

UrbanFlood



Internet Services for EWS

Work Package 6 – D6.6

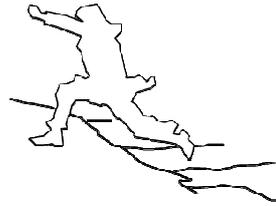
version 1.1 30.11.2012

November 2012



URBAN FLOOD

A project funded under the EU
Seventh Framework Programme
Theme ICT-2009.6.4a
ICT for Environmental Services and
Climate Change Adaption



Grant agreement no. 248767
Project start: December 1, 2009
Project finish: November 30, 2012

Coordinator

Urban Flood Project Office at TNO-ICT
Prof dr Robert J. Meijer

Eemsgolaan 3
PO Box 1416
9701 BK Groningen
The Netherlands

E : robert.meijer@tno.nl
T: +31 50-5857759
W: www.urbanflood.eu

DOCUMENT INFORMATION

Title	Internet services for EWS
Lead Author	Artem Ozhigin
Contributors	
Distribution	Public
Document Reference	UFD6.6v1.1.SIE

DOCUMENT HISTORY

Date	Revision	Prepared by	Organisation	Approved by	Notes
26.10.2012	1.0	Ozhigin	Siemens		First revision
14.12.2012	1.1	Ozhigin	Siemens	Rob Meijer	Review Marian Bubac: Updates upon review findings

ACKNOWLEDGEMENT

The work described in this publication was supported by the European Community's Seventh Framework Programme through the grant to the budget of the Project **UrbanFlood**, Grant Agreement no. 248767.

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 the UrbanFlood 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 UrbanFlood Consortium is liable for any use that may be made of the information.

© **URBANFLOOD CONSORTIUM**

TABLE OF CONTENT

SUMMARY	ii
ACRONYMS AND ABBREVIATIONS	iii
1 INTRODUCTION	4
1.1 UrbanFlood	4
1.2 Structure of this Document	4
2 URBANFLOOD EWS	5
2.1 Domain Model	5
2.2 Utilization of Internet Services	6
Basic Remote Dike Monitor	7
Operational DSS Viewer	7
Hosting Platform.....	8
Plug-in Technology	10
2.3 Case study: reference architecture of AI component	11
3 FUTURE OUTLOOK	14
4 REFERENCES AND BIBLIOGRAPHY	15

Acronyms and Abbreviations

C&D	Communication and Dissemination
CYF, CY	Cyfronet AGH, Cracow, Poland – UrbanFlood partner
CIS	Common Information Space
Defra	Department for Environment, Food and Rural Affairs, UK
DoW	Description of Work, annex to the UrbanFlood Grant Agreement
DSS	Decision Support System
EWS	Early Warning System
FLOODsite	EU funded project: Integrated Flood Risk Analysis and Management Methodologies. 6 th Framework Programme
FRM	Flood Risk Management
HRW	HR Wallingford
ICT	Information and Communication Technology
IJkdijk	IJkdijk Foundation, dike testing site in Groningen, the Netherlands
IWS	International Online Early Warning Workshop
LiveDijk	Test location for sensor technologies at Eemshaven, Groningen, the Netherlands
M&E	Monitoring and Evaluation
NEMO	Science museum in Amsterdam, the Netherlands
PU	Public (report)
RE	Restricted (report)
RTD	Research (in DoW)
SIE	Siemens, Germany. OOO Siemens in Russia is an UrbanFlood project partner
SME	Small and Medium sized Enterprise
STOWA, STO	Dutch acronym for the Foundation for Applied Water Research, Utrecht, The Netherlands – UrbanFlood Partner
TNO	TNO Dutch organisation for Applied Research, The Netherlands – UrbanFlood lead partner
UK	United Kingdom
US, USA	United States of America
UvA	University of Amsterdam
WP	Work Package (there are 7 work packages in UrbanFlood)

