Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 106–115

Procedia Computer Science

International Conference on Computational Science, ICCS 2011

Flood early warning system: design, implementation and computational modules

V.V. Krzhizhanovskaya[a,b,g,*], G.S. Shirshov[a,b], N.B. Melnikova[a,b,g], R.G. Belleman[a], F.I. Rusadi[a], B.J. Broekhuijsen[c], B.P. Gouldby[d], J. Lhomme[d], B. Balis[e], M. Bubak[e], A.L. Pyayt[a,f], I.I. Mokhov[f], A.V. Ozhigin[f], B. Lang[f], R.J. Meijer[a,c]

[a] *University of Amsterdam, The Netherlands*
[b] *St.Petersburg State Polytechnic University, Russia*
[c] *TNO, The Netherlands*
[d] *HR Wallingford, UK*
[e] *Department of Computer Science & ACC Cyfronet AGH, Poland*
[f] *Siemens , Russia*
[g] *National Research University ITMO, St. Petersburg, Russia*

Abstract

We present a prototype of the flood early warning system (EWS) developed within the *UrbanFlood* FP7 project. The system monitors sensor networks installed in flood defenses (dikes, dams, embankments, etc.), detects sensor signal abnormalities, calculates dike failure probability, and simulates possible scenarios of dike breaching and flood propagation. All the relevant information and simulation results are fed into an interactive decision support system that helps dike managers and city authorities to make informed decisions in case of emergency and in routine dike quality assessment. In addition to that, a Virtual Dike computational module has been developed for advanced research into dike stability and failure mechanisms, and for training the artificial intelligence module on signal parameters induced by dike instabilities. This paper describes the *UrbanFlood* EWS generic design and functionality, the computational workflow, the individual modules, their integration via the Common Information Space middleware, and the first results of EWS monitoring and performance benchmarks.

Keywords: UrbanFlood; early warning system; flood; modeling; dike stability; Flood Simulator; Virtual Dike; artificial intelligence; neural clouds

1. Introduction and generic design of early warning systems

Recent catastrophic floods around the world have spawn a large number of projects aimed at the development of stronger and "smarter" flood protection systems. Many projects, among which are FLOODsite, FloodControl 2015, International Levee Handbook [1], attempt to solve some of the flood control problems. One of the most challenging problems is the design of Early Warning Systems (EWS) for flood prevention and disaster management.

* Corresponding author. *E-mail address*: V.Krzhizhanovskaya@uva.nl

1877–0509 © 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
doi:10.1016/j.procs.2011.04.012

Design of an EWS requires developments in a number of technologies and areas of expertise:

- Sensor equipment design, installation and technical maintenance in flood defense systems;
- ICT for sensor data transmission, filtering and analysis;
- Middleware for connecting sensor data, relevant documents, analysis tools and modeling software
- Computational models and simulation components for analysis of dike stability, prediction of dike failure, simulation of possible flood dynamics and optimization of evacuation strategies;
- Advanced interactive visualization technologies;
- Development of a decision support system that will assist public authorities and citizens in choosing the right flood protection tactics and in managing emergency situations;
- Internet-based or dedicated remote access to the early warning and decision support systems.

The *UrbanFlood* EU FP7 project [2-4] is the first endeavor that works on all these development aspects of a fully integrated EWS capable of monitoring tens of thousands kilometers of flood defense systems. To scale up from a single dike to a pan-European system, several generic design aspects are addressed in the *UrbanFlood* system:

- Fast semi-automatic generation of new EWS's for new dikes, easily deployable and ready for full integration into the hierarchical higher-level EWS system;
- Adaptive system that can host multiple EWS for many individual dikes, or for different types of EWS for the same dike. A known issue is the compatibility of different standards, platforms and modules.
- Flexibility and modularity within one EWS: computational modules can be swapped for other available models, without the need to re-design the system architecture or modify the other modules linked in a scenario-based workflow;
- On-demand resource provisioning: computational Clouds & Grids, data storage, network broadband, throughput capacity of the AI filters and computational modules;
- Self-monitoring and robustness of the early warning systems, including the systemic health, modules performance, computing resources availability and benchmarking, response to user requests, etc.

The details of generic design and functionality aspects listed above are covered in [20].

Historically, flood protections systems were monitored by inspectors walking thousands of kilometers; each piece of a dike was inspected once in a few years. Modern technologies enabled non-stop monitoring with the use of remote sensing technology, satellite imaging, fiber-optics and electronic sensors installed in the dikes. The *UrbanFlood* project relies on *in situ* monitoring, especially on the developments and results of the IJkDijk experiments in the Netherlands (www.ijkdijk.eu). Sensor records from the dikes brought to failure are used for calibration and validation of the EWS.

The key challenge in building a reliable EWS is the development of advanced modelling and simulation tools for accurate predictions of dike (in)stability, and for simulation of dike breaching and flood propagation. For fast predictions and routine dike quality assessment, hydrological engineering models are employed. Some of the models had been developed within EU FP6 project FLOODsite [1], and within the *UrbanFlood* project they have been modified to take real-time streams of sensor data as input parameters and to communicate with other modules via the Common Information Space.

