NSF GRANT # CMMI-0824977
NSF PROGRAM NAME: Hazard Mitigation and Structural Engineering

Implementing a Multiagent-Based Self-Managing Structural Health Monitoring System on a Wind Turbine

Kay Smarsly
Post-Doctoral Fellow
Department of Civil and Environmental Engineering
Stanford University, Stanford, CA 94305-4020, USA

Kincho H. Law
Professor
Department of Civil and Environmental Engineering
Stanford University, Stanford, CA 94305-4020, USA

Dietrich Hartmann
Professor
Department of Civil and Environmental Engineering
Ruhr-University Bochum, 44780 Bochum, GERMANY

Abstract: Deteriorations, increasing costs and burdens of maintenance work as well as recent collapses of civil engineering structures underscore the need for robust, cost-effective structural health monitoring (SHM) strategies. This paper describes the design and prototype implementation of a multi-agent software paradigm for a self-managing structural health monitoring system. As will be highlighted in the paper, one of the key demonstrations of the prototype system is the timely detection of system malfunctions such as breakdowns of sensors, temporarily unavailable data acquisition units or server systems. Without such self-diagnostic and management capabilities as in current conventional monitoring systems, valuable monitoring data – necessary for an accurate and reliable safety assessment – can be lost as system malfunctions are usually not detected at an early stage. The performance of the prototype is validated through a long-term system deployment test on a 500 kW wind turbine in Germany.

1. Introduction: Many civil engineering structures, such as wind turbines considered here, are approaching or have already exceeded their originally estimated service life. Due to ageing, environmental factors and, often, inadequate maintenance, the conditions of these structures are progressively deteriorating, urgently needing inspection and repair works. For wind turbines, which are usually designed to work for 120,000 hours of operation throughout their design lifetime of about 20 years, at least 25 % of the total expenses during their lifetime are operation and maintenance costs (EWEA 2010, DWIA 2010). With approximately 90,000 wind turbines currently installed worldwide, the wind energy sector represents an emerging market of considerable economical importance (Lachmann *et al.* 2009). Installed with automated SHM systems, unnecessary replacements of wind turbine components as well as down time can be avoided, thereby leading to decreased lifecycle costs, reduced inspection time and increased performance of the structures.

New concepts and technologies are needed to develop reliable and cost-effective strategies for monitoring the integrity and health of the structures. Recent developments in the area of structural health monitoring have demonstrated that new methodologies are available to supplement current monitoring practice and enhance safety. Research in SHM represents an active field encompassing numerous research directions where various state-of-the-art reviews have been reported (Chang *et al*. 2003, Glaser *et al*. 2007, Liu *et al*. 2006). While most current research activities in SHM focus on the development, implementation and validation of damage detection algorithms, sensors and sensor networks, robust software platforms designed to facilitate collaborative efforts that support data archiving and usages, and automated execution of monitoring tasks have received little attention.

In this paper, an integrated monitoring concept is proposed for continuous assessment of structural performance and safety as well as self-diagnostics and

management of the structural health monitoring system. Specifically, the distributed multi-agent software paradigm, evolved from distributed AI research, is introduced as a conceptual foundation for the autonomous self-managing SHM system. After a brief overview of the proposed monitoring framework, this paper describes the design, the implementation and the validation of the agent-based SHM system. A wind turbine located in Germany serves as a reference structure for comprehensive system tests and continuous long-term field validation studies. Specifically, the self-management functionalities of the SHM system for detecting anomalies and system malfunctions such as sensor breakdowns or temporarily unavailable resources are illustrated.

2. Overview of the SHM System: The overall architecture of the distributed SHM system for the wind turbine is shown in Fig. 1. The SHM system consists of three subsystems: an on-site hardware subsystem M_1, a real-time monitoring subsystem M_2 and an on-demand monitoring subsystem M_3. Installed in the observed wind turbine, the hardware subsystem M_1 includes a dense array of sensors connected to data acquisition units, which are controlled by a computer located on-site in the structure. The subsystem M_2 is designed to autonomously execute crucial monitoring tasks, such as interrogating the data, storing the data persistently in a database server, and providing system access to the human users or the intelligent software agents ("artificial users"). The third subsystem, the on-demand monitoring subsystem M_3, allows human users and software agents remotely logging into the real-time monitoring subsystem M_2.

