10th International Conference on Hydroinformatics
HIC 2012, Hamburg, GERMANY

LEVERAGING EXISTING INFRASTRUCTURE FOR CENTRAL AUTOMATIC CONTROL OF MULTIPLE SEWER SYSTEMS

ARNEJAN VAN LOENEN (1), KLAAS-JAN VAN HEERINGEN (1), RONALD VAN NOOYEN (2), EDWIN VAN VELZEN (3)
(1): Operational Water Management, Deltares, Rotterdamseweg 185, 2600 MH Delft, The Netherlands
(2):Delft University of Technology, Stevinweg 1, 2628 CN, Delft, The Netherlands,
(3): Waterschap Hollandse Delta, P.O. Box 4103, 2980 GC Ridderkerk, The Netherlands

In this paper we discuss a project in which water quality improvements of surface waters are gained by upgrading locally controlled sewer systems. The paper focuses on the reuse and extension of existing sewer systems and hardware and software infrastructure in an experimental integrated automatic control system for a rural region in the Netherlands. The region contains five municipalities and one regional water authority (water board). The goal of the project is to improve the water quality through increased, technology-driven, cooperation between the authorities. The largest threat for water quality of open waters in the project area is the presence of combined sewer spills (CSOs). In this project we focus on reducing the number and volume of spills and changing the distribution of the spills to protect sensitive locations. The methods best suited require an extensive sensor network and central real-time control (RTC). Such a network is an expensive proposition and the same holds true for RTC. At the other hand, by re-using parts of the existing hardware and software infrastructure a non-viable project is made viable.

For the project the controller software was adapted to reuse fixed rate pumps, existing telephone lines to pumping stations were converted to high-speed ADSL connections and municipal and water board SCADA systems were online connected to an automatic central control system. This system itself was originally developed as a flood forecasting system (Delft-FEWS) but proved to be very suitable as an automatic central control system (Control NEXT). Preliminary results indicate that the number and volume of spills already have decreased significantly.

INTRODUCTION

More than one quarter of the Netherlands lies below sea level. Large areas rely on pumps to regulate surface water levels. It is a small country with a relatively high population density (400 per km^2). Possibly due to these facts all built-up areas have a sewer system. Almost every building, even those in remote areas has a sewerage connection. The sewer systems collect waste water from households and industries, but most systems also collect precipitation from streets and roofs. As the Netherlands are a flat country, sewer systems are mainly horizontally oriented: gravity discharge only appears in subsystems. Therefore, an extensive network of sewer pumps is maintained for transporting waste water to the Waste Water Treatment Plants (WWTP). These pumps have a limited capacity;

during storm events sewer systems can fill up quickly, resulting in contaminated spills to surface waters. In most cases these pumps are locally controlled, based on sewer level setpoints. As a result they only start working when local sewer levels are high; there is no coordination between the pumping stations.

In most municipalities, sewer systems are managed by the municipality; the WWTPs and surface waters are managed by the regional water authorities (water board). Usually there is no connection between the control systems. The increasing availability of system wide, real-time data acquisition and operational information systems enables application of advanced new control methodologies. In this project we used a controller called Composer, which is described in [1] and [4]. There are several papers on aspects of the use of this controller in this project: [5][6].

The project that forms the environment within the control system was designed provided objectives and boundary conditions for its implementation. Some of the objectives were reduction of the spill volumes and energy, cost aware operation of the entire group of waste water collection systems and doing this with limited additional investments. To limit additional investments existing (software and hardware) infrastructure was reused. Additionally, we expect that reusing existing infrastructure will have the positive side effect of easing the transition from the old to the new system for the managers and operators [3].