For advanced research into dike stability and failure mechanisms, a Virtual Dike software is developed, based on the concepts described in [22,23,24]. The role of the Virtual Dike is two-fold: (1) It can help experts to build and calibrate simplified engineering models by providing fundamental and detailed multiscale simulations of structural mechanics coupled to the fluid dynamics; and (2) It can generate the "virtual sensor" signals observed in a dike alarming conditions for training the AI modules. The Virtual Dike simulations are based on the finite element method and require substantial computing resources (supercomputers in case of 3D-*t* simulations of fluid-structure interaction). To this end, we have secured access to SARA supercomputing centre [15] and deployed some of the simulation components on SARA Clouds within the Big Grid initiative.

UrbanFlood EWS employs a multi-stage filtering process, with sensor anomaly detection is the first step, and more complex and realistic models invoked in the next steps, to provide more detailed information as required by the decision support system. The workflow of this multi-step simulation cascade is described further in Section 2. Computational models for dike reliability analysis, dike breaching and flood propagation are described in Sections 3, 4, 5 correspondingly. Section 6 presents the *Virtual Dike* for expert-level modeling. Section 7 describes EWS implementation and integration in the Common Information Space. EWS monitoring and performance results are shown in Section 8. Section 9 concludes the paper.

2. *UrbanFlood* early warning system computational workflow

Computational workflow of the EWS is organized as shown in Fig. 1: The **Sensor Monitoring** module receives data streams from the sensors installed in the dike. Raw sensor data is filtered by the **AI Anomaly Detector** that identifies abnormalities in dike behavior or sensor malfunctions. The **Reliability Analysis** module calculates the probability of dike failure in case of abnormally high water levels or an upcoming storm and extreme rainfalls. If the failure probability is high then the **Breach Simulator** predicts the dynamics of a possible dike failure, calculates water discharge through the breach and estimates the total time of the flood. After that, the **Flood simulator** models the inundation dynamics. For expert users, the **Virtual Dike** component is available. All these EWS modules are described in more detail in Sections 2-6.

Information from all the modules is fed into the **Decision Support System** (DSS), some elements of which are shown in Fig. 2. The primary user-interface to this DSS system runs on the *Microsoft Surface* interactive graphics device (further called *Surface* for short). The *Surface* is a "multi-touch" system, which means that multiple people can collaboratively work with the applications, using their hands and fingers as input devices. The combination of multiple people interacting with the application at the same time makes the use of this device very intuitive and helpful for crisis situations –exactly what Early Warning Systems are designed for. The simulation modules and visualization components are integrated into the Common Information Space. They are accessed from the interactive graphical environment of the multi-touch table or through a web-based application.

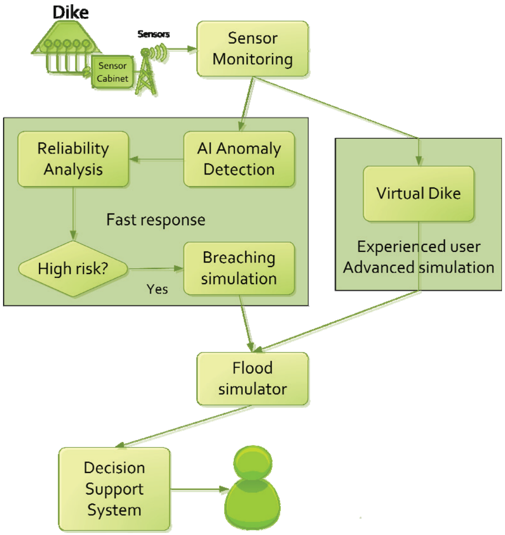


Fig. 1. *UrbanFlood* early warning system workflow

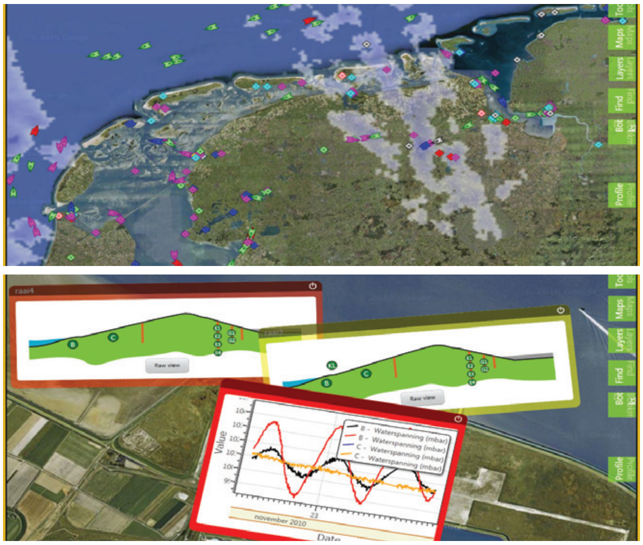


Fig. 2. Examples of decision support information modules accessed via the interactive multi-touch *Microsoft Surface* environment

3. Artificial Intelligence Anomaly Detector

Structurally stable dikes are considered to be in a "normal" state. For the real-life flood defense systems, there are no historical records of the "abnormal" sensor parameters in critical pre-failure conditions. For the "abnormality" detection, we use a one-side classification concept based on the Neural Clouds (NC) method [5]. NC approach allows detection of anomalies based on the historical measurements related to "normal" behavior used for training. After the training phase, Artificial Intelligence (AI) component processes the measurements from sensors in real-time and calculates the anomaly probabilities. One of the main advantages of the AI methods in environmental data analysis is the possibility to apply a model-free data-driven approach, which requires only the measured data.

