Information Agents Cooperating with Heterogenous Data Sources for Customer-Order Management

Dionisis Kehagias
Dept. of Electrical and Computer Engineering
Aristotle University of Thessaloniki
54124, Thessaloniki, Greece
diok@ee.auth.gr

Kyriakos C. Chatzidimitriou
Dept. of Electrical and Computer Engineering
Aristotle University of Thessaloniki
54124, Thessaloniki, Greece
kyrxa@ee.auth.gr

Andreas L. Symeonidis
Dept. of Electrical and Computer Engineering
Aristotle University of Thessaloniki
54124, Thessaloniki, Greece
asymeon@ee.auth.gr

Pericles A. Mitkas
Dept. of Electrical and Computer Engineering
Aristotle University of Thessaloniki
54124, Thessaloniki, Greece
mitkas@eng.auth.gr

ABSTRACT

As multi-agent systems and information agents obtain an increasing acceptance by application developers, existing legacy Enterprise Resource Planning (ERP) systems still provide the main source of data used in customer, supplier and inventory resource management. In this paper we present a multi-agent system, comprised of information agents, which cooperates with a legacy ERP in order to carry out orders posted by customers in an enterprise environment. Our system is enriched by the capability of producing recommendations to the interested customer through agent cooperation. At first, we address the problem of information workload in an enterprise environment and explore the opportunity of a plausible solution. Secondly we present the architecture of our system and the types of agents involved in it. Finally, we show how it manipulates retrieved information for efficient and facile customer-order management and illustrate results derived from real-data.

Keywords

Information Agents, Enterprise Resource Planning, Customer-Order Management

1. INTRODUCTION

Information agents acting on a cooperative environment represent a promising paradigm for constructing intelligent decision-making applications that manipulate data stored in legacy databases. The need for processing legacy data arises from the fact that many repositories have been developed over the past years in the enterprise IT environment, containing significant information. The majority of existing enterprise information systems come up with the problem of extracting available information as efficiently as possible, in order to perform decision-making, resource management, and prediction about financial trends at the lowest cost [13]. In this respect, many data manipulation frameworks, known as Enterprise Resource Planning (ERP) systems, have been developed in order to organize data produced on a periodic basis over an enterprise network. These systems automate and integrate the most important business processes in real time, creating large amounts of enterprise data [19].

However, in the best of its expenditure, an ERP system remains a data logging system, which keeps enterprise managers up-to-date, at any time of the inter-enterprise business workflow. The accumulation of large amounts of data provided by an ERP system, results in such a data overflow into enterprise databases, which humans cannot cope with, even utilizing the most advanced information extraction capabilities that the ERP system deploys. Moreover, traditional ERP systems cannot deal with tasks such as the effective resource planning and decision support [18], thus increasing the need for the development of a low-cost efficient information- manipulation system, especially for major and high-risk business processes.

In this paper we present a multi-agent system (MAS) that handles the business process of customer-order management. The process-flow of the latter involves the following phases:

1. A customer requests a new order to be processed, and the ERP operator inputs the order into the system. This activity is related to the functionality introduced by a specific agent type, which has been developed within our system and which is called *Customer Order Agent* (COA).
2. Fixed business policies are applied to the customer and the order. This phase involves two agent types, the *Recommendation Agent* (RA) and the *Customer Profile Identification Agent* (CPIA).
3. The system notifies the operator about the result. The

COA of our system is also related to this process.

The procurement of the required supplies to fulfill customers' orders, follows a different, but similar process (if the ERP system has a Supply Chain Management module installed):

1. Fixed business policies are applied to the inventories and to the suppliers. This functionality is implemented as a coordinated activity between the agents: RA, *Inventory Profile Identification Agent* (IPIA) and *Supplier Profile Identification Agent* (SPIA).

2. System notifies the operator about the result of the processing. This activity is carried out in our system by the COA.

3. A request for supplies is performed.

The agent types mentioned above, are described in detail later in this paper.

Our system confronts the problem of reducing the cost of information manipulation and decision-making over enterprise data, concerning the efficient transaction of customer-orders. For this purpose, the MAS presented in this paper is coupled with a decision-making mechanism for generating recommendations to the customer. All agents involved in it represent different business processes, which are activated in collaboration, in order to fulfill the customer-requested order. The MAS is built on top of a legacy ERP system, which contains data representing all business-process entities involved in the information processing procedure. These entities include customers, inventories and suppliers. Thus, the ERP system provides the main source of information for agents. Using specific techniques, described later in this paper, all information agents of the MAS perform efficient information processing that results in the generation of a set of recommendations for each requested order. Through our approach we intend to provide a low-cost, effective solution that overcomes data overload in an enterprise IT infrastructure and exploits the vast amount of available legacy data sources for successful customer-order management.