SYSTEM DESCRIPTION

The project area lies on the former island “Hoeksche Waard” a rural area just to the south of the large port of Rotterdam. It contains thirteen villages with 130 sewer districts. Five municipalities participated in the project. All the pumps in the project area are fixed rated pumps, their capacities vary from 15 m3/hr to 300 m3/hr. All pumping stations operate under local control. They function without an active data connection with their SCADA system. Each pumping station contains a dedicated computer, which applies the control algorithms and therefore acts as the local controller. In normal operation the SCADA systems daily connect to the pumping station computers to retrieve data and to synchronize control settings. The data connection is made by modem, using a telephone line. If required, the operator can manually set up a data connection with a pumping station to retrieve the pump status or to manually operate the pumps.

The water board operates five waste water treatment plants (WWTP) and 25 sewer collection pumps, which transport the waste water from the villages to the WWTPs. The sewer collection pumps operate locally, but can be controlled directly by the operators at the WWTPs.

To prevent sewer water to flooding the streets during high precipitation, each sewer district contains an emergency spill: the combined sewer overflow (CSO). Their spilling frequency depends on the characteristics of the districts: some CSOs are known to operate frequently, while others are never working. The level of the spill weir determines the maximum storage capacity of the sewer district.

CONNECTING THE COMPONENTS

The reuse of existing hard- and software infrastructure meant that several separate components needed to be connected: (1) a coordinator, a database and a number of computational nodes which actually run Composer on the side of the water board, (2) Composer, (3) the SCADA and the individual pumping stations at the municipalities, and (4) the SCADA and sewer collection pumps at the WWTPs. We will use the term *central controller* to refer to the software running at the water board. This software consists of Delft-FEWS [7] with extensions based on a new open standard called Control NEXT.

In the chain from one particular pump of the community to the central system at the water board we have the following components. At one end of our communications line we have the pumping stations with pumps, sensors and a computer that manages the pumps and collects sensor data. We will refer to the software running on the computer in the pumping station as the *local controller*. The computer at the pumping station is connected to a computer at the computing centre of the municipality by a telephone line. The computer at the municipality runs a SCADA system and a Control NEXT compatible data transfer service, a piece of software that acts as a bridge between the municipal SCADA system and the Control NEXT software running at the water board.

In the chain from one particular waste water collection pump to the central system at the water board we use almost the same components. The flow of messages is shown in Figure 1.

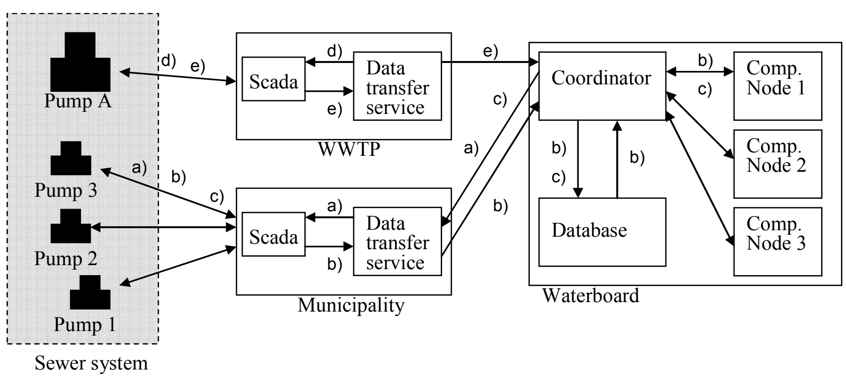

Figure 1: Data flow between the different components of the system, a) The coordinator sends a request for data that ends at the pumps managed by the community b) The collected data at the pumps is returned to the controller and from there sent to the computational node. Data is also stored in the database c) The central controller calculates the pump commands, which are sent back to the pumps. For analysis purposes the commands are also stored in the database. d) Parallel, a component frequently requests data from the sewer collection pumps, e) which are also returned to the coordinator and used as input for Composer.

The municipal SCADA system is in charge of communications with the local controllers, alerts for pump failures and visualization for municipal workers. It also writes current information to a file. This file is the boundary between the domain of responsibility of the municipal SCADA system and Control NEXT.