The NC classification algorithm uses pre-processed data and/or a set of features extracted from the data as inputs. First, data is clustered by an advanced k-means clustering algorithm [5]. Second, the clustered data is encapsulated using an extended radial basis functions network approach (Fig. 3). These two operations are applied to the training

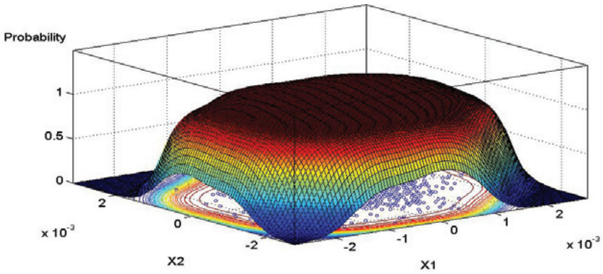


Fig. 3. Neural Cloud encapsulator visualized for two sensor parameters X1, X2. Dike "normality" confidence levels are shown by the colored 3D surface.

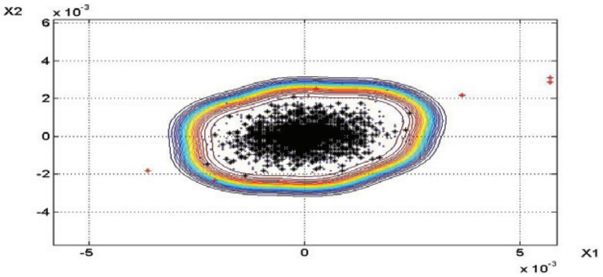


Fig. 4. 2D projection of the Neural Cloud encapsulator. Confidence levels are shown by the contour lines. Sensor parameters X1 and X2 laying within the red-colored area correspond to safe dike conditions.

data set forming an N-dimensional diagnostic parameter space. After training, a dike "normality" confidence value is calculated for each new set of measurements, by projecting it to the constructed data encapsulator (Fig. 4). The probability of normal behavior close or equal to 1 corresponds to the safe dike behavior (similar to the reference data), while values close to zero indicate a previously unknown and potentially abnormal behavior. The AI component is adaptive: in case the reported anomaly was not a critical issue and should be considered as normal, the AI component is retrained on the extended reference data set.

The first version of the AI component has been developed and integrated in the *UrbanFlood* EWS via the Common Information Space as explained in Section 7. A set of classification agents forming a committee of classifiers is constructed for dike abnormality detection. In present prototype, the committee consists of NCs constructed for every sensor measuring more than 1 parameter. After that, the calculated probabilities are used to estimate the condition of the whole cross-section or the whole dike. Further plans for the AI component development include construction of more sophisticated committees and implementation of additional methods for signal pre-processing and feature extraction. Visualization of the AI component in operation and abnormality detection results are shown in Fig. 5.

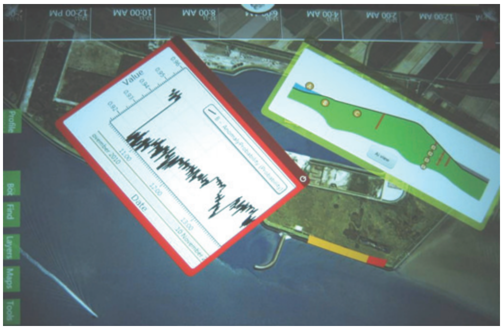


Fig. 5. Screenshot of a multi-touch table with the dike normality confidence level calculated by the AI component for one of the sensors located in a selected cross-section

4. Dike reliability analysis

To trigger warnings and further computational flood simulations, exceedence thresholds are specified, based on the probability of structural failure for the monitored dikes. For flood dike structures, the probability of failure is often defined conditionally upon the hydraulic loading (i.e. flood water levels and wave action), the resulting probability distributions are known as fragility curves. These are typically developed using standard reliability analysis, see for example [6,7,8]. In traditional structural reliability analysis, failure of a structure arises when the loads acting upon the structure, exceed its bearing capacity, or resistance. The probability of failure is often presented in the following generalized form [9]:

$$Pf = P[G(X) \leq 0] = \int .... \int_{G(X)\leq 0} f_X(x)dx$$

where G is the Limit State Function (LSF); X is the vector of random variables associated with the loads and resistance; and $f_X(x)$ is the multivariate probability density function of X. For the development of fragility curves, the probability of failure is derived, conditional on the loading conditions:

$$P_f = P[G(X) \leq 0 | L = l]$$

In some cases, the relationships between loads and the resistance are known explicitly and G (X) can be expressed in closed form or the problem can be suitably simplified such that this is the case. A range of methods can then be employed to solve the multi-dimensional integral. The most flexible and commonly applied approach is through Monte-Carlo simulation. Under the EU funded FLOODsite project (FP6) [10] we documented the limit state equations for a wide range of flood defense structures and their potential failure mechanisms. In other cases more complex models are required to describe the failure processes. When these models are computationally intensive, vanilla Monte-Carlo become impractical to implement. Response Surface methods can then be employed [9] to reduce the computational burden and implement the analysis. An example of this type of implementation in the context of flood defense is described in [11].

