Deformation monitoring in the internet of things. Implementation of a multi-platform software package for modern sensor networks in engineering geodesy

Philipp ENGEL

Hochschule Neubrandenburg – University of Applied Sciences, Faculty of Landscape Sciences and Geomatics, Neubrandenburg, Germany
E-mail: pengel@hs-nb.de

Abstract

The monitoring of structures and buildings for the detection of movements and deformations is an important field of activity in modern engineering geodesy, ensuring the standing and reliability of critical infrastructure through continuous observations. The need for advanced and cost-effective monitoring technology is increasing, especially for transport infrastructure, coastal protection, mining, and renewable energies. Major progress has been made in the past years in the areas of sensors, embedded computing, network communications, and geo data processing.

The emerging Internet of Things provides new perspectives for the automation of surveying. These advancements can contribute to a more comprehensive deformation monitoring and help to lower the acquisition and operating costs of permanent observations. Research on new approaches in engineering geodesy conducted at the Neubrandenburg University of Applied Sciences has the objective to develop an open-source multi-platform automatic deformation monitoring system. The on-going work involves a detailed inspection of data protocols for the inter-communication of power-constrained devices in low-latency networks, the usage of low-cost single-board computers for sensor control, and the persistence of acquired sensor values in document-oriented databases.

The monitoring software of the research project is written in the Python programming languages and integrates several open-source libraries and frameworks. Some components of the system are still in development, but small scale sensor networks can already be established. First practical tests include the observation of a 19th century gothic brick church and two highway bridges in Germany, using geodetic and geotechnical sensors. It has been shown that the application meets the requirements placed on a modern automatic deformation monitoring system.

Key words: engineering geodesy, deformation monitoring, sensor networks, software development, open source, embedded computing

1 INTRODUCTION

The permanent and automated deformation monitoring of buildings, infrastructure, and terrain is gaining in importance under national and international aspects of hazard prevention, operational safety, and damage documentation. Typical objects for permanent monitoring are structures underlying short or long term external influences, like buildings, slopes, dams, or even earth plates. Usually, the movement of these objects must be recorded and analysed over a certain period of time to distinguish between typical and untypical behaviour, to enable an early detection of possible weakening and to avoid major damage.

In order to understand the movement of an object and to provide preventive alarm messages, a comprehensive mathematical and physical modelling is required. This makes in necessary not only to detect geometric changes, but also to consider acting forces and influencing factors – which are the cause of the geometric changes. The primary task of a monitoring system must be the reduction of damage and thus the avoidance of risks to man and the environment.

The monitoring of deformations is common in modern engineering geodesy. Several proprietary hardware and software solutions are available. These include the commercial products of vendors of geodetic sensors, which offer systems tailored to their products, along with software developed by universities and engineering firms (Stempfhuber. 2009). At Neubrandenburg University of Applied Sciences open-source software for sensor networks in geodesy is in active research and development (see Fig. 1). Its main purpose is to provide a royalty-free modular platform for the realization of automated measurements in deformation monitoring.