2.1 On-Site Hardware Subsystem M_1: The wind turbine, a 500 kW gearbox-less structure, is instrumented with the on-site hardware system that includes sensors, data acquisition units and a computer. The sensors are installed inside the steel shaft of the wind turbine, outside the structure and on the foundation. Fig. 2 shows the installation of displacement transducers, accelerometers and temperature sensors. Three-dimensional PCB-3713D1FD3G piezoelectric accelerometers, manufactured by PCB Piezotronics, are placed at five levels in the wind turbine shaft. In addition, at the foundation of the wind turbine three single-axis PCB-393B12 piezoelectric seismic ICP accelerometers are installed. 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 adequately consider temperature influences on the displacement measurements. At two further levels both inside and outside the wind turbine, additional temperature sensors are placed, which are primarily used for determining 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. Additionally, for acquiring three-dimensional wind information, a Metek USA-1 ultrasonic anemometer is mounted on a telescopic mast near the wind turbine as shown in Fig. 3.

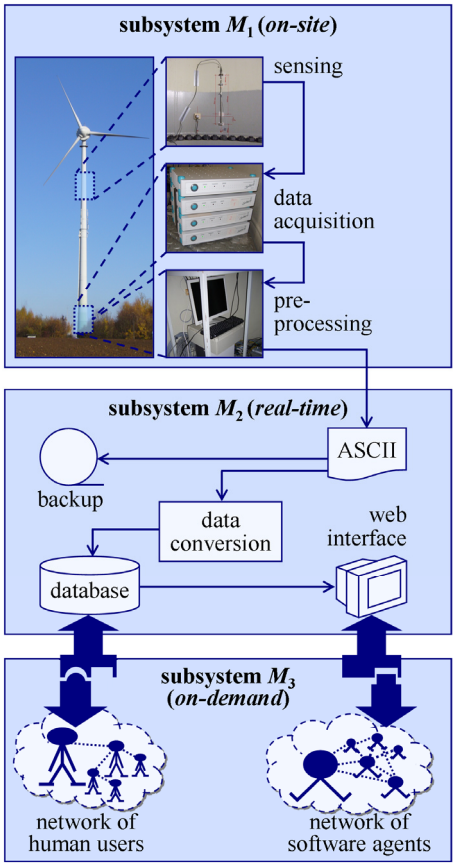

Figure 1. Architecture of the SHM system.

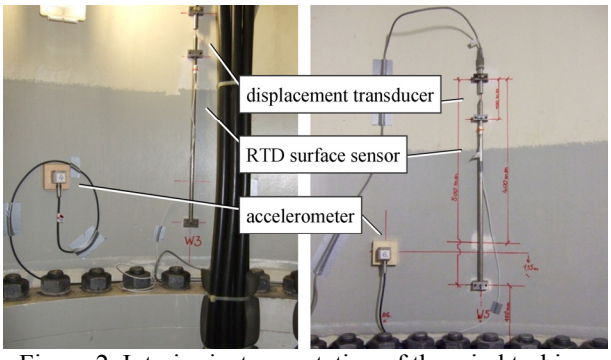

Figure 2. Interior instrumentation of the wind turbine being part of the on-site hardware subsystem.

Figure 3. Three-dimensional ultrasonic anemometer.

2.2 Real-Time Monitoring Subsystem M_2: The purpose of the subsystem M_2 is to perform in real-time the data archival and data processing tasks such as condensing, converting and storing of the acquired data sets. The data collected by M_1 is streamed to the monitoring subsystem M_2. There, the data is automatically backed-up, condensed, converted and stored into a central database system, which in turn supports access by participating (human or artificial) actors. The subsystem M_2 comprises (a) server systems for on-line data synchronization, data conversion and service-oriented data transmission, (b) RAID-based storage systems for periodic backups, (c) database systems for persistent data storage and (d) web interfaces for accessing and visualizing the monitoring data.