1 Introduction

1.1 UrbanFlood

UrbanFlood is a project investigating the use of sensors within flood embankments to support an online early warning system, real time emergency management and routine asset management. It is a project under the EU 7th framework Programme which started in December 2009 and runs for 3 years. Partners of UrbanFlood include TNO Information and Communication Technology, the University of Amsterdam and STOWA (Dutch acronym for the Foundation for Applied Water Research) from the Netherlands; HR Wallingford in the UK, ACC Cyfronet AGH in Poland and OOO Siemens in Russia

1.2 Structure of this Document

- Chapter 1** Introduces the project and its context as a domain model. Describes main relations within the model.
- Chapter 2** Considers general matters of Internet technologies utilization for building up EWS. Describes the system components developed within WP6 especially focusing on the practical application of Internet technologies.
- Chapter 3** Gives a future outlook to further evolution of Internet-based EWS technologies.

2 UrbanFlood EWS

2.1 Domain Model

Figure 1 shows very basic UrbanFlood EWS domain model which does not go into details of EWS internal structure (as it may differ) and all aspects of UrbanFlood hosting platform and Common Information Space (CIS). So it mainly shows the external outline of the EWS in the way general public may perceive it via UrbanFlood project demonstrations.

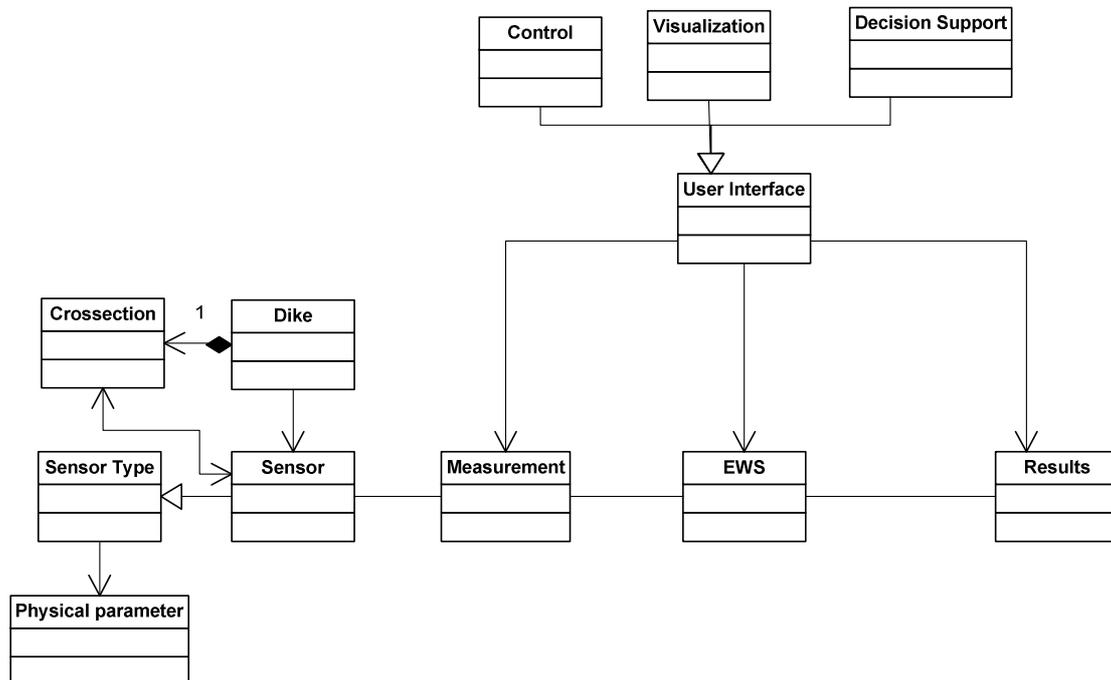


Figure 1 - High-level UrbanFlood EWS domain model

Typically, the basement of any EWS built within UrbanFlood framework is the data acquisition system consisting of the sensors installed in a monitored object. Since the project deals with flood monitoring the primary object is any flood defence referenced to as dike in the figure. Sensor installation in a dike may be organized as several groups referenced to a particular crossection. UrbanFlood have validated the application of different kinds of sensors for flood monitoring measuring temperatures, pressures, inclinations, displacements etc. Sensor measurements are passed to the inputs of one or more EWSs which calculate and emit results of different kinds (e.g. confidence values, stability factors, safety factors).

