

Implementation of a multiagent-based paradigm for decentralized real-time structural health monitoring

K. Smarsly¹, K. H. Law¹ and D. Hartmann²

¹Stanford University, Department of Civil and Environmental Engineering, 473 Via Ortega, Stanford, CA 94305-4020, USA; email: {smarsly, law}@stanford.edu

²Ruhr-University Bochum, Department of Civil and Environmental Engineering, Universitaetsstr. 150, 44780 Bochum, GERMANY; email: hartus@inf.bi.rub.de

ABSTRACT

Engineering structures such as wind turbines require continuous monitoring to ensure structural safety, to reduce the overall maintenance and repair costs and, ultimately, to achieve extended lifetimes and a greater economic viability. For that purpose, an automated SHM system for wind turbines has been developed and installed on a 500 kW wind turbine in Germany. During its operation, temporary malfunctions of the installed sensing units have been observed. These malfunctions, such as temporary sensor breakdowns which are well known from real-time SHM systems, might cause the loss of valuable monitoring data if not detected timely. A multi-agent system, which is capable of self-detecting system malfunctions and notifying the human individuals automatically, has been developed and integrated into the existing SHM system. The SHM system and its subsystem, the multi-agent system, have been in continuous operation since 2009. Since then, various malfunctions have automatically been detected and appropriate actions have been taken in a timely manner. As a result, the malfunctioning of the sensors did not lead to significant data loss, thus enhancing the quality of the SHM system.

INTRODUCTION

Degradation of engineering structures, increasing demands on structural safety and growing maintenance costs are emerging problems that are gaining attention in civil and environmental engineering. Engineering structures require continuous monitoring and frequent inspections in order to reliably assess their performance and safety, to provide timely maintenance support and to increase service life. This paper discusses the development and implementation of a self-managed monitoring system for a wind turbine structure. Wind turbines represent one of the most cost-effective sources of electrical power providing an environmentally friendly energy production. Accordingly, wind energy technology is fostered and large wind energy farms are being built in many countries. The world market for wind turbines has considerably

grown, for example, with approximately 16 GW of new capacity added worldwide in the first half of year 2010 (WWEA 2010).

The necessity of applying innovative structural monitoring systems has been recognized over the past several years. Wind turbine monitoring is considered worthy of substantial investments and supports by governments (BMJ 2009, MIWF 2009). Considerable efforts are currently being undertaken to design SHM systems for wind turbines. Novel sensing technologies and sophisticated monitoring techniques are being developed and applied to wind turbines (Jüngert 2008, Lading *et al.* 2002, Sutherland *et al.* 1994, Rolfes *et al.* 2007). However, robust software platforms designed to facilitate data interrogation, data archiving and automated execution of monitoring tasks have received little attention.

In a previous research project (Hartmann and Höffer 2010), a decentralized real-time SHM system for remote monitoring of wind turbines has been designed, implemented and installed on a 500 kW wind turbine in Germany. The decentralized SHM system integrates different information technologies and provides automated data acquisition, storage and interrogation functionalities. In addition, the SHM system provides remote access to the monitoring data including flexible data visualization and data export mechanisms. During the operation of the SHM system, it has been observed that temporary malfunctions occurred affecting the performance and the reliability of the SHM system. The causes for such malfunctions, well-known from real-time SHM systems, are manifold, e.g. communication problems when using long-distance lines, temporary power blackouts that affect the computer systems or simply hardware problems of the data acquisition units (DAUs) due to extreme weather conditions.

Evidently, if not detected timely the malfunctions might cause the loss of valuable monitoring data. A multi-agent system, which is designed to detect malfunctions automatically and to immediately send out alerts for corrective actions, has been developed and integrated into the existing SHM system. Multi-agent software technology allows creating modular and extensible decentralized systems based on cooperating software agents, which are self-contained computational entities capable of autonomously carrying out monitoring tasks, such as detecting DAU malfunctions, generating alert messages and sending notifications. Each software agent is designed for solving one single task. With a certain degree of flexibility and autonomy, a software agent can decide on its own which actions are appropriate to solve its task and which other cooperative agents are to be requested for assistance to achieve a specific goal. A multi-agent system is scalable and can be easily extended or modified by adding further specialized software agents.

