

Dedicated IT infrastructure for Smart Levee Monitoring and Flood Decision Support

Bartosz Balis^a, Tomasz Bartynski, Robert Brzoza-Woch, Marian Bubak, Daniel Harezlak, Marek Kasztelnik, Marek Konieczny, Bartosz Kwolek, Maciej Malawski, Piotr Nawrocki, Piotr Nowakowski, Tomasz Szydło, Bartosz Wilk, Krzysztof Zielinski

AGH University of Science and Technology
Faculty of Computer Science, Electronics and Telecommunications
Department of Computer Science
Al. Mickiewicza 30, 30-059 Kraków, Poland

Abstract. Smart levees are being increasingly investigated as a flood protection technology. However, in large-scale emergency situations, a flood decision support system may need to collect and process data from hundreds of kilometers of smart levees; such a scenario requires a resilient and scalable IT infrastructure, capable of providing urgent computing services in order to perform frequent data analyses required in decision making, and deliver their results in a timely fashion. We present the ISMOP IT infrastructure for smart levee monitoring, designed to support decision making in large-scale emergency situations. Most existing approaches to urgent computing services in decision support systems dealing with natural disasters focus on delivering quality of service for individual, isolated subsystems of the IT infrastructure (such as computing, storage, or data transmission). We propose a holistic approach to dynamic system management during both urgent (emergency) and normal (non-emergency) operation. In this approach, we introduce a Holistic Computing Controller which calculates and deploys a globally optimal configuration for the entire IT infrastructure, based on cost-of-operation and quality-of-service (QoS) requirements of individual IT subsystems, expressed in the form of Service Level Agreements (SLAs). Our approach leads to improved configuration settings and, consequently, better fulfilment of the system's cost and QoS requirements than would have otherwise been possible had the configuration of all subsystems been managed in isolation.

1 Introduction

Levees monitored with in situ sensors – so-called *smart levees* – are gaining momentum as a flood protection technology [1]. We present the ISMOP IT infrastructure for smart levee monitoring and flood decision support [2,3]. The main purpose of the ISMOP project is the development of a comprehensive system that supports decision makers in flood risk management by providing information on dynamics and intensity of processes that occur in river levees. The main concept is to enrich the river levees with sensors, mainly for measuring pore pressure and temperature, and then process the collected data using geophysical models. As a result of data processing, it will be possible to assess the likelihood of the levee breach, levee stability as well as determine not only the time and place but also the cause of a breach. For the purpose of the research, the experimental smart levee located in the Lesser Poland region, near the Vistula River (Figure 1) has been built. It is equipped with the IT infrastructure for its monitoring, in order to perform controlled flooding experiments and study the behavior of levees exposed to long-term water infiltration, typical for flood scenarios in Lesser Poland

where long-lasting flood waves frequently cause levee failures.

In this paper we describe the ISMOP IT infrastructure dedicated to smart levees and comprising a number of subsystems for data acquisition, transmission, storage, processing and presentation. While the prototype of the ISMOP IT infrastructure was built as a monitoring



Figure 1. The ISMOP experimental smart levee.

system for the experimental levee, it was designed to support decision making in large-scale emergency situations, affecting river areas protected by hundreds of kilometers of levees. Such scenarios require resilient and

^a Corresponding author: balis@agh.edu.pl

scalable IT infrastructures, capable of real-time acquisition and transmission of large volumes of sensor measurements, storage and retrieval of the collected data, and regular, timely data analyses through *urgent computing* services. Consequently, a levee monitoring system operates in two different modes: (1) a common 'normal' (non-emergency) mode in which the minimization of the operation costs is the priority, and (2) a rare 'urgent' (emergency) mode where reliability and timeliness are crucial.

The novel contribution of this paper is the *holistic approach to management of the system's configuration* during both normal and urgent operation. In this approach, a component, called the *Holistic Computing Controller*, calculates and deploys a globally optimal configuration for the entire IT infrastructure, based on cost-of-operation and quality-of-service (QoS) requirements of individual IT subsystems, expressed in the form of Service Level Agreements (SLAs). The proposed approach leads to improved configuration settings and, consequently, better fulfilment of the system's cost and QoS requirements than would have otherwise been possible had the configuration of all subsystems been managed in isolation.

The paper is organized as follows. Section 2 presents related work. Section 3 introduces the ISMOP IT infrastructure for smart levees. Section 4 explains the concept of the holistic approach to system optimization, and presents an experimental case study based on this concept. Section 5 discusses the results of the experiments. Section 6 concludes the paper.

2 Related work

Monitoring and decision support systems which focus on natural disasters often rely on urgent computing services in order to ensure sufficient supply of computer resources and the required quality of service in the event of a crisis. Urgent computing is an area of computation characterized by the presence of strict deadlines, unpredictability of crisis events and the ability to mitigate their impact through computational effort [4]. Examples of such applications include severe weather forecasting workflows [5], simulation applications [6], storm surge modelling applications [7], and wildfire forecasting workflow [8]. The core characteristic of urgent computing scenarios is not their resource-critical nature, but rather the temporal restrictions which necessitate procurement of on-demand computational and storage resources.

The operation of urgent computing systems for natural disaster prevention can be generally divided into two stages: data provisioning and data processing. Each of these stages is then subdivided into separate tasks, which carry QoS requirements. Existing approaches tend to focus on optimization of individual tasks or computational steps (e.g. resource allocation [9], data stream processing [10], or provisioning of storage resources [11]) rather than on optimizing the process as a whole. In particular, the correlation with data delivery has not been fully supported in target processing subsystems.

