

Data Management of On-Line Partial Discharge Monitoring Using Wireless Sensor Nodes Integrated with a Multi-Agent System

P. C. Baker, *Student Member, IEEE*, S. D. J. McArthur, *Senior Member, IEEE*, M. Judd, *Senior Member, IEEE*

Abstract—On-line partial discharge monitoring has been the subject of significant research in previous years but little work has been carried out with regard to the management of on-site data. To date, on-line partial discharge monitoring within a substation has only been concerned with single plant items, so the data management problem has been minimal. As the age of plant equipment increases, so does the need for condition monitoring to ensure maximum lifespan. This paper presents an approach to the management of partial discharge data through the use of embedded monitoring techniques running on wireless sensor nodes. This method is illustrated by a case study on partial discharge monitoring data from an ageing HVDC reactor.

Index Terms—Condition Monitoring, Multi-agent Systems, Intelligent Systems, Sensor Networks.

I. INTRODUCTION

CONDITION monitoring (CM) of power industry equipment has been the subject of significant research since the early 1990s and has proven to be instrumental in enhancing the performance of plant equipment, increasing equipment lifespan and reducing operational costs. Historically, time-based maintenance programs have been the main strategy for maintaining plant equipment, where equipment is inspected and serviced according to a schedule or a number of operational hours. This technique uses very little information relating to the health of the equipment to schedule maintenance, and failures can easily occur between maintenance periods. CM-based techniques take into account live and archived real-world data relating to the health of the equipment, with the aim of predicting failures before they occur [1].

Multi-agent systems (MAS) have been proven to benefit condition monitoring applications, providing a flexible and extensible framework for intelligent agents to cooperate towards the diagnosis of faults [2]. These systems are currently deployed onto PCs within substations, connected to Ultra-High Frequency (UHF) sensors for analysis of partial discharges (PDs) within ageing transformers. As more sensors are deployed to substations, the need for a suitable sensor deployment platform becomes more important to handle the increase in monitoring data.

Recent developments in miniaturisation of digital electronics devices have fuelled the development of wireless sensor networks (WSNs), which are increasingly being used for a wide-range of monitoring applications. These networks are

made up of a number of individual devices which 'self organise' to form an ad-hoc network with redundant links which pass sensor data back to data aggregation nodes connected to a wired network for analysis and action by engineers. These devices offer an integrated computing platform, encapsulating sensing, processing, communications and power components in a single compact package. Sensor networks have already been used for a number of applications, such as environmental monitoring, military applications, health applications, home automation, smart environments, as well as numerous industrial applications [3], but, to the authors' knowledge, sensor network technology has not been applied to power engineering applications.

An increase in monitoring sensors also leads to an increase in data. By analysing data at the source, bandwidth requirements can be reduced to a manageable level, ensuring that engineers receive only pertinent data. Also, as the processing capabilities of sensor nodes increases, the complexity of their applications will increase. This will allow power engineering applications to be deployed on inexpensive hardware where it was not previously possible.

This paper describes a new component to condition monitoring architectures, which can support a much larger number of monitoring sensors within a substation. By integrating a wireless sensor network with an existing multi-agent system, reactive analysis is carried out by an agent on the sensor device and statistical event data is passed back to a central gateway either on-demand, in regular intervals, or as determined by the sensing node. This paper uses archived data from an HVDC reactor, previously suspected of undergoing UHF partial discharge emissions. Large volumes of monitoring data are analysed, with uninteresting data discarded as soon as it is analysed, and interesting data logged and forwarded to a server-based multi-agent system for examination and dissemination by an engineer.

II. RESEARCH MOTIVATION

The system builds upon the COMMAS COndition Monitoring Multi-Agent System, which has been proven to be able to analyse and UHF diagnostic data, and incorporate output from other monitoring technologies [2]. While such systems are able to provide analysis and decision support to engineers, some deficiencies still exist. As novel sensors are developed and deployed onto plant equipment, a suitable processing and communications platform is required to capture and manage

the increased volume in sensory data. To date, the state-of-the-art in monitoring systems is able to manage data at the substation level, but there have been no publications regarding the management of this data at the device level.

In the case of a centralised condition monitoring system where data is aggregated in a central location, as more sensors are added, both the processing and communications load will rise as more data requires both transmission and analysis. Moving to a fully distributed monitoring system, embedded processing nodes with wireless network capability can host the lowest layers of a multi-agent system and analyse data as soon as it is captured. In doing this, the volume of data can be reduced to such a level that it can be intermittently transferred across a network link. While such systems can be implemented purely upon wired networks, the integration of the state-of-the-art in wireless sensor network technology with existing systems also removes the need for expensive and potentially dangerous cabling within a substation environment.

