Appl. Math. Inf. Sci. 6 No. 2S pp. 379S-385S (2012)

Applied Mathematics & Information Sciences
An International Journal

Implementing Statistical Agents on JADE Platform

Chien-Ho Wu [1], **Jung-Bin Li** [1] and **Tsair-Yuan Chang** [2]

[1] Department of Statistics and Information Science, Fu Jen Catholic University, Taiwan

[2] Department of Information Management, Ming Chuan University, Taiwan

Corresponding author: Email: tychang@mail.mcu.edu.tw

Received July 15, 2011; Revised Aug. 15, 2011; Accepted Sep. 2, 2011
Published online: 1 January 2012

Abstract: The advancement of internet and database technology makes it much easier than ever before to build databases that can accommodate huge volume of data connected by computer networks. In a modern business context, this scenario calls for the straight need to access and analyze data stored in distributed and possibly heterogeneous networked databases in order to generate business intelligence used to support effective business decision making. To address this need and alleviate the burden of possible constant human intervention in the process of generating business intelligence, we adopt agent technology and examine the pros and cons of agent technology in implementing statistical agents for data analysis on the JADE platform. JADE is chosen as the implementation platform since JADE is fully implemented, with FIPA compliance, in Java language, and thus can simplify the implementation of peer-to-peer agent based applications. The statistical agents implemented so far in this research can be deployed to different nodes on the JADE platform. The results from the implementation prove the technical feasibility of our design and give possible directions for future improvements.

Keywords: Statistical Agent, JADE, FIPA

1 Introduction

It is no argument that the survival and prosperity of a business relies very much on the effectiveness of its decision making process, which requires prompt availability of quality business intelligence. In a modern business context, data needed for the generation of business intelligence (BI) is very often huge and/or distributed over networked databases. In such a context it is never easy to conveniently locate and retrieve data needed for business analysis. Among other solutions, agent technology is very promising in facilitating the generation of intelligence for decision support [4].

In general, an agent is essentially a software component, capable of learning that behaves like a human agent. It has features such as autonomy, reactivity and social. Furthermore it can be mobile with the ability to travel between different nodes in a computer network [6]. With such features, agents can be deployed, on an ad hoc basis, to dedicated sites for not just collecting but also analyzing data for decision purposes.

In light of the fact that data required for the generation of business intelligence is often not located at a single site and JADE is the middleware aimed at supporting the development of distributed multi-agent systems, this research is set to exploit the agent technology on the JADE platform and demonstrate the mobility and technical feasibility of an JADE agent in supporting statistical data analysis.

The rest of this paper is organized as follows. Section 2 will justify and briefly account on the design of the implementation framework. Then the function of the statistical agent implemented to prove our idea is described in section 3. The results from our implementation are given in section 4. Finally section 5 presents the limitations and future research directions of this research.

2 Design of Implementation Framework

2.1. The Scenario

In modern business environment data required for the production of business intelligence is not uncommon stored in networked databases located at different sites. There is no doubt that data alone is of no use in supporting business decision making. It is through statistical analysis that decision information is able to emerge from the possible terabytes of data. As a result it is critical that we can deploy statistical agents to various sites to help efficiently collect data and generate decision information

2.2. The Objective

Although agents are deemed to have many essential features, the main objective of this research is focused on demonstrating the feasibility of dynamically deploying statistical agents on the JADE platform. Our efforts are directed to the evaluation of the mobility feature of an agent.

As shown in Fig. 1, statistical agents implemented in this research are able to be deployed on demand to conduct dedicated statistical analysis on remote sites and return the computed results to the local deploying site. In current implementation, statistical agents are independent of each other and access to remote databases is through database views. For security considerations no direct access to database tables is allowed.

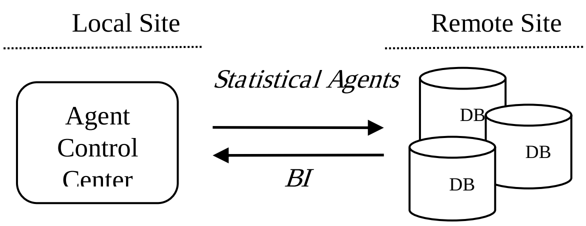


Figure 1: The conceptual structure of the implementation.

