes and achievements. During the course of the development, the objectives have been proactively adjusted in order to maximize the outcome. The revision has been a direct result of intensive discussion with the Theme 5 management, FLOODsite management team as well as relevant partners. Finally, consensus was reached with revised objectives and associated action plans. The revised objectives, while keep most of original contents, differ from the initially designed in the following aspects:

1. The new objectives stress further integration with the main site. As a result, a standalone, running independently site was dropped. Instead, a component-based approach was taken to provide various modules that can be directly or indirectly linked to the main site. The consistence is assured by the common structure (template) that has been used across the development of each individual task components.
2. The interactivity of contents remains as the central part of the new objectives. Dynamic and interactive contents become criteria for choosing the tasks inputs for further realisation.
3. The eLearning component that was mentioned in the initial plan as a separate implementation was also dropped with the standalone platform. However, useful contents developed by Task 30 were still available to main site can be used for the knowledge transfer purpose.

The revision of the objectives was mainly done in a methodological sense, which did not affect the main objectives in principle.

1.3 *Formation of the Research Team*

Task 30 is led by IHE with a research team undertaking research and development of the platform. The research team consists of researchers studying the effective transformation and dissemination of project outcomes, designing methodologies; and web developers whose main job was to follow the idea and design of the team researchers and development web applications/components accordingly. One of the main characteristics of Task 30 is that the close corporations with partners and users have been maintained all through the development cycle. As such, the team has received inputs, comments and feedback from relevant tasks under various themes of the project. The MT also has been contributing their ideas and opinions to the development that in turn affect and ensure the success of the task.

1.4 *Scope of the Document*

This document serves as Deliverable D32.2 to fulfil the commitment of Task 30. As the final report, the document attempts to demonstrate findings and achievements of Task 30, focusing on the new ideas and technological methods, tools used in the realisation of the objectives set for the task and beyond. The document also reveals the process of problem analysis and solution finding in terms of building such a complex platform on a non-traditional platform. The coverage of this document includes: development of the methodology; Application classification; design and implementation; and finally the conclusion and recommendation for future research.

2. Requirement Analysis

2.1 General Review

The very first step that was taken by Task 30 team was to detail an analysis of user requirements therefore the real development as use it as a blueprint. The nature of the task implicates the realisation of the work page falls into a kind of software development for which the proper requirement sits as a critical part of the entire process. However, there are several important differences between Task 30 and a conventional Web application development, which are:

1. Different audiences. Task 30 needs to produce various web contents for a variety of user groups, which is more complex as compared with a normal web application whose audiences are generally defined with a relatively narrow distribution. Task 30, however, has to cope with the requirements from basically three user groups: professionals, academics and the general public.
2. Different inputs. Task 30's main job is to disseminate outcomes and findings of others that may have already have been produced in a form very much different from Web contents. Moreover, the inputs varies greatly from task to task, for example, the inputs can be an existing model or an application of a number of models with plenty of interesting outputs. Unlike a normal web application that usually has more homogeneous input to be deal with, Task 30 apparently needs to take into account of the very huge difference in inputs as well as the component to be modelled themselves.
3. The ad-hoc structure. Most of classical web applications have a dedicated web environment or in other words, the whole application runs independently. This, however is not the case for Task 30 where the components realised need to be finally link to the main website. And the main website is literally out of the scope of Task 30 therefore no control available either to Task 30.
4. Choice of technology. As Task 30 needs to deal with outcomes for other tasks that are of different nature, it also has to find the optimal techniques according to different components. This in turn, means that on the one hand Task 30 has to utilise many available cutting-edge technology, it also needs otherwise to use a comprised method to implement some task inputs that it does not have any control at all.

In view of these aspects and the objectives of Task 30, it has been decided that the recognition and proper processing of the substantial heterogeneity in both task inputs and individual realisation, is indeed the first priority. In order to meet the demand of the realisation of the objectives with complex inputs and user requirement, a two-step approach was taken to firstly address the distinction of inputs in terms of classification of tasks; and then to reflect user requirement in terms of individual components implementation. The following sub-section is to discuss the process of partitioning contributing tasks according to the nature of their deliverables. The section of design and implementation is introduced afterwards to demonstrate how the user-centred approach is applied in realising each component to reach the goal set in the objectives.

2.2 Classification of Task Inputs for Modelling Facility

As stated in DOW, Task 30 need to get inputs, i.e., the research outcomes, recommendations for good practice from partners within the FLOODsite project, in order to effectively disseminate them through the Web platform. This is especially for the Modelling Facility (MF) of Task 30 and also applies to the Knowledge Map component that may need to have information from across all partners. A detailed review was conducted soon after the task was started. The review, as summarised by

Udale-Clarke (2006), was done on a task-by-task basis. The review proposed a list of tasks supposed to be able to inject inputs to Task 30 for disseminating together with the indications of the nature of deliverable they could provide. Further careful analysis of the list, taking into account the time line of their potential contribution, concluded that those contributing tasks should be divided into four main groups in accordance to the characteristics of their deliverables. The judgement was made mainly by looking into the contents and format of the inputs in connection with the consideration of the transformation into the Web platform. The four categories of tasks according to their inputs are listed below with typical tasks mentioned being examples:

1. Mainly static contents based inputs. A number of researches, mainly methodological studies, may produce outcomes that can be sufficiently supported and described by main static contents, i.e., text with diagrams. In this case, the transformation of such contents to the Web based one, should not be difficult and actually, the main site has already taken the lead to extract and process these static contents and use them appropriately. As such, these task inputs were mostly excluded from Task 30's implementation plan with some exceptions on tasks having limited modelling involvements that may take advantage of the Web platform for publishing. Examples of this group are contributions from Task 17 and 20.
2. Simple Model-centred inputs. There are several cases in the overall researches conducted by FLOODsite, which is about the description of the newly developed model and its application in the project. These materials normally have been well structured with an existing model also available. This is one of the typical cases that Task 30 should devote to implementing the alternative version of the model that can adequately demonstrate the functionality of the model under the Web environment. The main work in this regard falls in the re-designing the interfaces around the model (which is normally developed for desktop environment). The DiChiTop model of Task 1 is one of these kind examples whose implementation is discussed later in this document.
3. Complex model and application. More often is that the inputs are about a complex model and its application in the FLOODsite project. The models used in relevant tasks are not able to be "re-implemented" in the new Web platform due to the complexity of the models and/or the fundamental limitation of Web platform compared with the desktop environment on which they were initially developed. A pre-cooked, scenario-based solution is recommended for the transformation of these inputs onto the web platform, which actually emphasizes the interactivity of web version with some compromise in demonstrating the full functionality of models. Examples of this category are Task 4 and Task 6.
4. Large scale multiple model applications with highest complexity. Multiple models might be employed in some studies targeting at solutions and recommendations for complicated and research scenarios of the project. Apart from model-related material, these tasks may as well come with a great deal contents need to be restructured in an interactively to fully explore the complex outcomes based on the model outputs. Typical examples of this kind are Task 24 and Task 25/18 that are both comprehensive studies with a number of different models, methods.

Table 1 Classification of Inputs for Task 30

Input Classification	Complexity	Effort needed	On-/offline Run	Typical case
Static contents	Low	Less	Online	Many paper based deliverables
Simple model-centred inputs	Medium	Medium	Online possible	Task 1 (DiChiTop model)
Complex model & Application	High	High	Offline with pre-cooked scenarios	Task 6, Task 8
Large scale multiple model applications	Very high	Very high	Online and offline both possible	Task 24, 25/18

The above classification indeed has been examined though the whole life cycle of the project that means that certain inputs might change according to the variation of the inputs against time line. As Task 30 had no control of the development space of those contributors, some task inputs which were initially planned as in group 2, 3 or 4, might turn out to fall into group 1 as such were excluded from the final fully developed implementation that is underlined with adequate interactive and dynamic contents. For example, a number of pilot studies utilised different models and produce some results. However, these inputs to Task 30 were not necessarily to have enough dynamic contents for Task 30.