User interfaces allow for visualisation of different kind of data in a specific manner and to a different audiences, control over the operation of EWS and support data interpretation and decision making by domain experts.

Main goals of the UrbanFlood projects include:

- capability of monitor objects (dikes) located anywhere in the world
- creation of diverse structures of EWS for different objects using software components developed within the project as well as third-party software
- remote access to all data, including measurements, EWS operational results, EWS configurations and structures
- support of remote collaboration of experts.

Internet technologies became a key to achievement of all those goals.

2.2 Utilization of Internet Services

Internet services are widely used in the project as demonstrated in previous project deliverables. From the installation site usually data is sent to a data server via 3G connection. And here is the place where internet infrastructure comes into play for the first time. Although sensor data are sent over public 3G and internet they come to data server directly without actually getting public.

Then sensor data in the form of measurement reach the AnySense Database where they are stored and made available for external access in two ways – by means of HTTP REST interface allowing for fetching batches of measurements for any specified time period and by means of JMS queue intended for immediate measurements delivery to all data consumers in a publisher-subscriber manner.

From the topmost point of view, EWS consumes measurements using services of AnySense, performs prescribed processing workflow over them and emits results. All EWS parts, virtual appliances and services within UrbanFlood platform are managed by CIS [2]. CIS uses service-oriented paradigm to achieve interoperability of EWS parts and appliances in a loosely coupled way. Some of data produced by EWS can be stored back to AnySense as virtual measurements, others are made available over other services of UrbanFlood CIS.

All users interested in EWS operation are able to see their objects of interest which differ significantly via different user interfaces. User interfaces are based on internet technologies as well since EWS services are exposed to Internet and the whole infrastructure is distributed. They might be implemented as applications running on local devices although using internet services intensively for data access, visualization and manipulation or as a web-based application allowing to access data via web-browser.

Majority of CIS user interfaces are implemented as web-interfaces of different kinds. They are aimed at managing of different aspects of EWS and infrastructure operation. Operational internet DSS viewer [5] is implemented as a web-application using Microsoft Silverlight technology.

Multi-touch visualization application (the main visual showcase of UrbanFlood) runs as a desktop application under Windows 7, but all the data for visualisation are retrieved via

Internet from the services exposed by the EWS, CIS and the hosting platform. On the basis of the first multi-touch visualisation the scenario-based application for the Crystal exhibition centre (London) was developed.

So, in UrbanFlood Internet is used not only as a means of communication, but also as a way to provide human-machine interface to the stakeholders. Those stakeholders can use all benefits from modern Internet technologies utilized in the project allowing them to interact with each other across the globe.

More details on the practical experience of the Internet technologies application in different parts of EWS service platform developed in WP6 are given in further subsections.

Basic Remote Dike Monitor

The Basic Remote Dike Monitor (BRDM) [5] was the first step towards the remote monitoring of sensors installed in dikes. At the time of BRDM development there were no functional EWS available as it was only the very beginning of the project. This WP6 deliverable allowed for inspection of sensor data availability and values. The monitor was implemented in the form of web-based frontend towards TNO AnySense data acquisition and storage software which further served as the primary sensor data source for EWS.

Main capabilities of the basic remote dike monitor include numerical and graphical visualization of sensor data values, management of charts and diagrams, keeping notes in the form of web log and download of data files in different formats.

Underlying Internet technologies include REST services for data retrieval from AnySense, AJAX for dynamic web-interface.

Operational DSS Viewer

As the main UrbanFlood showcase of the project the interactive multi-touch table application (described in [9]) was developed. While providing intuitive user interface with wide scope of capabilities and innovative multi-touch user interaction it implied certain limitations on the devices it can run on— it required a multi-touch based device, preferably of large size and appropriate software support (Microsoft Surface Toolkit under Windows 7). Software installation and configuring was also not an easy task.