Under the EU funded FLOODsite Project, a software tool RELIABLE was developed to analyze the reliability of flood defenses and generate fragility curves. The tool includes a total of 72 failure modes [10] represented as simple Limit State Equations (LSEs), a flexible fault tree component, and a probabilistic failure analysis component based on Monte Carlo simulation (MCS). It is applicable to foreshores, dunes and banks; dikes and revetments; walls; and point structures, and accounts for hydraulic loading due to water level difference across a structure; wave loading; and lateral flow velocities. In the system described here, this software tool is being deployed in an online environment. Information from the sensor systems is used to determine values of parameters that are input to the LSE’s. The reliability analysis is then conducted in real time and the threshold probability of failures monitored. Threshold exceedence then triggers downstream modeling and warning activities.

5. Dike breaching and flood simulations

The Flood Simulator module predicts the flood dynamics once a dike is considered failed. This involves two calculations: estimation of the discharge through the breach and computation of the flood spreading. These two calculations are currently done separately, and it is envisaged that those tools could be swapped for other models in the future without changing the functioning of the Flood Simulator module.

The breach discharge calculation is a simplified model based on the following concept: the breach initiates with a given width, this width is unchanged as the breach invert level decreases linearly with time until reaching the toe level of the dike, then the breach invert level stays unchanged and the breach width increases according to one of the two following empirical equations:

$$W_{sand} = \frac{67\log t}{0.145} \text{ for a sand dike and } W_{sand} = \frac{20\log t}{0.08} \text{ for a clay dike,}$$

where W is the breach width (m) at time t. t is the time (hrs) after breach reached the lowest invert level. For all these stages, the discharge is calculated using the weir equation $Q = CLh^{1.5}$, where C=1.7, L is the breach width (m), h is the head of water on eroding crest (m) and is calculated using the load on the dike (water level) and the breach invert. An example of the breach width dynamics and related water discharge through this breach is shown in Fig. 6.

The flood spreading is computed by the rapid flood spreading model (RFSM), a simplified and computationally efficient model yet sufficiently robust for use in flood system risk models. The model was originally developed as a volume spreading approach with no temporal component [12,13]. This has however been extended to include the time domain (Dynamic RFSM or DRFSM).

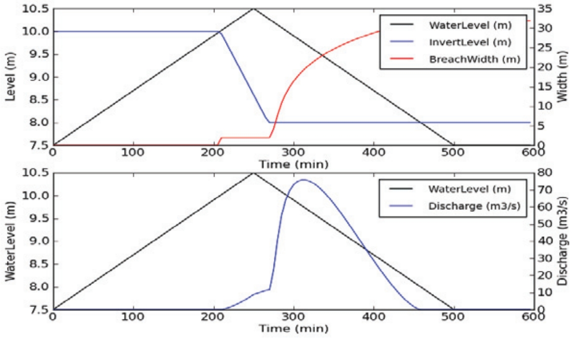


Fig. 6. Time dependence of dike breach width and water discharge. Water levels used in this example are not linked to the possible flood conditions

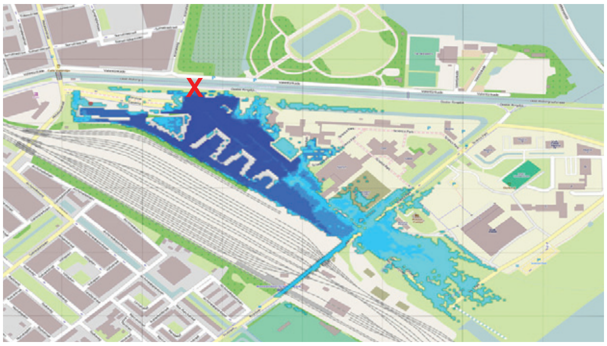


Fig. 7. Simulated inundation of Amsterdam Science Park, the University of Amsterdam campus. Marked with the red cross X is the location of a hypothetical breach in the Ring Dike

In a pre-processing stage, the domain is discretised in irregular shaped computational elements. These so called impact zones (IZs) are delineated around depressions in the topography. Input to this pre-process is the floodplain topography in the form of a Digital Terrain Model (DTM). Each IZ captures the underlying topography by the means of a table giving the volume of water stored in the IZ for different flood levels. This mesh allows to speed up the simulation by reducing the number of computational elements compared to the initial number of cells in the input DTM. The use of the level-volume relation means that the computation of the water level in an IZ is more precise than the use of an averaged ground level for situations where the IZ is not entirely flooded.

The model receives flood volumes discharged into floodplain areas from breached or overtopped defenses and then spreads the water over the floodplain according to the terrain topography (Fig. 7). Spreading of flood water is achieved by transferring water between IZs at each computational time-step. The discharge between IZs can be calculated by two methods, the Manning relationship (i.e. similar to diffusion wave models) or the weir relation. The computational time-step is constant. Water level, average discharge and average velocity are calculated in each IZ during the computation.

This approach can be used in probabilistic flood risk analysis where multiple runs are required, or in real time situations (flood forecasting), where the model run time is critical. Flood Simulator has been validated against available flood data and more advanced models.