2.3 Input Analysis for Knowledge Map

The purpose of the Knowledge Map is to allow efficient and effective knowledge networking and knowledge disclosure by making knowledge items (both internal and external) accessible, searchable, and transferable.

The Knowledge Map will provide access to descriptors of people, organisations, projects, training courses and documents related to FLOODsite via a relational data model.

Unlike in MF while needs a generally one-to-one information flow, the Knowledge Map requires information across the entire the research teams across the FLOODsite project. The information provided to Knowledge Map does not necessarily have dynamic contents but it does require the platform to provide maintenance to keep the related information up to date. In response to this demand, the KM component utilise database server to achieve centralised management and maintenance of the knowledge acquired by project partners. The detail of the design of the KM is discussed in Section 3.4.4.

2.4 General Guidance for Communication and Dissemination

The following guidance (Udale-Clarke, 2006) has been followed during the whole process of communication, implementation and dissemination.

1. **Keep it simple** – We will keep our outputs as simple as possible to maximise readability and understanding for the audience.
2. **Audience centred** – We will design our activities and outputs for the intended audience.
3. **Fulfil contract requirements** – We will ensure our activities and outputs will deliver our contractual requirements.
4. **Co-ordinated** – We will coordinate our activities and outputs in a suite of communication and dissemination actions defined by the FLOODsite project.
5. **Timely** – We will communicate our results as early as possible, in accordance with our communication and dissemination plan, but without jeopardising the successful completion of other parts of the project.
6. **Control of quality and content** – We will review all outputs as appropriate for quality and content.
7. **Disclaimer** – All project outputs will include an appropriate disclaimer of liability.
8. **Version control** – All documents will include a unique reference number and version number.

9. **Acknowledgement** – All outputs will include appropriate acknowledgements of the project funder, authors (with name and organisation) and contributors.
10. **No plagiarism** – We will seek permission to use, with acknowledgement, all material originating from others, whether or not they are participating in the project.

3. Design and Implementation

3.1 General Procedure

As previously mentioned, there are substantial differences among the components (MF and KM) as well as each individual task implementation of MF. The main challenge of Task 30, regarding the implementation, is how to find a consistent method that can overcome the huge heterogeneity in task contribution. In order to achieve this, a second classification is adopted to differentiate tasks with their own natures. This shall be done within a predefined, unified framework to ensure the consistency through all implementation. Figure 1 shows the concept of two-step classification with a unified framework for the design and implementation as a whole.

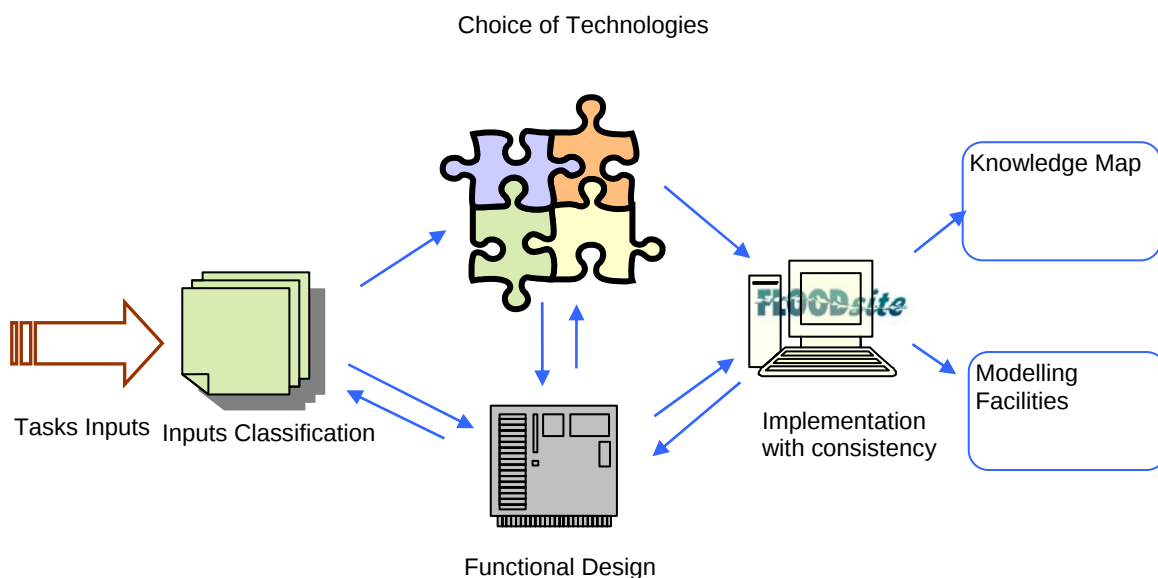


Figure 1 The two-step classification processes of Task 30 development

3.2 Choice of Platform

Task 30 aims to disseminate project findings and results through Web platform. Therefore, the choice of platform actually boils down to the choice of Web platform, in particular, the Web server and associated Operational system and Hardware. Technically, the difference of performance between different web servers becomes negligible when thinking of the web application for hour purpose. In view of the advantage of free software, the research team decided to use Apache as the web server for development and target deployment server. The main reason for this choice is that:

1. Performance and functionality consideration: Apache is a full-fledged, product-level web server with a good reputation of stability, scalability and excellent support for many server-side and client-side technologies.

2. Cost and Maintenance consideration: The server is open-source, free software comes literally free of cost. This is also important for future development and maintenance as it also has free community support from large number of user groups, developers over the Internet.
3. Hardware and OS support consideration: The server can run over a large number of different hardware platform as well as OSes. Although the development for the team is set up on the combination of Windows/PC, the products finally can transfer to various hardware and OSes like Linux and Unix. This gives the maximum flexibility for the integration with the main site afterwards.

The development of Task 30 used a variant of Apache – Xamp and MS-Windows as the main platform. The development environment is securely located behind the firewall of UNESCO-IHE with only Http access being allowed from outside. The topology of the platform and the network is shown in Figure 2.