As an alternative to the multi-touch table interface the Operational Internet DSS Viewer was developed [6]. It is implemented as a web-application based on Microsoft Silverlight technology (which is also limiting the range of devices it can be run on, but this limitation is much weaker than that for multi-touch technologies).

The viewer runs in a web-browser as well as the basic monitor although its features are much more extensive and approaching those of multi-touch application. Namely it offers the geographical overview of monitored locations, visualization of sensor placement within the

objects, real-time and historical overview of both sensor data and processed values. A screenshot of the viewer is shown in Figure 2.

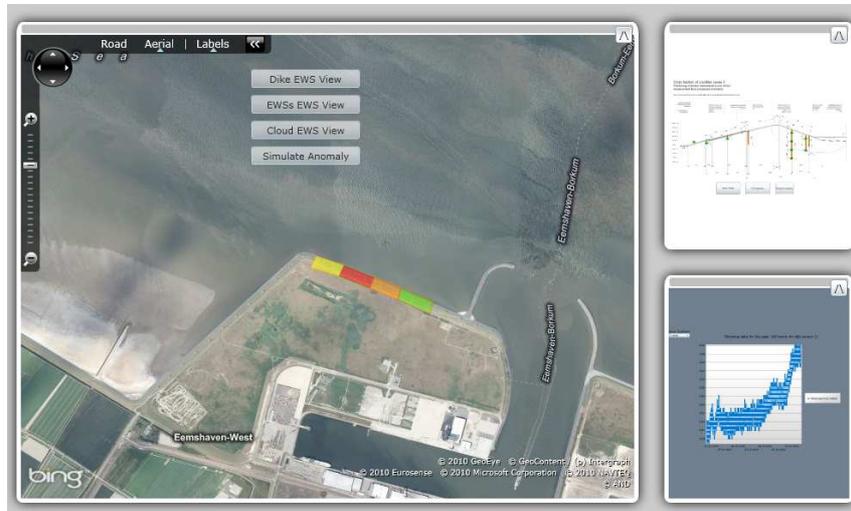


Figure 2 – Operational Internet DSS Viewer

Both multi-touch visualization software and operational Internet DSS viewer are already tightly integrated not only with data acquisition system but also with the whole infrastructure comprised of the CIS and hosting platform. Besides the raw and processed sensor data those viewers are able to retrieve information about the hosting platform performance metrics gathered from all virtualization servers as well as the structure and metrics of currently running EWS.

Possibilities to port the visualization software to other platforms were also surveyed and estimated in [8].

Hosting Platform

The central part of WP6 was the development of EWS hosting platform. The first overview of the platform is given in [7]. Main goal for the platform development was to create a distributed structure offering computational resources to the plug-ins acting as a low-level building blocks for the EWS under governance of CIS. As the basis technology of resource management within the platform the virtualization was chosen. Virtualization abstracts all of the underlying hardware details from running applications, provides comfortable environment and isolation for every application, prevents unintended disclosure of intellectual property and allows for easy management of computational resources as within the platform as well as on the plug-in level. Among many different variants of virtualization software the free edition of Citrix XenServer was chosen.

From the very beginning the platform was based on a distributed set of virtualization servers hosted by the project parties, so the only way to combine those distributed resources was using Internet technologies. Remote servers and server pools were integrated by means of a specifically developed software components shown in Figure 3. Communications between

the platform components is carried out by means of JMS (Java Messaging Service) technology. JMS provider is deployed within the CIS.