Complex solutions for job prioritization and pre-emption have been developed for HPC and grid computing infrastructures. Notable efforts such as the SPRUCE [12] project focus on enhancement of scheduling algorithms in order to acknowledge the presence of urgent tasks and provision the necessary computational and storage infrastructure to support said tasks. In the Common Information Space (CIS) platform [13] provided by the UrbanFlood early warning system [14] cloud services are used for on-demand deployment and autoscaling of warning system instances in emergency situations. The CLAVIRE platform [15] proposes a model-based approach to management of heterogeneous computing resources for urgent computing scenarios. In spite of this progress, however, existing work tends to deal exclusively with optimizing the data processing stage without resolving data delivery issues – the acquisition and staging of data sets required for computations (which are often massive) is taken for granted and the projects in question rely on the presence of local storage which is assumed to be secure and faultless. Using network protocols or distributed system techniques (e.g. web services, as in [7]) is alleged to bypass the challenge, but ongoing acquisition of external data (e.g. from numerous environmental sensors) and timely delivery of critical data components (including maintenance and adaptation of communication channels) are not explored sufficiently.

The necessity and usefulness of research dealing with data delivery has been recognized, e.g. in the LEAD project [5], the Urgent Data Management Framework (UDMF) [11] and SPRUCE, which proposes robust data placement mechanisms [16]. However, these solutions take advantage of existing data, whereas real-time monitoring calls for delivery of up-to-date measurements collected on the fly by a distributed sensor infrastructure - especially wireless sensors [10], which have become a common solution in recent years. Live environmental monitoring by a large number of heterogeneous nodes requires dedicated infrastructural and algorithmic solutions to ensure timely data delivery and aggregation [17].

Advanced solutions for providing efficiency and reliability to urgent data delivery have been developed. For instance, UDMF [11] introduced new capabilities into urgent computing infrastructures, including Quality of Service (QoS) or data policy management and monitoring. Urgent storage and data management tools provide QoS and SLA for data services [18].

While Tolosana-Calasanz et al. [10] recognized the significance of coping with data volumes generated by distributed sensor networks, as well as the SLA implications of having to aggregate and transfer such data, their QoS solution assumes that data is processed on the fly by a distributed computational infrastructure – which is not the case when data must be collected at a centralized location prior to processing, as in the ISMOP project.

To summarize, we have not been able to find an urgent computing system which relies on aggregation of data from heterogeneous and physically distributed devices while providing global QoS guarantees across the

entire infrastructure (as opposed to individual subsystems). The need for this type of holistic approach

frequent collection of temperature and pore pressure data from the affected areas, and turns on data- and model-

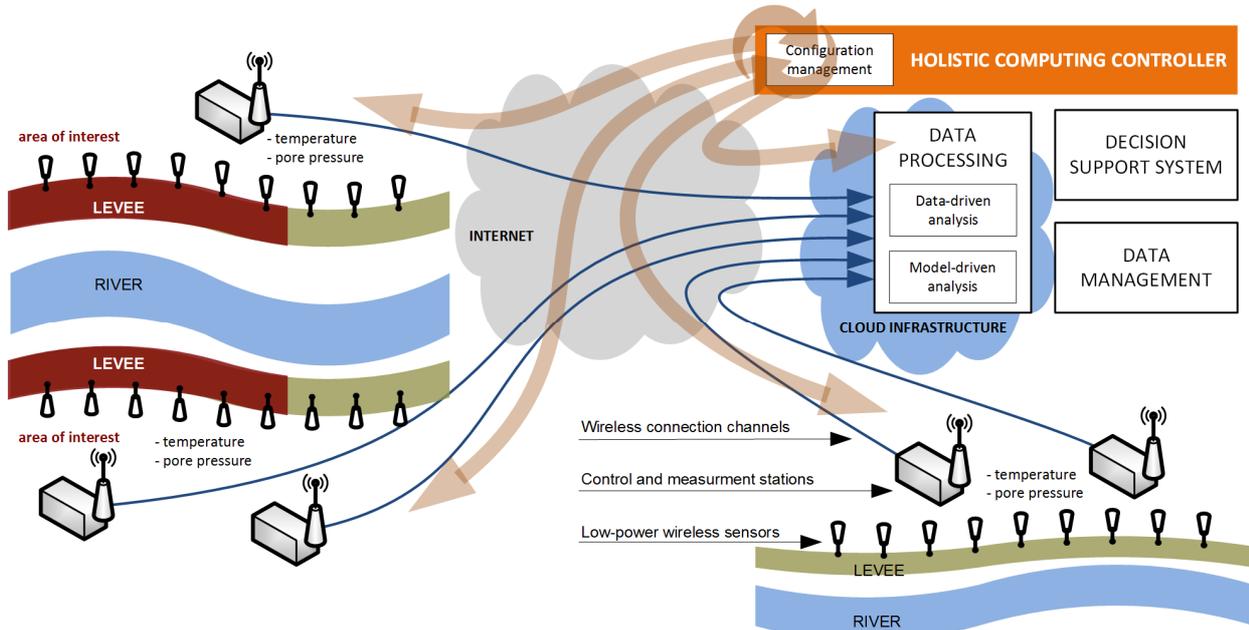


Figure 2: ISMOP smart levee monitoring and flood decision support system.

has been recognized in many domains reliant on complex event processing – including natural history, biology, education etc. [19]. Systems and their properties should be viewed as synergetic wholes, not as collections of parts [20] – whereas in the solutions presented above data processing and external data delivery are typically regarded as two separate domains. No mechanisms have been found that can leverage the overall system efficiency with the synergy of the processes.