2.3 On-Demand Monitoring Subsystem M_3: Unlike M_1 and M_2, which continuously monitor the structure and collect, process and archive the sensor data, the subsystem M_3 is operating according to an on-demand basis by a user, a software agent or an application tool. The on-demand monitoring subsystem M_3 is designed and implemented as a distributed multi-agent system *MAS*, which is composed of spatially distributed but functionally interconnected agent systems AS_i. Each agent system hosts specialized autonomous software agents a_i that are introduced to cooperatively perform specific monitoring tasks. The *autonomous* software agents are capable of controlling their own actions and taking their own decisions. For example, an agent can select to answer or to discard a request from another agent. Furthermore, an agent can respond *reactively* to a situation and trigger appropriate monitoring actions. For example, an agent may react to a detected anomaly based on observed data and trigger a notification to another agent. Last but not least, an agent may act as a *social* entity, communicating and interacting with other agents and/or human users to cooperatively perform a specific monitoring task.

Fig. 4 shows the overall organization of the distributed multi-agent system *MAS*. *MAS* includes a main agent system AS_{main} that provides the necessary infrastructure and management services, such as agent registration, security and other system functions. The main agent system hosts two special agents: (1) the "Agent Management System (AMS) agent" which is responsible for creating or terminating other agents, and (2) the "Directory Facilitator (DF) agent" which implements a yellow page service, allowing the agents to advertise their services and to look up services offered by other agents. By interacting with AS_{main}, distributed software agents can locate, interact and collaboratively work with other autonomous software agents. In addition, AS_{main} includes a "supervisor agent" $a_{s,main}$ for managing the execution of monitoring tasks and a "database agent" $a_{d,main}$ for providing remote access to the central database system – a MySQL database – residing in the monitoring subsystem M_2.

Once registered with the main agent system AS_{main}, an agent system AS_i can be launched remotely to carry out the monitoring functions. An agent system AS_i can be installed on any computer that has access to AS_{main}, for example, via the Internet. As a result, the capabilities of the multi-agent system *MAS* are extended by those functions offered by the software agents of the remote agent systems AS_i, specializing in solving specific sets of monitoring tasks.

3. Software Agent Implementation: To illustrate the design of the multi-agent framework for a self-managing SHM system, this section briefly describes the implementation of two software agents, namely an "interrogator agent" $a_{i,1}$ for analyzing the monitoring data and a "mail agent" $a_{m,1}$ for sending notifications to responsible individuals in case of detected anomalies. Also, the interactions between the agents are illustrated.

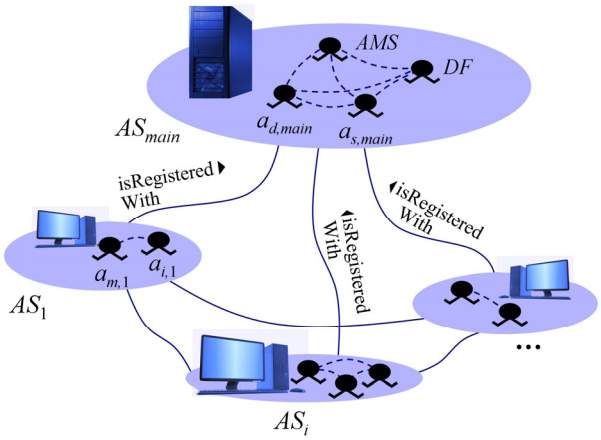

Figure 4. Distributed architecture of the multi-agent system.