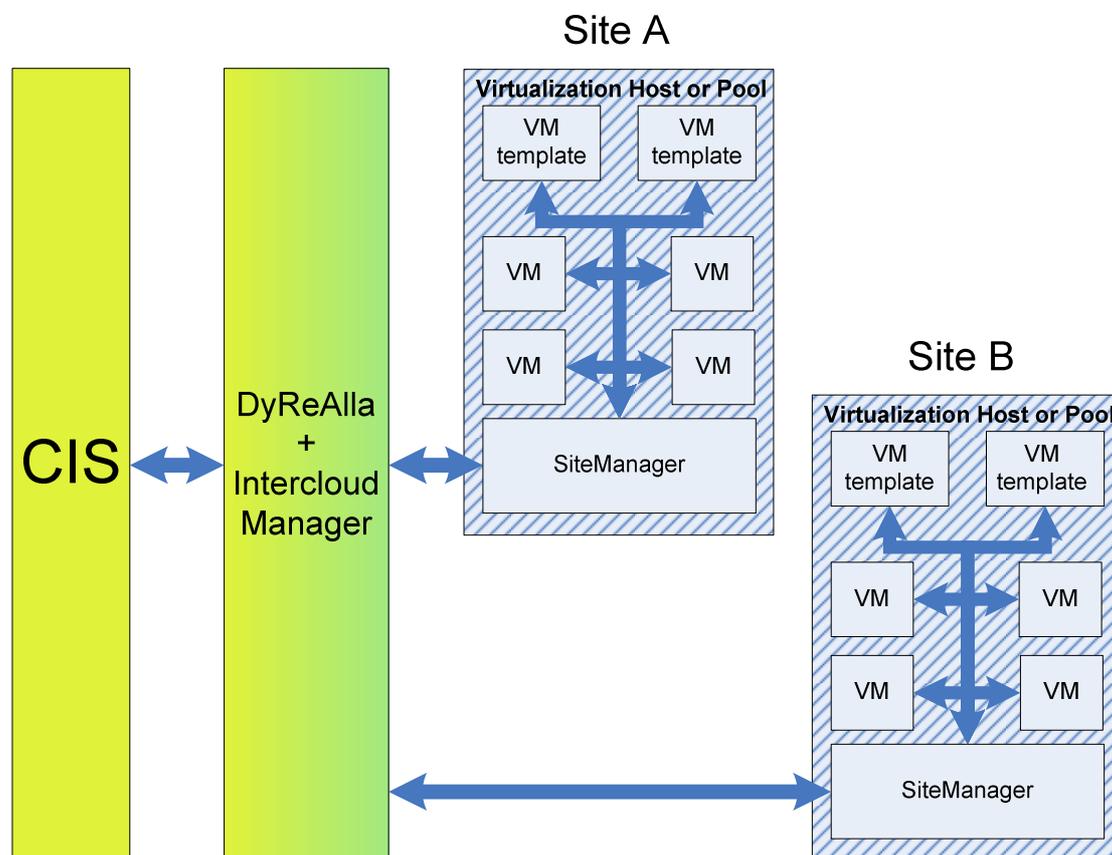


Figure 3 - Hosting platform structure

Those components are SiteManager, which is responsible for all operations within the site and InterCloudManager integrating all SiteManagers and providing interface to the DyReAlla component of CIS. DyReAlla performs all tasks concerned to the dynamic computational resource management in an intelligent way.

CIS itself uses a wide range of internet technologies as Camel, HTTP, JMS, BPEL, OSGi. Its components are running on the same virtualization infrastructure as EWS. CIS provides the following services:

- creation of EWS blueprints
- EWS execution and lifecycle management
- dynamic resource management
- component monitoring
- provenance.

In [8] main functions of the platform components were validated, it was demonstrated that the platform does not cause significant performance overhead to the running applications. Generally, the suitability of distributed cloud-like platform for running multiple EWS was proven.

Plug-in Technology

Plug-in technology defines how the application shall be wrapped into a virtual machine in order to be suitable for integration into UrbanFlood EWS. In details it is described in [7]. It does not imply any limitations on internal structure of application, operating system it shall run on and programming languages. The technology is specifying how a particular application shall interface the UrbanFlood platform and CIS instead.

Plug-ins in the form of virtual machine templates are stored on virtualization servers and transferred into virtual machines at the stage of EWS deployment. The structure of EWS is defined in a EWS blueprint which is created, stored and managed by CIS. CIS controls creation and launch of virtual machines sending corresponding hosting platform commands. Those commands are relayed to SiteManager modules which uses dedicated XenServer API to implement them.

