Mobile Agent Architecture Integration for a Wireless Sensor Medical Application

John Herbert, John O'Donoghue, Gao Ling,
Kai Fei
Computer Science Department,
University College Cork.
{j.herbert, j.odonoghue}@cs.ucc.ie

Chien-Liang Fok
Dept. of Computer Science and Engineering
Washington University in St. Louis
Saint Louis, Missouri 63130
liang@cse.wustl.edu

Abstract

Wireless sensor nodes are used to monitor patient vital signs in a medical application. To ensure proper patient care is provided, real-time patient data must be managed correctly in the context of relevant patient information and medical knowledge. The Data Management System (DMS) is an agent-based architecture that aims to provide flexible, effective data management within a Wireless Patient Sensor Network (WPSN). The DMS is built primarily on the sophisticated JADE agent platform. JADE runs on resource-rich platforms such as servers, PCs, PDAs and high-end mobile phones. The lightweight Agilla agent platform can run on resource constrained sensor nodes.

An integrated mobile agent based architecture combining Jade and Agilla is presented. This makes best use of the more sophisticated agent platform for high-level functionality and the lighter agent middleware for low-level sensor data collection. The resulting system is a unified agent architecture that runs on heterogeneous platforms on a wireless network.

1. Introduction

Current and future pervasive medical environments may produce large volumes of patient datasets. The DMS [1] deals with large volumes of static data (such as patient records in databases) and dynamic data (including patient vital sign readings). Software agents are employed within the DMS to manage and correlate multiple input/output streams of static and dynamic patient datasets. Software agent middleware is ideally suited to manage such data because of the inherent autonomy and mobility of agents.

The DMS architecture is built primarily on JADE [2], a FIPA (IEEE standard) compliant agent framework which is designed to execute on relatively resource rich devices. Wireless sensor nodes are constrained by severe resource limitations and unreliable network connectivity. Therefore if an agent environment is to be used on a wireless sensor node it will need to operate on very limited resources. The Agilla [3],[4] agent platform is an example of such a platform.

1.1 Hardware platform

The wireless patient sensor node is built upon a Tyndall25 [5] hardware platform. This is a stackable system, consisting of 25mm x 25mm layers which may be combined in a plug and play fashion. The patient sensor node is comprised of three layers: power; communication and processing; and the sensor interface. The sensor interface layer may be connected to three ECG patches, one blood pressure and one pulse rate sensor.

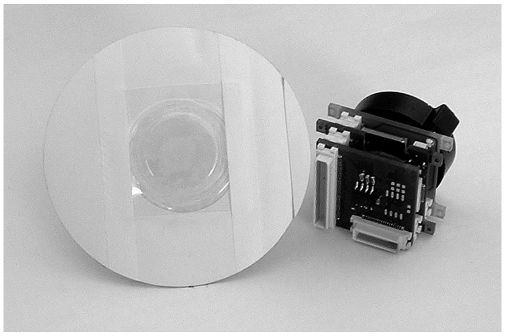

Figure 1. ECG Sensor and the Tyndall25 mote.

1.2 Agilla

The Agilla middleware executes on TinyOS (an event based operating system designed for wireless sensor networks [6]).

Agilla was initially developed for the Mica mote [7]; it has now been successfully ported to the Tyndall25.

Agilla provides the functionality for data processing, communication and supports agent mobility. Agilla provides two types of agent migration: weak migration and strong migration. With weak migration only the code is transferred. For strong migration everything is relocated (heap, program counter, and stack) and the agent resumes execution from its previous node state. An appropriate migration method may be used to respond to dynamic context requirements (patient status, location) thus providing an effective run time solution.

2. DMS Architecture

The DMS architecture within the medical domain has many components dealing with data gathering, data correlation of static and dynamic patient datasets, and presenting appropriate data to end-users. Important components include: data collection; data integration; data analysis; and data presentation.

Data Collection

This involves gathering data such as: readings from wireless patient sensors; data input by staff through their PDAs and PCs; patient records held on servers.

Data Integration

The DMS must dynamically manage and integrate static and dynamic medical data in a context and situation aware manner.

Data Analysis

The DMS generates value-added information based on the data and the meaning of this data in relation to its previous and current context. An ontology model is built using Protégé. Expert system rules are applied (using Jess [8]) to manage the context variables. The aim of this component is to intelligently analyse the large amount of application data. This provides the basis for improved diagnosis at the patient point of care.

Data Presentation

Data is filtered and, depending on the current context aware variables, relevant patient medical data is sent to the end user.

2.1 Use of Agilla in the DMS

Agent platforms have been applied in a number of medical environments [9] and [10]. The integration of software agents is encouraged by the prediction expressed in [11] “that mobile agents capable of discovering, extracting, interpreting and validating context will make significant contribution to increasing efficiency, flexibility and feasibility of pervasive computing systems*”*.