The Control NEXT data transfer service at the WWTP connects independent at set interval via OPC [9] to the SCADA system and requests predefined data. The data are sent automatically to Control NEXT.

Control NEXT is the operational extension of the present Water Information System and is responsible for the communication with the Control NEXT data transfer services. visualization for operators at the water board, data storage and validation and for the generation of control commands.

NEW AND UPGRADED SOFTWARE

Existing software was reused wherever and whenever possible. This included (parts of) the software of the pumping station computers, the SCADA systems, the coordinator and the central controller.

The pumping station software in place at the start of the project itself needed to be adapted to allow for the implementation of real time control (RTC): an update was required to be able to receive and implement pump settings sent to it. At the same time safety mechanisms were implemented, adding to the existing intelligence build in for optimal pump operation. This makes the pumping station computer a local supervisor.

The RTC controller chosen at the start of this project (Composer) had an optimization routine that was not optimal for controlling fixed-rate pumps. Therefore a different optimization routine was inserted.

As an integration platform Delft-FEWS [7] was applied. It provides the coordination and data exchange needed for RTC. The existing platform was extended with functionality to make real-time connections with the SCADA systems by applying web services. Also additions to allow for calculations within the RTC loop and an optimized synchronization for real-time tasks were implemented.

NEW AND UPGRADED INFRASTRUCTURE

Data connections between the SCADA systems and the pumping stations used analog modems over telephone lines: connections needed to be re-established for every communication attempt. For real-time control purposes, these connections were too slow. Installation of ADSL modems provided always-on connections based on the IP/TCP protocol over the same phone lines.

The municipal SCADA system was originally designed to communicate with the pumping stations a limited number of times per day. For an RTC implementation much more frequent contact is required. Therefore also the SCADA systems needed to be upgraded to be able to handle real-time connections with the pumping stations and with the central controller.

The water board already owned a Water Information System based on Delft-FEWS, which was used to collect telemetry data. This system was extended with the Control NEXT platform, adding operational functionality.

The controller for the sewer systems, Composer, was based on an existing design that had already been built for Rotterdam ([1],[4]). This controller tries to avoid CSO events by making optimal use of the storage in the system as a whole. This is implemented by central automatic control of the pumps. The limited pump capacities and the presence of fixed-speed pumps make it unlikely that the system can be brought to the desired state in one step, so a quadratic penalty function is used. The original implementation used in Rotterdam was not compatible with Delft-FEWS and it assumed variable speed pumps so a new implementation, especially suited for fixed speed pumps, was written.

MOTIVATION FOR USE OF INPUT AND OUTPUT FILES

The dual use of Control NEXT as a database/decision support system and as overall manager for the communication between several different controllers and their controlled systems has clear advantages; it gathers all responsibility for data storage and validation in one spot and allows one standard interface between controller and intermediary.

However, it also introduces a single point of failure. All code executed directly within Control NEXT must be extremely reliable. In a project such as this, where the controllers are being developed within the project this creates a problem. Control NEXT is written in Java, so there are three basic options of communication between Control NEXT and the controller.

1. Write the controller in a language supported by the Java Virtual Machine (JVM) and have a jar file with a factory object to be called by Control NEXT code to deliver a controller with a specified interface. This means the controller runs as part of Control NEXT.
2. Let the controller run as a separate process and communicate with Control NEXT through sockets or a web interface.
3. Make the controller an external program and let Control NEXT start the controller at each time step, communicate through files with a time-out.

In the third case separation is optimal. As Control NEXT already contained a module that implemented option 3 ([8]) for the control of models used in the decision support part, it was decided to use the same module to access the controllers. This module also allows storing time stamped controller configuration files ([8]). One positive side-effect of using files for data transfer turned out to be the possibility to check the actual data transfer between the components.

CONTROL CYCLE

The implemented communication scheme is the following. The SCADA system at the community computing centre retrieves in a continuous loop data from all its pumping stations. In practice this means that for every pumping station approximately every ten seconds new data are available. It does this independently from any actions taken by the central controller. Data are written in one file per pumping station: previously collected data is overwritten.