The specification outlines the set of interfaces which are used by a plug-in to communicate. Vast majority of these interfaces are based on different Internet technologies, including JMS, Webservice, FTP, NFS, HTTP, REST.

CIS treats all plug-in interfaces as services [2], it is capable of hierarchically assemble those services and plug-ins into EWS parts providing larger and higher level pieces of functionality. Extensibility of the CIS allows for adoption of other interface protocols and technologies in addition to those mentioned in initial version of plug-in specification. CIS also collects and stores metadata and configuration data for whole EWSs, EWS parts and plug-ins. Detailed sequence diagrams illustrating the interactions of hosting platform, plug-ins and CIS can be found in [7].

Figure 4 shows main interactions of a plug-in within a EWS. CIS manages instantiation and execution of plug-ins based on the blueprint of EWS and component / EWS part metadata. DyReAlla is responsible for decisions on which sites particular plug-ins shall be deployed. It interacts with platform infrastructure (InterCloudManager) to get performance metrics and to start virtual appliances. SiteManager creates virtual appliances from the templates and provides initialization via xenstore. Upon startup plug-in retrieves configuration information from UFoReg. After the configuration the plug-in starts to perform operations and data exchange. All activities within EWS are monitored by a self-monitoring system represented with ErlMon.

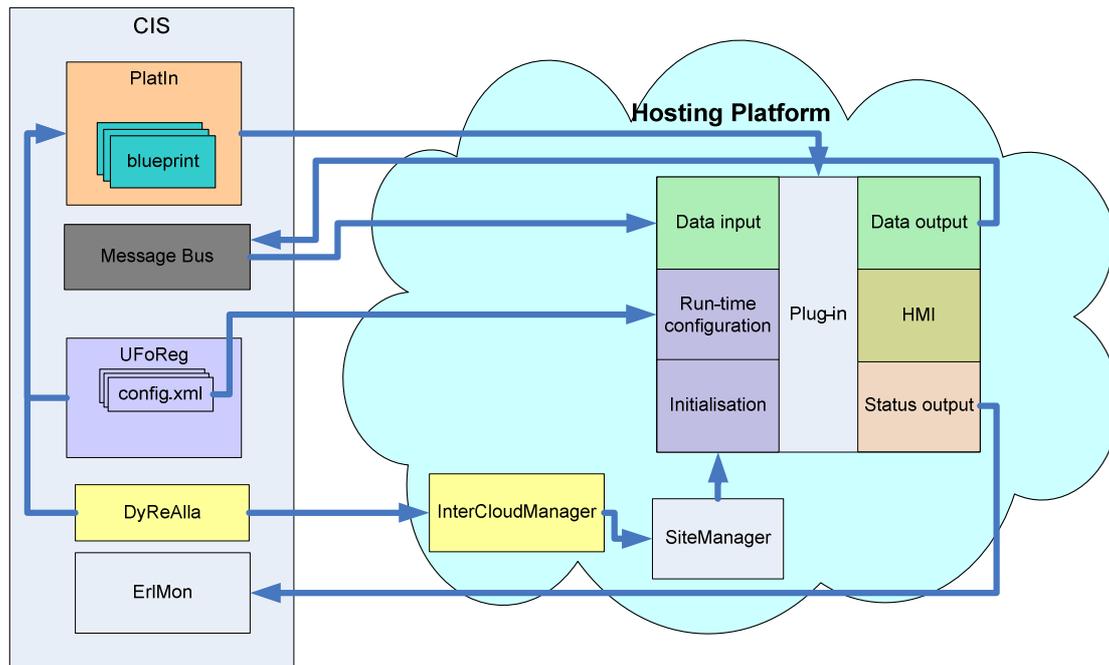


Figure 4 – Interactions of Plug-Ins, Hosting Platform and CIS

Several plug-ins were created in the course of the project (HRW Reliable, HRW Hydrograph, HRW DRFSM, AI). In section 2.3 of this document there is a case study on the design of AI component.