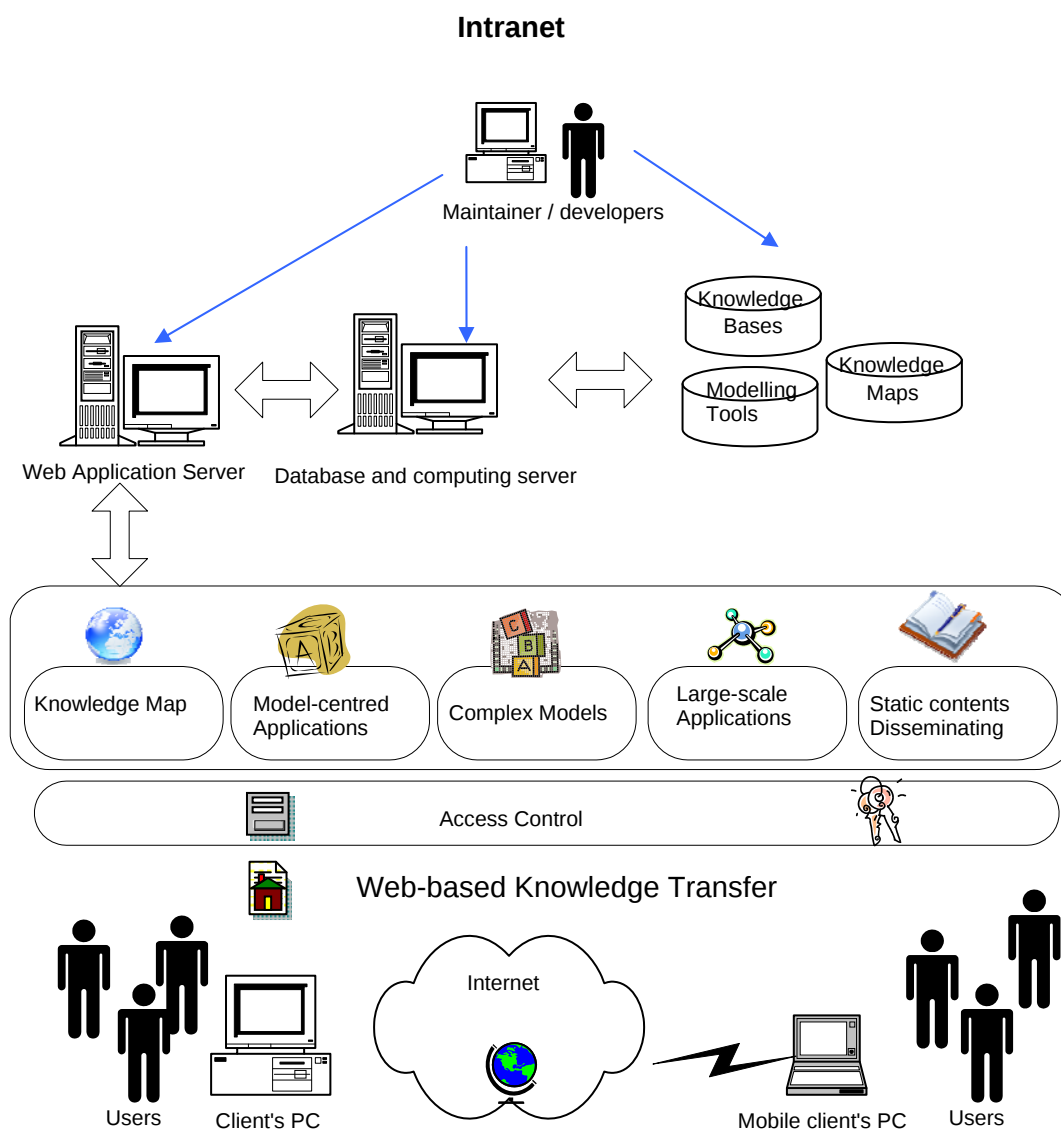


Figure 2 The topology and network structure of Task 30.

3.3 Selection of Technology for implementation

One of the technical goals of Task 30 is demonstrate the project result using dynamic and interactive web contents. To achieve this, there are several cutting-edge web-development technologies needed, mainly falling in these categories:

1. **Server-side scripting:** The server-side scripting is one of the long-history technologies used by web server to fulfil user's request with the contents dynamically generated from data sources, i.e., database server, using scripts run at the server side. The Common Gateway Interface (CGI), ASP and PHP are three important choices. For Apache server, CGI and PHP are natively supported. Task 30 make the use of server-side scripting mainly for two purposes: (1) Providing dynamic contents together with client-side scripting; (2) Dynamically executing the well-wrapped models across the network in response to user's request and pass back the result of the model run as part of the contents.
2. **Client-side scripting:** These are programs run at the client side to respond user input. As to the web platform for Task 30. The client side mainly refers to the web browsers, e.g., IE and Firefox etc. The computer languages available are VBScript (for IE only) and JavaScript (widely supported by all main-stream browsers with subtle variations). Task 30 chose JavaScript as the client-side scripting tool in order to support users access with different browsers. Apart from some routine use like user form validation, JavaScript is also extensively used for nearly all tasks implementation in connection with the Ajax to produce interactive and dynamic contents.
3. **Ajax:** The Asynchronous JavaScript and XML (Ajax) is the core part of all technologies used in Task 30 in terms of implementing interactive web applications. It allows for the partial updating of WebPages asynchronously at background without interrupting the foreground actions of current pages. The data exchange between the client and the server is conducted at the background which enables fast responding to user action and therefore to support the interaction at a higher level (without always reloading the entire page as done in server-side only applications. As such, the Ajax technique has been used for all implementations.
4. **Web-based GIS:** The KM component of Task 30 explicitly requires the Map support for its particular presentation of knowledge distribution and networking on a geo-reference basis. There are many choices for desktop GIS applications but it becomes very limited when moving application on to Web. Google Maps is selected as the GIS service provider upon which the KM is built. The key advantages of using Google Maps are: state-of-the-art technology, scalability, rich set of programming interfaces (Google Maps API) and easy to maintain and free of charge. The Google Maps service is nearly completely based on Ajax (JavaScript and XML) providing a fast responsive UI with enough flexibility of customisation. Apart from KM where the Web-based GIS plays a key role, there are many other occasions in other components where the Google Maps is extensively used (customised) to present geo-referenced information like location of river basins, inundation areas, embankment etc.
5. **Media streaming:** In respect of knowledge transfer, the animations and videos have indispensable position in the whole plan of the implementation. As revealed by the implementation section, Task 30 has organised specific events to record expertise from project partners with video contents. To deal with bottom-neck situation of network speed that may concern the users of Task 30 contents, Adobe® Flash® based media streaming technology is adopted into the development in which all video contents are stored into the sever and are provided through the Video on Demand (VOD) methods.
6. **Database:** Database plays a central role in data management of Task 30 development and implementation. An open-source database server MySQL® is set up to enable the centralised

management and maintenance of the Task 30 web platform. Most of the dynamic, updateable information is stored in the database server and can be provided through the 5 technologies above building the corresponding web contents upon users' request. The MySQL[®] database server also features the cross-platform availability, industrial level performance and wide availability of community support. This also makes the later integration with the main web site a lot easier.

3.4 Functional Design

3.4.1 Design as per task classification

Although Task 30 is supposed to develop web applications/components based upon other inputs, the development in most cases is indeed a complete re-creation even with well-prepared contributions from project partners. This is because the underlying difference of the Web platform compared with traditional desktop based environment.

Apart from the fundamental platform difference, the functional design shall take into account the very different nature of each task as described in user requirement analysis. Obviously, this results in a second classification based on the first one, especially for each MF tasks.

1. Functional design of task with mostly static contents. As the inputs lack of dynamic contents, the functional design is relatively straightforward – reorganise the material to suit for the web platform with extracting of structure, changing the format of pictures etc.
2. Representing the (online) model-centred task. The purpose of this kind of task input is to demonstrate the utility of the model itself. The design was made around the implementation of an alternative Web version of the model and its functionality. The design includes following aspects:
 - Design of Web UI (user interface) of the model that will be used as a bridge to receive the user inputs and display the result of the model run with the data set by the user.
 - Design of an extra wrapper layer to package the model (executables) so that model can be executed automatically on the server with the inputs (files) passed from WebUI. The result of the model also needs to be sent back to the Web UI.
 - Restructuring the static contents to fit the design of the model.
 - Treatment of multi-user concurrent environment with specially designed structures. The models to be presented normally come from a desktop environment having no capability to deal with the typical multi-user concurrent request on a Web platform. This is a critical step to ensure the model can be executed online successfully without interfering with individual instances for concurrent requests.
3. Design for complex (offline) models and applications. Models used in FLOODsite project are not necessarily suitable to be moved to Web platform with the concept of wrapper introduced in class 2 above. For those tasks and their results, different designs are needed to facilitate the transforming of the materials, which involves:
 - Making proper plan of running the complex model offline, aiming to obtain the rich results covering the important aspects/feature of the model that are supposed to be useful for users.
 - Design the Web UI for the results of the pre-cooked scenarios, taking into account as well other materials as a whole.
 - Restructuring the static contents together with the feature-rich materials from the offline run.