2.3. The JADE Platform

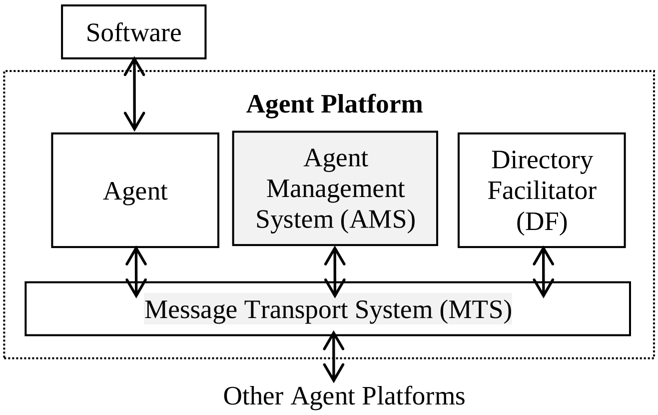
JADE, fully implemented in Java, is a middleware and an agent development platform conforming to FIPA standards for intelligent agents. The reference architecture of a FIPA-compliant agent platform is shown in Fig. 2. The Agent Management System (AMS) exerts supervisory control over access to and use of the agent platform [9]. Only one AMS can exist in a single JADE platform, and each JADE agent must register with an AMS in order to receive a valid agent identifier (AID).

A JADE agent has a physical lifecycle. The states of an agent in its life cycle include initiated, active, suspended, waiting, transit and unknown [9,21]. An agent may be registered at a number of transport addresses at which it can be contacted.

The Message Transport System (MTS) [16] controls all the exchange of ACL messages [19] between agents. The Directory Facilitator (DF) [9] provides yellow page service in the platform. A DF can be federated with other DF's and create a complex domains and sub-domains of yellow pages.

According to FIPA specifications, AMS and DF communicate by using the FIPA-SL0 content language [17], the fipa-agent-management ontology [9,16], and the fipa-request interaction protocol [18]. While the content language defines the syntax of the content of a message, the ontology provides a set of concepts and symbols to express the message. The predefined sequences of messages are known as protocols.

Figure 2: The reference architecture of FIPA agent platform.



The software component includes all non-agent collections of executables accessible through an agent. Agents may access software, for example, to acquire new security protocols or algorithms.

JADE relies on what is called agent containers to provide the runtime support to agent execution. Agent containers can be split among several hosts, thus making JADE a distributed agent platform.

2.4. The Implementation Framework

Fig. 3 is the framework for our current implementation. The local deploying site will communicate with the remote site to collect connection credentials and the schema for data to be analyzed.

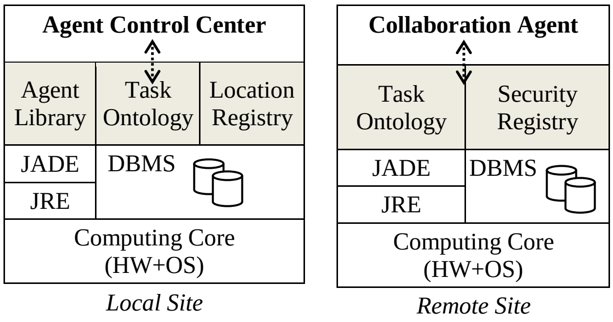


Figure 3: Implementation framework.

Detailed descriptions of the main components of Fig. 3 are given as follows.

Agent Control Center. This is the main GUI of the implemented system and it is also a JADE agent. Agent control center plays the role as a portal and a task manager.

Collaboration Agent. This agent, residing on the remote site running in the background, is in charge of coordinating the initiation and logging of a statistical task deployed by the agent control center.

Agent Library. This is the library containing statistical agents that can be deployed to remote sites.

Task Ontology. In this research, the main purpose of the task ontology [1,10,14,15,20] is to facilitate the exchange of schema and security credentials between the local deploying site and the remote site. Fig. 4 is the structure of the task ontology. The task ontology is not to define the conceptualization of terms but to prescribe what constitutes the meta-information of a statistical task. This task ontology must be shared and agreed upon by the participating agents.

In Fig. 4, the symbol ◆ indicates that the values of the constituent elements are mandatory, and the symbol ◇ indicates that the constituent elements may assume default values if their values are not prescribed.

In JADE, an ontology for a given domain is essentially a set of schemas defining the structure of the predicates, concepts and agent actions that are pertinent to that domain. Schemas are then implemented by proper Java classes. An agent must register the defined ontology and the selected content language to use the ontology.