Most of the components within the DMS are implemented with the more sophisticated JADE agent platform. It is clear that Agilla is appropriate for the important task of data collection from the wireless patient sensor nodes. The basic function of the data collection agent is to retrieve sensor readings from the wireless patient sensor nodes. One may configure the agent to take readings at regular intervals. However during patient monitoring the rate at which these readings are taken may need to be altered. For example, the patient's condition may change or medical staff may make a reading request. The agent paradigm is ideally suited to the reactive/proactive nature of patient monitoring. Therefore extending the agent architecture to the actual patient sensor nodes is essential for a comprehensive solution.

4. Integration of Agent Domains

The concept of integrating a pure agent platform with Agilla was first introduced in [1], with work also reported in [12]. Implementation of a JADE-Agilla integration must fulfil the following minimum requirements: communication via an interface; agent injection from resource rich JADE to the restricted Agilla environment; migration of agents in the sensor network; retrieval of sensor readings by Agilla agents and return of results to the main JADE environment.

JADE to Agilla interaction is achieved via method calls defined in a generic interface (cf. Figure 2). This provides a straightforward way of injecting Agilla agents. While arbitrary Agilla code can be injected into the wireless sensor network it is useful to have a number of Agilla agents to perform standard tasks such as retrieving sensor readings at regular intervals, or retrieving sensor data in response to an asynchronous request.

Combining agent injection with a library of useful Agilla agents extends the agent architecture to the wireless sensor nodes in a seamless manner. A library of the most common functions has been created. Examples include functions to retrieve patient data periodically and to take an immediate blood pressure reading.

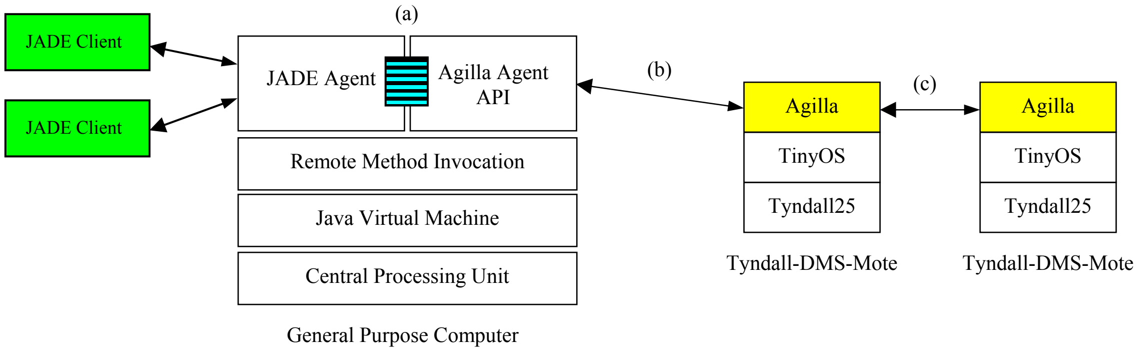

Figure 2. Overview of JADE-Agilla Agent Integration. JADE on a PC interacting with Agilla agents on a Tyndall25 patient sensor node (a) Generic interface, (b) Agent injection. (c) Agent Migration.

4.1 Examples of the JADE-Agilla Integration

1) Agilla agent injection through JADE

```
/*JADE Java Class for injecting an Agilla agent*/
public AgentInjector(String source,
                     boolean connect, int col,
                     boolean createGUI,
                     boolean debug) throws Exception
```

2) Agilla migration

Agilla agents may propagate throughout the sensor network. For example, an Agilla agent may travel through the network, collecting specific patient sensor readings. When it returns to the base station node it may perform some rudimentary analysis and report back to the JADE agent.

```
//Agilla code for agent propagation
//Move Agilla agent to node (0,0)
Pushloc 0 0
wmove //A weak version of smove

//Move Agilla agent to node (1,1)
Pushloc 1 1
wmove  //A weak version of smove
```

3) Agilla sensor reading

The basic function of the Agilla agent is to retrieve sensor readings from the wireless patient sensor nodes. Different situations may require different sensor acquisition strategies.

```
//Define sensor type(s)
BloodPressure = 1
ECG  = 2

//Agilla code for sensor sampling
//Reading a specific sensor
Push BloodPressure
sense //Take a BloodPressure sensor reading
```

5. Evaluation

The results of two experiments are presented to indicate the performance of the JADE-Agilla integration (cf. Figures 3 and 4). In experiment one a JADE client (JC1) requests a sensor value to be sampled once every second from AC1 (cf. Figure 3). This experiment was executed in isolation. In experiment two JC1 requests a sensor value once every second from AC1, JC2 from AC2, and so on. Results for 20 interactions indicate the system performance and the difference between the two experiments (cf. Figure 4).