3.1 An Interrogator Agent: The interrogator agent analyzes the data collected from the sensors installed on a structure to detect potential anomalies. Many statistics-based algorithms for detecting anomalies, such as Grubb's test, Hampel's test, Walsh's test, Dean-Dixon-Test, Chauvenet's criterion, Nalimov-Test, Peirce's criterion, etc., have been proposed in the literature (Sachs and Hedderich 2009). Besides statistical procedures that are targeted in detecting anomalies, other simple procedures, such as observing "inconsistent" or "abnormal" data values for specific sensor types, can also be implemented in the interrogator agent. To access the sensor data, the interrogator agent interacts with the database residing in the monitoring subsystem M_2, via the database agent $a_{d,main}$. Using the programming interface based on the Java Database Connectivity (JDBC), the interrogator agent can simply submit an SQL query, implementing simple computational procedure such as computing mean values and standard deviations, directly to the database. Alternatively, sensor data can be retrieved from the database and analyzed by the interrogator agent. The security of database requests and data transmissions are provided by the database system, which requires password and username as well as secure drivers to be specified by an agent trying to access the database system.

3.2 A Mail (Notification) Agent: Upon the detection of an anomaly, notifications to the responsible individuals are issued by the mail agent $a_{m,1}$ on behalf of the interrogator agent $a_{i,1}$. The mail agent $a_{m,1}$ first collects all relevant data from the monitoring system, which is necessary for composing the emails. The mail agent $a_{m,1}$ creates the emails using the information provided by $a_{i,1}$ about the observed anomaly. Furthermore, metadata, such as the email addresses of the recipients are acquired from the configuration files located in AS_{main}. Once the emails are composed, they are sent by $a_{m,1}$ to the email clients (the human experts) as recommended by AS_{main} using the Simple Mail Transfer Protocol (SMTP), which ensures secure email messages using username- and password-based authentications.

3.3 Agent Interaction: In a multi-agent system, the actions undertaken by an agent are based on its local behaviors and the interactions with other agents. An agent behavior defines how the agent perceives, analyzes and, in particular, reacts to various external events (e.g. sensor malfunctions, detected anomalies or unavailable monitoring data), as well as interacts with other agents. Fig. 5 shows three agent behaviors, `CyclicInterrogation`, `MailInteraction-Initiator` and `MailInteractionResponder`, that are needed for executing the "anomaly detection" described within the agent system AS_1. The behaviors `CyclicInterrogation` and `MailInter-actionInitiator` belong to the interrogator agent $a_{i,1}$. The third agent behavior `MailInteraction-Responder` is an integral part of the mail agent $a_{m,1}$.

As shown in Fig. 5, the anomaly detection executed by $a_{i,1}$ is initiated by its `CyclicInterrogation` behavior. The `CyclicInterrogation` behavior periodically requests and analyzes the monitoring data with respect to anomalies and inconsistencies. Upon detecting an anomaly, $a_{i,1}$ requests the email notification service and uses its `MailInteraction-Initiator` behavior for contacting mail agent $a_{m,1}$. Email notification to responsible individuals is handled by the mail agent's `MailInteractionResponder` behavior. After receiving and accepting a notification request, $a_{m,1}$ informs $a_{i,1}$ about its decision on agreeing or refusing to handle the request. Once agreed, $a_{m,1}$ processes the request by preparing and sending the email notifications to the users responsible for handling the detected anomaly, as described earlier. If the emails are sent successfully, $a_{m,1}$ informs $a_{i,1}$ and the monitoring procedure is finalized; otherwise, a failure message is returned from $a_{m,1}$ to $a_{i,1}$.

