Multi-Agent System For Nuclear Condition Monitoring

Christopher J. Wallace
Institute for Energy And Environment
University Of Strathclyde
Glasgow, UK
cwallace@eee.strath.ac.uk

Gordon J. Jahn
Institute for Energy And Environment
University Of Strathclyde
Glasgow, UK
gjahn@eee.strath.ac.uk

Stephen D.J. McArthur
Institute for Energy And Environment
University Of Strathclyde
Glasgow, UK
smcarthur@eee.strath.ac.uk

ABSTRACT

This paper presents experience of Multi-Agent Systems (MAS) within the nuclear generation domain, with a particular emphasis on Condition monitoring (CM). A prototype system, which used a MAS for the management of CM observations and the generation of a user interface, was developed however it was decided that this application lacked the required speed and efficiency. Extending the developed ontology, an alternative MAS system was developed to perform CM analyses and storage and management of data, without the user directly interacting with the MAS.

The paper discusses the development of both systems, relating key features of MAS which meet the particular needs of CM applications, but also highlights remaining issues, such as a lack of an existing ontological framework and the problem of data security, that are particularly prominent in the nuclear domain.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous

General Terms
Design, Reliability, Security

Keywords
condition monitoring, nuclear, generation

1. INTRODUCTION

The use of Intelligent Agents for condition monitoring has been shown for a variety of applications in power systems [7][13][15][10][12] due to the flexible and distributed nature of Multi-Agent Systems (MAS). As lifetime extensions are granted to nuclear power stations around the world, there has been an increased interest in the implementation of Condition Monitoring (CM) to satisfy regulatory bodies and supplement knowledge of reactor state between statutory outages and inspections.

This paper describes the development of MAS for addressing nuclear condition monitoring challenges and considers the requirements, benefits and challenges that arise from the use of agent technologies for continuous offline monitoring of the cores of the UK’s Advanced Gas-cooled Reactors (AGR).

1.1 Nuclear Agents

Previous work on intelligent agents in the nuclear industry has largely considered the use of agents for the automated control of future power stations [18][9] and for online monitoring of systems related to the reactor [19]. The application described in this paper however, is designed for engineering support in order to detect anomalous behavior in reactor cores before there are safety concerns or adverse impacts on operations. The use of a MAS to achieve this, it will be argued, is a way for agents to prove themselves in safety critical applications, while the MAS platform allows for the addition of further AI analysis techniques encapsulated as agents.

There are, however, key practical challenges associated with the adoption of new technologies such as MAS within the nuclear engineering domain. These are addressed in section 7 before the paper concludes that whilst condition monitoring in the nuclear domain with MAS is possible and beneficial, there remain challenges to be overcome.

2. NUCLEAR CONDITION MONITORING

The continued operation of nuclear power plants beyond their original design lives is dependent on the ability of the operator to demonstrate the structural integrity of critical components. In the UK, the graphite cores of AGRs are the principal life limiting components [17] and are inspected every three years. To supplement these inspections, a condition monitoring strategy has been introduced at some stations, the results of which are considered at quarterly Monitoring Assessment Panel (MAP) meetings, where the observations are graded and recorded. There are a number of analyses associated with key core components, which form the basis of the MAP discussion. These analyses generally involve the extraction of a data set prior to the MAP, analysis of the data by a suitably qualified engineer and presentation of results at the MAP for consideration by a panel of experts. These experts then consider the data presented with a view to identifying temporal and spatial colocation that might indicate a root-cause issue with the reactor core itself.

The initial requirement for reactor core CM within the AGR fleet was a solution to the problem of the management of CM analysis results and observations. The nature of slightly different hardware configurations at different sta-
stations requires that the approach be flexible to allow different analyses to be undertaken.

3. MAPS AND IMAPS

The MAP process was trialed at two power stations from 2005, and is a formal process for collating and analyzing data. As a new process, there were no supporting systems and it was recognized that there was the opportunity to trial MAS within this environment. This project was known as IMAPS, the Intelligent Monitoring Assessment Panel System, which delivered an agent-based platform for the storage and analysis of AGR observations.

