Risc-Kit Db Tool

Sommario

[Introduction 2](#_Toc392000434)

[Requirement Analisys 2](#_Toc392000435)

[Architecture 2](#_Toc392000436)

[System Design 2](#_Toc392000437)

[Server Side Components 2](#_Toc392000438)

[Client Side Components 2](#_Toc392000439)

[Database 2](#_Toc392000440)

[Project Scheduling 2](#_Toc392000441)

# Introduction

Risc-KIT (Resilience-Increasing Strategies for Coasts – toolKIT) is a FP7 Project that will deliver ready-to-use methods, tools and management approaches to reduce risk and increase resilience to low-frequency, high-impact hydro-meteorological events. The WP1 of the project includes Task 1.3 includes development of a webGIS interface to manage the data that will be made publicly available at the end of the project. The database will be kept live after the project conclusion as part of the OpenEarth system maintained by Deltares. The database will build on the metadata catalogue developed during the MICORE Project and will now be embedded in the OpenEarth database. The database will contain forcing factors (waves, tides, winds, rainfall), observations and hindcast data (from existing ERA-40 and HIPOCAS databases), IPCC climate scenarios, observed impacts and quantitative and qualitative socio-economic, cultural and environmental data. To ensure the long-term continuity of the database, the link with continuing efforts like the ISDR-GAR databases, CRED Database and EEA-Clearing House will be pursued. This document is about the proposed Architecture, Technology and Planning of this task.

# Requirement Analysis

This section is about the Tool use cases analysis.

|  |  |
| --- | --- |
| UC | UC\_RKT\_010 – Creation of an event |
| Actors | Partner User |
| Preconditions | The user makes login |
| Sequence | 1) The user creates a new event;  2) The system creates an Event ID;  3) The user selects the country;  4) The user selects the region (regions to be defined using the EUROSTAT NUTS Definitions);  5) The user inserts the Event Start Data [required]  6) The user can insert the start hour [optional]  7) The user selects the measure unit of the duration [days or hours];  8) The user inserts the value of the duration;  9) The user specifies if the duration is exact or approximated;  10) The user can specify an optional Event Description;  11) The User selects the type of Wave Height to insert choosing between:  a. Mean;  b. Max;  c. Significant;  12) The user inserts the value in Meters  13) The user can upload an xml INSPIRE metadata file and a Time-Series File of wave heights  14) The user selects the type of wave period to insert choosing:  a. Mean;  b. Significant  c. Peak;  15) The user inserts the value in Seconds  16) The user can upload an xml INSPIRE metadata file and a Time-Series File of wave periods  17) The User chooses to insert wave direction in degrees or in a clustered way using sectors (e.g NE, SE, etc);  16) If the user chooses degrees from North:  a. The user insert a number between 0 and 360;  17) Otherwise  a. The user chooses one of: N NE E SE S SW W NW [show arrows]  18) The user can upload an xml INSPIRE metadata file and a Time-Series File of wave directions;  19) The user inserts the max water level in meters;  20) The user can upload an xml INSPIRE metadata file and a Time-Series File of water levels;  21) The user specify the type of wind intensity:  a. Maximum values;  b. time average.;  c. Maximum gust  22) The user inserts the wind intensity value (m/s);  25) The user can upload an xml INSPIRE metadata file and a Time-Series File of wind data;  26) The User chooses to insert wind direction in degrees or in a clustered way;  27) If the user chooses degrees from North:  a. The user insert a number between 0 and 360;  28) Otherwise  a. The user chooses one of: N NE E SE S SW W NW [show arrows]  29) The user can upload an xml INSPIRE metadata file and a Time-Series File of wind direction;  30) The User choose if the event is also a River Flooding was occurring or not;  31) If the event included a River flooding the user inters:  a. Peak Water discharge in m^3/s;  b. The user can upload an xml INSPIRE metadata file and a Time-Series File of discharge;  c. Flood height in m;  d. The user can upload an xml INSPIRE metadata file and a Time-Series File of water level;  32) The user inserts the number of Reported casualties;  33) The user inserts a free text associated to reported casualties;  34) The user can upload an xml INSPIRE metadata file and a shape file or dbf file with the location of reported casualties;  34) The user inserts a text with damage to buildings;  35) The user inserts a Total cost estimation in Euro;  36) The user can explore the cost details:  a. Direct Cost  b. Business Interruption Cost  c. Indirect Cost  d. Intangible Cost  e. Risk mitigation Cost  37) The user can upload an xml INSPIRE metadata file and a shape file or dbf file with the location of reported damages;  38) The user inserts a text field describing the measure taken to respond to the event; |
| Out conditions | The event is added to the database hosted on Open Earth |
| Open Issues | Q: Where can country and regions be fetched?  A: From EuroStat.  Q: which format is required for metadata?  A: Inspire XML format. The system can suggest a link to the inspire editor on line.  Q. which format is required for time-series data?  A: any raw format, keeping with Open Earth protocol. However, viewing tools in the database will only support NetCDF files, hence it is encouraged for users to also provide a script to transfer raw data to NetCDF |