When focusing on technological solutions, the simplest interpretation of holism would be shifting control mechanisms from the local plane to the global plane. This approach is evident in Smart Cities [21] and Software Defined Networks (SDN) [22] paradigms. In both cases, business logic is shifted from the infrastructure layer (e.g. a local traffic lights system or a network switching appliance) to the control layer (management center or business applications supported by network controllers). A large-scale decision system may thus take advantage of global knowledge and perform global – rather than local – optimization. Such an approach is also embodied in the system presented in this work.

3 ISMOP system for smart levees

The ISMOP system for smart levee monitoring and flood decision support is shown in Figure 2. The operation of the system is defined by two operating loops: (1) **Decision-support loop** controlled by *human decision-makers* observing the status of the levees in the *monitored areas of interest* and, depending on the current situation, setting the operating mode for these areas to *urgent* or *normal*. Changing the mode to *urgent* enables

driven analyses that assess the levee state and predict their future behavior. (2) **Management loop** controlled by the *Holistic Computing Controller* (HCC) component which continuously monitors the IT infrastructure and, if necessary, initiates its reconfiguration in order to optimize non-functional properties of the entire system.

Figure 3 shows a detailed architecture of the ISMOP IT infrastructure comprising the Computing and Data Management System (CDMS) and the Data Acquisition and Pre-processing System (DAPS). DAPS and CDMS are further divided into subsystems described in sections 3.1 and 3.2, respectively. The DSS is presented in Section 3.3. The Holistic Computing Controller is introduced in Section 4.

3.1 Data Acquisition and Pre-processing System

DAPS has a multilayer structure composed of three layers which acquire and transport data obtained from sensors to the Computing & Data Management System. The following layers are defined:

- **Measuring Layer** which is composed of sensors measuring physical parameters of the environment (temperature and pore water pressure). Values of these parameters are transmitted over a wired or wireless network to edge computing devices.
- **Edge Computing Layer** which is a collection of distributed computing devices which control Measuring Layer operation and perform preprocessing of measurement data (such as data compression, encryption, filtering etc.) In more advanced systems this is the place where event

processing could be effectively deployed. The operation of this layer is referred to as Fog Computing.

(EDLCs) also known as supercapacitors (SCs). The station uses a renewable energy source and is equipped with solar cells for charging its batteries.

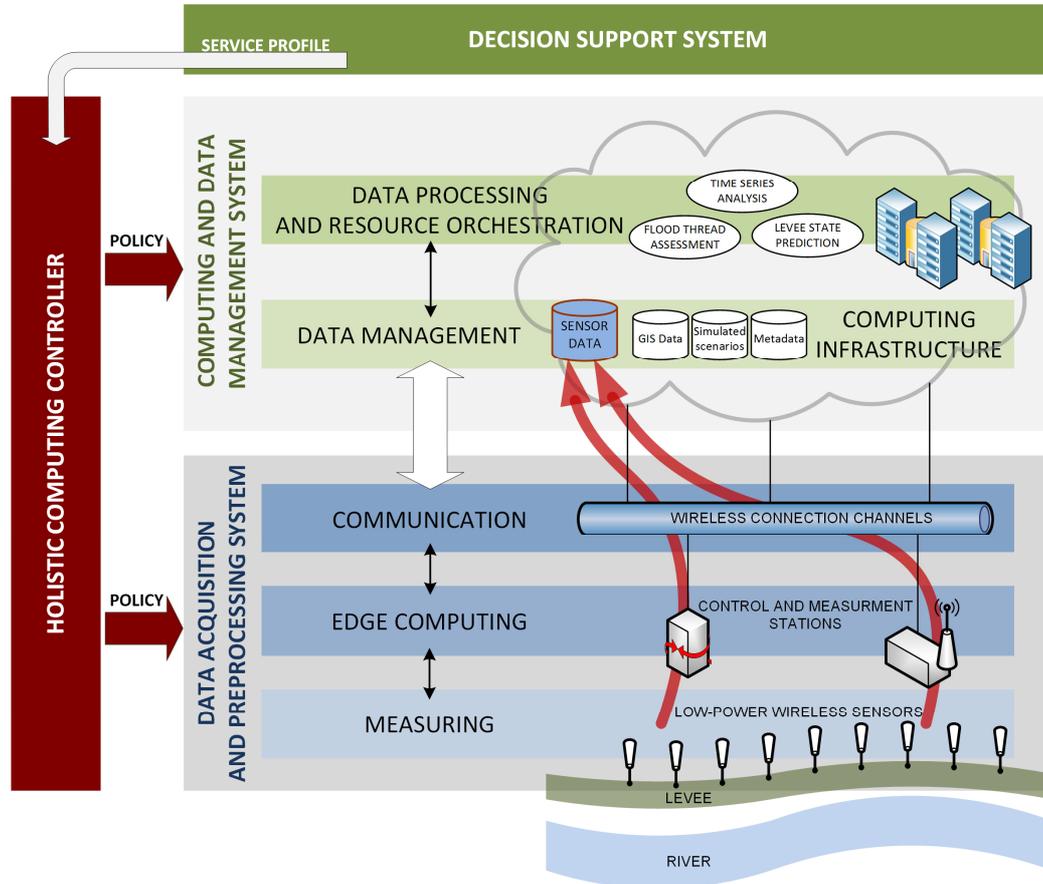


Figure 3. Architecture of the ISMOP IT infrastructure.