The primary means of supporting the MAP process is by allowing observations of different types to be recorded and plotted on a graphical plan view of the reactor, originally using a whiteboard. The simplicity of this approach for allowing the identification of spatially colocated events and the easily understandable relationship between the actual core and data plotted on such a view meant that this basic interface was carried forward into the MAS.

As the system was being developed alongside the formal station-based process, there was also a need to allow existing manual interpretation to be included within the system as soon as possible, allowing data to be recorded and analyzed as soon as possible. The initial prototype of this system involved the creation of an archiving agent along with the user agent. This met the initial brief of providing a working system for storing the data and presenting this back to engineers.

The high-level block diagram of the system envisaged is shown in figure 1.

Figure 1: IMAPS High-Level Block Diagram

Due to the specific requirements in this case, the final delivery to end-users was to be provided through a web-browser that posed the challenge of allowing asynchronous user behavior to interact with an agent with the MAS that would be regularly communicating with other agents.

The approach taken to solve this problem was to implement a MAS based on the Foundation for Intelligent Physical Agents (FIPA) standards. This was carried out using the Java Agent DEvelopment framework (JADE) [2] and an agent for each user was executed within a web server; this user agent provided an interface between user inputs provided via a web browser and the agent based system and also with the rendering of information back to the user.

The initial system contained simply the user agents and the archive agent which provided a translation between FIPA Semantic Language (SL) and the Structured Query Language (SQL) used by the database fulfilling the fact storage requirement.

An issue that arose with this technique was of the amount of data requiring to be presented to users. The screenshot in figure 2 shows a typical rendering for a single reactor. As each station has two reactors, the normal case is to present two reactors simultaneously.

Each reactor plan shown has around 400 channels including larger fuel channels and smaller interstitial channels used for control and other purposes. This means that at any given time, the system is rendering around 800 data points. Later iterations of the system also used the border color to represent the presence of inspection information taking each view as above to representing around 1,600 data points.

Figure 2: Sample data rendering from IMAPS. Two cores per station are available to the user.

IMAPS [11] has a relatively small domain model to contend with. There is a type hierarchy associated with the various observation types and other concepts must be defined to support the reactors, the station and the different grades that can be assigned to the observations. A base ontology was sought for this application, but no suitable candidate ontology was found and, instead, one was developed specifically for the application. The frames-based ontology shown in figure 3 was used for the initial testing of the system.

Following experimentation with this system, the prototype structure was found to be unable to meet response time expectations. Upon investigation, this was found to be due to the number of data movements having to occur in order that the information be rendered to the user. The following steps were taking place:

- The user agent parses the request from the web browser
- The user agent translates the parsed request into a FIPA SL query
Figure 3: Initial IMAPS ontology, with two zoomed examples of concepts and how they relate to concepts by predicates such as \textit{isa}.

- The user agent envelopes the FIPA SL query and sends it to the archive agent.
- The archive agent parses the SL query translating the content and predicates into an SQL query.
- The archive agent executes the SQL query and recreates abstract objects.
- The abstract objects are used to render an SL response to the query.
- The SL response is enveloped and sent back to the user agent.
- The user agent parses the data back into objects for use by the renderer.

Due to the structure of the messaging within the agent, this process involves repeated handling of the same data in slightly different forms. This both increases the use of system resources and slows down the data handling. Whilst the advantages of this approach tend to outweigh the disadvantages by making it available in an agent-based society, the fact that this use case is so important to the end-user experience meant that further investigation was warranted.

Analysis revealed that the agent-based approach to retrieving the data was taking around seven times longer than directly retrieving this data from the database. This problem was ultimately caused by JADE operating on very long strings whilst building the SL response and enveloping the message.