This paper presents a multi-agent system designed for structural health monitoring applications. First, a decentralized real-time SHM system installed on the wind turbine in Germany is described. The design and implementation of the multi-agent system and its integration with the existing SHM system are discussed. Examples are presented to demonstrate the practicability and the reliability of the multi-agent system, which has been in continuous operation since 2009.

A DECENTRALIZED REAL-TIME SHM SYSTEM FOR WIND TURBINES

A decentralized real-time SHM system is designed and prototypically implemented on a wind turbine located in Germany to continuously assess its performance and operational conditions. As shown in Fig. 1, the SHM system consists of a hardware monitoring subsystem M_1 and a software monitoring subsystem M_2 . Installed in the observed wind turbine, the hardware subsystem M_1 includes an array of sensors connected to data acquisition units, which are controlled by a computer located on-site in the structure. Remotely connected to the hardware subsystem, the software subsystem M_2 consists of software modules that are designed to autonomously execute crucial monitoring tasks, such as storing and converting the collected data sets. Both subsystems are briefly described in the following.

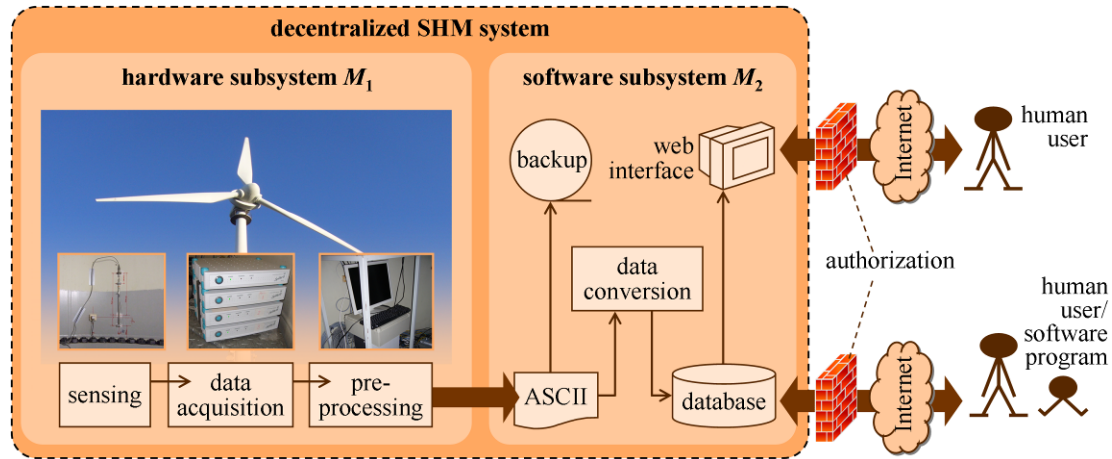


Figure 1. Architecture of the SHM system.

Hardware subsystem

The hardware subsystem M_1 collects structural and environmental data for assessing the conditions of the wind turbine. To this end, the wind turbine is instrumented with sensors that are installed both inside and outside the steel shaft as well as on the foundation of the structure. The sensors are controlled by data acquisition units which are connected to a computer located in the wind turbine.

As shown in Fig. 2, six inductive displacement transducers, type HBM-W5K, are mounted at two different levels inside the shaft. The displacement transducers are complemented by Pt100 resistance temperature detectors to account for temperature influences on the displacement measurements. Additional temperature sensors are placed at two other levels in and outside the wind turbine to measure the temperature gradient of the wind turbine shaft. For the temperature data acquisition, three 4-channel Picotech RTD input modules PT-104 are installed and, through RS232 to USB interface converters, connected to the computer located in the wind turbine.