III. BACKGROUND

Over the past ten years, as sensor network research has gained momentum, so has research into embedded, sensor-network based agents. In terms of agent research, sensor networks are the realisation of the hardware platform required for wide-scale embedded agent deployment.

The term 'Embedded Agent' was first coined by Callaghan et al, in the field of ubiquitous computing and buildings automation. It is defined as an *embedded computer with the inclusion of an autonomous intelligent control entity* [4]. Embedded agent research has also been carried out in other fields, such as military applications and automobile control [5].

An 'Intelligent Agent' is an autonomous software entity that exhibits; 1) reactivity; 2) pro-activeness and; 3) social ability [6]. In terms of software development, reactive agents are relatively easy to craft where agent behaviour occurs in response to some sensory input, but both agent pro-activeness and social ability are more difficult to implement. In a resource constrained environment such as a wireless sensor network, this becomes even more difficult, as agent toolkits such as JADE [7] are not available to provide a framework with which to develop agents.

A. Mobile vs. Static Agents

Mobile Agents are a type of software agent which also include the property of mobility. They are able to traverse a network from host to host as they see fit, to best complete their goals. In terms of wireless sensor networks, it is not feasible for an engineer to visit each node every time code is to be updated, so some kind of code migration mechanism needs to be used. Mobile agents provide this mechanism through code migration; for instance, if an agent is to be replaced on a particular remote node, a new agent can be dispatched from a central location to take the place of the old agent, calling for its deletion on arrival to its new location. Mobile agent technology has previously been applied to power engineering applications by authors such as Wu et al [8] and Kezunovic [9]

Mobile agents for sensor networks has become a popular research area, and to the authors knowledge there are at least four research groups that have published work regarding mobile agent-based sensor networks [10], [11], [12], [13]. The focus of these platforms has been the use of mobile agents to navigate through a sensor network, performing local data analysis on each node before returning to a central location for both code update and data retrieval purposes. Such agents are written in assembly-style code, as they operate within a virtual machine making both algorithm development and code maintenance a non-trivial task. Even though considerable research efforts have been made in the field of mobile agents, their viability as a useful technology is still a hotly debated topic, and implementation of true 'embedded intelligent agents' on sensor networks has yet to be seen.

Without the ability to migrate, agents are confined within the bounds of their host platform. While static agents within a sensor network cannot migrate between nodes, they still have all the inherent advantages of agents. It should be noted that code update within a sensor network is not limited to mobile agents; TinyOS features an in-built code update mechanism called Deluge [14], which allows code to be disseminated throughout a sensor network when required.

As mobility was not a prerequisite for this application, and complex processing techniques were required, it was decided against using an existing mobile agent platform, but instead write a static data processing agent instead.

IV. APPLICATION OVERVIEW

The application processes archive monitoring data from a switched HVDC reactor, with the aim to identify partial discharge emissions using a statistical function. In this case, archived data is used in place of on-line data, which is processed by a wireless sensor node to determine interesting partial discharge events.

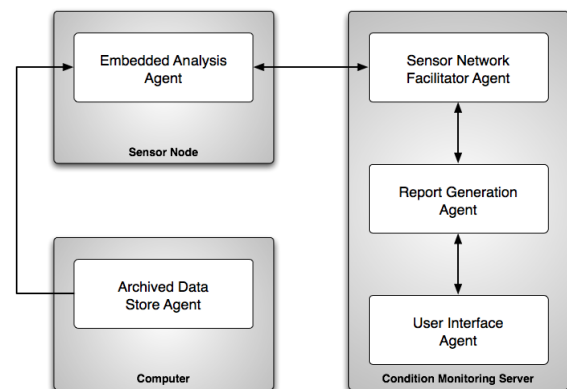


Fig. 1. Block diagram of embedded monitoring system.

As partial discharges are identified by the sensor node, the PD event data is stored in local memory to be forwarded across a wireless link at a later time. The data is forwarded to a multi-agent based condition monitoring system, where it can be further processed and disseminated to engineers. A

block diagram of the overall system architecture can be seen in Figure 1.