|  |  |
| --- | --- |
| UC | UC\_RKT\_020 – Add media file to the event |
| Actors | Partner User |
| Preconditions | The user is logged and an event exists |
| Sequence | 1) The user click on "Add Media Files" (pictures and documents);  2) The user selects the file on his own pc;  3) The user uploads the selected file;  4) The user localize the uploaded file on the map;  5) The user inserts a description;  6) The user insert the date; |
| Out conditions | The georeferenced media file is added to the event and saved in Open Hearth svn |
| Open Issues |  |

|  |  |
| --- | --- |
| UC | UC\_RKT\_030 – Add a geographic file to the event |
| Actors | Partner User |
| Preconditions | The user is logged and an event exists |
| Sequence | 1) The user clicks on "Add GIS Files";  2) The user selects the file on his own pc;  3) The user uploads the file;  4) The user can upload an xml INSPIRE metadata file; |
| Out conditions | The geographic is added to the event |
| Open Issues |  |

# Architecture

This section is about the proposed system architecture. The tool and the database can be included in the category of environmental past-events data sets, including information about geolocalization, physical characteristics and related impacts. The European Community issued a directive (2007/60/EC of 23/10/2007) on the assessment and management of flood risks; this directive considers the maintenance of a EU Database of flood events, obtained merging single states databases under a unique UE scheme. The European Commission – DG Enviroment published a “Technical Support in relation to the implementation of the floods directive – A User guide to the floods reporting Schemas”, describing a detailed xml scheme with all the information needed to fulfill the directive. This scheme has some information tailored to the flood events but also many impacts-related information that can be considered useful also for other environmental risks. The proposed system architecture takes in account this experience to create a Database that should be adoptable at EU level.

The main goals of the system can be considered:

* Collect physical, geographical and impacts information about events;
* Integrate all the related existing storage;
* Promote use of collected data by researchers community;
* Give users access to collected data through a web user interface;
* Give users access to collected data through standard protocols;
* Promote adoption of a European Standard to collect same information;

The proposed system architecture is represented in the following diagram:



The core of the project is the Central Risc-KIT Db, that is hosted on the Deltares Open Earth Platform. The limit of the RiscKIT Tool system is marked by the blue rectangle, including the web tool and a GeoServer. The limit of Open Earth platform is the green rectangle, hosting the svn server, the Thredds Data Server and a Postgree/PostGIS database server.

Data of the MICORE database is already hosted in the OpenHeart Platform.

Authorized users, till the end of RISC-KIT only the Case Study Site managers, can interact with the database using the web tool application to insert a new event, localize it and specify all the needed parameters.

Georeferenced data can also be published through an Open Source GeoServer using WMS and WFS Standard: this allows external users to ingest collected data in their own GIS applications.

Uploaded raw data will be published, using svn protocol, on the Deltares OpenEarth platform; this will permit the widespread availability of the data in the external research community.

On demand, data could also be exported in a XML file format, derived by the scheme defined by European Commission for the floods directive. Resources about the floods directive can be obtained at this address:

<http://icm.eionet.europa.eu/schemas/dir200760ec/resources>

The Floods Directive Reporting Schema, defined in the “Document No.2: Floods Directive reporting: User Guide to the reporting schema v5.0”, mainly defines a n-n relation between events and locations; other tables are dedicated to the description and characterization of the consequences of the event, including informations about Human health and Social, Enviroment, Cultural Heritage, Economic Activity.

# System Design

Proposed System is composed of this main components:

* A central database;
* One Web Application;
* One GeoServer Installation;

The design of this kind of system nowadays has to consider the so-called 2.0 Web standards. The choice of technologies to use is influenced by the will to choose Open Source solutions. The design is described in the following schema:



The solution is a typical 2.0 architecture.