At the central controller the coordinator runs with an interval of 60 seconds the following sequence of operations. It sends out a query to the data transfer services at the municipalities and waits for the responses. When this query arrives within the time-out period at the data transfer service, the service collects the latest data of the pumping stations, provided by the SCADA system. The data are transmitted back to the coordinator.

Independently, the data transfer service at the WWTP requests every minute data from the SCADA system and sends this data to the coordinator.

The coordinator stores the result of the query and of the data transfer service at the WWTP in the database and simultaneously sends the data to the computational node. The computational node writes the data to a file and starts Composer as an external program. Next it waits until either the output files of the controller appear or a time out occurs. It then reports back to the coordinator. The results of the call are stored in the database and simultaneously send as a set of pump commands to the data transfer service at the municipality. The data transfer service writes the command to a file that is directly read by a process running on the SCADA system.

TIMING BOTTLENECKS

The central controller operates in cycles of one minute. These cycles are triggered by the coordinator and for each municipality there is an individual cycle. Initially the delay between t_0 , the start of a cycle, to t_1 , the time at which the response of a pump to the command becomes available at the central controller, exceeded the data collection time step of 60 seconds. An analysis of the system revealed several bottlenecks.

One bottleneck was found in the SCADA system software of the municipality. It ran into problems because the software was designed for much lower query rates. Improvements in this software resulted in much better response times (in some cases the original software took more than 15 seconds to retrieve the data).

Another bottleneck was located within the prototype Control NEXT implementation. This still used the standard mechanisms from Delft-FEWS to let the coordinator access a computational node. These were intended for use for model runs or decision support purposes where a series of 1 to 5 second delays between the steps in a procedure is not critical. Creation of a special purpose connection between the coordinator and the computational node resulted in a reduction of the time taken from the initial call of the computational node to the arrival of the result at the coordinator by 20 seconds.

Other improvements included fine-tuning of settings of components, so that a minimum of data is transferred between the components. After implementing the improvements, the time from a water level measurement in a pump station up to a control command based on this measurement, was decreased to about 30 seconds.

SAFETY MECHANISMS

For reasons of safety and efficiency it was decided to distribute responsibilities over the different components. The local controller would be in charge of safety and the well - being of the pumps. It already contained well-developed safety mechanisms that could best be reused. It would protect against dangerously high and low water levels by overriding the commands received from the central controller in case of threshold crossings.

It would also translate a request to run one or more pumps into a command to switch specific pumps on or off. In the case of a command to switch on one pump in a station with multiple pumps, the local controller will for instance select the pump with the least number of on/off cycles since midnight. This results in a distributed (alternating) load of available pumps, with that making the best use of the intelligence in the pumping station computer.

One final safety mechanism concerns the connection with the *central controller*. If the connection with the controller is lost for a certain amount of time, the pumping station will switch back to local control. When the connection is regained, central control will directly take over.

IMPLEMENTATION IN AN ACTIVE SYSTEM

The real-time control needed to be added to the systems while they were on-line. This meant ongoing processes were not to be disturbed, adding an extra dimension to the implementation of the RTC [9]. Since existing hard- and software would be reused, developments on the components could be carried out parallel. Testing components and data connections could take place in small separate steps, making the implementation of this complex system a clear and plain step by step process. Setting up the data connections, extending Composer and examining the physical infrastructure all took place in parallel project routes. At the time of implementing the pilots, the components and data connections only had to be put together.

For the sewer operators, hardly any changes were visible, lowering the threshold to hand over control to Control NEXT. The assurance that several safety mechanisms were in place and had been tested and that the central controller could be overruled at any time, minimized operator concerns [3].