Six three-dimensional PCB-3713D1FD3G piezoelectric accelerometers, manufactured by PCB Piezotronics, are placed at five levels in the wind turbine shaft. At the foundation of the wind turbine, three single-axis PCB-393B12 piezoelectric

seismic ICP accelerometers are installed. For acquiring wind information, a Metek USA-1 ultrasonic anemometer is mounted on a telescopic mast near the wind turbine. Fig. 2 shows the instrumentation of the wind turbine, exemplarily illustrating the interior of the shaft at the level of 21.3 m as well as the anemometer installed.

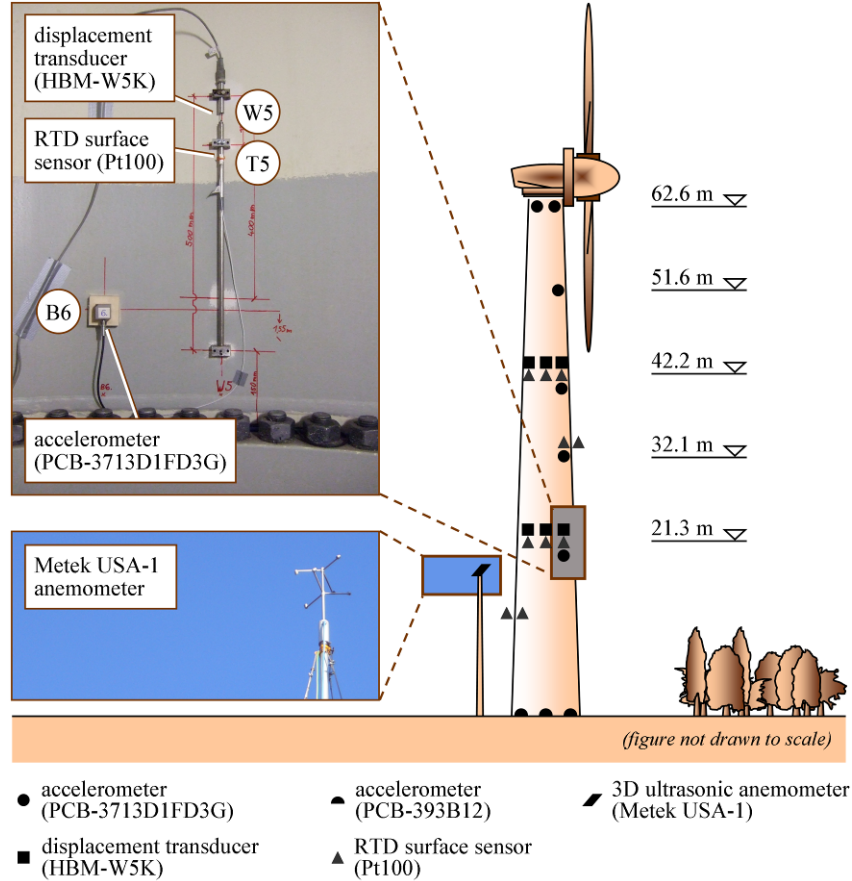


Figure 2. Instrumentation of the wind turbine (hardware subsystem).

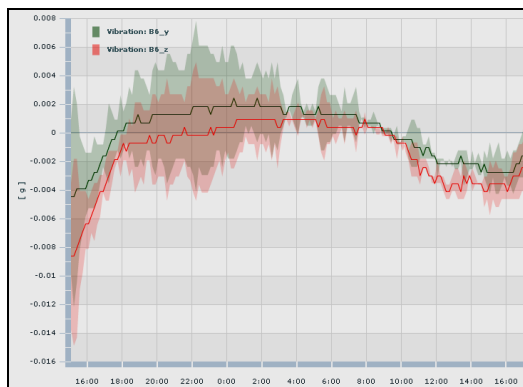
Software subsystem

The data collected and pre-processed by the hardware subsystem M_1 is forwarded to the software subsystem M_2 using a permanently installed DSL connection. The purpose of the software subsystem is to perform data archival and data processing tasks such as condensing, converting and storing of the acquired data sets. Furthermore, the subsystem is design to allow authorized users to remotely access the SHM system independent of their geographical location. The subsystem M_2 comprises (a) server systems for on-line data synchronization, data conversion and data transmission, (b) RAID-based storage systems for periodic backups, and (c) a MySQL database system for persistent data storage. A web-based interface is available for remotely accessing and visualizing the monitoring data.