Location Registry. The location registry is a text file containing the MTP addresses of remote sites.

Security Registry. The registry contains user names, passwords and views that can be used by statistical agents to connect to the remote site. The registry is not encrypted in current implementation.

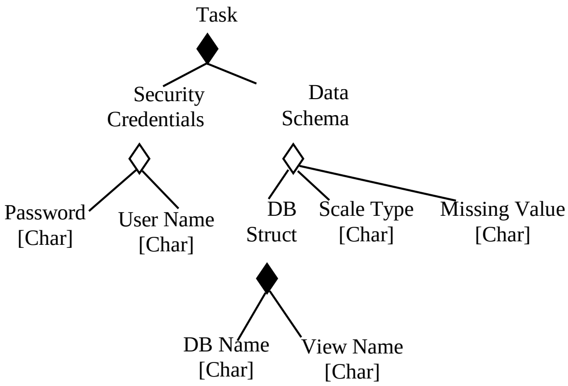


Figure 4: Task ontology.

The interaction between the local deploying site and the remote site is shown as a sequence diagram [11] in Fig. 5. The purpose of the interaction is to exchange the meta-information about a task so that a statistical agent can be deployed to the remote site. The results from a particular statistical computation at the remote site are wrapped in ACL messages sent to the local deploying site.

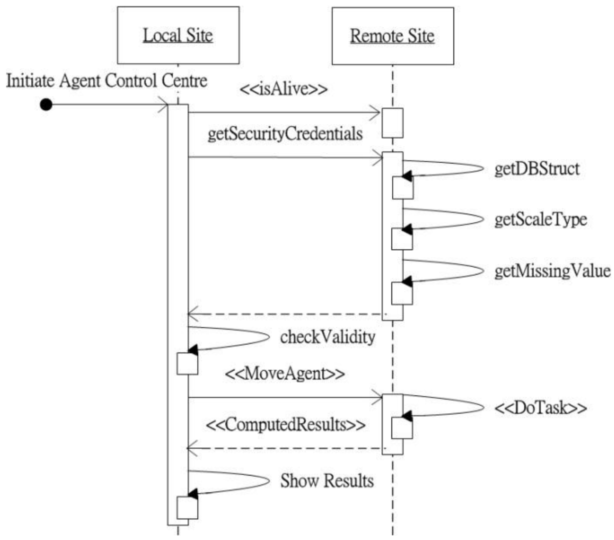


Figure 5: The sequence diagram for the interaction between the local and remote sites.

3 The Function of an Illustrated Statistical Agent

3.1. The Idea of the Scaling Procedure

Research in the field of social sciences often asks subjects to rate their opinions on a scale of ordered categories such as Likert-type scale concerning, for example, sensory testing or attitude scaling. Typically, arbitrary integer scores are used to represent those categories and data thus produced are analyzed by parametric methods on the assumption either that the necessary criteria are satisfied or that the test is sufficiently robust for it not to matter. However, datasets derived from integer scoring schemes often do not conform to distributions for which such assumptions are justified. A scoring scheme based upon assumptions of normal distribution theory is preferable over simple integer scoring schemes.

Snell transformation [5] is a method of determining numerical scores for the ordered categories of subjective scales. The scores so determined are suitable for use in methods of analysis dependent upon assumptions of normality.

The Snell transformation method assumes that there is an underlying continuous scale of measurement along which the scale categories represent intervals, as shown in Fig. 6 [3,5]. For illustration, category s_k in Fig. 6 corresponds to the interval (x_{k-1}, x_k).

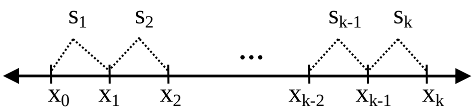


Figure 6: The underlying continuous scale of measurement.

In general, after the intervals are derived, mid-points of the intervals are taken as scores of the corresponding scale categories. It shall be noted that the intervals computed by the Snell transformation method may not be of equal range. This feature is favored by those who argue that integer scoring scheme is not sound and could be fundamentally flawed from the perspective of cognitive science.

The statistical agent designed for the current implementation can convert an integer-coded Liker-type dataset on the remote site into a dataset in real numbers using the Snell transformation method.