4. Large scale applications and pilot studies with multiple models. The inputs from those tasks normally come with a well-defined structure already. The main steps of involved in the functional design is to craft and restructure the material to

- fit the consistent Web UI as for group 1, 2 and 3.
- change the way from the paper media targeted presentation to the interactive one used by web contents.
- use techniques employed for groups 1, 2, 3 as it needs considerations for many detailed structures.

3.4.2 Common practices in MF functional design

There are common procedures applied for each individual task design. The following two are main steps involved in every design:

1. User-requirement analysis, which is not the one conducted for the materials from contributors. The analysis here is to ensure the design fit users from different groups: professionals, academics and general public. It is not always possible to have three different designs for those groups since the materials is limited, but as a rule of thumb, the design should always reflect the difference among the group requirements and therefore trying to organise the contents wisely.
2. Restructuring the existing materials. This procedure needs to be applied to all task materials in which the interactivity has been buried in their original format. The re-construction of the materials can look into and explore the interactivity inherited in. Also there are many other contents need to be stripped away in order to fit in the UI and functional design.

3.4.3 Keep consistency in visual effect and underlying functions

The heterogeneity inherited in MF components has to be addressed properly to maximize the outcome from individual contribution. The consistency in visual effects and function implementation is also a key factor for successful. An extra measure was taken to ensure such a consistency across all tasks. This measure is based upon a common template that was introduced to ensure the consistent implementation and smooth link to the main site.



Figure 3 A typical use of the common template to maintain the consistency in visual effects and structure

The template was set to target a component-based implementation in which every task inputs will be wrapped into a popup window. The component can be linked as a whole (the window) or though

particular functions as indicated in the left hand side of the window. The look-and-feel, is maintained by the same CSS (Cascading Style Sheet) adapted from the main website of FLOODsite. A typical application of the template is shown in Figure 3.

3.4.4 Functional Design of Knowledge Map

The function of KM is heavily dependent on the database server and Google Maps API. The functional design undergoes the following steps.

1. Database structure design to store information that will be presented by the KM components. The table will contain the information of knowledge produced by the whole project and beyond. The Google Maps API also needs to access the database to present information on the map.
2. The query interface design to facilitate users to seek and locate desirable information and knowledge through proper combination of query criteria.
3. Web UI design following the common template used in MF but reserving large area for the map which provide GIS functionality through the underlying Google Maps services in connection with the knowledge database. The Web UI of KM is shown in Figure 4 where the text-based result is displayed together with geo-referenced, clickable information shown in the Map on the upper side.

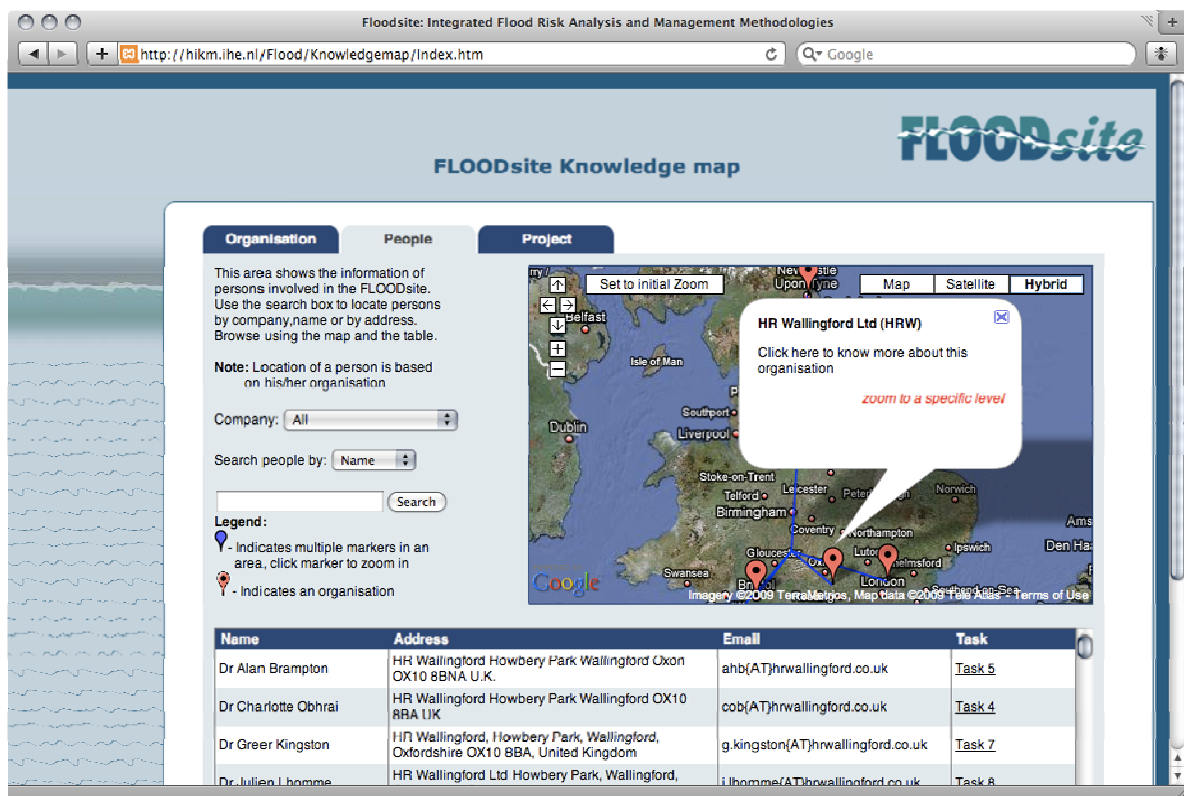


Figure 4 The Web UI of the Knowledge Map where information can be queried in various ways. Also shown here is the use of Google Maps service as its underlying Web-GIS layer.

3.5 Implementation

3.5.1 General procedure

The implementation of components of Task 30 started immediately after the complete of the functional design for individual components based on the relevant inputs from corresponding fellow tasks. The developers, coordinated and guided by the research team then produce the initial implementation with preferred techniques suitable for the component. The implementation undergoes three steps:

1. Initial implementation based on the existing materials supplied and the functional design.
2. Internal evaluation to pinpoint the problem due to insufficient/ineffective material supplied. The request was then send to corresponding partners and/or MT to ensure the acceptance of further (improved) inputs. The initial implementation was then upgraded accordingly. This step could last for a long period depending the progress of the contributors.
3. Finalise the implementation with all available materials. The development now enters into the iteration cycle with improvements and bug corrections being continuously made according to: (1) internal/external test; (2) feedback from the contributing tasks (the knowledge provider) and (3) the feedback from the users (the knowledge consumers). In this regard, this step also serves as a QA (quality assurance) procedure to guarantee the quality of the final products.

3.5.2 Technical Aspects of Implementation

Technically, the process of implementation involves the following measures to carry on the development:

1. Formation of the developing team. Developers with required knowledge and experiences in those technologies listed previously were selected into the team together with researchers.
2. Building the development platform. A dedicated computer server was set up with necessary software installed. Database server was also made ready for the web development. The registration key for using Google Maps service was applied. Decision on the choice of development tools was reached to use Adobe[®] Dreamweaver[®] as the main tool for the site developing. Security measures including network firewall, access authentication was also introduced to protect the development site in general and some sensitive material in particular.
3. Collaboration with the development of the main website. The aforementioned template and CSS from main site were used to maintain the consistency across all components as well as with the main website. Task 30 also worked closely with developers of the main web site to make sure a smooth transition of the development results (implementation) onto the main site.