In order to meet the design goal, the agent interaction was removed from the process with the server directly parsing the query, translating to SQL and executing in the database before streaming results – without ever storing them – direct to the client. Render times in this case were brought to around 300msec for the same data on the same hardware platform. Whilst this meant that the prototype was now without the multi-agent design in terms of retrieving data, it was retained for data logging and this hybrid approach was essential to allow the research and development of further analyses.

4. EXTENDING IMAPS INTO ANALYSIS

In order to consider how the prototype system described in the previous section can be extended to provide and support routine analyses, the requirements for the analysis processes are considered along with the changes that are then required in the IMAPS MAS.

4.1 Requirements

Existing CM of the AGR fleet is based on a model of extracting data or information from reactor systems and data stores as required, in order to perform analyses or explain anomalous analysis results according to a monitoring schedule. By delegating tasks and analyses currently performed by engineers to intelligent agents, it is possible to automate the collection, analysis, correlation and verification of data as it becomes available.

This design of CM system allows for faster completion of analyses and requires considerably less engineer time, and allows more time for investigation of non-trivial anomalies, which can then be considered in more detail prior to the MAP meeting.
The current CM regime manually records anomalous observations for particular components in the core, however a more comprehensive monitoring system could monitor the data for all components for further analysis and trending of anomalies not currently detected by existing monitoring analyses, or results dismissed by current analysis limits that may be changed later to meet an enhanced safety case.

Finally, the monitoring of AGR cores is still a relatively young process, compared to the existing inspection protocols, and as such it is essential that a CM system be flexible enough to include additional data sources and further analyses as required. The use of a MAS with a carefully defined ontology allows for this extensibility by encapsulating any relevant data interface or analysis within an agent that utilizes the ontology. This provides a platform within which Artificial Intelligence (AI) techniques can be deployed to perform further analyses with very minimal impact on the existing system. Furthermore, such analyses may extend the current work performed by engineers by not only considering known causal relationships between events, but also statistical relationships using clustering or structural health monitoring.

4.2 Ontology and Development

Based on the success of previous MAS developed for power applications\[5\], and the original prototype agents and ontology developed for IMAPS along with the maturity of the platform, JADE was selected for further development.

The ontology developed for the initial version of IMAPS provides the basis of a nuclear ontology for this work, however since the initial emphasis of the project was the management of CM observations, expansion of the ontology was required in order to facilitate the management, storage and analysis of reactor data. The ontology was created and developed using the Prot´eg´e ontology editor\[3\] and a plugin which allowed the automatic generation of Java code, an example of which is shown in figure 4. Each concept in the CM system is therefore represented by a Java class and takes advantage of object oriented concepts such as inheritance, resulting in a class structure of the same form as the ontology.

The Prot´eg´e bean generator used has also been extended to support the Java Persistence API (JPA), allowing it to add annotations which are included in the generated code and are used in mapping of the ontology to a database for storage and retrieval of data and observations.

4.3 Storage

The archive agent is essentially a small messaging and translation layer added to a standard SQL database system providing storage. For database storage, an object relational mapping library called Hibernate is used, which maps Java classes (generated from Prot´eg´e) directly onto an SQL database. Data and observations that are created by the MAS and stored in the database can then be retrieved as Java objects, which allows for simplified querying using the Hibernate Query Language (HQL) which allows object oriented queries against the methods available in the objects stored. For example, the query select r from Reactors r returns all instances of the Reactor fact in the database, along with providing access to related objects.

This format is significantly simpler than a typical SQL query which would depend on the table structure of the database and may involve complex operations to retrieve the same data. The use of object relational mapping therefore adds portability to the system and with only slight modification, the database can be changed to any supported SQL database.

Coupled to an SL translator that can understand arbitrary SL queries and map predicates onto SQL queries, this approach supports other agents presenting the archive agent with any valid SL for processing.