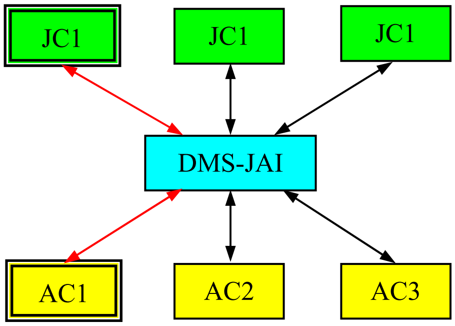

Figure 3. (a) Single JADE client (JC1) and single Agilla client (AC1) only. (b) Multiple JADE clients (J1-J3) and multiple Agilla clients (A1-A3) interacting simultaneously.

It should be noted that a number of factors affect the performance and its variability. In particular separation and orientation affects the wireless communication between nodes. Variability is introduced by scheduling differences due to the execution queues for agents (i.e. built-in time delays to ensure task execution).

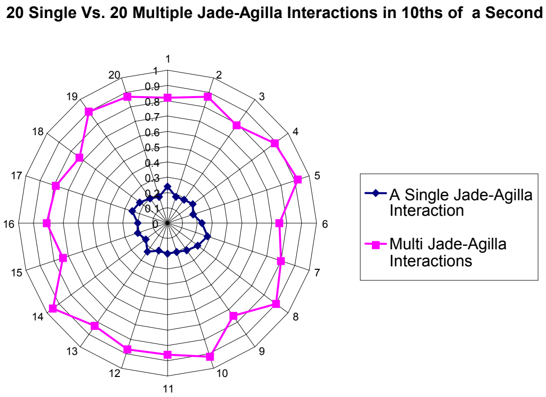

Figure 4. Time taken to complete (a) Single JADE-Agilla interactions in isolation (b) 20 Multiple JADE-Agilla interactions in parallel.

6. Conclusion

The agent (and in particular mobile agent) based software architecture may improve the management of patient medical data within a pervasive wireless patient sensor network, and may thus contribute to better patient care. The problem is that the distributed environment contains both resource-rich and resource-constrained computing platforms. The more lightweight Agilla environment has been ported to the Tyndall25 wireless node, and supports agents on the wireless patient nodes where a more sophisticated but heavyweight agent environment (such as JADE) would not be supported. The more powerful computing nodes may support the full JADE agent environment. To maximise the benefits of both the general purpose JADE and specialised Agilla agent platforms, an effective integration was achieved.

The integrated JADE-Agilla environment has been evaluated under a number of scenarios, and is shown to provide a mobile agent infrastructure that extends to the wireless nodes. The integrated architecture supports the agent-based proactive and reactive behaviour demanded by a medical application in a pervasive environment.

Acknowledgments

This work is funded by the Boole Centre for Research in Informatics at UCC and SFI(Science Foundation Ireland) through the NAP (National Access Program) at the Tyndall Institute, UCC.

7. References

[1] J. O’Donoghue and J. Herbert, “Data Management System: A Context Aware Architecture For Pervasive Patient Monitoring” Proceedings of the 3rd International Conference On Smart homes and health Telemetric - ICOST 2005

[2] F. Bellifemine, A. Poggi, and G. Rimassa, “JADE – FIPA Compliant Agent Framework”, In Proceedings of PAAM, 1999.

[3] C.-L. Fok, G.-C. Roman, C. Lu, "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications" In Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'05), Columbus, Ohio, 2005, pp. 653-662.

[4] C.-L. Fok, G.-C. Roman, C. Lu, "Mobile Agent Middleware for Sensor Networks: An Application Case Study" In Proceedings of the 4th International Conference on Information Processing in Sensor Networks (IPSN'05), Los Angeles, California, 2005, pp. 382-387.

[5] B. O'Flynn, S. Bellis, et al, "A 3-D Miniaturised Programmable Transceiver" Proc. XXVIII International Conference of IMAPS, 26-29 September, 2004, Poland Chapter Wroclaw, p. 343-347

[6] Philip Levis, Sam Madden, et al, "TinyOS: An Operating System for Wireless Sensor Networks", Aarts (Eds.), Ambient Intelligence, Springer-Verlag, 2004.

[7] Mica Mote, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA.pdf

[8] Jess, the Java Expert System Shell: http://herzberg.ca.sandia.gov/jess/

[9] O’ Sullivan, T., O’ Donoghue, J., Herbert, J., and Studdert, R.: CAMMD: Context Aware Mobile Medical Devices. International Journal of Universal Computer Science, Special Issue on Pervasive Health Management: New Challenges for Health Informatics, (2005)

[10] Marcela Rodriguez, Jesus Favela, Victor Gonzalez and Miguel Muñoz, "Agent Based Mobile Collaboration and Information Access in a Healthcare Environment", Proceedings of Workshop of E-Health: Applications of Computing Science in Medicine and Health Care.

[11] Arkady Zaslavsky “Mobile Agents: Can They Assist with Context Awareness?” 2004 IEEE International Conference on Mobile Data Management (MDM'04) p. 304

[12] D. Massaguer, "Multi mobile agent deployment in wireless sensor networks," Master's thesis, University of California, Irvine, 2005