2.3 Case study: reference architecture of AI component

AI component developed within WP3 [4] is a core component of EWS allowing for fast detection of anomalous behaviour of the object. Behaviour is considered anomalous if it deviates from the previous observations. Previous observations as a pattern of normal behaviour are fed to the AI as training dataset. Training-based detection functions of AI are based on the neural network technologies. Besides neural networks AI has many different algorithms that can be applied to the data. Research conducted in WP6 showed that AI customization towards its application for any particular monitored object requires definition not only of the data processing parameters, but also definition of the set of input data, logical grouping of different measurements and design of the structure of data processing workflow.

First versions of the AI component were rather rigid in the structure, requiring modification of the source code to support changing structures. The challenge appeared in the matter how to support full-fledged runtime configurability of data processing chain within AI. The following architecturally-relevant requirements were formulated:

- Capability to receive and send data using protocols supported by UrbanFlood platform
- Easy integration of new protocols and data formats

- Configurability based on the runtime composition of desired workflow from a set of pre-defined processing components
- Independence of processing component implementation on used protocols and data formats
- Easy implementation and integration of new data processing components
- Support of training data set retrieval and training of neural cloud components

After a survey on technologies which can support the implementation Spring framework was chosen [10]. Spring is a Java framework mainly aimed at the development of distributed web-applications of enterprise level, and we used its core part to build a kind of component-based application.

The central part of Spring which also became the core of AI implementation is called inversion of control container. It allows for configuring and managing of Java objects called also beans. It controls the whole lifecycle of the components starting from their creation, configuration and wiring by means of dependency injection.

Specifically for AI a meta-model of components was developed based on publisher-subscriber relation for data propagation. This relation was implemented in the implementation of abstract class representing arbitrary data processing component using observer pattern. Basic decomposition of the AI component is shown in Figure 5

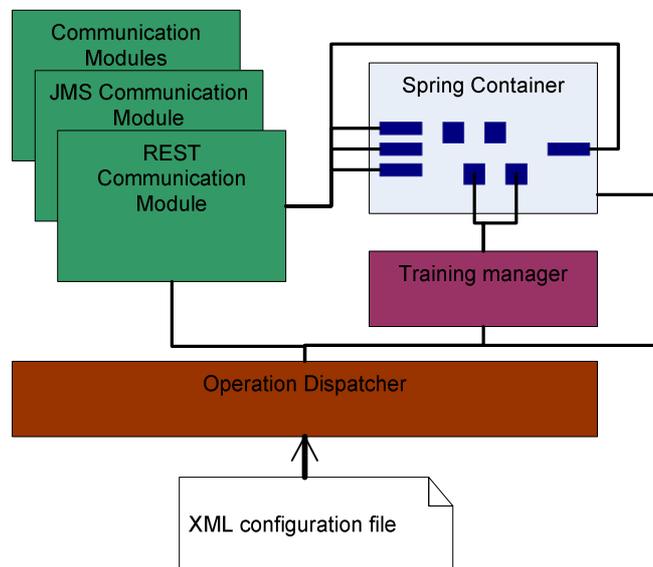


Figure 5 - AI component decomposition

Operation dispatcher is the main entry point for the AI. It reads xml-based configuration file retrieved from CIS (UFoReg) and creates the Spring container, which in its turn creates all the beans defined in the configuration and wires them appropriately. Then according to configuration the communication module is instantiated. Communication modules

implement certain communication protocols, currently there are 2 modules available: for JMS and for REST.

After initialization of communication module, training manager is activated. It fetches neural cloud beans out of the container, retrieves necessary training data via communication module and performs training sequentially feeding downloaded data to the data processing workflow established in the container.

When training is completed, communication module is connected to the Spring container and starts to receive incoming online data, feed them into the container, getting analysis results out of the container and sending them back to the other parts of the EWS.

Now 14 kinds of processing components are implemented which can form processing chain. This amount includes 2 special types of components called InputTerminal and OutputTerminal which are used to attach communication modules to the container.