3.5.3 Selected Examples of Implementation

The implementation of all tasks can be accessed through the main web site (<http://www.floodsite.net>) which has links to the components implemented by Task 30. It is also convenient to use the Task 30's own web site <http://hikm.ihe.nl/floodsite> to access all the components. Several typical components (tasks) are listed below for reference as well.

1. Knowledge Map

This component aims to disseminate the information of expertise, research topics, institutions and researchers across the FLOODsite project and beyond. The database and Google Maps API technology are used in the implementation. Users can access the information through the geo-referenced locations or search from the database directly with the record located on the embedded map. Figure 5 shows the interface of the current implementation of Knowledge Map.

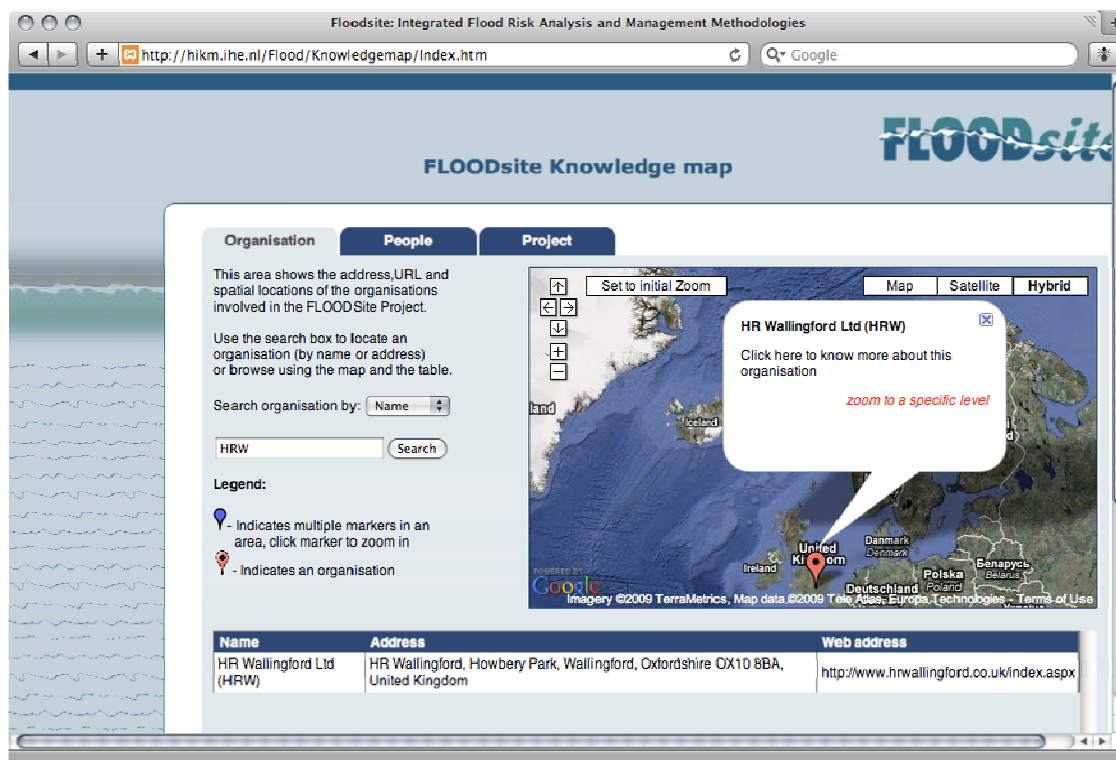


Figure 5 The implementation of the Knowledge Map (example shown here is the query of professional organisations)

2. Task 1 (The DiCHiTop Model)

Task1 involves demonstration of the DiCHiTop model - a tool used in flash flood forecasting and warning system. This tool in fact represents one typical group - the model-centred one as introduced previously. The implementation incorporates general introduction and brief explanation of the principles behind the DiCHiTop model. Then scenarios of model outputs obtained from combination of different time-steps and rain factors (Figure 6) are demonstrated by output graphs that show the measured, simulated and base discharge and rainfall values (Figure 7).

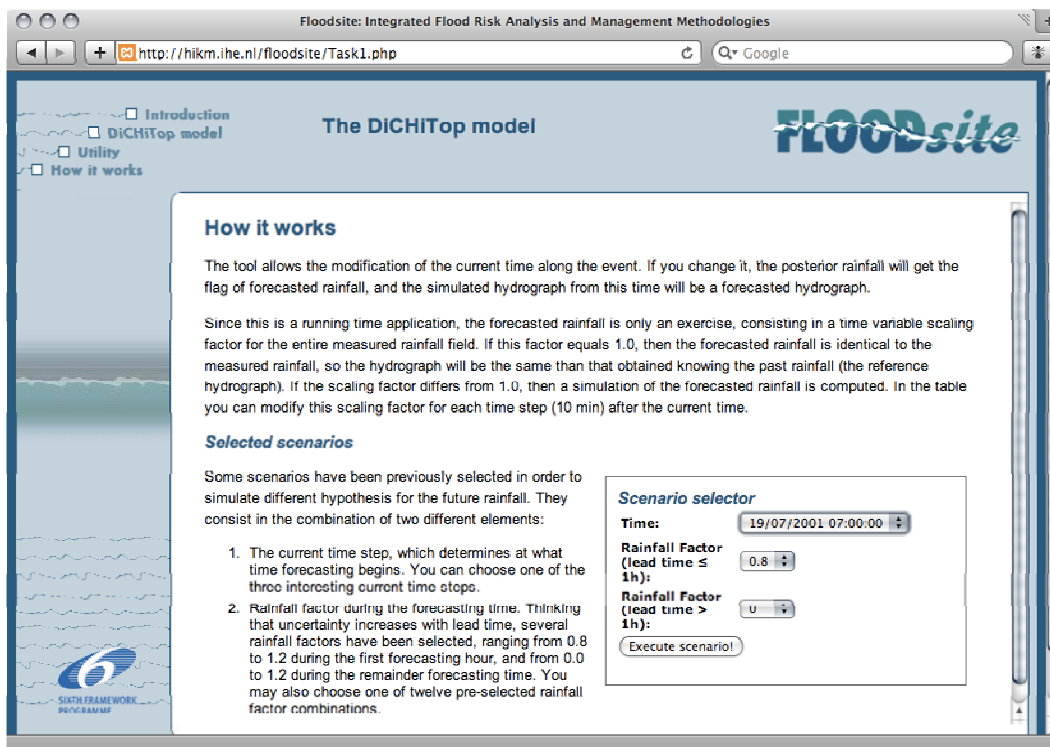


Figure 6 The DiCHiTop model page with interactive selection of model parameters.