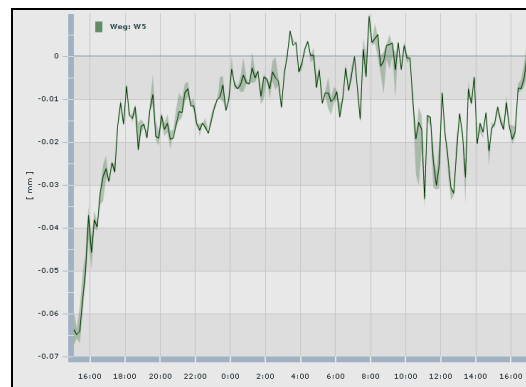
Remote access to the data is provided in two ways: through a web interface and through a direct database connection. The web interface offers GUIs for remotely

visualizing, exporting and analyzing the monitoring data. Through the database connection, the monitoring data can be directly accessed by software applications such as MATLAB and other tools.

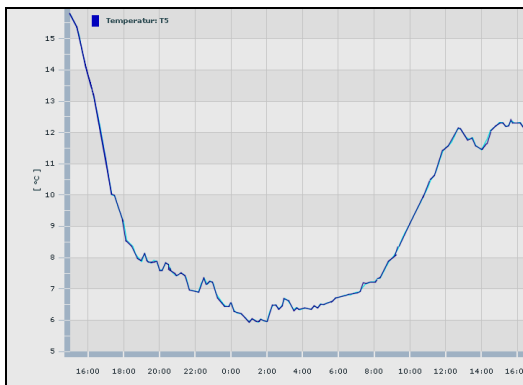
To illustrate the remote access to the SHM system using the web interface, Fig. 3 shows a user's GUI for visualizing wind turbine data, collected on October 22, 2010, in near real-time. In Fig. 3a, a typical acceleration time history is displayed, collected by accelerometer B6 shown in Fig. 2 at the level of 21.3 m. Fig. 3b, 3c and 3d show, respectively, the displacement time histories (collected by sensor W5), temperature data (acquired by displacement transducer T5,) and wind speed (recorded by the anemometer). Besides remotely visualizing the sensor data, the user can analyze the sensor data with embedded functionalities provided by the web interface.



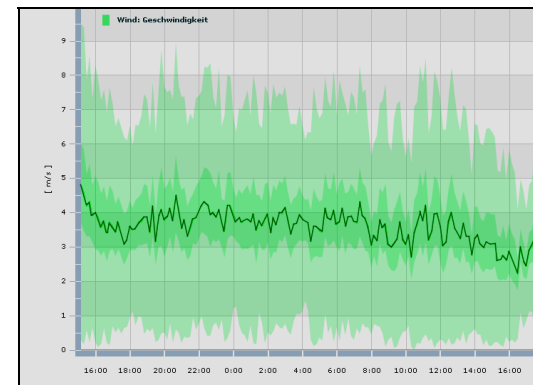
(a) Horizontal accelerations (sensor B6).



(b) Vertical displacement (sensor W5).



(c) Temperature (sensor T5).



(d) Wind speed (anemometer USA-1).

Figure 3. Data sets remotely accessed via the web interface provided by the SHM system (collected on October 22, 2010, during one day).

Malfunctions observed during operation

A typical malfunction of a temperature DAU, observed during operation of the SHM system is elucidated in Fig. 4. As can be seen from Fig. 4, the temperature DAU malfunction is implicitly indicated by anomalies within the data sets collected by the DAUs themselves. A malfunction, or cause for alert, that is characterized by a large number of identical measurements due to an internal system crash of the DAU at time

t_0 is shown in Fig. 4. As a result of the malfunction occurred, the DAU cannot record and store the actual temperature measurements. Instead, the last normally collected value (sensed right before the occurrence of the malfunction) is stored repeatedly by the DAU.

