UrbanFlood WP5
Common Information Space (CIS) after Year 1

Marian Bubak, Bartosz Baliś and WP5 team
ACC Cyfronet AGH, Kraków, Poland

{bubak,balis}@agh.edu.pl
Plan

- Motivation, goals, tasks, and the team
- Objectives for Year 1
- Results of requirement analysis
- Common Information Space
  - architecture
  - implementation status
- CIS-based Flood EWS
- Summary of achievements
- Plans for Year 2
Motivation & Goals

- Facilitate creation, deployment and robust operation of Early Warning Systems
- **Early Warning System**: any system working according to four steps:
  - Monitoring
  - Analysis
  - Judgment
  - Action
- Example: *environmental monitoring*
- Also: cloud infrastructure monitoring, EWS self-monitoring
Tasks and the Team

- WP5 leader: Marian Bubak
- T5.1: Execution framework
  - Integration platform: Marek Kasztelnik, Bartosz Balis
  - Data access: Piotr Nowakowski, Tomasz Gubala
  - Resource allocation: Tomasz Bartynski
  - Development of CIS-based EWSs: Bartosz Balis, Marek Kasztelnik, Jeroen Broekhuijsen, Artem Ozhigin
- T5.2: Event infrastructure
  - Communication Bus: Bartosz Balis, Marek Kasztelnik,
  - Self-monitoring: Grzegorz Dyk, Bartosz Balis
- T5.3: Provenance logging (Metadata)
  - Generic metadata service: Tomasz Gubala, Piotr Nowakowski, Adam Belloum
- T5.4: User & developer GUls
  - Jeroen Broekhuijsen, Tomasz Gubala
Timeline

Milestones:

M12: The CIS integrates the (distributed) software modules of the EWS
M24: The CIS manages (online) computing resources
M33: The CIS has been demonstrated to SEIS and/or INSPIRE representatives
Requirement analysis

- Existing sources of live and archive sensor data
- Existing legacy applications
  - Heterogeneous platforms
- Complex scenarios emerge, involving:
  - Real-time processing of sensor data
  - Decision-making, automatic or by human interaction
  - Compute-intensive simulations
  - What-if scenarios
- High priority (urgent) computing
- Sharing of limited resources
System requirements

- Application integration: enable data exchange between diverse components
  - Legacy, heterogeneous
- Orchestration of components into *composites*
  - Scientific / business workflows
  - Integration patterns
- Dynamic resource allocation management
- Metadata management
  - Domain (applications, data)
  - System information (resources, data sources, running tasks)
  - Provenance
Overview of CIS architecture
PlatIn: the Integration Platform

- Start and stop EWS capability implemented *(Every EWS part delivers management WS)*
- Integrated with UfoReg *(receiving EWS metadata)*
- EWS parts can be created using BPEL or Camel integration technology
UFOReg: the UrbanFlood Registry

- Currently includes three data models:
  - Sensor metadata model
  - EWS model (describing EWSs and their appliances)
  - Cloud model (describing the available hardware resources and virtual system images, and recent cloud state)

- Stable prototype deployed at Cyfronet and populated with initial sensor/EWS/Cloud data
- Basic GUI available
- RESTful API interfaces are in place to enable integration of UFOReg with DyReAlla and visualization components
DyReAlla: Dynamic Resource Allocation service

- Integrated with cloud infrastructure *(receiving reports about resource availability and load metrics)*
- Integrated with UfoReg *(storing cloud status data)*
- Not in the first milestone / release of CIS
Paradigms and technologies used

- Enterprise Service Bus (Open ESB)
- Business Process Management (BPEL)
- Enterprise Application Integration framework (Apache Camel)
- Message-Oriented communication middleware (ActiveMQ)
- Robust NoSQL database (MongoDB)
Early Warning System – the CIS view

- Each EWS is composed of: (1) Appliances, (2) Parts, (3) External components

- **Appliance**: application component exposed as a service and wrapped into a virtual image
  - Typically legacy EWS-specific application

- **EWS Part**: composite integrating and orchestrating control and data flow between appliances (and possibly other parts)
  - Also can be exposed as a high-level service

- **External component**: external producer or consumer of data, not managed by CIS
EWS Part

- **Generic operations:**
  - Start, stop
  - Change alert level

- **Invokes appliances**
  - A well-defined high-level service exposed via part-specific interface