- **Communication Layer** which provides bidirectional communication between Edge Computing devices and CDMS. This layer performs routing operation and selects the most suitable communication technology and routes to transfer preprocessed data to the central system.

We have developed a functional prototype of the control and measurement station (Figure 4) which operates on all three layers, with particular focus on the Edge Computing Layer. The prototype consists of a specialized hardware platform and an embedded software solution (Figure 5). The control and measurement station's hardware platform is based on a modern low-power ARM Cortex-M4 microcontroller. In future, in order to achieve better control over hardware configuration, the utilization of Field Programmable Gate Array (FPGA) [23,24] technology can be considered. The station connects to the wireless sensor network's edge router and acquires data from its sensors. The acquired data is subsequently preprocessed, serialized and transmitted to higher layers of the system using the MQTT protocol [25]. Due to limited physical access to the control-measurement station hardware, we also consider substituting traditional lead-acid batteries with electric double-layer capacitors



Figure 4. ISMOP control and measurement station.

The control-measurement station uses multi-level power management features. Whenever the microcontroller operates in idle mode, the onboard real time operating system enables reduced power consumption. This mode allows the system to preserve only a fraction of total power, but supports rapid resumption of the normal operating mode. The microcontroller can also enter a deep power-saving mode in which its microprocessor core is disabled. This mode is utilized periodically when

the system has no operation to perform. The third power saving mechanism concerns controlling power supply of the peripheral modules, including sensors and communication subsystems. The station can transmit data with one of two available interfaces. By default, it uses its built-in GPRS connectivity to transmit data directly to the CDMS. The alternative path is based on XBee communication. In the latter scenario one station disseminates data to other stations until data reaches a station with sufficient cellular network coverage, enabling transmission to CDMS via GPRS.



Figure 5. Specialized hardware platform and embedded software solution.

3.2 Computing and Data Management System

CDMS comprises the following subsystems:

- **Computing infrastructure** includes physical computers along with management software, such as cloud middleware. The computing infrastructure provides the capability to dynamically allocate computing resources required for data processing, e.g. in the form of Virtual Machine (VM) instances.
- **Data management subsystem**, responsible for reliable data storage and access to data required by the decision support system, in particular sensor data received from the Communication layer. The primary responsibility of the data management subsystem is to ensure data availability and access quality (e.g. throughput and latency). The system comprises, among others, the Data Access Platform (DAP). This component collates and stores data obtained from the acquisition and preprocessing system. DAP provides a uniform logic model, storage infrastructure and access interfaces to various types of sensors, including their metadata and measurements. It is implemented as a Ruby on Rails application, which obtains input data by way of Apache Flume sinks. Flume (which is located between the communication layer and the data management subsystem) is used in order

to provide an added layer of data security since all crucial measurements and metadata items can be cached and queued for later retransmission in case of network problems. Externally, the Data Access Platform presents a selection of RESTful interfaces enabling authorized users to register new sensors, alter their properties and query for measurements using a variety of filtering options. Internally, DAP can persist its data using a variety of storage technologies. PostGIS is currently in production use, while experimental support for InfluxDB time series storage is under development.

- **Data processing and resource orchestration subsystem**, where data analyses are performed in order to assess the current levee state and forecast future behavior. These analyses may include anomaly detection, simulation of future levee state, and computation of flood threat levels for individual levee sections.

3.3 Decision support system

The main goal of the decision support system in the presented infrastructure is to visualize different aspects of the collected measurements and data analyses in order to improve levee monitoring procedures and enable a more informed decision making process. Sample views of different visualization aspects are depicted in Figure 6. Geospatial data of monitored levees and measurement devices are shown in the top left corner. A cross-section of one of the measurement profiles with a temperature gradient is visualized in the top right corner. The bottom left part of the image shows a profile with measurement devices superimposed onto the levee contour and, finally, in the bottom right corner measurement plots of selected devices are depicted. Different views of the monitored objects and data sets allow for more informed analysis of levee performance for both present and historical data. Additionally, the user interface contains several facilities to improve the analysis process. The experiment panel has a global time slider which synchronizes all the views with the current timestamp selection. Vertical and horizontal intersection views carry dedicated wizards which allow users to modify configuration details at any point and, wherever possible, an auxiliary mini-map is used to mark positions of the devices whose data is being visualized. To keep track of the current flooding experiment a dedicated widget visualizes the current water level along with the expected level to conveniently observe possible deviations.