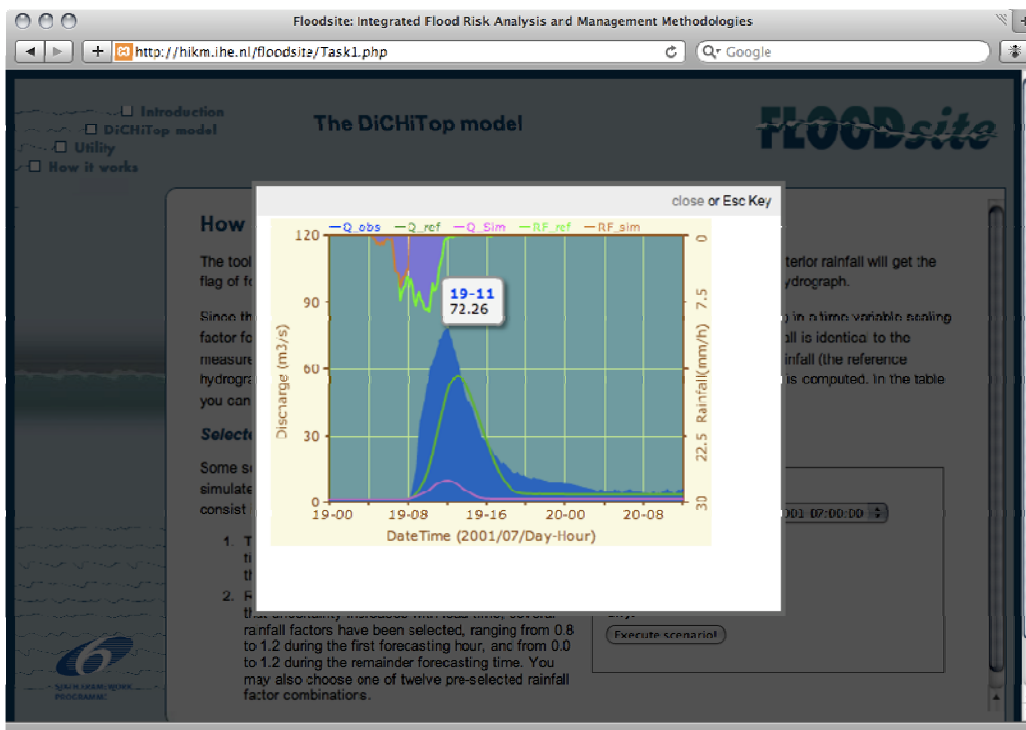


Figure 7 The result page of the DiCHiTop model where the user can track hydrographs with dynamic information. The model result is show by the LightBox.

3 Task 4

This is an online presentation of different modes of failure for flood defence structures. The implementation is to transform paper based publication with the interactive web contents, as it should be. The work is based on the FLOODsite report T04-06-01. The contents of the report has been restructured as follows:

1. Introductory explanation of flood defence structure types, common loads and resulting failure modes
2. A failure mode matrix in which the failure modes are arranged based on the structure type and hydraulic load conditions.
3. Individual failure mode templates accessed based on user's preference
4. Entire PDF file of FLOODsite report T04-06-01

The access to the failure-mode matrix is designed in a cascading way following which the user can track down to the final definition of a particular from the very first level. Figure 8 shows such a process.

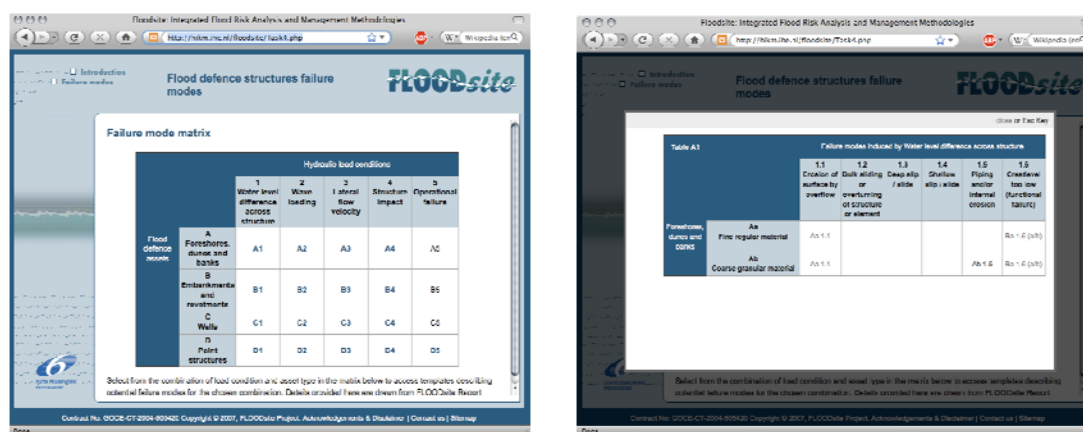


Figure 8 Cascading of interactive selection in Task 4

4 Task 6

Task 6's implementation falls in another category: a complex model needing to be represented with pre-cooked scenarios as discussed before. The implementation is a demonstration of the HR BREACH model that is a numerical model for simulation of breach initiation and growth in embankment dams and flood embankments. The contents has been formulated to include:

- Introductory explanation of the HR BREACH model
- Interactive Scenarios: where example model outputs for different embankment types, failure modes and erosion type are shown. The results include breach outflow (discharge), breach width and reservoir water level with time. In the results page, the user is given freedom to choose different model parameters that affect the model outputs; the graphs dynamically change based on the user's choice. Figure 9 depicts the process of the scenario selection.

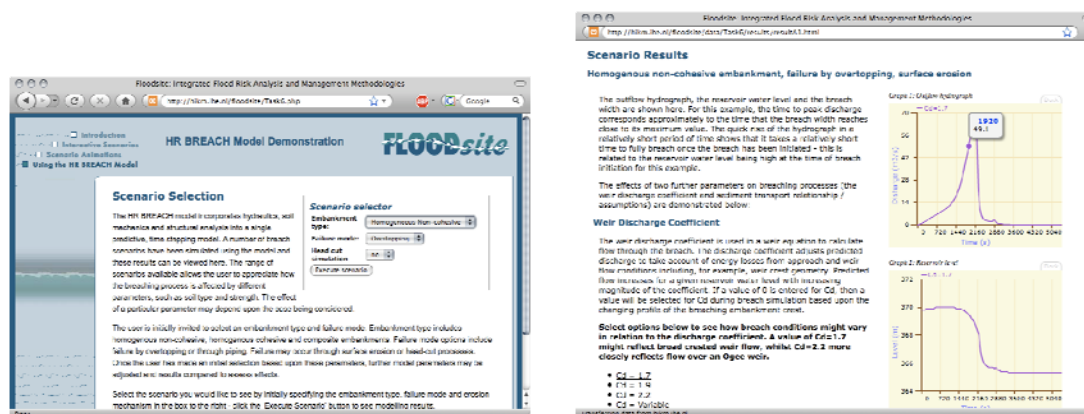


Figure 9 Breach mode selection and the model result changing with additional parameters.

- Scenario Animations: Video demonstration of the model interface while simulation is on progress for different scenarios (Figure 10).
- Demonstration with video and sound track of the various stages of data entry and model run.

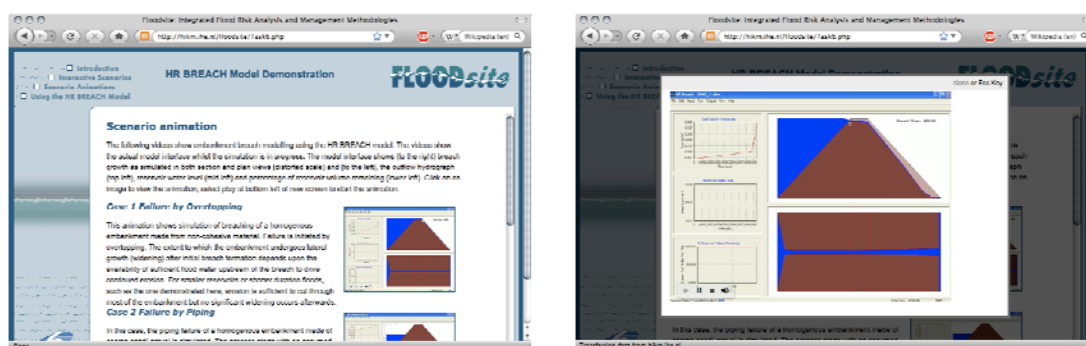


Figure 10 The animation presentation of selected model run.