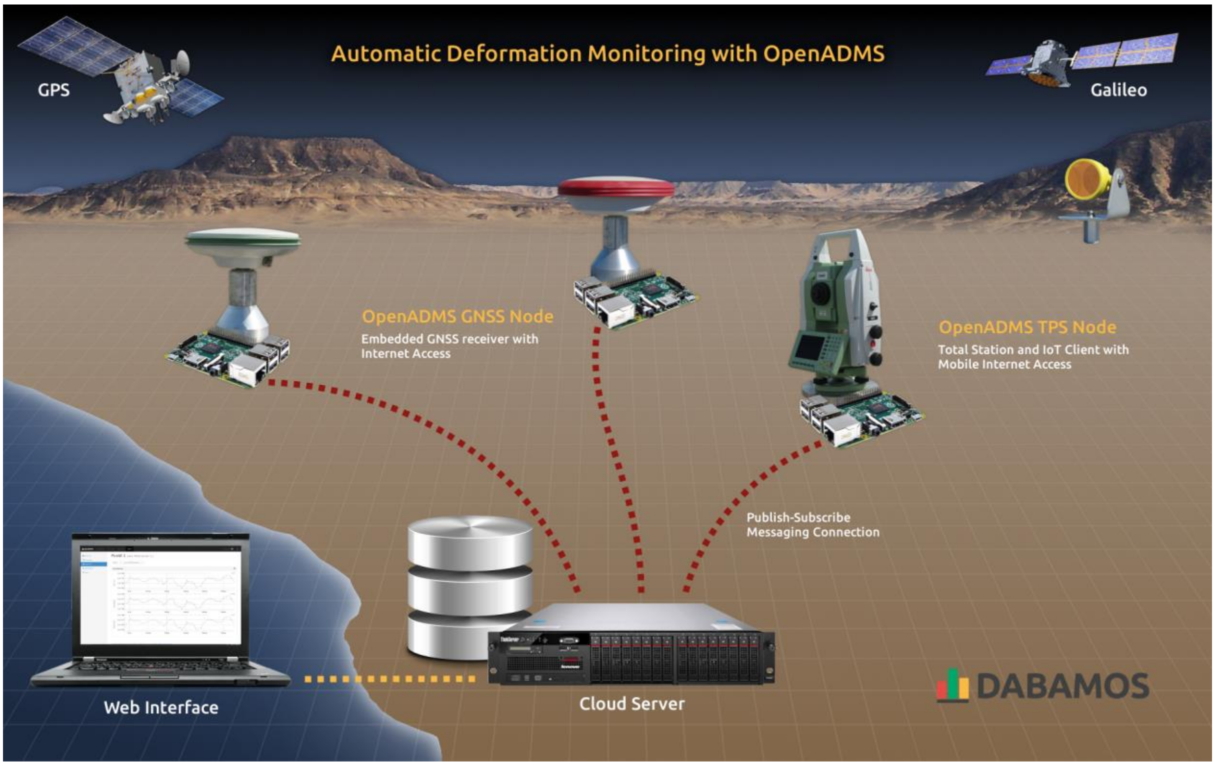

Fig. 1 Schematic overview of the described monitoring system architecture

2 PURPOSE

The automation of geodetic observations started already the 1980s, when the emerging personal computers and analogue signal modems where used by engineers to control electronic sensors and record sensor data on magnetic storages (Pelzer. 1988). At this time, temporary measurements were increasingly replaced by permanent and time-continuous observations.

The prerequisite for automatic measuring digital systems in geodesy was the availability of suitable software for sensor control and data processing. These deformation monitoring systems were designed as isolated applications for single personal or industrial computers, with no or limited network functionality. Even today, the more modern systems lack features like open network and data interfaces for machine-to-machine communication to third party software, making it difficult to integrate them in sensor networks (Sohraby. 2007).

Especially in academia, systems with open access and available source code are desired to modify them according to the needs of research. In order to coordinate large-area measurements, monitoring systems must be integrated into sensor networks. The capability of the Internet of Things to interconnect devices has simplified the technical requirements for a comprehensive networking in engineering geodesy – establishing sensor networks has become easier and less expensive. The research project therefore aims to discover new ways in deformation monitoring by using advancements in sensors, software programming, hardware development, and network engineering.

3 METHODS

A system for the permanent monitoring of arbitrary objects, for instance, buildings or sections of terrain, must satisfy certain requirements. The actual behaviour of an object has to be observed to detect regular, i.e. daily, monthly, or annual, periods. Occurring repetitions in these periods depend on the Nyquist frequency. In addition, it is necessary to register the mean influencing factors, which cause the movement of the object, like meteorological values or external forces.

Flexible concepts in terms of hardware and software architecture can be beneficial to allow changes on the configuration or even the expansion of the system by additional parts on demand, by minimal costs. A further requirement is an alarming function that must be triggered on special events, such as the exceeding of thresholds. It is also important to note that the required accuracies of the whole system depend on the expected movements of the observed object. Therefore, the system must be economically legitimate, but also deal with the infrastructural or cultural significance of the monitored object.