## Server Side Components

On the server side Apache is used as a Web Server while Tomcat is the application Server.

The DatabaseManagement System is PostgreSQL with PostGIS; PostgreSQL has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL object-relational database. PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium (OGC).

The server side programming language is Java; the Business and Data layer are developed using the OpenSource Hibernate ORM. The role of ORM is to map object-oriented domain to a traditional relational database. Hibernate is one of the well-known and widespread adopted ORM for Java, used by tens of thousands of Java developers.

The system API is based on a RESTFul web services layer, using official (and Open) Java Jersey implementation. The acronyms REST means REpresentational State Transfer. It consists of an software architectural style: it means that it is not properly a concrete system, neither an established standard. REST is a set of guidelines aiming at the realization of a system architecture. It specifies features, such as the uniform interface. If these features are applied to a Web Service, they imply good properties (i.e., scalability) on the Web, improving the performance. Data and functionality can be accessed via the Uniform Resource Identifiers (URIs), which normally consists of links on the Web. The main advantage from adopting the URI scheme is that it is already well-defined and no new implementations are requested.

A concrete implementation of a REST Web service follows four basic design principles:

* Use HTTP methods explicitly.
* Be stateless.
* Expose directory structure-like URIs.
* Transfer XML, JavaScript Object Notation (JSON), or both.

In comparison with SOAP REST is easier to use for the most part and is more flexible. It has the following advantages when compared to SOAP:

* No expensive tools require to interact with the Web service
* Smaller learning curve
* Efficient (SOAP uses XML for all messages, REST can use smaller message formats)
* Fast (no extensive processing required)
* Closer to other Web technologies in design philosophy

## Client Side Components

On the client side a standard HTML5 + css solution is preferred. Nowadays the web evolution requires the focus not only on the use cases but also on the UI usability; the ability to write complex user interface code can be a racing condition using native web language like Javascript, not tailored for big and reach client side applications. Looking at trends in the developers network AngularJS framework is reaching an exponential interest.

AngularJS is an open-source JavaScript framework, maintained by Google, that assists with running single-page applications. Its goal is to augment web-based applications with model–view–controller (MVC) capability, in an effort to make both development and testing easier.

The library reads in HTML that contains additional custom tag attributes; it then obeys the directives in those custom attributes, and binds input or output parts of the page to a model represented by standard JavaScript variables. The values of those JavaScript variables can be manually set, or retrieved from static or dynamic JSON resources.

Angular follows the MVC pattern of software engineering and encourages loose coupling between presentation, data, and logic components. Using dependency injection, Angular brings traditional server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the backend is reduced, leading to much lighter web applications.

The exchange of JSON data supported by Jersey server platform, assists a View-Driven design of the Server side RESFFul web API that helps building rich and smart web client applications.

# Database

The RiscKIT Database has to store different data types:

* Descriptive information about the Event, including localization, date etc.;
* Physical information about the described event;
* Information about the consequences;

For all datasets metadata information will be present according to the INSPIRE directive requirements.

The conceptual structure of the Database can be represented in the following scheme:



This is a conceptual scheme that will be updated and fixed during the detailed system design and implementation.

For the ingestion of raw data the following formats will be supported:

* Csv file for time series like wave height or wind intensity;
* Other text file formats;
* Multimedia images formats;
* Documents (doc, docs, pdf…);
* Shape or dbf files for event localizations;

As stated before, the Database content will be exportable in XML format.

# Project Scheduling

The project scheduling is represented in the following scheme:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Date** | *01-mag* | *15-mag* | *01-giu* | *15-giu* | *01-lug* | *15-lug* | *01-ago* | *15-ago* | *01-set* | *15-set* | *01-ott* | *15-ott* | *01-nov* |
|
| System Architecture | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| System Design | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Client Input Interface development | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Server Input Interface development | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Client navigate Interface development | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Server navifate interface development | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| System tuning and Fixing | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Project Event** | |  |  |  | Deltares Visit | | Intake UI Beta | |  |  | Full Beta Version | |  | Release |

The main Events are:

* 15 May: final architecture of the database
* 15 June: visit of Deltares staff and WP Leader to premises in Genova for interface Open Earth
* 15 July: data intake interface ready and sent to partners for start-up of data intake
* 15 September: Beta version of Database and webGIS available to partners for comments and debugging
* 30 October: Official Release V1 made publicly available

During the project’s life the database will continue to be populated by partners but results will be made available publicly.