This experiment uses archived data, previously captured using 3 matched UHF sensors distributed around the equipment. Data was captured using a portable 4-channel oscilloscope, which was configured to output a 100ns sample containing 500, 16-bit samples. The fourth channel was used to measure the DC waveform, and is unused in the experiment. The data was captured over 6 days in 2005, represented by over 20000 data files, providing a wide sample with which to test the application.

V. SYSTEM ARCHITECTURE

The system architecture is divided into three main sections; 1) The sensor network and data acquisition system; 2) The JADE-based multi-agent system and; 3) the sensor network facilitator agent, which interfaces between sensor nodes and FIPA-based agents. These three components will be discussed in full below.

A. Sensor Node

The sensor node used is a Crossbow MicaZ MPR2400CA 'mote' (see fig. 2). This device features a 7.3 MHz ATmega128L processor, 128KB Program Memory, 4KB Configuration Memory and 512KB of EEPROM for permanent measurement data storage. The device also features a 2.4Ghz IEEE 802.15.4-compliant radio. Programming these devices can be carried out using standard tools which are not limited to any single operating system or architecture.

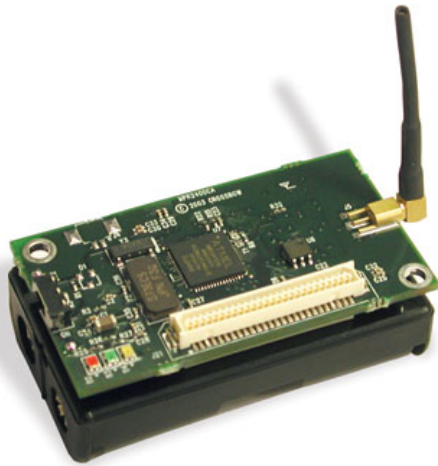


Fig. 2. Crossbow MicaZ Mote, © Crossbow Technology Inc.

To bridge the gap between the sensor hardware platform and its host applications, a sensor network operating system (OS) is used to handle system resources, and abstract the lowest layers of the hardware. TinyOS [15], [16] is such an OS, created at U.C. Berkeley specifically for the 'Berkeley Mote' family of sensor nodes. Since its creation, it has been open-sourced and is developed by a number of different research and industry groups, becoming the 'de-facto' operating system for sensor network research.

The programming language developed for TinyOS, nesC [17], provides an event-driven model with all hardware timers and interrupts delivered as asynchronous events. Each hardware and software component is bundled as a discrete module which is 'wired' together at compile time, allowing minimum execution code to be included in each application, and natural code reuse. There is also a 'flexible concurrency model', allowing multiple application threads to run in parallel, to meet the needs of any executing application. Together, these offer a much simplified development environment compared to traditional microcontroller programming. In addition to using the base TinyOS libraries, the Capsule object storage system [18] is used to allow easy file storage and retrieval. This energy-optimised component allows storage of streams, files, arrays, queues and lists through a simple API.

B. Embedded Data Analysis

Several techniques can be used to manage UHF data, by identifying erroneous data and discarding it. Two such techniques are wave-shape analysis and peak-to-peak voltage identification. Partial discharges are known to have a specific fingerprint in the time-domain (See fig. 3), which allows for differentiation between PD events and other UHF noise. Peak-to-peak voltage can also be used as a simple threshold, where low-voltage events are discarded.

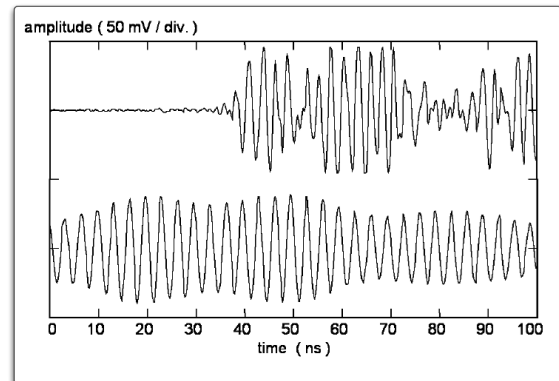


Fig. 3. A partial discharge trace has a distinct inception (top), while a trace from a non-partial discharge event (switching resonance) does not (bottom).