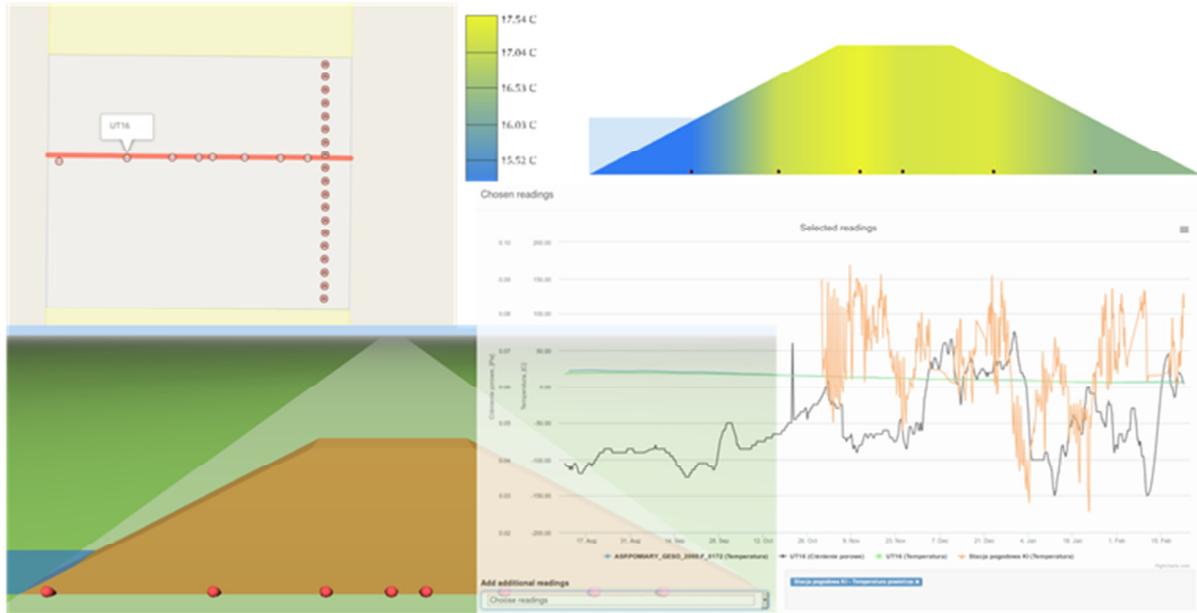


Figure 6. Dedicated user interface for smart levee monitoring visualized measurement and geo-spatial data to support the process of decision making and performing flooding scenario experimentation.

Special attention while building user interfaces was devoted to responsiveness as the data set of collected measurements (millions of readings) is quite large and unoptimized data queries could take minutes, which is unacceptable for end-users. It was assumed that the maximum data query response time should not exceed ten seconds in order for the interface to remain sufficiently responsive. While collecting data it soon became clear that additional work was needed to improve the response time of data queries. One example of improving the interface responsiveness is asynchronous data loading while changing the time constraints of readings charts and limiting the number of readings in accordance with requested chart zoom levels. This also required modifying the data storage subsystem to take additional query parameters into account.

The main user interface of the presented infrastructure relies on a web platform which currently supports rich capabilities in terms of graphical user interface development. The only user-side requirement is a modern web browser installed on any type of device. Another advantage of using a web-based approach is a wide selection of off-the-shelf graphical components, which improves the process of rapid prototyping and UI delivery.

4 Holistic system management

4.1 Non-functional properties of the system

The IT infrastructure supporting the flood decision support system is characterized by a number of *non-functional properties* related to cost and Quality-of-Service (QoS) parameters. The most important of these properties are as follows:

- **Operating cost (OPC):** expenses required to maintain operation of the system. The total OPC is

the sum of expenses for all individual subsystems, for example the cost of data transfer over the cellular network, the cost of renting computing resources from the computing infrastructure, etc.

- **Energy Efficiency (EE):** an indicator showing how energy efficient the system is – especially important in the context of limited energy availability in the lower layers as it directly influences the system’s lifetime.
- **Data measurement interval (DMI):** a specification of how often sensor parameters are captured by the measuring subsystem. The lower the value of DMI, the more frequently the measurements are captured and, consequently, the more accurate data analyses can be. However, low DMI also contributes to increased energy consumption.
- **Data processing interval (DPI):** a specification of how often data analyses, such as the assessment of the current and projected state of the monitored levee, are conducted in the data processing subsystem.
- **Data processing time (DPT):** time required to complete a single data analysis for a given area of interest.

These non-functional properties are mutually exclusive and cannot all be optimized at the same time. Moreover, their relative importance will vary depending on the system’s mode of operation. In the *urgent mode* in which the flood threat is high the decision support system should provide regular flood threat assessments for the affected levees in a reliable and timely fashion, e.g. every 30 minutes. In such a case the system should be optimized towards performance and fault tolerance at the expense of operating cost. However, most of the time the

system will operate in its *normal* (non-emergency) *mode* in which optimization of operating costs is the priority.

The system must therefore be able to dynamically manage its configuration in order to adapt to the circumstances and optimize the fulfillment of the crucial non-functional properties, possibly at the expense of others. This is the task of the Holistic Computing Controller, described in the following section.

4.2 Holistic Computing Controller

The Holistic Computing Controller (HCC), shown in

Figure 3, regularly calculates and updates the configuration of the entire system (i.e. all subsystems) in such a way as to **maintain functional operation** while **optimizing non-functional properties** of the system, given the **current context** in which the system operates. The algorithm performed by the HCC is as follows.

We assume that the system has k *configurable properties* $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}, \dots, \mathbf{p}_k)$. For each of these properties there is a set of possible *configuration options* that can be chosen $\mathbf{O}_i = (\mathbf{o}_1^i, \mathbf{o}_2^i, \dots, \mathbf{o}_{n(k)}^i)$ ($i = 1..k$). We also have information about the current *system context* (e.g. battery levels, weather conditions) $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_l)$.

We define *system configuration* as a vector of configuration options chosen for each of the configurable properties: $\mathbf{s} = (\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_k) \in \mathbf{S}$, where $\mathbf{o}_i \in \mathbf{O}_i$, while \mathbf{S} denotes the set of all possible configurations of the system.