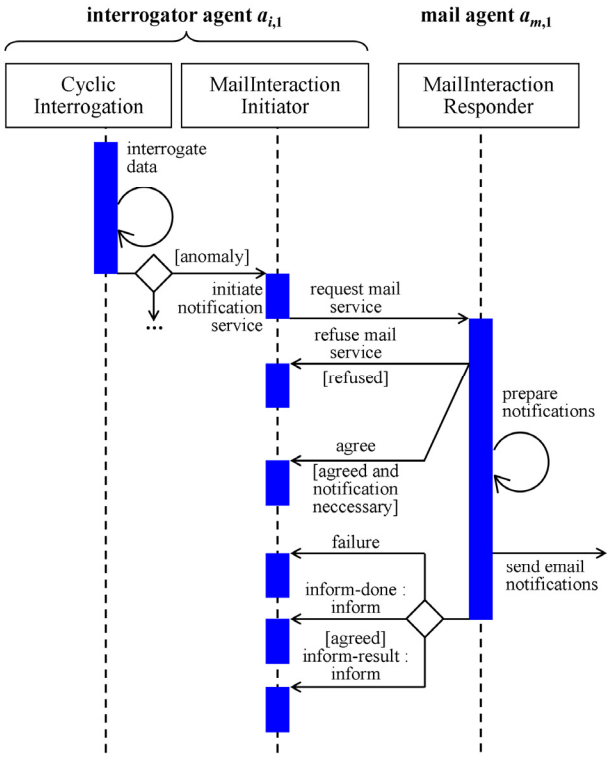

Figure 5. Agent interaction for email notification executed in case of a detected anomaly.

The interaction and communication of the multi-agent system *MAS* is implemented in accordance with the specifications issued by the Foundation for

Intelligent Physical Agents (FIPA), an IEEE Computer Society standards organization promoting agent-based technology and interoperability of agent applications. Adhering to the FIPA specifications not only ensures extensibility and interoperability but also provides considerable advantages with respect to performance and robustness of a multi-agent system. The FIPA Agent Communication Language (ACL), which is based on the speech act theory, ensures common understanding between communicating agents (FIPA 2002). FIPA ACL includes an extensible library of communicative acts and a comprehensive representation of communicative intentions (such as "inform", "request", "refuse", "propose", etc.) that can be used to express the communicative acts and intentions of an agent.

4. Validation of the Self-Managing SHM System: The proposed multi-agent framework for a self-managing SHM system has been implemented and installed on the 500 kW wind turbine (Fig. 6). The three subsystems, as depicted in Fig. 1, are installed at different locations. The on-site hardware subsystem (including sensors, data acquisition units and computer) is installed with the wind turbine located in Dortmund, Germany. The real-time monitoring subsystem and the centralized MySQL database system are installed at the Institute for Computational Engineering at the Ruhr-University Bochum, Germany. For the on-demand multi-agent system *MAS*, the main agent systems AS_{main} is installed at the Ruhr-University Bochum, Germany. For the validation tests presented in this paper, the interrogator agent $a_{i,1}$ and the mail agent $a_{m,1}$ for detecting malfunctions and generating alerts are located at Stanford University, USA. The email server, used by the agent $a_{m,1}$ at Stanford University, resides at the Ruhr-University Bochum.

The SHM system prototype and the multi-agent monitoring system have been installed and in service since December, 2009, continuously monitoring the operation of the wind turbine. Since then, the system has been able to detect different malfunctions that occurred. Malfunctioning in the SHM system and components, such as breakdowns of sensors, temporary unavailability of a data acquisition unit or a disconnected server system, occurs often in a real-time SHM system and could cause significant loss of valuable monitoring data. To give an example, a malfunction on the temperature DAUs (which collect data at a frequency rate of 1 Hz from 10 temperature sensors) undetected for a week would cause the loss of more than 6,000,000 temperature readings. The value and reliability of a monitoring system can be substantially hampered if such malfunctions are not detected early.

Figure 6. Reference structure.

To illustrate the self-management operations, the following discussion focuses on the self-detection of the malfunction of the temperature data acquisition units (DAUs). For the temperature DAUs, malfunctions are implicitly indicated by anomalies within the data sets, characterized by a large number of identical measurements. To detect such an anomaly, the interrogator agent $a_{i,1}$ sends a query at certain time intervals, extracts and analyzes the temperature data sets stored in the centralized MySQL database (Smarsly *et al*. 2004). The monitoring sequence proceeds according to the implemented agent interactions as illustrated in Fig. 5. Fig. 7 shows an example of the data set collected from the temperature DAUs on January, 2010. As shown, the anomaly is 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. The anomaly has led to an undesirable interruption of the automated data acquisition process.