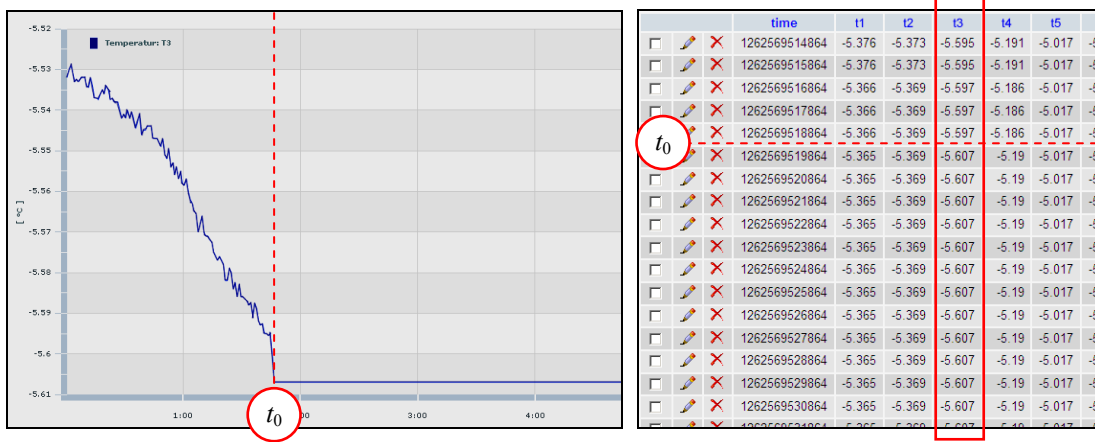


Figure 4. Visual and numerical representation of a DAU malfunction illustrated by temperature sensor T3 through the web interface (left) and the direct database connection (right).

A MULTI-AGENT SYSTEM FOR AUTONOMOUS MALFUNCTION DETECTION

The purpose of the multi-agent system, which is integrated into the SHM system, is to detect malfunctions in the DAUs automatically and to inform the responsible individuals immediately about the defects so that the affected DAUs can be restarted remotely or replaced in a timely manner. The agent-based paradigm has been chosen as a technological basis for the implementation of the self-detection and self-management functions. A software agent can flexibly carry out its autonomous actions and cooperatively communicate and work with other agents. Software agents are characterized by a number of properties (Wooldridge and Jennings 1995):

- **autonomy:** agents operate without any direct intervention by humans and exhibit control over their actions
- **social ability:** agents interact with other agents (and possibly humans) via agent-communication languages
- **reactivity:** agents perceive their environment (for example, the physical world or a SHM system) and respond to the changes that occur in the environment
- **pro-activeness:** agents do not simply act in response to their environment, but they are able to take initiative and to apply goal-directed behavior

The developed multi-agent system comprises a set of collaborating software agents implemented to accomplish the tasks required for malfunction detection. First,

the multi-agent system hosts two special agents, namely the “Agent Management System (AMS) agent” and the “Directory Facilitator (DF) agent”. The AMS agent is responsible for the overall management actions, including creating and terminating other agents. The DF agent implements a yellow page service, allowing the agents to advertise their services as well as to look up other agents that offer the services needed to cooperatively accomplish a specific monitoring task. The AMS and DF agents together provide the overall configuration description of the multi-agent system. Furthermore, an “interrogator agent”, implemented for analyzing the monitoring data with respect to malfunctions, and a “mail agent” for sending notifications to the responsible individuals are included in the multi-agent system.

Data interrogator agent

The interrogator agent analyzes the sensor data collected by the DAUs to detect malfunctions. As in the example described earlier, one malfunctioning symptom for a temperature DAU can be characterized by a large number of identical measurements. To detect such an anomaly, the interrogator agent at certain time intervals extracts and analyzes the temperature data sets stored in the centralized MySQL database. A set of configuration files (specifying interrogation parameters, database URL, database driver, sensor specification, scheduled interrogation intervals, etc.) is pre-defined in the multi-agent system (Smarsly *et al.* 2010).