A lightweight analysis application for this platform was previously developed, which processes partial discharge data from a UHF signal acquisition system. The acquisition system used to capture the UHF emissions outputs 4002 byte files, each with 3 channels of data representing three UHF sensors with each channel burst containing 500 samples. Prior to analysing the data, it had to be split into its constituent channels, as a sensor node can only deal with a single channel of data due to the underlying hardware constraints. The host platform has 4KB of RAM, so it is not possible to load an entire 4002 byte burst into memory. If this system were to be put into on-line operation, there are two approaches that can be taken to solving this problem: either a processing node could be used for each channel, or a more powerful node can be used. Either way, this does not present a large engineering problem.

C. Embedded Data Manager Agent

The Embedded Data Manager consists of a number of discrete components, which are analogous to agent behaviours (see fig. 4). This embedded application can be considered an 'embedded software agent', as it runs autonomously with the goal of identifying partial discharges in the data it receives, although it does not yet exhibit planning or reasoning capabilities.

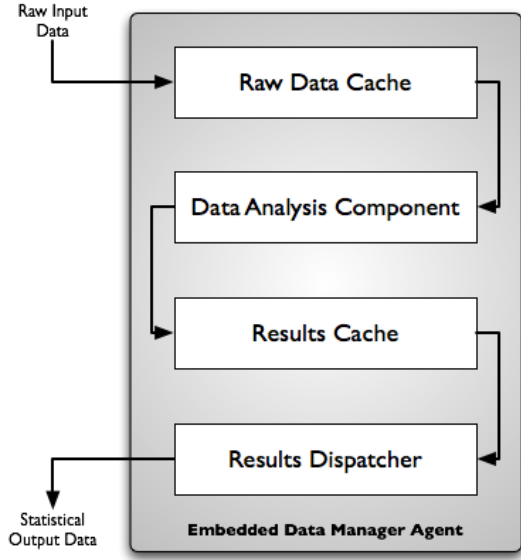


Fig. 4. Block Diagram showing Embedded Data Manager Agent Components with data flow.

1) *Raw Data Cache Component*: As the processing agent receives a 500-sample event snapshot, the raw data is stored within the raw data cache component. This component uses a 'Queue' storage structure, provided by the Capsule module. The queue size is set at compile-time, as TinyOS does not currently have support for dynamic memory allocation. This component operation is simple: as a data file is received by the sensor node it is pushed onto the queue for temporary storage, and as processing is completed, the file is removed from the end of the queue, freeing up space for other files.

By wrapping the data cache around the Queue component, it becomes easy to substitute a different data structure if required. If the system was to use an 'Array' rather than a 'Queue', indexed access to data files would be available, allowing any 'interesting' raw data to be stored if required.

2) *Data Analysis Component*: As new files become available on the node, they are dispatched to the data analysis component for partial discharge identification. The algorithm used to analyse the data was previously written to run within a JADE-based agent, and was re-written from Java, into nesC. As the incoming data is in integer format and the algorithm does not calculate any complex statistical features, fixed point arithmetic is used removing the need for any optimisations specific to the embedded platform.

The analysis technique identifies partial discharges by comparing the energy in the first 50 samples to the energy in the last 450 samples as follows:

$$\frac{\sum_{i=0}^{49} x_i^2}{\sum_{i=50}^{499} x_i^2} \geq \frac{1}{20}$$

where x_i is the i th sample. In this case, if the ratio of power between the beginning and end of the sample is greater than 1/20, a partial discharge is identified and a flag is set which triggers the results cache component.

3) *Results Cache Component*: Once data has been analysed and a partial discharge found, the summary data from the analysis is stored within the results cache component. The cache component works in a similar way to the data manager, where it can load, create and delete entries in permanent memory using the Capsule module. In this instance, an indexed array is used to allow access to specific event data, rather than a queue where only the first and last items can be accessed.

There are two types of results stored by the results cache. Firstly, each partial discharge will register a record with its timestamp and maximum peak-to-peak voltage. Secondly, the date and value of the maximum peak to peak voltage is stored, plus the total sum of all peak to peak voltages, giving an indication of the total partial discharge power emitted. By storing summary information on the health of the equipment, it is possible to provide an approximate level of health without providing exact details of individual partial discharge events. The results cache is designed to be a cyclic buffer, with the oldest data being overwritten each time the buffer fills.

4) *Results Dispatcher*: Wireless communication between sensor nodes is the largest consumer of battery power, and for this reason, the less data that is sent across a wireless link the better. The results dispatcher is designed to send data back to a basestation, either at predefined intervals, or after a certain number of events have been registered.