5 Task 24

The task 24 - Long term flood risk management planning, provides an unique opportunity for Task 30 to exercise all the capacity and technology of the development, as it is one of most comprehensive task implemented with all the methods mentioned previously, i.e., Google Maps API providing Web GIS interface for spatial information query; restructure static pages into interactive one; complex model simulation and comparison; simulations of DSS process, to name a few. Figures 11—13 attempt to demonstrate the rich features of this implementation with limited snapshots.

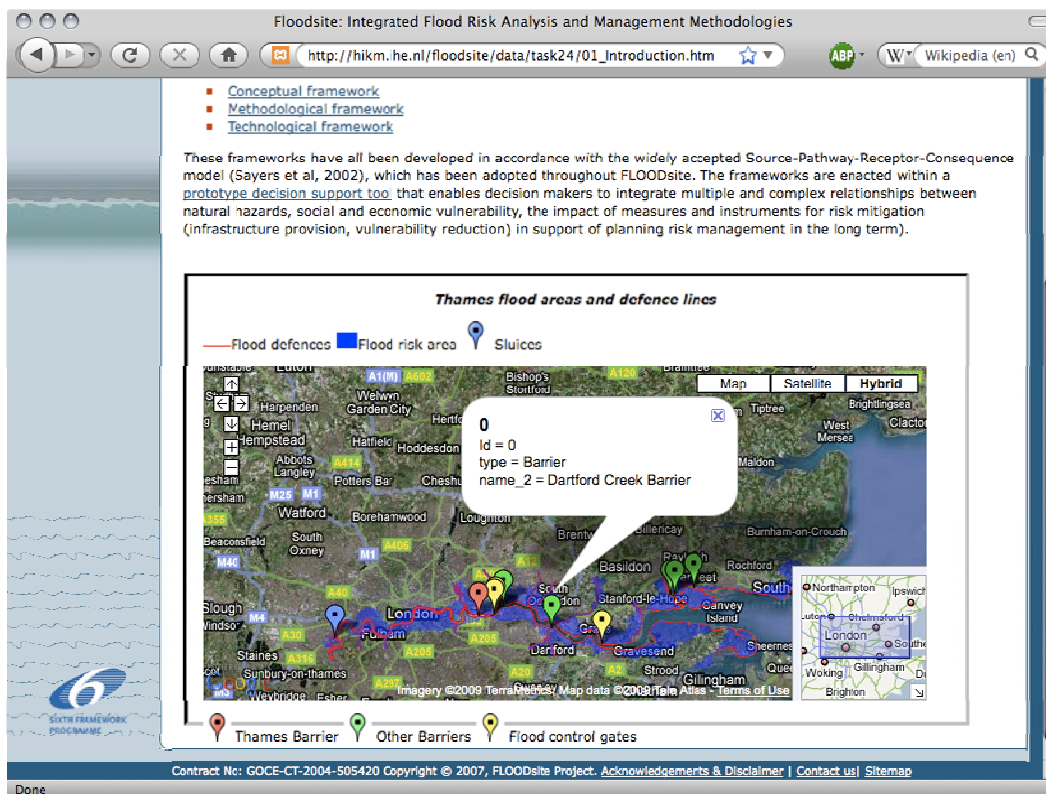


Figure 11 Google Maps API provides extra entry for users to acquire geo-referenced information.

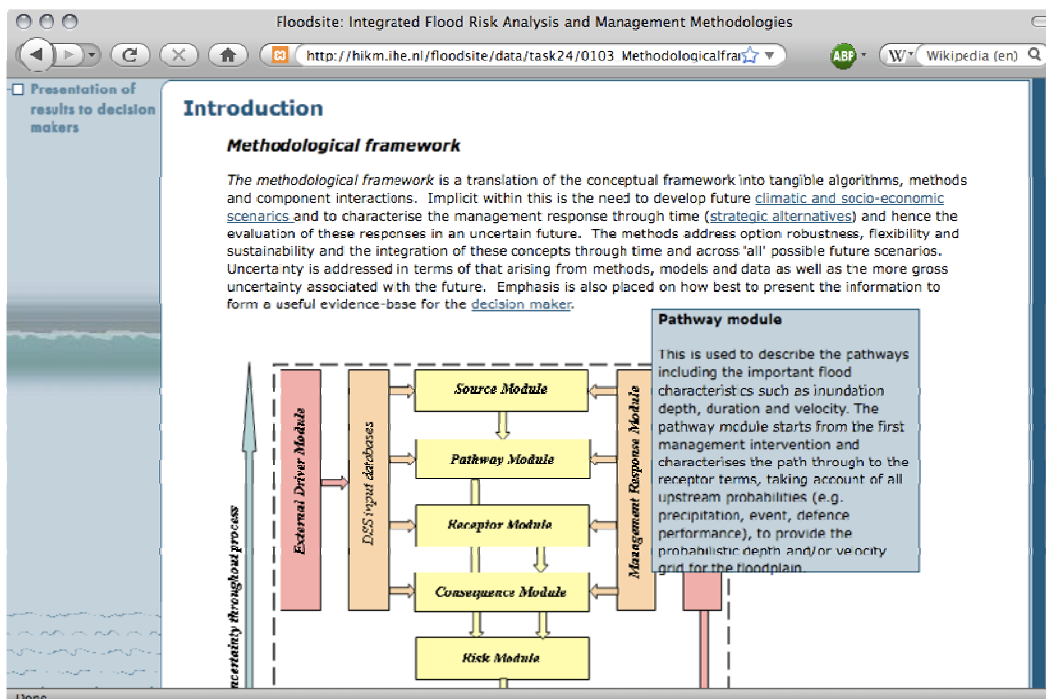


Figure 12 Dynamic contents with text and images.

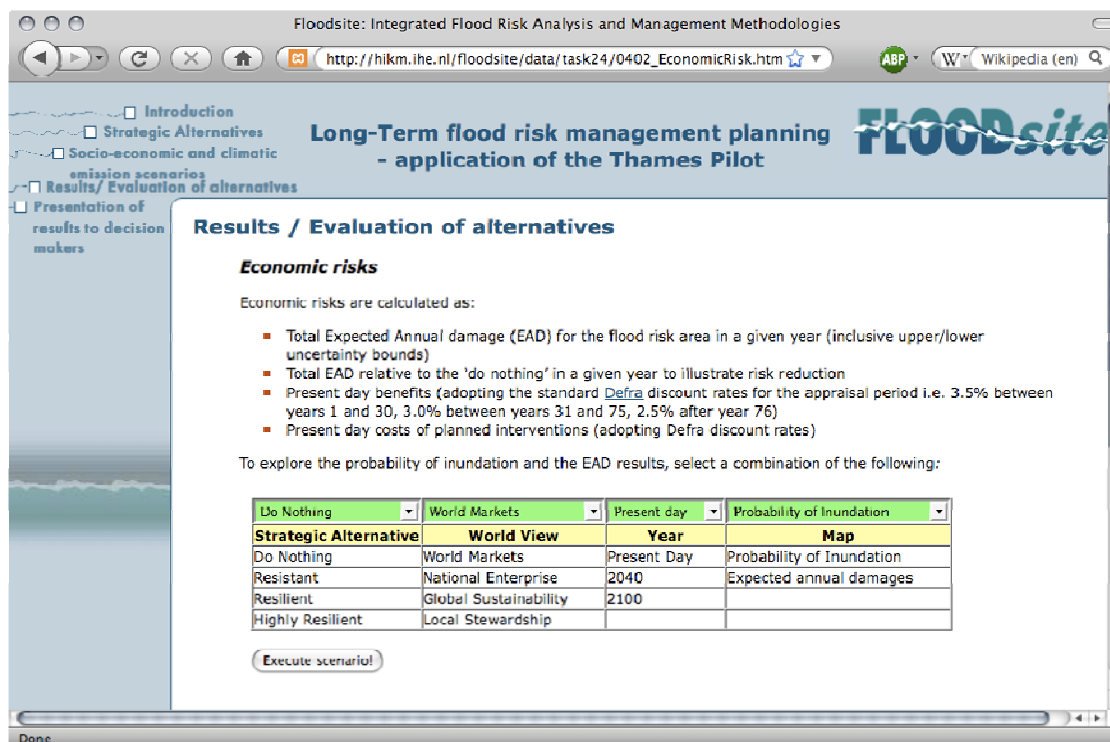


Figure 13 DSS simulation with an online model implementation.