DISCUSSION: ADVANTAGES AND DISADVANTAGES OF REUSING EXISTING INFRASTRUCTURE

Presented was a project in which existing infrastructure was reused in order to set up a sewer control system. The advantages of this approach are clear: replacing the (digital) infrastructure would be very costly and time consuming. Testing of all the new components would take a lot of time. Reuse of existing infrastructure, also means that sewer operators are already familiar with large parts of the new systems. Another advantage is that the original situation with local control remains available as a fall-back position. Finally, reusing existing hard- and software made it possible to implement RTC without taking the system off-line.

Of course, reusing existing infrastructure also has disadvantages. All relevant components of the original system were reused and additional components were added. If one of the components fails, the whole control system fails. This could make the system more vulnerable. Reusing components also imposes the risk that we try to adapt old technology, when setting up a new component with new technology might be wiser.

CONCLUSIONS AND OUTLOOK

We presented a project in which existing infrastructure was reused in order to set up a real-time control system for sewer systems. By reusing hard- and software and adding functionality to these components, a functioning system was developed that efficiently operates the sewer systems in two villages. The system was successfully implemented, both technically as organizationally. This proves that the concept of reusing infrastructure offers opportunities. The first results after half a year of operation show that the decrease in CSO was significantly larger than expected on beforehand.

ACKNOWLEDGMENT

We were supported by grants from Agentschap NL, The Netherlands. This agency provided a subsidy from the "Innovatieprogramma Kaderrichtlijn Water" of the Dutch Ministry of Transport, Public Works and Water Management. This work is part of the project "Integrale Sturing Afvalwaterketen en watersysteem Hoeksche Waard".

REFERENCES

[1] Breur, K.-J., van Leeuwen, P. E. R. M. and Dellaert, N. P., "Deriving discharge strategies to reduce CSO in urban drainage systems", *Water Science and Technology* Vol. 36, No.8-9, (1997), pp 259-263.

[2] Dicken, C.R., "From hard desk to soft desk -the changing operator interface", *PEOPLE IN CONTROL. International Conference on Human Interfaces in Control Room, Cockpits and Command Centres,* Bath. IEE Conf. Publ. No.463, (1999), pp 96-100.

[3] Greer,N.J., "New for old – effectively managing a control system changeover". *PEOPLE IN CONTROL. International Conference on Human Interfaces in Control Room, Cockpits and Command Centres,* Manchester,UK. IEE Conf. Publ. No. 481, (2001), pp 36-41.

[4] Leeuwen, P. .E. R. M. van, *Sturing van Watersystemen*, Delft University Press, Delft, The Netherlands (2003, In Dutch).

[5] Nooijen, R. van; Kolechkina, A. G., van Leeuwen, P. and Velzen, E. van, "Agile Design of Sewer System Control", *Proceedings of the 18th IFAC World Congress*, Milan, Italy, (2011) pp 14776-14182

[6] Nooijen, R. van; Kolechkina, A., Heeringen, K.-J. van and Velzen, E. van , "Implementation of central control for multiple sewer systems", *Proceedings of the twelfth International Conference on Urban Drainage*, Porto Alegre, Brazil, (2011)., PAP005323 pp 1-8.

[7] Reggiani, P.; Kwadijk, J.C.J.; Werner, M.G.F.; Dijk, M.J. van; Schellekens, J.; Kappel, R.R. van; Sprokkereef, E., "Modelling of hydrological responses; Delft-FEWS: an open shell flood forecasting platform". *EGS Congress, Nice* (2003)

[8] Werner, M., "Open model integration in flood forecasting" *Practical Hydroinformatics*, Water Science and Technology Library, Vol 68, (2008), pp. 495–505.

[9] Willdridge, J.R., “Control room reinstrumentation:‘on line’”, *PEOPLE IN CONTROL. International Conference on Human Interfaces in Control Room, Cockpits and Command Centres,* Bath. IEE Conf. Publ. No.463, (1999), pp 137-141.

[10] Hong, X.; Jianhua, W., “Using standard components in automation industry: A study on OPC Specification”, *Computer Standards & Interfaces*, Vol 28, (2006), pp.386–395.