Implemented architecture allows for independent development of new types of processing components and communication modules since they follow clearly defined structure and interfaces. Configuration file consists of two main parts: the first one defines communication configuration in a protocol-specific way and the second contains bean definitions as per Spring specification. As for all other UrbanFlood components (see [3]) the AI configuration file is stored in UrbanFlood registry (UFoReg) which is a part of CIS and is retrieved upon AI start up via HTTP.

Application of Spring framework allowed to achieve significant flexibility and extensibility of AI component. Advanced capabilities of the Spring framework could facilitate implementation of many attractive features of a component in future development. These features include:

- Integration with databases for data caching and data processing by means of data access module
- Rich web application for component-specific data visualization and component configuration and fine-tuning by means of model-view-controller module
- Provision of remote services to third party applications via remote access module.

Reference architecture developed for AI component can be reused for creation of other modular and flexible Java-based applications which can then be wrapped into virtual appliances to be used in UrbanFlood platform to build up EWSs. Detailed description of AI architecture and design can be found in [4].

3 Future outlook

The UrbanFlood project is addressing possibilities to build up a platform for easy development and deployment of internet-based early warning systems validating the approach on flood protection structure monitoring. Main goals of the EWS are forecasting or early detection of disturbances aimed at minimization of negative sequences of protection structure failures. Current progress of the project demonstrated that EWS benefit very much from using Internet in different ways:

- as data transmission media
- as means for data presentation via different kinds of web-based HMI
- as the means to integrate different applications in a service-oriented manner
- as the means to connect experts from different part of the world.

Internet has even bigger potential which can be utilized in future development of EWS though. Wide spreading of mobile internet and internet-enabled devices (smartphones, tablet PCs and communicators) facilitates quick delivery of warning information from the EWS to general public. But the communication can go in the opposite direction as well – information can be easily gathered from the field by means of digital cameras of mobile devices and in conjunction with exact positioning which is also supported by majority devices in the market may help to deliver evidence of the situation development to the experts and decision makers.

Social media as a group of Internet-based communication can also be used to deliver warning information. Some web-based geographic information systems provide an open API to mark events on a map, e.g. Google Alerts.

Internet and internet services have also gateways to some more traditional or conventional services as landline phones, mobile phone short message service etc.

As soon as detected or predicted disturbance might affect people's lives it is extremely important not only to make a decision on mitigation measures, but also to convey the message to all affected. Internet technologies open wide range of possibilities to do that. To support all those capabilities future EWS shall be integrated tighter into public services of the Internet.

4 References and bibliography

- [1] TNO (2009a). UrbanFlood proposal Annex I – “Description of Work”
- [2] UrbanFlood D5.5 Common Information Space Technology for Europe
- [3] UrbanFlood D5.2 Specification of the architecture and interfaces of the Common Information Space
- [4] UrbanFlood D3.3 Experiences with online dike monitoring systems
- [5] UrbanFlood D6.2 Basic Remote Dike Monitor
- [6] UrbanFlood D6.2 Operational DSS viewer
- [7] UrbanFlood D6.3 Specification of plug-in technology
- [8] UrbanFlood D6.5 Running multiple EWS on distributed clouds.
- [9] UrbanFlood D4.1 The design and prototyping of a decision support system for flood control
- [10] Spring framework web-site <http://www.springsource.org>

Annex 1 Assessment of Deliverable D6.6

Relevance	<i>D6.6 Summarizes the achievements of WP6 focusing on the utilization of Internet technologies</i>
Efficiency	<i>N/A</i>
Effectiveness	<i>The objective of D6.6 was to summarize all of the development results of the tasks 6.2 – 6.4.</i>
Impacts	<i>D6.6 is one of the required deliverables. It contributes to objective 3 and 4.</i>
Sustainability	<i>See paragraph 3 of this document, describing the potential directions of further improvements.</i>
Coherence	<i>Paragraph 2 of this document describes the UrbanFlood EWS and utilization of Internet services.</i>
Quality	<i>Where appropriate the references to the validation and testing results are given throughout the document</i>