3.1 SOFTWARE DEVELOPMENT

The development of a modern application from scratch is based on the selection of an adequate programming language. For the rapid implementation of geodetic sensor networks in embedded environments it is preferable to use a platform-independent language that already contains intrinsic or third-party libraries for sensor communication (Breuer. 2012), data processing, network access, and so on.

For instance, software written in the Python programming language can be run on many platforms, as its instructions are executed by an interpreter without the need of compiling the source code into machine-language before, like it is necessary with C++ or Fortran.

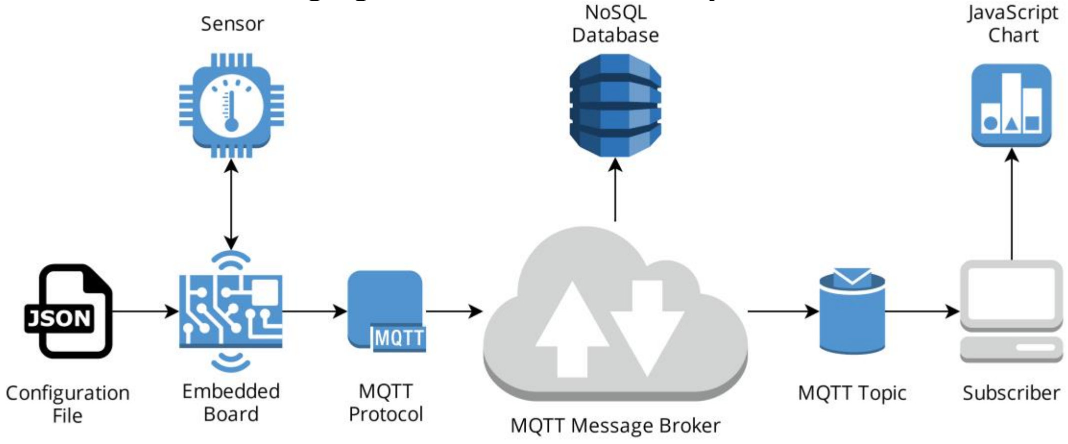

Fig. 2 Distributing observation data in JSON format with the MQTT protocol

Therefore, Python software is compatible to all operating systems for which an interpreter is available, like Microsoft Windows, GNU/Linux, Unix, or Android. The environment of the language also provides a broad range of pre-existing modules for many use cases. For these reasons, Python 3 was chosen for the development of the monitoring system.

3.2 SENSOR CONTROL HARDWARE

The advent of small single-board computers, like the Raspberry Pi, has made it possible to create low-cost middleware for observation and control tasks in short time, using rapid prototyping techniques. These embedded boards are full-featured computers, with ARM or MIPS processor, memory, and I/O interfaces (Peci et. al. 2014). On top of them, any application, like the discussed monitoring software, can be executed. The concept of embedded sensor nodes is also applicable on portable Android devices, like smart phones and tablet computers. They are able to provide the necessary interfaces for sensor link connections as well as mobile Internet access over 3G/4G. Re-using such generic devices can lower the effort and the costs of the implementation of sensor networks.

3.3 OBSERVATION DATA EXCHANGE

Appropriate network protocols and semantic descriptions of data have to be chosen in order to enable smart devices to communicate within the Internet of Things. Several communication standards are available and in active usage, some of them specifically for the requirements of sensor networks, but none of them can be seen as universal in a way that covers all use cases.