4.4 Design Methodology

The design process of MAS has been discussed in the literature\[20\], and in particular for power systems\[14\]. The initial methodology for the MAS design roughly followed the same general path as McArthur et al\[14\] and several aspects of the Gaia methodology\[21\], in particular for the development of the Service Model, which defines the roles for which agents are created and the capabilities they can offer the system. The emphasis on defining the services offered by the agents reflects the desire of the system, initially at least, to automate the tasks and procedures already defined and performed by engineers. These roles were defined formally using the structure of the Gaia services model, an example of which is shown in figure 5.

The approach of using this service model to map between tasks and agents within the MAS is now demonstrated through a brief case study showing how a simple analysis has been implemented.

5. CASE STUDY: CHANNEL POWERS
An example of an existing analysis which can be conveniently performed by an agent based system, is the regular comparison of channel power models, which quantifies and attempts to explain differences between thermal and neutronic models of channel power\[8\]. Excessive deviations between the thermal and neutronic models of channel power could potentially indicate inadequate cooling of fuel, however these can far more commonly be attributed to innocuous local events such as refueling or other normal reactor operations that affect the flux profile across the core.

Figure 6: An example of an engineers monitoring procedure mapped onto a set of agents.

Using a set of pre-defined limits, the power discrepancies are graded initially, and then an engineer is tasked with determining whether any reactor operations, such as refueling, adequately explain the discrepancy. Using a looser set of limits, the engineer then re-grades the discrepancies as appropriate using any available information.

5.1 Agent Approach

Mapping the procedure followed by the engineer to a set of algorithms and with the appropriate knowledge of available data sources, a set of agents as shown in figure 6:

- Parses the data (Data Collection Agent)
- Stores the data (Database Agent) in a database in a form consistent with the ontology.
- A notification is sent to the Channel Power analysis Agent
- The Channel Power analysis Agent retrieves the data
- The Channel Power analysis Agent sends queries to available data source interface agents (the IMAPS Agent in this case) for any data or observations within a defined spatial or temporal range of each fuel channel power discrepancy for which it analyzes.
- After correlating all possible events with the data, the gradings are calculated and then stored in the database for verification by the engineer.

Other data sources may be utilized in this particular type of analysis, the availability and nature of which may vary between stations, however provided the analysis agent is aware of relevant data sources and how the data from each should be interpreted, an arbitrary number can be added.

6. FUTURE FUNCTIONALITY

The future development of the system will include the development of agents to perform other MAP analyses as well as an investigation of other design approaches to the AGR CM requirements. The design presented in this paper is very task focused, with the agents designed around the required functionality already provided by the engineer. It can be argued however that this approach is inherently limited by attempting to emulate the engineer, as it places very little emphasis on the autonomy of the agents (outside of the prescriptive CM they perform) and by extension any emergent behavior that could potentially enhance the CM regime.

6.1 Component Monitoring

For this reason, recent work has begun investigating the viability creating agents for major components in the core (fuel channels, control channels, coolant pumps etc) that can be delegated the responsibility of managing their own data and analyses. Though this would involve the creation of around 300-400 agents per reactor core, it has been shown\[6\] that the JADE platform scales quite well to systems of several hundred agents. It is likely in the interests of performance however that multiple agent containers would be used, for example an analysis container on one machine that performs numerically intensive algorithms and component agents in a separate container.

These agents would be able to request appropriate analyses as data becomes available to them, keep snapshots of the most recent states of the component and track changes of various parameters. They would also have the ability to share this data with other components, and in certain circumstances, such as an anomalous result in an analysis, suggest to other component agents particular analyses that they might wish to perform. This rolling model of CM would allow for greater depth of analysis of parameters for which there are often secondary causes and effects, for which the availability of data is not synchronized. The system would of course still be required to perform at least the minimum number of analyses required for the MAP.

Individual component health monitoring will become more important as the core ages and a more detailed view of the state of groups of components or the whole core is required.

It is envisaged that this may present itself as an agent hierarchy with different agent societies responsible depending on the component grouping. This could manifest itself as a layered architecture as shown in figure 7, substituting layered analyses into the original vision outlined in figure 1.