3.2. The Theoretical Model

The scoring procedure proposed by Snell satisfies, as far as possible, the following two parametric assumptions:

1. The residual deviations are normally distributed
2. The residual variances are homogeneous.

Since logistic distribution agrees closely over most of its range with the normal curve and can simplify the resulting model, the underlying continuous distribution function for the transformation method is based on a logistic model.

As shown in Fig. 6, for a scale of k categories, it is defined that category s_j corresponds to an interval (x_{j-1}, x_j). If there are m groups of observations then the underlying distribution function for each group, denoted $P_i(x_j)$ or P_{ij} for short, is defined as follows.

$$P_i(x_j)=\frac{1}{1+e^{-(a_i+x_j)}} \qquad (1)$$

$P_i(x_j)$ is a logistic probability distribution function with mean equals $-a_i$ and variance $\pi^2/3$. The probability of an observation of group i in category s_j is thus equal to $P_i(x_j)$- $P_i(x_{j-1})$.

3.3. The Approximate Scaling Procedure

Although there is no easy solution to the maximum likelihood estimates of x_j, we can achieve approximate estimates of the parameter x_j by referring to equations (2) and (3) [3].

$$\hat{x}_{k-1}-\hat{x}_{k-2}=\ln\left(\frac{N_{k-1}}{\sum_{i=1}^{m}(n_{i,k-1}+n_{ik})\hat{P}_{i,k-1}-N_{k-1}}+1\right) \qquad (2)$$

where $n_{i,k-1}$ denotes the number of observations of group i in category s_{k-1}, and N_{k-1} the total number of observations in category s_{k-1}. $\hat{P}_{i,k-1}$ is the theoretical proportion of an observation of group i in category s_{k-1} or below.

$$\hat{x}_j-\hat{x}_{j-1}=\ln\left(\frac{N_j}{\sum_{i=1}^{m}(n_{ij}+n_{i,j+1})\hat{P}_{ij}+\frac{N_{j+1}}{e^{(\hat{x}_{j+1}-\hat{x}_j)}}-N_j}+1\right) \qquad (3)$$

$$for \quad j=2,\ldots k-2$$

For the purpose of achieving an approximate solution, the theoretical proportions $\hat{P}_{ij}$ in equations (2) and (3) can be replaced by the observed accumulative proportions p_{ij} provided that there are no obvious irregularities in the data that may cause significant discrepancies between the theoretical and observed proportions. The estimates of x_j thus obtained will be close to the true values for $\hat{x}_j$. The observed proportions p_{ij} can be obtained by equation (4).

$$p_{ij} = \frac{\sum_{u=1}^{j} n_{iu}}{\sum_{t=1}^{k} n_{it}} \quad where \quad i = 1 \; to \; m \qquad (4)$$

Since the choice of the origin for the continuous scale of measurement is arbitrary, the value of x_1 can be set to 0. Once the class boundaries x_j are estimated, mid-points are taken as scores for categories. For the extreme category s_k, we take (x_{k-1}+1.0) or (x_{k-1}+1.1) respectively as its score according to whether the average of the observed proportions in category s_k (i.e. $1- p_{i,k-1}$) is less than 0.10, or lies between 0.10 and 0.20. A point with a similar distance below x_1 is taken as the score for s_1. An algorithm with linear complexity for the scaling procedure can be found in [3]. Readers may refer to [5] for illustrative examples.

4 Implementation

As mentioned in section 2.3, a JADE platform contains one or more agent containers which can be split among different hosts. The role of agent containers is to provide the runtime support to agent execution. Main container and peripheral container are the two types of agent containers that exist in a JADE platform. JADE relies on the main container to coordinate all other nodes and keep together the whole platform. Each JADE platform must have one and only one main container [6,7,8,9].

4.1. The Architectural Elements

The JADE architectural elements for our implementation are shown in Fig. 7. To avoid possible confusions and to be more realistic in simulating a distributed computing scenario, it is not suggested to run multiple JADE platforms by starting more than one main container with different local ports on a single host.

Since the latest version of JADE built-in agent mobility service supports only intra-platform mobility, the test bed for our current implementation comprises of a single JADE platform distributed over two computing hosts running on windows and Linux (CentOS) operating systems respectively.

To facilitate the set-up of the test bed, we have adopted the concept of virtual machines which allows a computer to run multiple operating systems and their applications simultaneously. As such, the remote host of the test bed is realized by a Linux virtual machine created by using the VMware software [2,12,13].

Each and every peripheral container must register with a main container to join a particular JADE platform. Agents of our design are run in the peripheral containers.