The paper is structured as follows. Section 2 reviews related work. Section 3 describes the architecture of our application and illustrates the types of agents involved in it. In section 4, a presentation of the core information processing mechanisms is given. Finally, section 5 concludes the paper and outlines future work.

2. RELATED WORK

Many approaches have been introduced for enterprise management and development of decision support systems. Especially, in the last few years, many researchers have involved agents into business processes for realizing decision-making capabilities. In this context, two known frameworks [4, 20] have been developed for modeling supply chains in order to capture their dynamics and provide a re-usable development platform, to be exploited by managers for detailed analysis and evaluation.

Another system for Supplier Relationship Management (SRM) [7] uses neural networks and case-based reasoning for benchmarking suppliers by both quantitative and qualitative criteria. Even though the system integrates Customer Relationship Management and Supplier Relationship Management, it does not deal with any account-inventory management, such as demand forecasting, at all. Moreover, this approach requires additional effort from human experts, in order to gather quantitative and qualitative information from available data about customers and suppliers, because it does not rely on attributes already stored in legacy data sources.

Two successful implementations of a multi-agent system supporting intelligent integration, but in a different domain than ours, can be found in [12, 16]. These two applications provide an intelligent framework for production planning in an enterprise, by the use of expertise agents acting in a co-operative way. Both systems are situated on top of a legacy system gathering information and providing real-time decisions for different manufacturing-management scenarios.

All of the aforementioned systems have been developed for the provision of customer-oriented services and share common characteristics. From a software-engineering perspective, only [12, 16] are constructed as complete multi-agent systems. These also deploy a relatively high degree of intelligence into agents, which is not always a mandatory characteristic of a MAS. For instance, work described in [4] and [20] is also based on agents, without incorporating any AI algorithms. On the other hand, several implementations have given special focus on the use of advanced AI techniques. In particular, a specific mechanism has been developed in [7] that enables agents to learn suppliers profiles from their recorded attributes.

Even though there is an effort in pursuing Customer Relationship Management (CRM) dynamically in time, most systems of this kind do not capture the complete dynamics of the entities in an enterprise (e.g Integration with Supplier Relationship Management modules). Our implementation uniquely combines agent technology, fuzzy techniques and data mining algorithms for building a facility that performs CRM [17]. The special focus of our approach is given in a two-fold way: a) as the application of fuzzy algorithms and data mining techniques for information processing, and b) as an information-flow procedure, beginning from the legacy data-source and ending up to the end-user, which is realised as a set of coordinated communicative activities between agents, according to a concrete MAS architecture.

3. THE MULTI-AGENT SYSTEM

3.1 Architecture

Our system can be viewed as a set of five cooperating layers; each one of them lies on top of the other. The five layers, illustrated in figure 1, are the following:

1. The *data source layer* includes the external data repository, which is incorporated into the legacy ERP system.

2. The *software middleware layer* comprises of all software interfaces that provide connectivity between the external data and the MAS.

3. The *information processing layer* is the core layer of the system. It handles the manipulation of information received by the previous layer and the execution of the appropriate mechanisms that identify data attributes used for system recommendations.

4. The *decision-making layer* is responsible for collecting the outcome of information processing performed

in the previous layer and generating specific recommendations about how the order should be transacted. This layer is implemented as a rule-based agent.

5. The *graphical user interface layer* enables the system user to enter input data, regarding the details of the order to be transacted. This layer is also responsible for notifying users about the recommendation generated by the system.

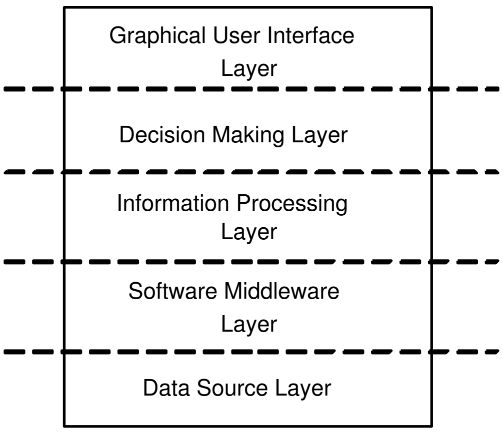

Figure 1: The five cooperating layers that represent the functionality of the system.