An interactive visualization application was designed for the Surface which allows a new simulation to be interactively defined: a breach can be added by simply touching on the desired location on a map, pop-up windows allow the characteristics of the dike to be defined, the hydrograph can be interactively adjusted and the total simulation time as well as simulation step length. Once defined, the simulation is delegated to the CIS by the touch of a button. The output of the flood simulator is visualized on the Surface represented as an animated time series of water level in each Impact Zone. The simulation results are visualized on top of a topological representation of the area which can be selected from a range of sources, including Google Maps, Bing Maps, Yahoo! Maps, and OpenStreetMap. Interaction methods allow the visualization to be navigated (zoom, scale, rotate) and introspected (such as a graph that reflects water level over time in the touched Impact Zone). In Fig. 7 the flooding of the Science Park area of Amsterdam is shown. For this experiment we obtained DTM data from “Actueel Hoogtebestand Nederland” (AHN, http://www.ahn.nl/) of the Amsterdam Watergraafsmeer area. The total area used in the simulation measures 4.4 by 2.8 km with a resolution of 25 m^2 per pixel.

6. Virtual Dike

Virtual Dike is an advanced multiscale multi-model simulation lab for expert users and model developers [18]. This virtual lab is used for validation of all the models involved in the modeling cascade, and serves as a research field for experiment planning and understanding the underlying physical processes influencing dike stability and failure. Comparison of simulation results with the experimental data allows determining the material properties and computational model parameters that best represent real-life dikes, with all their inhomogeneities and special features. In the first stage of the project, we have studied the structural stability of the LiveDike, a sea dike in Eemshaven. LiveDike is protecting a seaport in Groningen. This dike has been equipped with sensors, and data stream is available in real-time. Pore pressure and dike inclination sensors are placed in four dike cross-sections. These cross-sections have been simulated in 2D models under tidal water loading. Simulation results have been compared with the pore pressure sensors data in order to calibrate model parameters.

The modeling approach is based on a coupled fluid-structure interaction with non-linear dike material properties. The fluid part of the model describes the dynamics of flow through porous soil, with Richards equation for wetting and drying of the area above phreatic surface. The Van Genuchten model [17] is used to describe the properties of unsaturated soil. As pore volume expand with the expansion of the media, water content increases. This process is taken into consideration with additional source term in the right hand side of the Richard’s equation that dscribes relative permeability by Van Genuchten equation. The structural part of the model describes deformation dynamics of the dike under tidal pressure load, gravity and volumetric pore pressure load obtained from flow simulation. Linear elastic constitutive relation is used for describing sand and clay properties [14].

Dike stability is evaluated by the Mohr-Coulomb failure criterion [16]. Harmonic approximation of water level sensor signal is applied as input parameter for the flow boundary conditions (to define pressure level at the sea side). At the land side, water stays at the constant average see level. The remaining boundaries are treated as impermeable.

In the structural task, corresponding transient pressure is applied at the sea side and constant pressure below average sea level at the land side. The base of the dike is fixed. The remaining boundaries are free from structural loads. The equations are solved by the finite element method, with the computational mesh refined in the area of a dynamically changing phreatic surface. A more detailed description of the Virtual Dike models and simulation results can be found in [18], and parallel performance results in [19].

The resulting transient fields of pore pressure, water content and structural deformations have been obtained. Structural displacements field in one moment in time is shown in Fig. 8. The displacements are composed of (a) the static soil settlement under gravity load (maximal at the top of the dike) and (b) transient displacements resulting from the tidal pressure at the seaside and volume pore pressure load. Total displacements are maximal at the top of the dike due to gravity settlement component.

A comparison of real and simulated water pore pressures for one sensor location is shown in Fig. 9. A real sensor signal is shown with a bold line; a "virtual" (simulated) sensor signal is shown with a thin line. The amplitude of simulated pore pressure oscillations is higher than the real amplitude, which indicates that the soil permeability around this sensor must be reduced in the model. That would also make real and simulated pressure oscillations more synchronous in this point. Thus local inhomogeneities shall be included into the model in future simulations, in order to obtain good agreement of pore pressure dynamics for all sensors.

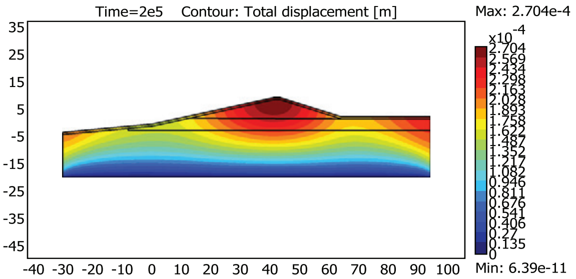


Fig. 8. Distribution of structural displacements in a cross-section of the LiveDike in Eemshaven

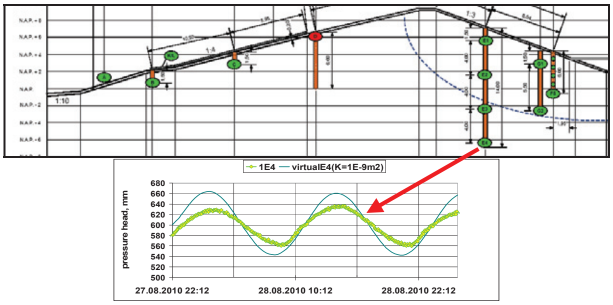


Fig. 9. Pore pressure dynamics obtained from real and "virtual" sensors

7. EWS implementation: integration in the Common Information Space

In order to organize individual components into a working Early Warning System, *UrbanFlood* provides the *Common Information Space* (CIS) [20], a generic framework for creating and hosting any EWS systems which work according to a four-step cycle:

1. Monitoring: data is collected in real time from sensors and fed to the EWS.
2. Analysis: one or more EWS-specific applications/workflows are invoked to perform analysis of sensor data streams, such as anomaly detection or simulation based on expert models.
3. Value judgment: the outcome of this analysis is judged according to EWS-specific rules in order to estimate the risk level or determine the necessity of taking action.
4. Advice/act: if mandated by the value judgment, further EWS-specific applications/workflows may be invoked in order to recommend actions for users interacting with a decision support system, or automatically take action to influence the system.

The design and implementation of CIS draws upon our previous experience in development of the ViroLab Virtual Laboratory [21]. EWS implementation is schematically shown in Fig. 10. Each EWS component, discussed in Sections 3-6, has been wrapped as a virtual appliance, i.e. a virtual machine image which contains a minimum platform and the component exposed as a service. The appliances are deployed on physical resources organized in a Cloud and managed through a Cloud management layer.

The main services of the CIS include: (1) *Component invocation*: invoking the appliances through a Service Bus, therefore supporting a variety of communication protocols; (2) *Workflow orchestration:* implementation of data- and control-flows between EWS components, such as the workflow described in Section 2; (3) *Data exchange*: enable the components to exchange (publish and consume) data through a common message bus; (4) *Self-monitoring*: monitoring of the EWS itself in order to detect failures and, ultimately, provide self-healing capabilities; (5) *Resource orchestration*: dynamic management of virtual machine instances and their allocation to available resources in order to improve performance and ensure timely execution of critical tasks.

CIS is organized as a set of software services. The most important are:

- Integration platform (PlatIn): CIS core technologies for component integration, data exchange and workflow orchestration.
- Metadata registry (UFoReg): a generic service to host and query metadata; currently it hosts metadata describing currently active Early Warning Systems, sensor data, and available resources.
- Dynamic resource allocation service (DyReAlla): a service for dynamic allocation of resources to running Early Warning Systems.

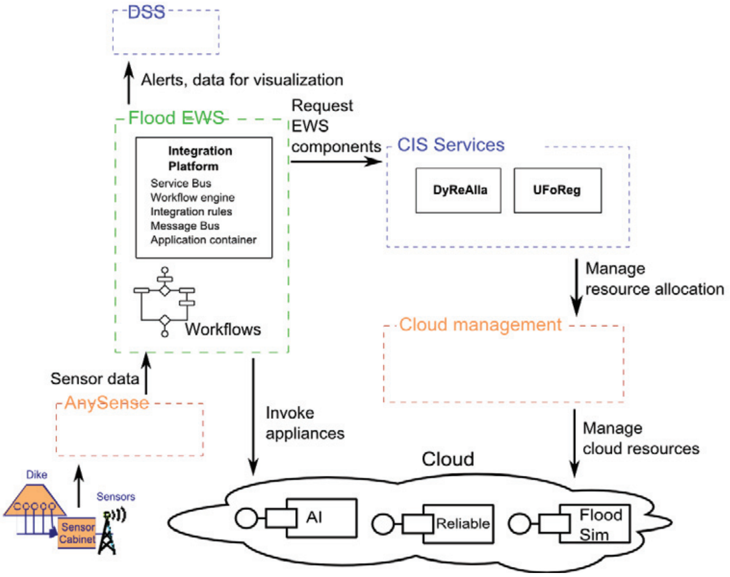


Fig. 10. Flood EWS implementation: EWS components are deployed in the Cloud as virtualized appliances. The Common Information Space Integration Platform invokes the components and implements workflows to orchestrate data- and control-flows between them. Additional CIS services: DyReAlla and UFoReg manage dynamic resource allocation and metadata.

Robustness is a key requirement for Early Warning Systems. Consequently, the current implementation of CIS relies on several mature and stable technologies. At the same time the framework does not impose any particular technology for developing Early Warning System components, therefore avoiding the risk of vendor lock-in. This is achieved by adopting the Service-Oriented Architecture (SOA) principles. EWS parts are loosely coupled services with well defined interfaces described in WSDL, which can be accessible through a variety of protocols including SOAP, REST, JMS or FTP. These services do not exchange data directly but through a message bus, therefore there are no direct dependencies between them. As a result, they can be developed independently using different technologies. For example, currently we support both Apache Camel and OpenESB / BPEL for component integration and workflow orchestration. Apache ActiveMQ, a JMS implementation, is used as the message bus.

To increase robustness even further, the same mechanics of the CIS are used to create an EWS that instead of having sensors in an embankment, has sensors in the other EWSs, thereby monitoring the state of the appliances, workflows and data streams in an EWS. This sensor data is part of the four step process mentioned at the start of this section. After it has been fed into the self-monitoring EWS, it is processed by the AI to provide anomaly detection in the various signals. If anything is out of the ordinary, the DyReAlla component is notified and action can be taken. Specific models that take into account the behavior of the appliances are an ongoing research.