D. Sensor Network Facilitator Agent

One of the key advantages of multi-agent systems is their flexibility and extensibility. The sensor network facilitator agent allows multi-agent systems to extend their reach out onto sensor networks, by providing an interface between sensor nodes and FIPA-compliant agents. FIPA-based agents use an "ontology", which defines the data model and the inter-agent communication vocabulary, a "communications language", such as FIPA ACL [19], which defines speech act-theory based communications protocols, and a "content language", such as FIPA SL [20], which defines the semantics of each message payload. These elements combined, create a rich messaging platform for inter-agent communication. Unfortunately, it is not feasible to use all of these components with the current generation of sensor network nodes, as the overhead introduced in each message is too large. FIPA-based messages would quickly fill the device memory, not taking into account the resources required to generate, parse and process such messages.

For this reason, a sensor network facilitator agent is required to translate lightweight sensor network messages to a messaging format understood by FIPA-based agents. This allows resource-constrained sensor network applications to run

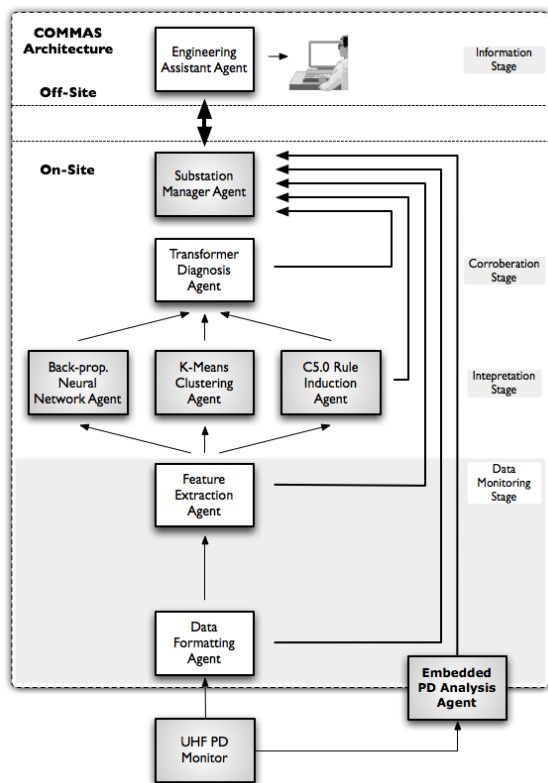


Fig. 5. The COMMAS architecture, with Embedded PD Analysis Agent.

without large communications libraries, but still communicate with intelligent agents which process and disseminate sensory data. This type of agent has previously been developed, albeit for mobile agent frameworks [21]. This facilitator agent provides incoming messages from the sensor network to any agents that have subscribed to the service. This allows multiple agents to receive the incoming sensory data as soon as it becomes available.

It should also be noted that the sensor network facilitator may also be configured to interoperate with IEC 61850 applications. By including the IEC 61850 data model in the agent ontology, wireless sensor nodes may be addressed as IEC 61850 logical nodes, furthering the interoperability between systems.

VI. FURTHER WORK

In the future, more elements of the COMMAS architecture will be moved down onto the sensor network level. Figure 5 shows the COMMAS architecture for UHF partial discharge activity monitoring, with an additional 'Embedded PD Analysis Agent' sitting on the edge of the system. This demonstrates how the new embedded agent would integrate with the existing system, between the PD monitor and the software-based multi-agent system.

The addition of such components will allow embedded data interpretation and corroboration to take place sooner, further reducing the required bandwidth and amount of monitoring data passed up to engineers. It is also planned to integrate other novel sensors into this architecture, and investigate the

feasibility of further miniaturisation of the UHF data capture process.

VII. CONCLUSIONS

This paper has demonstrated that components from the data layer of condition monitoring systems can be moved down into an embedded environment. This research has shown that the addition of sensor network technology to condition monitoring applications is an important step in the evolution of such systems.

The current architecture has been shown to offload some of the data analysis to an embedded agent, albeit a simple reactive agent with no reasoning or planning capabilities. To extend this particular application, an embedded agent will be used for each channel, with a federator agent used to consolidate their results in-situ.

Subsequent versions of this architecture will focus on creating a rich platform for embedded intelligent condition monitoring agents, by further integrating sensor networks with existing CM systems and developing embedded agents with pro-active and social abilities.