The goal of the HCC is to optimize the system's non-functional properties $\mathbf{q} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m)$, where each $\mathbf{q}_i = \mathbf{f}(\mathbf{s}, \mathbf{c})$ is a *function of the system's configuration and context*. To this end, HCC solves the following multi-criteria optimization problem:

$$\text{minimize } \mathbf{q} = (\mathbf{q}_1(\mathbf{s}, \mathbf{c}), \dots, \mathbf{q}_m(\mathbf{s}, \mathbf{c})) \in \mathbf{Q}$$

subject to **func and sla**
 constraints

where \mathbf{q} is a vector of *objective functions*, \mathbf{Q} being the *objective space*.

$\mathbf{s} \in \mathbf{S}$ is a vector of *decision variables* (system configuration), \mathbf{S} being the *decision space*.

\mathbf{c} is a vector of input (non-decision) variables (system context).

func are *functional constraints* imposed on the decision space, for example "configuration options \mathbf{o}_2^3 and \mathbf{o}_3^4 are mutually exclusive".

sla are constraints imposing restrictions on the objective space. Typically these are low and high boundaries for the quality properties, for example *minimal system lifetime*.

The result of the optimization is a Pareto-set of feasible configurations that optimize the non-functional properties of the system:

$$\mathbf{S}^{OPT} = (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_p)$$

A question remains – which of the Pareto-optimal configurations should be chosen and actually deployed in the system? Currently HCC applies a simple heuristic: it assumes a certain importance hierarchy among the objectives and chooses the configuration that produces the best value of the most important objective. If several configurations are equally good, the second most important objective is taken into account, etc. The importance hierarchy depends on the mode of operation. In the urgent mode it is as follows (from most to least important): DPI, DMI, EE, OPC; in the normal mode the corresponding order is: OPC, EE, DMI, DPI. We are currently working on a more complex heuristic based on the AHP method [26].

4.3 Experiments

In order to practically validate the proposed holistic approach to system optimization we have performed a series of experiments using prototype implementations of hardware and software components of the ISMOP IT infrastructure. First, we have identified a number of key configurable properties for all subsystems of the ISMOP IT platform. These properties and their configuration options are presented in Table 1. A notable configuration property is *Data aggregation*: it specifies for how long the sensor data collected by the measurement subsystem can be buffered in the edge computing subsystem before being transmitted to the data management subsystem. High aggregation time saves a considerable amount of energy by minimizing the activity of the wireless network interfaces.

Subsystem	Configurable properties
Data processing and resource orchestration	Processing interval: 5,15,60,720,1440 min Scheduling algorithm: time-optimized, cost-optimized
Computing infrastructure	VM allocation policy: aggressive, conservative
Communication	Transmission protocol order: (Xbee-SMS-GPRS), (GPRS-SMS-Xbee)
Edge computing	Encryption: on, off Data aggregation: off, low, high
Measurement	Accuracy: high, low Measurement interval: 1, 5, 15, 60, 720, 1440 min

Table 1. Configurable properties and their configuration options for all subsystems of the ISMOP platform.

Next, we have chosen three most important non-functional properties as objective functions: operating cost (OPC), energy efficiency (EE), and Timeliness

(TML). The latter property is an aggregated measure of the system’s performance, responsiveness and capability to deliver timely results. TML is related to other more basic properties described above: Data Transfer Interval (DTI), Data Processing Interval (DPI), and Data Processing Time (DPT).

The goal of further experiments was to study the effect of configuration options described in Table 1 upon key non-functional properties of the system. The results of this investigation are summarized in Table 2. Each configurable property either has no effect on a given objective function (crossed-out fields), or can contribute to its low, medium or high value. In addition, we have discovered that the effect on energy efficiency also highly depends on current weather conditions, hence there are two separate columns for this objective. This phenomenon stems from the fact that the control and measurement station utilizes solar cells for charging its batteries.

Based on these findings, we have created simplified models of the objective functions, where each of them maps a configuration and context to a value between 0 and 1, where 0 denotes the minimum (lowest possible), while 1 is the maximum (highest possible) value.

Configurable property	Config. option	Objective functions / context			
		OPC	EE / Sunny	EE / Cloudy	TML
Processing interval	5,10 60 720,1440	HI MED LOW			HI MED LOW
Scheduling algorithm	Cost-opt Time-opt	LO HI			LO HI
VM allocation policy	Aggress. Conserv.	HI LOW			HI LOW
Transmission protocol ord.	Xbee... GPRS...	LOW HI	HI HI	HI LOW	LOW HI
Aggregation	Off Low High		LOW MED HI	LOW MED HI	HI MED LOW
Measurement interval	1,5,15 60 720,1440		LOW MED HI	LOW MED HI	HI MED LOW

Table 2. Effect of configuration on non-functional properties OPC, EE and TML (objective functions).

4.4 Results

Having developed the models of the objective functions, we proceeded to implement a prototype of the HCC and used it to calculate optimal configurations of the system in two different situations: 1) normal mode; 2) urgent mode. The normal and urgent modes differ with respect to the constraints and the objective hierarchy, as shown in Table 3. In the normal mode, measurements are infrequent (once every hour), while computations assessing levee

conditions are invoked rarely (every 24 hours). In the urgent mode both measurements and computations are frequent (occurring every 15 and 30 minutes respectively). Note the constraint imposed on the aggregation time: it basically states the maximum time sensor data can be buffered in the edge computing subsystem before it is required by the data processing subsystem. Finally, the hierarchy of objectives is quite different in both modes: low cost of operation is the most important in the normal mode, while timeliness is the priority in the urgent mode.