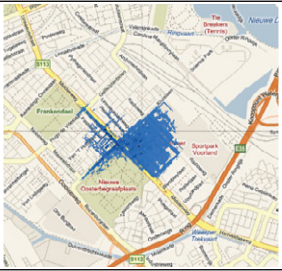
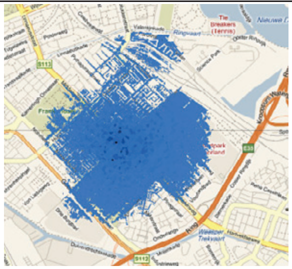
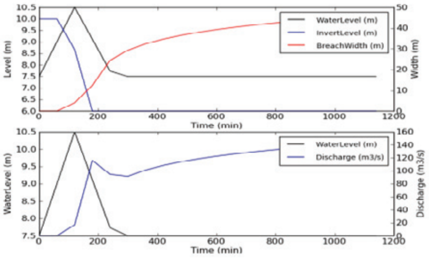
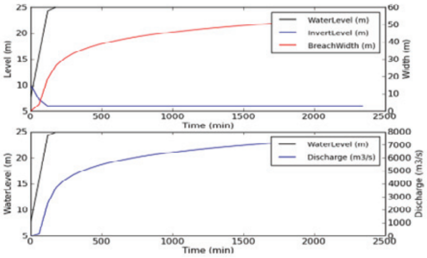
The EWS components have been deployed in a distributed manner across three partner sites, and some modules have been ported to the SARA Supercomputing Center Clouds [15]. The cloud is hosted on a 128-core cluster with the following characteristics: 16 compute nodes; CPU dual quad-core 2.2 GHz; 24 GB RAM per node; 500 GB local hardisk; 100 TB backup storage, network 1 Gb/s per node; 20 Gb/s aggregated connection from the cluster to storage. SARA uses OpenNebula open source cloud computing management toolkit and KVM virtual machines.

8. Performance results of the interactive flood simulation and visualization system

Despite the complexity of the system architecture, simulations are computed and visualized within a timeframe of less than one minute. This allows the system to be used for interactive testing of different flooding scenarios ("what-if" experiments). Table 1 illustrates this for two scenarios that used the same simulation parameters but the hydrograph in the second scenario generates more water influx than the first, as illustrated at the bottom of the table.

The hydrograph affects the flooded area and consequently the size of the simulation output data that needs to be transmitted every time step. The table shows that the time for each simulation is well below one minute and the time between updates is within seconds, even for scenarios in which the demand for computational resources is significantly higher.

Table 1. Performance measurement of the interactive flooding simulation and visualization for two scenarios with a different hydrograph.

	Scenario 1		**Scenario 2**	
Simulated time (minutes from T0)	*Update duration (seconds from previous time step)*	*Data Size (KB)*	*Update duration (seconds from previous time step)*	*Data Size (KB)*
300	+4.78	18	+7.46	70
600	+1.70	47	+4.06	263
900	+1.35	53	+5.24	350
1200	+1.54	56	+5.66	381
1500	+1.31	63	+6.15	404
1800	+1.09	69	+6.22	427
2100	+1.63	77	+6.06	448
Total	13.40s	373 KB	40.85 s	2342 KB
Flood image at time step 2100 min				
Hydrograph				

9. Conclusions and future work

We described a prototype of the *UrbanFlood* Early Warning System (EWS) that includes an AI module for sensor data anomaly detection, and a cascade of models for dike stability analysis, dike breaching and flood propagation. All the components of the EWS are connected in an intelligent workflow via the Common Information Space, and accessed interactively from an end-user decision support system interface. In addition, we presented the *Virtual Dike*, an advanced modelling tool developed for fundamental studies of dike stability coupled to the flow dynamics. We have simulated the LiveDike in Eemshaven under the sea tides and flood conditions. Further, we experimented with the Flood Simulator in a scenario of the RingDijk breach, in which case the University of Amsterdam team would be flooded in their Science Park campus.

The EWS has been implemented and demonstrated to dike stakeholders and flood management authorities during the *Joint UrbanFlood & SSG4Env Workshop on Monitoring and Flood Safety*. A very positive response and interest in the DSS system showed that the ideas and implementation approaches were chosen appropriately.

In the next stage of the project, we will enhance the functionality of the *UrbanFlood* EWS, taking into account the feedback from targeted groups of end-users. The simplified dike breaching simulation will be substituted by a more accurate model, and a new model of city evacuation will be added for assisting in flood emergency management. The EWSs will be equipped with self-monitoring and self-generation capabilities, so that for example parallel EWSs can be generated on request of the dike stakeholders.

The *Virtual Dike* simulator will model heterogeneous soils inside the dikes and real sea-storm conditions. Next, we will model dike failure mechanisms based on the IJkDijk experiments, including slope instability, piping and surface erosion. In addition, we plan to explore the system identification theory approach to build simplified models based on advanced modeling and sensor data analysis. Finally, the *Virtual Dike* will be integrated into the Common Information Space to receive sensor input automatically and to produce real-time simulation results for displaying them on a multi-touch table or a web-based system interface.