Task 25/18

This is another implantation of large-scale model applications whose implementation utilise all technologies as mentioned. There are two case studies implemented by Task 30: Flood event management in the river Shelde Estuary and long-term flood risk management planning application to the Sheldt Pilot. Figure 14 displays one of many interactive pages developed for the task where user can indeed get result of text, clickable diagram and even spatial information on the map.

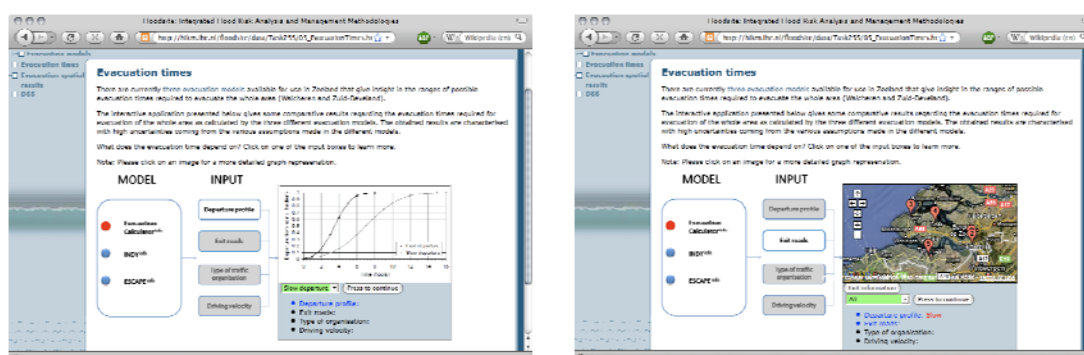


Figure 14 Pages with dynamic contents and a range of interactivities in Task 25/18 implementation to facilitate exploring of the model result.

Discussion and Recommendation

The importance of effectively disseminating research findings and project to the broad range of audiences has been increasingly recognised by more and more large-scale research projects. These projects are normally initialised aiming to solve a series of challenging questions in a much broader context, in contrast to sporadic, isolated projects running at small scale where the research area and audience are both very limited. In addition, many research questions of the large-scale project are more and more posed closer than ever to the general public with topics set to have huge potentials of direct or in-direct influence their lives as the findings and results may affect the public policy making. In this sense, the projects need to be brought into the views of a broader range of audiences, not only the researchers themselves, but also those having direct or indirect links through a variety of ways.

The FLOODsite project, is one of such projects also facing the challenges of disseminating and transmitting the findings, results, messages and recommendations in an effective way, which nowadays preferably is to using Web platform. Task 30 as part of Research Theme 5 of the project, is proposed to meet such specific demand in terms of Web-based Knowledge Transfer.

The overall objective of Task 30 is to support the communication of the findings of the whole FLOODsite project and to promote uptake of the FLOODsite framework and methodologies by the three main target groups: public, professional and educational. During the development the target audiences from these groups were explicitly taken into account.

Task 30 developed a Web-based platform initially called E-FLOOD on which the project outcome, team expertise, findings are disseminated through a number of components that are defined into two groups: (1) Knowledge Map which provides a Web GIS interface for user to access descriptors of people, organisations, projects, training courses and documents related to FLOODsite; (2) Modelling Facility supplies web accesses to the tools and modelling systems with suitable web-enabling interfaces developed in Themes 1 to 3 and demonstrated/tested in Theme 4.

The development of Task 30 underwent three main stages: input analysis, functional design and implementation. The input analysis was conducted to discriminate and re-organise the contributing projects according to the deliverables/materials supplied. The functional design further separate the components into several groups subject to the underlying inputs from other tasks and the technology suitable for the implementation. Task-wide and project-wide consistency in visual effects and functional structure was maintained by using a template providing a unified functional structure and Web user interface. Differentiating target audiences is achieved using this template in connection with the underlying material supplied by fellow tasks. The implementation employed cutting-edge web development technologies to realise the design for each components for which the dynamic contents and interactivity are highlighted as the focuses. A three-step implementation cycle was used to ensure the development proceed in time while keeping the quality up to the expected level. Internal/external tests were the major tool to improve the technical quality of the implementation; effective communications with contributing tasks, MT, user groups were also important to realise the user-centred goal for all implementation.

In order to take advantage of the Web platform, technologies are carefully chosen to: (1) keep the development technically up-to-date; (2) use deliberate combination of cutting-edge technique to suit the ad-hoc transformation of traditional desktop modelling tools to the web platform; (3) ensure the requirements on dynamic contents and interactivity be met. The core WEB 2.0 technology, i.e., Ajax, XML, JavaScript played a key role in the development for the dynamic contents with rich interactivities. Google Maps service was selected as an important technical ground to support Web GIS operations needed by the Knowledge Map and many other components. Server side scripting with PHP, media streaming use Adobe Flash and database server are also selected to support the online model run, multimedia disseminating and centralised data management respectively.

Apart from the final implementation of the Web-based knowledge transfer platform together with feature-rich components that are sufficiently fulfil the commitment on the goal of the task, Task 30 also made substantial achievements in terms of developing web-based knowledge transfer system to facility the disseminating complex, heterogeneous project outcomes to various audiences in different contexts. In this sense, the main achievements of task 30 are further enriched with: (1) the multi-step cascading classification method to unify the very-different-in-nature materials in the same framework; (2) the techniques developed to represent existing traditional models and their applications on the Web platform; and (3) the management practice of the task in terms of effective communication and coordinating to support the smooth development in a multi-disciplinary, highly inter-project dependent context.

The achievements made in Task 30 shed the light for further development in Web-based knowledge transfer and dissemination. In this sense, the following recommendations based on the experience obtained during the Task 30 development may be of more inspiring value to other or future development in this area:

- (1) Well planned intra-project coordination plays a key role and therefore the effectiveness of this kind of practice may dominate the entire progress of development. Difficulties are frequently encountered due to the pre-mature delivery of the material for disseminating. In other words, the findings and knowledge should be presented in a mature way before moving on for web-based dissemination.
- (2) Traditional research results are presented in a way assuming the academics and/or professionals are the only knowledge receptors. A practice standard of inclusion of wider audience in the presentation should be formulated and adopted by each research project/tasks beforehand. This will certainly increase its entire value for a border acceptance.
- (3) The development of such web-based disseminating should be able to sufficiently take into account of maintenance issue and to make sure the product can be moved/linked to the different site without too much extra effort.
- (4) Last but not least, the free and open-source technologies should be the first choice for the development. Active, community-based support and cost-effective implementation make them become more and more the mainstream technologies for web development.

4. Acknowledgements

The authors would like to express their gratitude to Mark Morris (HRW), Erik Mosselman (Deltares) for very fruitful collaboration in this project. Special thanks go to the UNESCO-IHE participants Adrian Almoradie and Tseganeh Gichamo who did so much during the implementation of the platform.

5. References

1. UDALE-CLARKE, (2006), *Web-based knowledge transfer (Specification of the E-Flood Platform)*, FLOODsite Deleverable D301.1.