The Sensor Web Enablement is an initiative of the Open Geospatial Consortium (OGC) to define services for the integration of sensor networks into the World Wide Web. One of its parts is the *Sensor Model Language* (SensorML), which describes sensor nodes and observations processes in a syntactical and semantic way, to enable inter-machine communication. This standard is extended by further ones, like the *Sensor Observation Service* (SOS) for Web-services querying observation data, or *Observations and Measurements* (O&M), an XML implementation for sensor data encapsulation. A major drawback of them is their large scope that makes them complex and hard to implement in all their aspects and prevents their actual dissemination. Furthermore, the broad coverage leads to the problem that those standards are often outdated at the time their specifications are published, due to the fast advancement of Internet technologies. Thus, the actual implementation of the SWE specifications has been adopted by a few services only (OGC. 2017).

Fig. 3 OpenADMS Web application for remote control of sensor networks

The JavaScript Object Notation (JSON) is a simple data-interchange format, similar to XML, but easier to read and write by humans and machines. For many programming languages specific libraries are available that can be used to transform internal data structures to JSON and vice-versa. The format makes it possible to process sensor data across languages, platforms, and applications. For instance, sensor data obtained by a monitoring system can be exported to JSON easily and then transferred over various network protocols to be used by NoSQL databases, Web services, or browser-based charts.

Many quasi-standards for message exchange have evolved in the Internet of Things, for instance, raw socket connections, message-oriented middleware, Cloud-based message queuing services, WebSockets, or Representational State Transfer. Not all of them can be mapped onto sensor networks. With the lightweight MQ Telemetry Transport (MQTT) protocol small devices can communicate with each other or with external systems by using a publish/subscribe messaging model. It is ideal for constrained networks with low bandwidth, high latency, data limits, and fragile connections (Lampkin et. al. 2012). Messages are despatched by a broker, a service that distributes the messages in a network in real-time. The sending devices do not need to know the receiver of a message nor its address. This concept makes MQTT suitable for distributed sensor networks.

4 RESULTS

The software platform consists of three independent applications. The *Open Automatic Deformation Monitoring System* (OpenADMS) is used for sensor control and observation data processing. It is designed as a system service rather than a desktop application. The operating system will start OpenADMS at boot time and run it in the background. For this reason, the monitoring system does not contain an integrated graphical user interface, instead, one is provided by secondary software of the project.

4.1 MONITORING SOFTWARE

Modern monitoring software has to complete several tasks. It can be seen as beneficial if the programme logic is split into separated modules, which solve these tasks independently. Despite a few core modules for configuration and management, all other parts can be loaded dynamically at run-time and combined according to the specific requirements of the user. The implemented modules are used for the scheduling of observations, sensor communication, the extraction of values out of raw sensor responses, the inspection of error codes returned by sensors, processing of measurements of total station positioning systems, alarming by email or short message, Web-based control, or real-time on-line publishing to third-party applications. Further ones can be written by the user on top of a prototype module.

Observation data and other messages can be exchanged freely between modules, using the publish/subscribe-based MQTT lightweight messaging protocol. An external message broker, for instance, Eclipse Mosquitto or RabbitMQ, despatches the messages and provides guaranteed delivery (Videla. 2012). All data is encapsulated in JSON before it is forwarded to the broker. By using the JSON standard, observation data does not need to be converted before storing it in document-oriented NoSQL databases or displaying it as dynamic JavaScript charts in web browsers.

A critical part of developing a monitoring application is the testing of all components, to find software bugs and to prove the overall stability. In computer programming, so-called unit testing is used to find problems at early stage by determining input and output of each function, verifying the correct processing of data and preventing errors caused by later additions or changes of code. Since it is often not possible to check all parts of the monitoring system with the whole range of sensors, virtual sensors are supported to simulate physical measurement instruments through software-defined methods.

4.2 WEB AND CONTROL APPLICATIONS

A simple user interface, OpenADMS Control, runs on desktop computers and is used to start a local monitoring, for testing or short term observations, whereas OpenADMS Web is projected to run on network or Internet servers to orchestrate sensor networks consisting of OpenADMS instances. The web application is written in Python, JavaScript, and HTML5 (see Fig. 3). The interaction and data exchange between the programmes is done on top of the MQTT messaging protocol. So far, only a few basic functions have been implemented yet, like authorisation, starting and stopping of measurements, or the visualisation of observation data in JavaScript charts.