REFERENCES

- [1] Y. Han and Y. H. Song, "Condition monitoring techniques for electrical equipment—a literature survey," *IEEE Transactions on Power Delivery*, Vol. 18, No. 1, pp. 4–13, January 2003.
- [2] S. D. J. McArthur, S. M. Strachan, and G. Jahn, "The design of a multi-agent transformer condition monitoring system," *IEEE Transactions on Power Systems*, Vol. 19, No. 4, pp. 1845–1852, November 2004.
- [3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, Volume 38, Issue 4, pp. 393–422, 2002.
- [4] V. Callaghan, G. Clarke, A. Pounds-Cornish, and S. Sharples, "Buildings as Intelligent Autonomous Systems: A Model for Integrating Personal and Building Agents," *The 6th International Conference on Intelligent Autonomous Systems (IAS-6)*, Venice, Italy, 2000.
- [5] M. Huhns, "Networking embedded agents," *Internet Computing*, IEEE, vol. 3, no. 1, pp. 91–93, 1999.
- [6] M. Wooldridge, *Intelligent Agents*, G. Weiss, Ed. MIT Press Cambridge, MA, USA, 1999.
- [7] "Jade - Java Agent Development Environment," 2007. [Online]. Available: <http://jade.tilab.com>
- [8] D. P. Buse, P. Sun, Q. H. Wu, and J. Fitch, "Agent-based substation automation," *Power and Energy Magazine*, IEEE, vol. 1, no. 2, pp. 50–55, 2003.
- [9] A. Kezunovic, X. Xu, and D. Won, "Improving circuit breaker maintenance management tasks by applying mobile agent software technology," *Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, vol. 2, 2002.
- [10] H. Qi, S. Iyengar, and K. Chakrabarty, "Distributed multiresolution data integration using mobile agents," in *IEEE Aerospace Conference*, 2001.
- [11] L. Szumel, J. LeBrun, and J. D. Owens, "Towards a mobile agent framework for sensor networks," in *Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, May 2005, pp. 79–87.
- [12] C.-L. Fok, G.-C. Roman, and C. Lu, "Rapid development and flexible deployment of adaptive wireless sensor network applications," in *24th International Conference on Distributed Computing Systems (ICDCS'05)*, 2005.
- [13] D. Georgoulas and K. Blow, "Making motes intelligent: An agent-based approach to wireless sensor networks," *WSEAS on Communications Journal*, pp. 515–212, 2006.
- [14] J. W. Hui and D. E. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM Press, 2004, pp. 81–94.
- [15] "TinyOS," 2007. [Online]. Available: <http://www.tinyos.net>

- [16] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System architecture directions for networked sensors," in *Architectural Support for Programming Languages and Operating Systems*, 2000, pp. 93–104.
- [17] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. E. Culler, "The nesC language: A holistic approach to networked embedded systems," *Proc. of the ACM SIGPLAN*, 2003.
- [18] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, "Capsule: an energy-optimized object storage system for memory-constrained sensor devices," *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 195–208, 2006.
- [19] Foundation for Intelligent Physical Agents, "FIPA ACL Message Structure Specification," 2002, available from <http://fipa.org/specs/fipa00061>.
- [20] —, "FIPA SL Content Language Specification," 2002, available from <http://fipa.org/specs/fipa00008>.
- [21] J. Herbert and C.-L. Fok., "Mobile Agent Architecture Integration for a Wireless Sensor Medical Application," *Web Intelligence and International Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on*, pp. 235–238, 2006.



Pete Baker is a Research Student within the Institute for Energy and Environment, at the University of Strathclyde in Glasgow, Scotland, UK. He received his B.Eng. (Hons) Degree from the University of Strathclyde in 2004. His research interests include wireless sensor networks, systems integration and multi-agent systems for condition monitoring.



Dr Stephen McArthur (M93, SM07) received his B.Eng. (Hons) and PhD degrees from the University of Strathclyde in 1992 and 1996 respectively. He is a Reader in the Institute for Energy and Environment, within the Department of Electronic and Electrical Engineering. He is Director of the British Energy Advanced Diagnostics Centre and Co-director of the Strathclyde Centre for Applied Intelligent Systems. His research interests include intelligent system applications in power, condition monitoring, fault diagnosis and multi-agent systems.



Dr Martin Judd received his BSc in Telecommunications from the University of Hull and PhD in Electrical Engineering from the University of Strathclyde in 1985 and 1996 respectively. He is a Senior Lecturer in the Institute for Energy and Environment at the University of Strathclyde. His research interests include electromagnetic theory, propagation and measurement and high frequency diagnostic techniques for power plant.