The connection to the database being part of the software subsystem M_2 is realized through a programming interface based on the Java Database Connectivity (JDBC). JDBC is an industry standard for database-independent connectivity between Java applications, such as the software agents considered here, and various database systems. The security of database requests and data transmissions are provided by the database system, i.e. the MySQL database, which requires password and username as well as secure drivers to be specified by the agent.

Email notification agent

Upon detecting a malfunction, responsible individuals are immediately notified. On behalf of the interrogator agent, notifications are issued by the mail agent via an agent-based email messaging service in a two-step process. In the first step, the mail agent collects all relevant data from the monitoring system, which is necessary for composing the emails. The mail agent creates the emails using the information about the observed anomaly that are provided by the interrogator agent. Metadata, such as addresses of the recipients or the email server to be used, are acquired for the automated emailing; these metadata are stored in configuration files located in the multi-agent system. In other words, the email content is created based on the anomaly information received from the interrogator agent and the email header is automatically created based on the metadata obtained by the mail agent from the configuration files resided in the multi-agent system. Once the emails are composed, they are sent by the mail agent to the email clients (the individuals responsible for the malfunctioning devices). Secure email messages are ensured by username- and

password-based authentications that the mail agent, like a human user, needs to specify when trying to access the mail server.

Decentralization and extensibility

One distinct advantage of the agent-based approach is that the software system can be easily extended to accommodate additional functions. For example, self-detections of other types of malfunctions (that are not characterized by a large number of identical values) or interrogations of other measurement data (such as acceleration data) can be implemented as additional specialized agents that are integrated into the multi-agent system. Also, damage detection algorithms, such as fast Fourier transforms or autoregressive models, can be included into the multi-agent system as software agents. Software packages, such as MATLAB, can be integrated into the multi-agent system by “wrapping” the particular software package as an agent.

Software agents can be distributed over several computers and geographically situated at different locations. Communications between distributed agents can be carried out through agent messages via computer networks, such as the Internet. Additional agents can be launched remotely to perform specific structural monitoring functions as long as they are registered to the multi-agent system.

To ensure extensibility and interoperability, the multi-agent system is implemented in compliance with the specifications issued by the Foundation for Intelligent Physical Agents (FIPA). FIPA, the IEEE Computer Society standards organization for agents and multi-agent systems, promotes agent-based technology, interoperability of agent applications and the interoperability with other technologies. The FIPA agent communication specifications, for example, ensure a common understanding of two or more communicating agents based on the FIPA Agent Communication Language (ACL) as used for the communication within multi-agent systems (FIPA 2002).

VALIDATION OF THE MULTI-AGENT SYSTEM

The multi-agent system has been integrated into the existing SHM system to monitor the temperature DAUs instrumented on the wind turbine. The SHM system and the multi-agent system are installed at different geographical locations as follows:

- The hardware subsystem M_1 (including sensors, DAUs and on-site computer) is installed with the wind turbine located in Dortmund (Germany).
- The components of the software subsystem M_2 , including the MySQL database system, reside on different computers at the Institute for Computational Engineering at the Ruhr-University Bochum (Germany).
- For the validation tests, the multi-agent system is distributively located at different institutes: The interrogator agent and the mail agent for detecting malfunctions and generating alerts are located at Stanford University (USA) while the email server, used by the mail agent at Stanford University, is situated at the Ruhr-University Bochum.

For detection of malfunctions in the temperature DAUs, the interrogator agent scans the collected data sets, based on experience, for 150 consecutive identical values. Fig. 5 shows two malfunctions, detected from the data sets collected by the temperature DAUs in January, 2010 (Fig. 5a), and in July, 2010, (Fig. 5b). These malfunctions are characterized by a large number of identical measurements, which, as a result of a DAU-internal system crash, are repeatedly stored by the DAU instead of regular measurements.