System mode	Constraints (SLAs and functional constraints)	Objective hierarchy
Normal	DMI ≤ 1 hour DPI ≤ 24 hours Aggregation time < DPI – DPT	OPC, EE, TML
Urgent	DMI ≤ 15 min DPI ≤ 30 min Aggregation time < DPI – DPT	TML, EE, OPC

Table 3. Two different modes of system operation, their associated constraints, and objective hierarchy.

System mode	# Configurations meeting constr.	# Pareto-optimal configurations
Normal	408	102
Urgent	192	64

Table 4. Sizes of the decision space (configurations meeting constraints) and the Pareto set (Pareto-optimal configurations) in both system modes.

Table 4 shows the sizes of the entire decision space (number of all possible configurations meeting constraints) and the number of Pareto-optimal configurations for both system modes. The single optimal configurations for both modes finally selected by the HCC are shown in Table 5. Note that the optimal value of OPC is 0.0 (minimum cost), while for EE and TML it is 1.0 (maximum energy efficiency and timeliness, respectively). In the normal mode the HCC chose a configuration which results in the best OPC (0.0). Since there were 12 such configurations, HCC chose the one with the highest EE.

Profile	Configuration	Objective function (best value)		
		OPC (0.0)	EE (1.0)	TML (1.0)
Normal	Process. interval 1440 Scheduling alg. cost-opt VM all. policy conser. Protocol order Xbee... Aggregation high Meas. Interval 60	0.00	0.68	0.16
Urgent	Process. interval 5 Scheduling alg. time-opt VM all. policy aggres. Protocol order GPRS... Aggregation none Meas. Interval 1	1.00	0.33	1.00

Table 5. Optimal configurations selected by the HCC in two situations: 1) normal mode, 2) urgent mode.

5 Discussion

Thanks to the holistic view of the system, HCC can perform global optimization of its configuration which leads to better configuration settings than would have been chosen had all subsystems been configured in isolation, based on SLAs pertaining only to these subsystems. For example, only the holistic view of the system allowed setting the Edge computing subsystem's *Data aggregation* property to *high* – i.e. at least 12 hours – in the normal mode (Table 5), even though the *Measurement interval* was much lower – only 60 minutes. This was possible because the HCC could take into account the constraint imposed on the *Processing interval* (a property of a completely different subsystem) which, in this case, was much higher than the *Measurement interval*. This, along with the rule that $Aggregation\ time < DPI - DPT$, enabled finding the configuration which maximizes OPC. Without the HCC, the *aggregation* property would have been set to *none* by the edge computing subsystem because there would have been no indication that sensor data was not required immediately by the upper subsystems. This, in turn, would have resulted in greater power consumption at the edge node as communication would have to be initiated more frequently. For such a configuration, the values of OPC, EE and TML would have been equal to 0.0, 0.34, 0.32, respectively. Given that the OPC is the most important property in the normal mode, clearly it would have been a worse configuration compared to the one achieved with the holistic approach.

6 Conclusion

This paper presented the ISMOP IT infrastructure and flood decision support system designed to support decision making in large-scale emergency situations. The presented holistic approach to system optimization allows the system to achieve better non-functional properties during operation in both urgent and normal modes than would have otherwise been possible with the isolated approach. Future work involves:

- additional experimental studies in order to develop more precise models of the objective functions,
- design of a better heuristic for selection of the best configuration from the Pareto set,
- investigation of additional configurable properties and their interplay across different subsystems,
- implementation of an efficient solver for multi-criteria optimization within the Holistic Computing Controller,
- further development of the prototype components of the ISMOP system.

Acknowledgments

This work is partially supported by the National Centre for Research and Development (NCBiR), Poland, project

PBS1/B9/18/2013; AGH statutory research grant no. 11.11.230.124 is also acknowledged.

The authors are grateful to Professor Rob Meijer (University of Amsterdam and TNO, The Netherlands) for fruitful discussions.

7 References

1. Krzhizhanovskaya V.V., Shirshov G.S., Melnikova N.B., Belleman R.G., Rusadi F.I., Broekhuijsen B.J., Gouldby B.P., Lhomme J., Balis B., Bubak M., Pyayt A.L., Mokhov I.I., Ozhigin A.V., Lang B., and Meijer R.J. (2011). Flood early warning system: design, implementation and computational modules. *Procedia Computer Science, Proceedings of the International Conference on Computational Science, ICCS 2011, Vol. 4*, pp. 106–115.
2. Brzoza-Woch R., Konieczny M., Kwolek B., Nawrocki P., Szydło T. and Zielinski K. (2015). Holistic Approach to Urgent Computing for Flood Decision Support. *Procedia Computer Science, Vol. 51*, pp. 2387–2396.
3. Balis B., Kasztelnik M., Malawski M., Nowakowski P., Wilk B., Pawlik M. and Bubak M. (2015). Execution management and efficient resource provisioning for flood decision support. *Procedia Computer Science, Vol. 51*, pp. 2377–2386.
4. Leong S. H. and Kranzlmüller D. (2015). Towards a General Definition of Urgent Computing. *Procedia Computer Science, Vol. 51*, pp. 2337–2346.
5. Droegeleier K. K., Clark V. Ch, R., Gannon D., Graves S., Ramamurthy M., Wilhelmson R., Brewster K., Domenico B., Leyton T., Weber D., Wilson A., Xue M. and Yalda S. (2004). Linked environments for atmospheric discovery (lead): A cyberinfrastructure for mesoscale meteorology research and education. *20th Conf. on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology, Bulletin of the American Meteorological Society*, pp. 901–911 .
6. Cui Y., Moore R., Olsen K., Chourasia A., Maechling P., Minster B., Day S., Hu Y., Zhu J., Majumdar A. and Jordan T. (2007). Enabling very-large scale earthquake simulations on parallel machines. *Computational Science - ICCS 2007, Lecture Notes in Computer Science, Springer Berlin Heidelberg, Vol. 4487*, pp. 46–53.
7. Bogden P. S., Gale T., Allen G., J. MacLaren, Almes G., Creager G., Bintz J., Wright L. D., Graber H., Williams N., Graves S., Conover H., Galluppi K., Luettich R., Perrie W., Toulany B., Sheng Y.P., Davis J.R., Wang H. and Forrest D. (2007). Architecture of a community infrastructure for predicting and analyzing coastal inundation. *Marine Technology Society Journal, 41(1)*, pp. 53–71.
8. Mandel J., Beezley J. D., Bennethum L. S., Chakraborty S., Coen J. L., Douglas C. C., Hatcher J., Kim M. and Vodacek A. (2007). A dynamic data driven wildland fire model. *Computational Science - ICCS 2007, Lecture Notes in Computer Science*,