- Sends messages to and receives from a message bus
Loose coupling

- Parts are loosely coupled
- Communicate via the message bus
- No direct dependencies between parts
- Facilitates extendibility of the EWS
  - Addition of new parts that further process already published data
  - Connecting new visualization front-ends
  - Can even be done at runtime
Attention (alert) level

- General state of every EWS shared by all its parts
- A Part may send 'alert level change' (raise or decrease) message
- **Alert Level Manager** (general-purpose specialized EWS part) receives the messages, calculates a new alert level, and sends 'alert level set' message
- Other EWS parts may adjust their operation depending on the alert level
Flood Early Warning System

- Monitoring of dikes using wireless sensors
- AI-based detection of sensor signal anomalies
- Dike failure prediction
- Simulation of inundation due to failure
- Visualization and user interactions on Multi-touch Tables
Flood EWS implemented with CIS

Legend:
- EWS Part
- CIS Technology
- Appliance
- External component
- UFoReg (CIS metadata registry)
Self-monitoring & cloud monitoring as Early Warning Systems

Each EWS has:

- Its own instance of **PlatIn**, the CIS integration platform,
- Specific appliances in the cloud
- EWS Parts: Composites which orchestrate the execution of appliances

Multiple EWSs:

- EWS1, EWS2: two instances of Dike Monitoring EWS
- EWS3: Monitoring and management of other EWSs (**DyReAlla** and **UFoReg**)
- EWS4: Cloud monitoring and management
Deliverables, milestones & meetings in Y1

- **Deliverables:**
  - D5.1 Common Information Spaces (description of state of the art and future developments) (M8)
  - D5.2 Specification of the architecture and interfaces of the Common Information Space (M9)
  - D5.3 Orchestrating the information flow in a Common Information Space (M12)

- **Milestones:**
  - M12: The CIS integrates the (distributed) software modules of the EWS

- **Meetings:**
  - Kick-Off meeting in Groningen
  - Consortium meeting in Wallingford (definition of first EWS)
  - Integration meeting in Cracow (first EWS prototype)
  - Integration meeting in Amsterdam (first EWS production run)
  - **Teleconferences:** every Wednesday (current issues discussions, progress reports)
Publications & presentations

Plans for Y2

Task 5.1
- Improvement of functionality of all modules (e.g. support for self-monitoring)
- Implementation of the dynamic resource allocation mechanism
- Further implementation and refinement of components of the CIS-based flood EWS.

Task 5.2
- Development of a robust software sensor network for self-monitoring
- Improvement of scalability and flexibility of self-monitoring infrastructure
- Improvement of robustness of the communication services.
Plans for Y2 (ctd)

Task 5.3
- Continuous optimization of storage, gathering and publishing techniques inside UFoReg
- Support for the provenance model
- Continuous extending domain models when requested by partners and required by the CIS/EWS design

Task 5.4
- Implementation of graphical dashboards for users and developers for monitoring status and health of CIS and EWSs
- Integration of CIS services and CIS-based EWS services with graphical user interfaces of the Decision Support System
- Development of auxiliary developer tools for automation of CIS-based EWS development.
Means to high-quality software

- Adoption of mature, industry-quality, standards-based approaches and technologies
  - Apache, MongoDB, OpenESB, BPEL, Passenger
- At the same time adoption of platform and technology-independent design in order to avoid vendor lock-in
  - SOA, loose coupling between components
  - Communication based on well-defined protocols
- Self monitoring inherent part of the CIS design
- Test driven development
Hardware contributed by Cyfronet

- **Computer host (Y1)**
  - 2 dualcore processors Intel Xeon CPU 5150 @2.66GHz, 4 GB RAM, 500 GB storage
  - Deployed services
    - CIS (ActiveMQ JMS, Glassfish server, UFoReg, DyReAlla, Health monitoring)
    - EWS parts (message filters, converters etc)

- **4 nodes (from February 2011)**
  - 2 quadcore processors Intel Xeon CPU L5420 @2.50GHz, 16 GB RAM, 120 GB local storage, access to shared storage over iSCSI
  - Dedicated for hosting UF appliances (Flooding simulator, Reliable, Hydrograph, DRFSM, AnySense, AI)
urbanflood.cyfronet.pl
www.urbanflood.eu
dice.cyfronet.pl/cis