4.3 EMBEDDED CONTROL SYSTEMS

Since 2012, several sensor control units based on versions of the Raspberry Pi single-board computer have been built and tested as part of permanent geodetic observations. They are compatible to various operating systems, like Linux distributions or Unix derivates. Internet access is provided either by WiFi, Ethernet, or 3G/4G. Enhanced with an uninterruptible power supply unit, they were able to run for years without intermittent reboots. For testing, the OpenVPN virtual private network is installed on a central FreeBSD server to establish a secure connection to sensor control units in remote areas.

5 CONCLUSIONS

OpenADMS is already in active use for the permanent observation of three buildings. In Neubrandenburg, a 19th century gothic brick church is monitored by GNSS receivers and inclinometers to determine the influence of the bell movements on the church tower. Remote maintenance is achieved by a direct 5 GHz wireless link between the church and the university. In Lower-Saxony, two monitoring systems based on OpenADMS have been installed on motorway bridges in order to observe them permanently using robotic total stations.

Both the hardware and the software of the project are still in active development. Many of the planned features, like the support for cloud-computing platforms (Emeakaroha et. al. 2015) or GNSS receivers, are not yet implemented. However, it has been shown – especially during the first field tests – that the software can meet the requirements placed on a modern deformation monitoring system. It is therefore important to continue the development in the coming years. Future announcements regarding the progress of the project will be made on the official web page (https://dabamos.de/).

Funding Acknowledgement: This work was supported by the German Federal Ministry of Education and Research.

REFERENCES

Breuer, C. 2012. Sensorkommunikation bei automatisierten Monitoringsystemen. Unterschiedliche Methoden zur Steuerung von Messsensoren im Bereich Überwachungsmessungen. In *AVN – Allgemeine Vermessungsnachrichten*, vol. 4.

Emeakaroha, V. C. ; Fatema, K. ; Healy, Ph. ; Morrison, J. P. 2015. Contemporary Analysis and Architecture for a Generic Cloud-based Sensor Data Management Platform. In *Sensors & Transducers*, vol. 185.

Lampkin, V. ; Leong, W. T. ; Olivera, L. ; Rawat, S. ; Subrahmanyam, N. ; Xiang, R. ; Kallas, G. ; Krishna, N. ; Fassmann, S. ; Keen, M. ; Locke, D. 2012. Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. IBM Redbooks.

OGC. 2017. Open Geospatial Consortium. Implementations by Specification. Retrieved from http://www.opengeospatial.org/resource/products/byspec/

Peci, L. M. ; Berrocoso, M. ; Fernández-Ros, A. ; García, A. ; Marrero, J. M. ; Ortiz, R. 2014. Embedded ARM System for Volcano Monitoring in Remote Areas: Application to the Active Volcano on Deception Island (Antarctica). In *Sensors*. 2014;14(1):672-690.

Pelzer, H. 1988. Ingenieurvermessung. Deformationsmessungen. Massenberechnung. Ergebnisse des Arbeitskreises 6 des Deutschen Vereins für Vermessungswesen (DVW) e. V. Herausgegeben von Hans Pelzer. Wittwer, Stuttgart.

Sohraby, K. ; Minoli, D. ; Znati, T. 2007. Wireless Sensor Networks. Technology, Protocols, and Applications. Hoboken, NJ : John Wiley & Sons.

Stempfhuber, W. (2009). Geodätische Monitoringsysteme – Stand der Technik und Abgrenzung der gegenwärtigen Systeme. In *Zeitabhängige Messgrößen – Verborgene Schätze in unseren Daten. Beiträge zum 85. DVW-Seminar am 7. und 8. September 2009 in Kassel*, vol. 59. Augsburg : Wißner.

Videla, A. ; Williams, J. J. W. 2012. RabbitMQ in Action: Distributed Messaging for Everyone. Manning Publications Co.