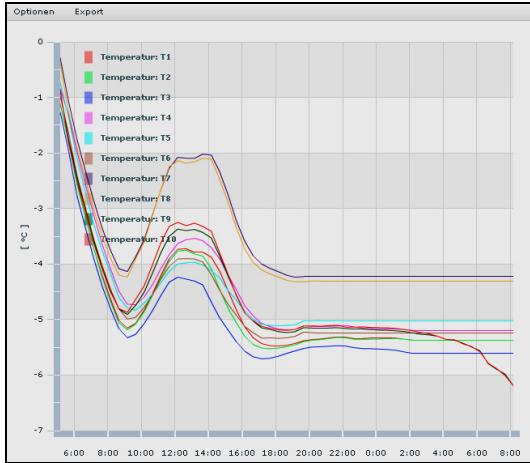
Once a malfunction is detected, the multi-agent system generates an alert to notify the authorized users immediately about the malfunction so that appropriate measures can be taken in order to avoid the loss of valuable monitoring data. In detail, the interrogator agent requests the mail agent to inform the responsible users via email about the malfunction discovered. As a result, the mail agent generates an email to submit it to the recipients as specified in the multi-agent system's configuration files. As shown in Fig. 5c, the email content includes detailed information on the detected malfunction and is generated using the analysis results provided by the interrogator agent. The email header is assembled based on the information taken from the configuration files.

SUMMARY AND CONCLUSIONS

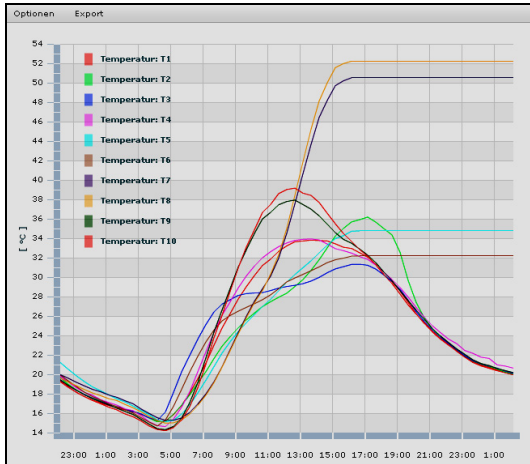
In this paper, a multi-agent system for the autonomous detection of SHM system malfunctions has been presented. The multi-agent system has been integrated into an existing decentralized SHM system for real-time monitoring of a 500 kW wind turbine in Germany. Since its initial deployment in December, 2009, the multi-agent system is in continuous operation and has reliably detected all malfunctions of the temperature data acquisition units of the SHM system¹. Furthermore, the multi-agent system has immediately notified the technicians by email, whereupon the affected data acquisition units were remotely restarted by the technicians in a timely fashion. Consequently, considerable amounts of valuable sensor data, which could otherwise be lost as in conventional SHM systems, were saved and are available for assessing the performance and operational conditions of the wind turbine.

Due to its adaptability and modularity, the multi-agent system can be easily extended and integrated with other existing SHM systems. Software agents and the multi-agent system can be installed on any computers at any locations away from the SHM system. Treated as software agents, different algorithms for malfunction detection and monitoring functions can easily be added into the multi-agent system. In summary, this research has successfully demonstrated that the multi-agent software paradigm can be advantageously deployed for long-term structural health monitoring practice under real-world conditions.

¹ For long-term operation, the multi-agent system has been installed at the Institute for Computational Engineering in Bochum.



(a) Malfunction occurred on January 3, 2010 (affected sensors: T1, ..., T8).



(b) Malfunction occurred on July 19, 2010 (affected sensors: T5, T6, T7, T8).

Von: monitoring@inf.bi.ruhr-uni-bochum.de
An: smarsly@stanford.edu
Betreff: [Generated Email] Wind Turbine Safety Report
Datum: 20.07.2010 03:33:20

This email has autonomously been generated
and was sent by a software agent. Do not reply.