Once the malfunction is detected, the SHM system prototype generates an alert to inform the authorized users immediately about the anomaly. The interrogator agent $a_{i,1}$ requests the mail agent $a_{m,1}$ to inform the human users via email about the malfunction discovered in the temperature acquisition process. Fig. 8 documents the interaction between the participating agents to carry out the agent-based email notification process. It should be noted that, although the mail agent $a_{m,1}$ and the interrogator agent $a_{i,1}$ are the main

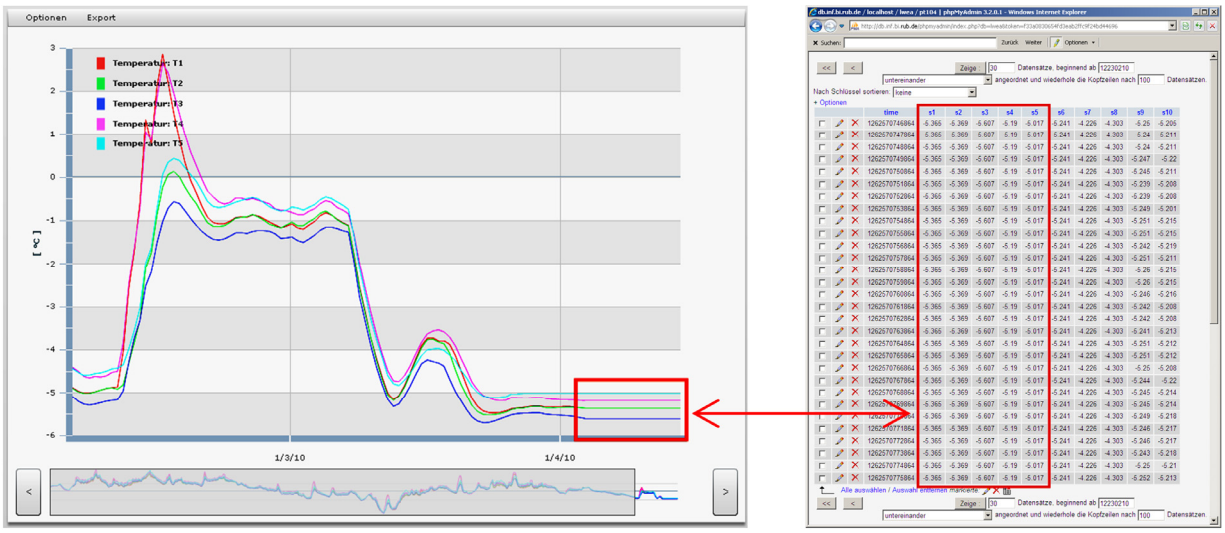

Figure 7. Graphical and numerical representation of a detected anomaly using the web interface (left, source: Hartmann and Höffer 2010, modified) and the database connection (right).

actors, additional information (such as agent identifiers, agent services provided, etc.) are supplied by the Directory Facilitator agent in the main agent system. Consequently, the malfunctioning temperature DAUs received immediate attention and the affected DAU was remotely restarted by the responsible technician early to prevent potential loss of valuable measurement data.

The multi-agent framework has shown to be very valuable in self-managing the SHM system. The malfunctioning of the commercial temperature DAUs described, which was mainly caused by the cold winter 2009-2010 in Europe (–20°C and less), was one of the many examples. Since the installation of the system, not only the exceptional events (such as power blackouts) can be detected, but also regular events are taken into account (such as software updates of the operating system that requires reboot of the computer and, hence, temporary termination of all software applications). For all these events, the SHM system prototype, incorporated with self-management functions, has been able to reconfigure and restart itself.

5. Summary and Discussions: This paper has introduced a modular multi-agent framework for the development of a robust, self-managing SHM system. The proposed concept has been illustrated with the continuous real-time monitoring of a wind turbine for almost a one-year period. This long term validation experiment has demonstrated that the multi-agent system is capable of reliably detecting anomalies and system malfunctions such as sensor breakdowns or temporarily unavailable resources. Without such self-diagnostic and self-management capabilities as in current conventional monitoring systems, large amounts of valuable monitoring data can be lost. The robustness, fault-tolerance and performance of a multi-agent system, as an integral part of the SHM prototype, have been demonstrated. Because of its modularity and extendibility, multi-agent technology is a promising paradigm for the implementation of structural health monitoring systems, allowing continuing upgrades and new advancements to be incorporated in an autonomous distributed manner.

