A Development and Execution Environment for Early Warning Systems for Natural Disasters

Bartosz Baliś, Tomasz Bartyński, Marian Bubak, Grzegorz Dyk, Tomasz Gubała, Marek Kasztelnik
Motivation: Early Warning Systems against natural disasters

- Complex distributed systems
  - Require compositions of resources into complex scenarios
- Mission & time-critical
- Data analysis & simulations → data- & resource-intensive computing
- "Spiky" behavior (variable workload)
Example: Urban area flood simulation
Goals for the Common Information Space for EWSs

- A platform facilitating development, deployment and execution of EWSs
- **EWS development**
  - EWS reference model
  - EWS development framework
- **EWS deployment**
  - EWS blueprints
  - EWS-factory-as-a-service
- **EWS execution**
  - CIS runtime services for resource allocation, self-monitoring, self-healing, mission-critical operation, and urgent computing
UrbanFlood
Online Early Warning System Factory
Common Information Space

Domain resources exposed as **Basic Services**
Data, sensors, apps wrapped as appliances and deployed onto clouds, ...

**Composite Services (Parts)**
Building blocks for EWSs
Orchestrate domain resources towards complex application scenarios (e.g. area flood simulation)

**Early Warning System**
A number of Parts deployed, connected, and configured for a specific setting (e.g. a dike section)
**CIS: metadata registry**

**UFoReg: Metadata Registry**

"Memory" of the system

Metadata about EWSs, services, dikes, sensors, cloud resources, etc.

**GUI for EWS composition & management**

**High-level RESTful APIs**

EWS discovery, configuration & execution
CIS: Architecture and EWS reference model

PlatIn: integration platform + workflow
Programming framework for service development
Runtime management of EWS instances
Runtime orchestration of workflows

Using enterprise integration patterns approach (Apache Camel technology)
CIS: Optimization of resource allocation & self-healing

**DyReAlla + ErlMon**

- **On-demand resource provisioning**
  - From local resources & clouds

- **Horizontal scaling of infrastructure**
  - More instances of appliances are added based on EWS importance level

- **Load balancing**
  - Proxy (http traffic)
  - Task queues (lazy evaluation)

- **Self-healing**
  - On-line availability monitoring
  - E-mail notifications about problems
  - Automatic restart of failed components
CIS as a system factory

System factory
System blueprint

Service factory

Virtual Appliance

Virtual appliance store

Invoke

Service

Systems instance

Service

Configurations

Configurations
Flood Early Warning System

• Protection of urban areas against floods due to dike failures
• Real-time monitoring of embankments using wireless sensors
• A cascade of models for:
  ◦ Anomaly detection
  ◦ Risk assessment
  ◦ Impact prediction
Flood EWS workflow

Cascade of models invoked for different alert levels

- Level „0”: AI-based anomaly detection
- Level „1”: Dike stability calculation
- Level „2”: Simulation: prediction of inundation, evacuation & life loss

→ Organized by the Common Information Space into a working Early Warning System
Dike anomaly detection

- Anomaly detection using neural cloud classification
Dike reliability analysis

- Calculation of probability of dike structural failure
- Based on reliability analysis approach
- Fragility curves depending on the current water level
Dike breach & flood simulation
Life Safety Model

- Loss of life estimate due to flood

Initial state of the world →

Post-event summary → (red color = deceased)
Virtual Dike

- Computer model of a dike
- Simulation of dike behaviour
- Useful for „what-if” experiments
- Also for testing the real EWS („emergency drill”)

Flood EWS: implementation

Dike with wireless sensors deployed

System control center

AnySense
(sensor data service and archive data repository)

Control (e.g. simulation requests)

Sensor data

CIS Message Bus

Simulation requests

Flood Simulation Part

Simulation results

Life safety simulation Part

Simulation requests

Virtual dike Part

Sensor data

Anomaly probability

Dike failure probability

Alert level change

Sensor data / Anomaly probability

Sensor Anomaly Monitoring (AI appliance)

Dike Stability Monitoring Part
Example: Flood Simulation Part

System control center

Flood Simulation Service

Inputs:
- (Link to) terrain elevation data
- (Link to) dike properties
- Breach location
- Water levels (time series)

Outputs:
- (Link to) inundation prediction data

Map Service (WMS)

Inundation data service (WCS)

Inundation data

Area map

Data archiver

Write simulation results

Inundation data sets

Terrain elevation data service (WCS)

Terrain elevation data sets

CIS Runtime

Invoke

Flood Simulation Service (WPS)

Publish Inundation data

Message Bus

Platin service (EWS management)

UFoReg (Metadata registry)

ErlMon service (Self-monitoring)

DyReAlla service (resource allocation)

Manage resource allocation

HR Wallingford Dynamic RFSM

HR Wallingford Hydrograph

OS

OS
CIS as a geo-ICT technology

- Adoption of OGC standards for service interfaces
- Parts can be developed and deployed as Web Processing Services (WPS)
- Capability to consume spatial data services compliant with WCS (Web Coverage Service) and WFS (Web Feature Service)

→ CIS as a factory for spatial processing services
Conclusion

CIS contributes conceptually and technologically in the following areas:

- **CIS as a factory for Early Warning Systems**
  - Proven by the implementation of the Flood EWS

- **CIS as a geo-ICT technology for spatial data processing services**
  - CIS adopts guidelines of the INSPIRE directive and leverages OGC standards for service interfaces

- **CIS as a runtime infrastructure for resource-intensive mission-critical systems**