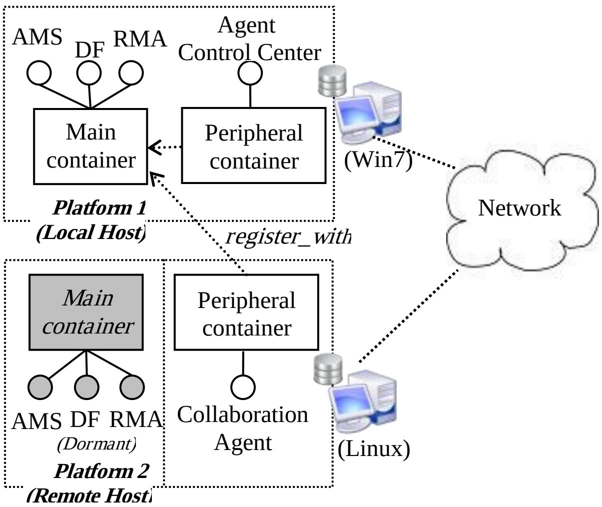


Figure 7: The JADE architectural elements for the implementation.

4.2. The Control Flow of Agent Control Center

Fig. 8 is the streamlined activity diagram [11] depicting the flow of control of the agent control center. This flow of control is mainly implemented as a composite behavior of the control center agent. In fact the deployment of an agent is achieved by creating an agent in the agent container at the remote site through AMS.

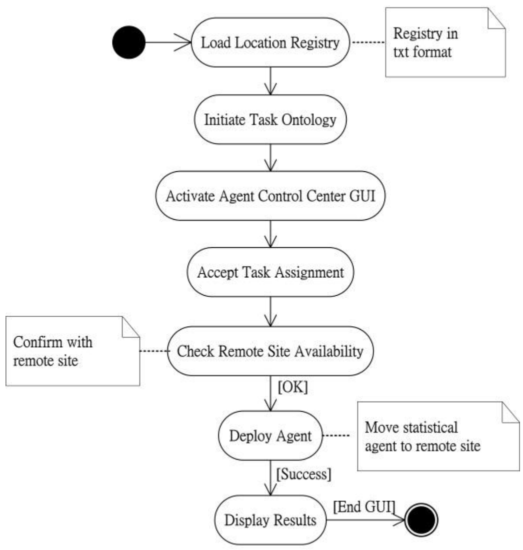


Figure 8: The activity diagram for the agent control center.

4.3. Network Configurations

Since agents need to exchange messages to complete a dedicated task, it is a must that agents involved with the execution of the task are able to identify each other. In particular, the IP addresses of the underlying hosts must be known to each other so that agents will be reachable. This implies that some network configuration chores are indispensable. Nonetheless, firewalls should also be configured or turned off so that network access from or to the collaborating platforms will not be blocked due to security constraints.

5 Final Remarks

This research has implemented, on the JADE platform, a statistical agent that can be deployed to remote sites. The implemented agent can collect, compute and return computed results back to the local deploying site. The implementation so far has the following limitations:

1. The agent is not designed to hop around from a remote site to another remote site autonomously. The agent implemented can only be deployed to a dedicated site for a specific assignment.
2. The role of the task ontology is to facilitate the exchange of security credentials and data schema. This preliminary arrangement may not be appropriate for supporting the execution of a complex task.
3. We assume that the remote site is active and connected through computer network with the local deploying site continuously. For now, no measures are adopted to deal with broken links.
4. JADE, as a pure Java implementation platform, is very good for implementing agents of high portability. However, for the sake of supporting statistical data analysis, its integration with statistical libraries deserves more attention.

In the near future this research can be extended in the following directions:

1. Elaborate in more detail on the various types of ontology that may be used to support statistical data analysis.
2. Extend the ability of an agent to use existing statistical libraries on local or remote sites for data analysis.
3. Streamline the workflow and explore the possibility of hopping around sites to complete a complex statistical analysis.

A bottleneck in using JADE as a distributed computing platform is the difficulty in creating a test bed. This research found that creating a distributed test bed on a single physical host by adopting the idea of virtual machines comes in handy for the research community.

References

[1] B. Smith, Ontology in *Blackwell Guide to the Philosophy of Computing and Information*, Edited L. Floridi, Oxford: Blackwell Publishing Company, 155-166, (2003).