This method of performing analyses at different levels of component grouping should allow for both mirroring the manner in which analyses are carried out by experts – increasing trust and transparency within the system – but should also allow for competing theories regarding localized and global hypotheses for each reactor. These can be presented and assessed by the reactor agents to present a consolidated case, with a risk profile, enabling the engineer to make an informed judgment based on the outputs of the MAS.

7. CHALLENGES

The use of ontologies is intended to allow for uniformity of development and communication, however the question of how to manage revisions of ontologies or the interaction of different ontologies\[4\] is a problem to which there is not a
analyses

Reactor

Station Hardware

User Agent

Archive Agent

Fact storage

Engineers, MAP Members, etc.

Fuel Grab Load

Agent

Trace Agent

1

Quadrant

4

Quadrant

analyses

.  .  .

Reactor

1

Reactor

4

analyses

Channel

analyses

Figure 7: Proposed High-Level Architecture

well defined solution. The IEEE MAS working group[1] has already proposed the adoption of a common upper ontology for MAS in the field of power engineering, and though ontologies are application and domain specific, it remains unclear as to whether a nuclear ontology merits development, as no appropriate re-usable ontology was available during the course of this work.

Each reactor data set or CM observation object that the MAS handles, uniquely describes some fact, however there is a significant overhead in checking the data store each time a new fact is recorded to check whether the fact is already stored. Similarly, where datasets overlap in time, the archiving agent is required to determine how much duplicate data exists and how to store the new data set. How to achieve this without adding significant overhead to the processing of data remains an issue.

7.1 Nuclear Domain Challenges

The particular importance of safety in a Nuclear Power Station means that deployed technologies are required to be proven and their impact well understood. This results in a relatively slow adoption of new technologies, so while a MAS may be in principle capable of performing some of the tasks of an engineer, it is likely that any initial deployment of a nuclear CM MAS will be very much in a supporting role and will only be used for verification of the existing engineers analysis. It will also likely be held separately from the main IT infrastructure with several well defined data interfaces for the purposes of verifying data integrity and security.

The need to protect the operational plant and associated sensor and control systems from any interference by unauthorized parties has historically been met by complete segregation of operational and corporate networks. Whilst this ensures malware and control signals cannot pass from, for example, Internet e-mail to the plant, it also reduces interoperability within the plant and results in increased difficulties in accessing online data for analysis.

With the implementation of CM across fleets of nuclear power stations, the question of data security has become increasingly prevalent as operators endeavor to provide access to relevant data while maintaining the integrity of their IT security infrastructure. MAS by their very nature depend on the availability of network access to facilitate communication between agent containers on different remote machines. For the purposes of the work described in this paper, the test system was not deployed and the collection of data was simulated through access to remote systems (such as IMAPS), databases and remote file shares. In an operational environment, security is of major importance and access to data will be much more stringent. One possible solution involves the definition of information security zones with the plant and using “data diodes” that often claim to provide 100% one way data passage. As explained in [16] however, such devices are far from a “silver bullet” due to the inability of the receiver to provide feedback, such as acknowledgements, to the sender and this could result in missing data.

8. CONCLUSIONS

This paper has described the application of MAS to the condition monitoring of a nuclear reactor. Initial work implemented an agent based system for the management of condition monitoring observations, however it was found that using agents to generate a user interface was inefficient and caused excessive delays in updating the display. An alternative use of MAS was developed, by extending the existing ontology, to perform time consuming analyses currently performed by engineers. This system can provide a consistent structure for the addition of many analyses and distributed data sources, which are common features of the monitoring of older nuclear reactors. A MAS was found to be a good choice of platform for this problem, however some issues associated with the application of an agent based system in the nuclear domain remain, particularly the lack of an existing ontological framework and issues associated with data security.

9. ACKNOWLEDGMENTS

This work was funded by EDF Energy, the views presented here do not necessarily represent the views of EDF Energy.

10. REFERENCES