Acknowledgements

This work is supported by the EU FP7 project *UrbanFlood*, grant N 248767; by the *Leading Scientist Program* of the Russian Federation, contract 11.G34.31.0019; and carried out in collaboration with AlertSolutions, BiG Grid for e-Science, Deltares, IJkDijk Association, Rijkswaterstaat, Ministerie van Verkeer en Waterstaat, SARA Computing and Networking Services, UvA IBED-CGE, UvA GIS Studio, WaterNet, Waterschap Noorderzijlvest.

References

1. FLOODsite http://www.floodsite.net, Flood Control 2015 http://www.floodcontrol2015.com, The International Levee Handbook project http://www.leveehandbook.net/about-project.asp
2. *UrbanFlood* EU FP7 project http://www.urbanflood.eu
3. V.V. Krzhizhanovskaya. A roadmap to multiscale modeling of flood defense systems: from sand grain to dike failure and inundation. Proc. of ASME 2010 Computers and Information in Engineering Conf. IDETC/CIE 2010, Montreal, Canada. Paper # DETC2010-28967
4. B. Gouldby, V.V. Krzhizhanovskaya, J. Simm. Multiscale modelling in real-time flood forecasting systems: From sand grain to dike failure and inundation. Procedia Computer Science, V. 1, N 1, ICCS 2010, May 2010, p. 809
5. Lang, B., Poppe, T., Minin, A., Mokhov, I., Kuperin, Y., Mekler, A., Liapakina, I.: Neural Clouds for Monitoring of Complex Systems. Optical Memory and Neural Networks (Information Optics). 2008. V. 17. N 3, pp. 183–192, (2008)
6. M. Schultz, B. Gouldby, J. Simm, J. Wibowo, Beyond the Factor of Safety: Developing Fragility Curves to Characterize System Reliability. ERDC SR-10-1. G. a. S. Laboratory, USACE, 2010.
7. S. Vorogushyn, B. Merz and H. Apel. "Development of dike fragility curves for piping and micro-instability breach mechanisms." Natural Hazards and Earth System Sciences 9(4): 1383-1401, 2009.
8. J. Simm, B. Gouldby, P. Sayers, J-J. Flikweert, S. Wersching, M. Bramley, Representing fragility of flood and coastal defenses: Getting into the detail, Flood Risk Management: Research and Practice – Samuels et al. (eds), 2009.
9. R. Melchers, Structural Reliability Analysis and Prediction. New York, Wiley, 2009.
10. W. Allsop, A. Kortenhaus, M. Morris. Failure Mechanisms for Flood Defense Structures. FLOODsite Report. T04_06_01. 2007.
11. G. Kingston, M. Rajabalinejad, B. Gouldby, Van Gelder P H A J M. "Computational intelligence methods for the efficient reliability analysis of complex flood defense structures " Structural Safety in press, 2010.
12. Gouldby B., P. Sayers, J. Mulet-Marti, Maam Hassan and D. Benwell (2008). "A methodology for regional-scale flood risk assessment." Proceedings of the Institution of Civil Engineers-Water Management 161(3): 169-182
13. J. Lhomme, P. Sayers, B. Gouldby, P. Samuels, M. Wills and J.R.. Mulet-Marti. Recent development and application of a rapid flood spreading method. Proceedings of the FloodRisk 2008 Conference, Oxford, Taylor and Francis Group, London, 2008.
14. A.R. Koelewijn, M.A.Van. Monitoring of the test on the dike at Bergambacht: design and practice. Proceedings of XIII ECSMGE 2003, Prague, Czech Republic.
15. SARA Computing and Networking Services, http://www.sara.nl/
16. A. Verruijt. Soil Mechanics. Delft University of Technology, 2001. 315 pages.
17. M.T. van Genuchten. Soil Science Society of America Journal 44: 892-898.
18. N.B. Melnikova, G.S. Shirshov, V.V. Krzhizhanovskaya. Virtual Dike: multiscale simulation of dike stability. Proceedings of the International Conference on Computational Science, ICCS 2011. Procedia Computer Science 00 (2011) 000–000.
19. N.B. Melnikova, G.S. Shirshov,V.V. Krzhizhanovskaya, N.N. Shabrov. Virtual Dike and Flood Simulator: Parallel distributed computing for flood early warning system. Proc. Intl Conf. on Parallel Computational Technologies PAVT-2011, pp. 1-9, 2011, Moscow (In print).
20. B. Balis, M. Kasztelnik, M. Bubak, T. Bartynski, T. Gubala, P. Nowakowski, J. Broekhuijsen. The UrbanFlood Common Information Space for Early Warning Systems. Proc. Intl Conf. on Computational Science, ICCS 2011. Procedia Computer Science 00 (2011) 000–000.
21. M. Bubak et al. Virtual Laboratory for Collaborative Applications. Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine, and Healthcare, chapter 27, pp. 531-551. Medical Information Science Reference, 2009.
22. V.V. Krzhizhanovskaya. *A Virtual Reactor for Simulation of Plasma Enhanced Chemical Vapor Deposition.* PhD thesis. Publ.: PrintPartners Ipskamp, Enschede. 2008.
23. V.V. Korkhov, J.T. Moscicki and V.V. Krzhizhanovskaya. IEEE Systems Journal, V. 3, Issue 1, pp. 121-130. 2009.
24. V.V. Krzhizhanovskaya, V.V. Korkhov. Proc. PACO-2006, Moscow, Russia. Publ: Institute of Control Sciences RAS, 2006, pp. 917-932.