This message is sent by the 'InterrogatorAgent'.
Content:

Analysis properties:
Database driver: com.mysql.jdbc.Driver
Database table: pt104
Number of values observed per sensor: 86400
Number of significant values: 150
Sensors analyzed: t1;t2;t3;t4;t5;t6;t7;t8;t9;t10;

Anomaly detected:
DAU id: pt104
Date: Mon Jul 19 17:57:59 CEST 2010
Sensor: t5
Errornumber: 34.764
(...)

Properties:
IP address: 134.147.216.69
Internet host: tunsq1
Hap: tunsq1: 1099/JADE
Responsible agent: mailAgent@tunsq1:1099/JADE
Agent container: tunsq1:1099/JADE
Container state: Ready(4)
Agent platform: tunsq1:1099/JADE
Mail host: delta.inf.bi.rub.de
Port: 25
Debugging: true
Protocol: smtp
Sender: monitoring@inf.bi.ruhr-uni-bochum.de
Receivers: 2
Local config file: config/mail.shm

(c) Email notification generated and sent in direct consequence of malfunction (b).

Figure 5. Visual representation of two detected malfunctions and email notification generated and sent by the software agents.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of the German Research Foundation (DFG) through the grants SM 281/1-1 and HA 1463/20-1. This research is also partially supported by the National Science Foundation (NSF) through grant CMMI-0824977.

REFERENCES

Foundation for Intelligent Physical Agents – FIPA (2002). SC00061:2002 *FIPA ACL Message Structure Specification*. Alameda, CA, USA.

- German Federal Ministry of Justice – BMJ (2009). “*EEG – Gesetz für den Vorrang erneuerbarer Energien.*” (BGBl. I S. 2074 and BGBl. I S. 643), Berlin, Germany.
- Hartmann, D., and Höffer, R. (2010). “*Lifespan assessment of wind energy converters through system identification (Lebensdauerabschätzung von Windenergieanlagen mit fortlaufend durch Systemidentifikation aktualisierten numerischen Modellen).*” Research project funded by the German Research Foundation (DFG) through the research grant HA 1463/20-1, Ruhr-University Bochum, Bochum, Germany.
- Jüngert, A. (2008). “Damage Detection in wind turbine blades using two different acoustic techniques.” *The NDT Database & Journal (NDT)*.
- Lading, L., McGugan, M., Sendrup, P., Rheinländer, J., and Rusborg, J. (2002). “*Fundamentals for Remote Structural Health Monitoring of Wind Turbine Blades – a Preproject.*” Report, Risø National Laboratory, Roskilde, Denmark.
- Ministry of Innovation, Science, Research and Technology of the State of North Rhine-Westphalia – MIWF (2009). “*Das Cluster EnergieForschung.NRW.*” Report, Düsseldorf, Germany.
- Rolfes, R., Zerbst, S., Haake, G., Reetz, J., and Lynch, J.P. (2007). “Integral SHM-System for Offshore Wind Turbines Using Smart Wireless Sensors.” *Proceedings of the 6th International Workshop on Structural Health Monitoring*, Stanford, CA, USA.
- Smarsly, K., Law, K.H., and Hartmann, D. (2010). “A Multi-Agent-Based Collaborative Framework for a Self-Managing Structural Health Monitoring System.” *ASCE Journal of Computing in Civil Engineering* (submitted).
- Sutherland, H., Beattie, A., Hansche, B., Musial, W., Alread, J., Johnson, J., and M. Summers (1994). “*The Application of Non-Destructive Techniques to the Testing of a Wind Turbine Blade.*” Sandia Report SAND93-1380, Sandia National Laboratories, Albuquerque, NM, USA.
- Wooldridge, M., and Jennings, N.R. (1995). “Intelligent Agents: Theory and Practice.” *Knowledge Engineering Review*, **10**(2), 115-152.
- World Wind Energy Association – WWEA (2010). “*16 Gigawatt of Wind Power added in First Half of 2010.*” Press release. [Online]. Available at: <http://www.wwindea.org> [Accessed October 26, 2010].