- Springer Berlin Heidelberg*, **Vol. 4487**, pp. 1042–1049.
9. Leong S. H., Frank A. and Kranzlmüller D. (2013). Leveraging e-infrastructures for urgent computing. *Procedia Computer Science*, **Vol. 18**, pp. 2177–2186.
 10. Tolosana-Calasanz R., Banares J., Rana O., Pham C., Xydias E., Marmaras C., Papadopoulos P. and Cipcigan L. (2014). Enforcing quality of service on opennebula-based shared clouds. *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pp. 651–659.
 11. Cope J. and Tufo H. (2008). Supporting storage resources in urgent computing environments. *IEEE International Conference on Cluster Computing*, pp. 348–353.
 12. Beckman P., Nadella S., Trebon N. and Beschastnikh I. (2007). Spruce: A system for supporting urgent high-performance computing. *Grid-Based Problem Solving Environments, IFIP*, **Vol. 239**, pp. 295–311.
 13. Balis B., Kasztelnik M., Bubak M., Bartynski T., Gubała T., Nowakowski P. and Broekhuijsen J. (2011). The UrbanFlood Common Information Space for Early Warning Systems. *Procedia Computer Science*, **Vol. 4**, pp. 96–105.
 14. Balis B., Bartynski T., Bubak M., Dyk G., Gubała T. and Kasztelnik M. (2013). A Development and Execution Environment for Early Warning Systems for Natural Disasters. *13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pp. 575–582.
 15. Knyazkov K. V., Kovalchuk S. V., Tchurov T. N., Maryin S. V. and Boukhanovsky A. V. (2012). Clavire: e-science infrastructure for data-driven computing. *Journal of Computational Science*, **3(6)**, pp. 504–510.
 16. Cope J., Trebon N., Tufo H. and Beckman P. (2009). Robust data placement in urgent computing environments. *IEEE International Symposium on Parallel Distributed Processing (IPDPS)*, pp. 1–13.
 17. Pieta A., Lupa M., Chuchro M., Piorkowski A. and Lesniak A. (2014). A model of a system for stream data storage and analysis dedicated to sensor networks of embankment monitoring. *Computer Information Systems and Industrial Management, Lecture Notes in Computer Science, Springer Berlin Heidelberg*, **Vol. 8838**, pp. 514–525.
 18. Kosinski J., Nawrocki P., Radziszowski D., Zielinski K., Zielinski S., Przybylski G. and Wnek P. (2008). SLA monitoring and management framework for telecommunication services. *30 Fourth International Conference on Networking and Services (ICNS)*, pp. 170–175.
 19. Levins R. and Lewontin R. (1985). *The Dialectical Biologist*, Harvard University Press.
 20. Varela F. J. (1978). On being autonomous: The lessons of natural history for systems theory. *Klir G. J. (Ed.), Applied General Systems Research, NATO Conference Series, Springer*, **Vol. 5**, pp. 77–84.
 21. Harrison C., Eckman B., Hamilton R., Hartswick P., Kalaganam J., Paraszczak J. and Williams P. (2010). Foundations for smarter cities. *IBM Journal of Research and Development*, **54(4)**, pp. 1–16.
 22. Software-defined networking: The new norm for networks (2012). *Tech. rep., Open Networking Foundation*, <https://www.opennetworking.org>.
 23. Brzoza-Woch R. and Nawrocki P. (2016). FPGA-based web services – infinite potential or a road to nowhere? *IEEE Internet Computing*, **20(1)**, pp. 44–51.
 24. Bachara P., Brzoza-Woch R., Długopolski J., Nawrocki P., Ruta A., Zaborowski W. and Zielinski K. (2015). Construction of hardware components for the Internet of Services. *Computing and Informatics*, **34(4)**, pp. 911–940.
 25. Szydło T., Nawrocki P., Brzoza-Woch R. and Zielinski K. (2014). Power aware MOM for telemetry-oriented applications using GPRS-enabled embedded devices – levee monitoring use case. *IEEE Federated Conference on Computer Science and Information Systems, Annals of Computer Science and Information Systems*, **Vol. 2**, pp. 1059–1064.
 26. Saaty T.L. (1990). How to make a decision: the analytic hierarchy process. *European Journal of Operational Research*, **48(1)**, pp. 9–26.