[2] B. Ward, *The Book of VMware: The Complete Guide to VMware Workstation*, San Francisco: No Starch Press, (2002).

[3] C. Wu, An Empirical Study on the Transformation of Likert-scale Data to Numerical Scores, *Applied Mathematical Sciences*, Vol. 1 No. 58, 2851-2861, (2007).

[4] C. Wu, Y. Shao, J. Liu and T. Chang, On Supporting Cross-platform Statistical Data Analysis Using JADE, *Studies in Computational Intelligence*, Vol. 214, 271-277, (2009).

[5] E. Snell, A Scaling Procedure for Ordered Categorical Data, *Biometrics*, Vol.20 No.3, 592-607, (1964).

[6] F. Bellifemine, G. Caire and D. Greenwood, *Developing Multi-Agent Systems with JADE*, Chichester: John Wiley & Sons, (2007).

[7] F. Bellifemine, G. Caire, T. Trucco and G. Rimassa, *JADE Programmer's GUIDE*, CSELT S.p.A., (2010).

[8] F. Bellifemine, G. Caire, T. Trucco, G. Rimassa and R. Mungenast, *JADE Administrator's GUIDE*, CSELT S.p.A., (2010).

[9] FIPA, *FIPA Agent Management Specification*, Geneva, Switzerland, (2004). (URL: http://www.fipa.org/specs/fipa00023/SC00023K.pdf).

[10] G. Heijst, S. Falasconi and A. Abu-Hanna, A Case Study in Ontology Construction, *Artificial Intelligence in Medicine*, Vol.7, 227-255, (1995).

[11] J. Rumbaugh, G. Booch and I. Jacobson, *The Unified Modeling Language Reference Manual*, Massachusetts: Addison-Wesley, (2010).

[12] R. Bastiaansen, *Rob's Guide to Using VMware*, Books4Brains, The Netherlands, (2005).

[13] R. Troy and M. Helmke, *VMware Cookbook: A Real-World Guide to Effective VMware Use*, Sebastopol: O'Reilly, (2010).

[14] S. Staab and R. Studer (Eds.), *Handbook on Ontologies*, Berlin: Springer-Verlag, (2004).

[15] W. Quine, On What There Is, *The Review of Metaphysics*, Vol.2, No.5, 21-38, (1948). (URL: http://www.jstor.org/stable/20123117)

[16] FIPA, *FIPA Agent Message Transport Service Specification*, Geneva, Switzerland, (2002). (URL: http://www.fipa.org/specs/fipa00067/SC00067F.pdf)
[17] FIPA, *FIPA SL Content Language Specification*, Geneva, Switzerland, (2002). (URL: http://www.fipa.org/specs/fipa00008/SC00008I.pdf)
[18] FIPA, *FIPA Request Interaction Protocol Specification*, Geneva, Switzerland, (2002). (URL: http://www.fipa.org/specs/fipa00026/SC00026H.pdf)
[19] FIPA, *FIPA ACL Message Structure Specification*, Geneva, Switzerland, (2002). (URL: http://www.fipa.org/specs/fipa00061/SC00061G.pdf)
[20] FIPA, *FIPA Ontology Service Specification*, Geneva, Switzerland, (2001). (URL: http://www.fipa.org/specs/fipa00086/XC00086D.pdf)
[21] FIPA, *FIPA Agent Management Support for Mobility Specification*, Geneva, Switzerland, (2002). (URL: http://www.fipa.org/specs/fipa00087/DC00087C.pdf)

Chien-Ho Wu obtained his PhD degree in Informatics from the University of Manchester, U.K. He is an Associate Professor at the Department of Statistics and Information Science of Fu Jen Catholic University, Taiwan. His research interests are in the areas of distributed intelligent systems, semantic engineering, object-oriented technology and applied statistical data analysis. He is a permanent member of both CSIM and EBMS.

Jung-Bin Li received his PhD degree from National Chiao Tung University of Taiwan. He is an Assistant Professor at the Department of Statistics and Information Science, Fu Jen Catholic University. His research interests include data mining, machine learning, electronic commerce and mobile applications.

Tsair-Yuan Chang is presently an assistant professor of Information Management Department at the University of Ming Chuan, Taiwan. He received his PhD degree in Operation Research & Systems from the Warwick Business School at the University of Warwick, U.K. His research interests relate to ontology modeling, semantic web technology, object-oriented technology and case-based reasoning.