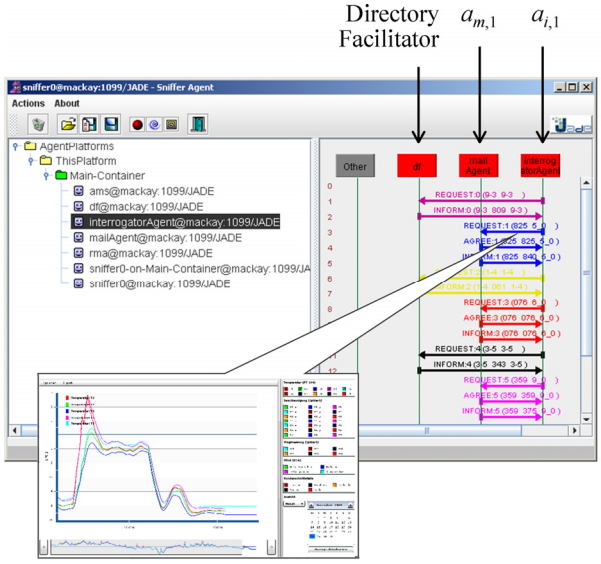

Figure 8. Agent interactions within the email notification procedure.

6. Acknowledgments: 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 US National Science Foundation (Grant No. CMMI-0824977).

7. References:

[1] ASCE – American Society of Civil Engineers (2009). "2009 Report Card for America's Infrastructure." [Online]. Available at: http:// www.infrastructurereportcard.org [Accessed March 25, 2010].

[2] P.C. Chang, A. Flatau, and S.C. Liu, (2003). "Review Paper: Health Monitoring of Civil Infrastructure." *Structural Health Monitoring*, **2**(3), 257-267.

[3] DWIA – Danish Wind Industry Association (2010). "Wind Energy Economics." [Online]. Available at: http://www.talentfactory.dk [Accessed May 3, 2010].

[4] EWEA – The European Wind Energy Association (2010). "Operation and Maintenance Costs of Wind Generated Power." [Online]. Available at: http://www.wind-energy-the-facts.org [Accessed May 3, 2010].

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

[6] S. Glaser, H. Li, M. Wang, J. Ou, and J.P. Lynch, (2007). "Sensor Technology Innovation for Advancement of Structural Health Monitoring: A Strategic Program Plan of US-China Research for the Next Decade." *Smart Structures & Systems*, **3**(2), 221-244.

[7] D. Hartmann, and R. Höffer, (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.

[8] S. Lachmann, M. Baitsch, D. Hartmann, and R. Höffer, (2009). "Structural Lifetime Prediction for Wind Energy Converters based on Health Monitoring and System Identification." *Proceedings of the 5th European and African Conference on Wind Engineering*, Florence, Italy.

[9] S.C. Liu, M. Tomizuka, and G. Ulsoy, (2006). "Strategic Issues in Sensors and Smart Structures." *Structural Control and Health Monitoring*, **13**(6), 946- 957.

[10] L. Sachs, and J. Hedderich, (2009). *Angewandte Statistik*. 13th ed., Springer-Verlag GmbH, Heidelberg, Germany.

[11] K. Smarsly, I. Mittrup, J. Bilek, and M. Bloch, (2004). "Implementation of a Software Agent for the Administration of Complex Data in Civil Engineering (in German)." *Proceedings of the Forum Bauinformatik 2004*, Braunschweig, Germany.

[12] K. Smarsly, and D. Hartmann, (2009). "AMBOS – A Self-Managing System for Monitoring Civil Engineering Structures." *Proceedings of the 16th EG-ICE Conference of the European Group for Intelligent Computing in Engineering*, Berlin, Germany.