In order to model our MAS we used the Agent-Object Relationship Modeling Language (AORML) [22], an extension of UML that supports the representation of relations between agents, as well as other agent-specific characteristics that cannot efficiently represented by the use of plain UML. The architecture of our system is illustrated in figure 2, on an *external AORML diagram* where six different types of agents are depicted. The implementation of our MAS is based on the Java Agent Development Environment (JADE) [6], ensuring compatibility to FIPA standards, as they defined in [5, 8]. All agents, acting within the MAS, exchange certain Agent Communication Language (ACL) messages to each other, enabling the coordination of actions for the execution of the requested order. The ERP system is depicted as a rectangle box in the lower-left part of figure 2. Having now an idea of how the agents are arranged, we give, in the remaining section, a brief description of their roles and tasks they carry out.

3.2 Types of involved agents

3.2.1 Data source, ERP agents

By the ERP agent type, we represent a class of agents that implement the Software Middleware layer of figure 1. These agents can be resembled to transducers [10], because they are responsible for retrieving data from the legacy ERP system and transforming it into an agent-comprehensive format. The ERP agents handle all queries posted by the information agents residing on the third level. Specifically, they establish communication with the ERP database, through the Java JDBC application programming interface and transform the retrieved data to XML documents. Parts of these documents become content of specifically constructed ACL

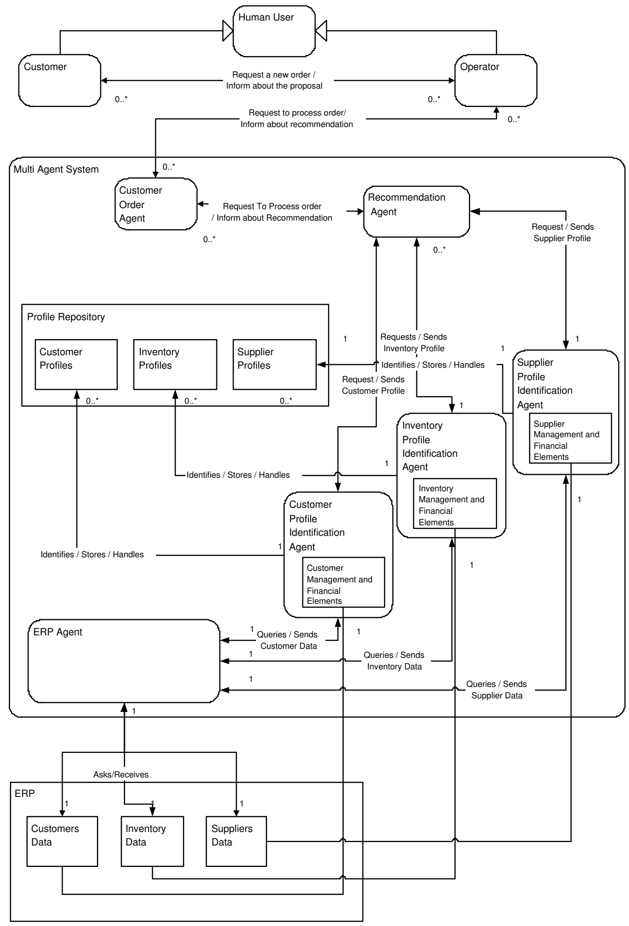

Figure 2: External AORML diagram representing the architecture of the system.

messages, which are sent to the information agents of the third layer. In our implementation we used FIPA-SL0 [9] as the ACL. All agents represented by this agent-type are the only agents of the system that need to be configured properly in order to meet the specific characteristics of the legacy ERP.

3.2.2 Customer Profile Identification Agents

The *Customer Profile Identification Agents* (CPIAs) are capable of identifying specific attributes in data retrieved from the legacy database that characterize the customer's profile. In this way, a CPIA classifies any customer who posts a request for a new order. This procedure is quite essential for the generation of a recommendation that best suits the current order. The process takes place on a monthly basis and the profiles are stored inside a profile-repository for future use, serving as a long time memory. A CPIA belongs to the Information Processing Layer, shown in figure 1.

3.2.3 Inventory Profile Identification Agents

Agents of this type are specifically designed to identify the profile of company's products. An *Inventory Profile Identification Agent* (IPIA) retrieves prices and availabilities about a specific product from the legacy database and constructs a profile, which is based on statistical measurements about this product's demand in the past years, or other related products' concurrent demand. Moreover market basket analysis techniques are applied on available transactions. In particular IPIA plays a two-fold role in the MAS and the product profile is an integration of two different processes:

1. It fetches information on price, stock, related suppliers and statistical data about the demand faced for each product in the order.

2. It provides recommendations on additional items to buy, based on association rule extraction techniques, by the use of the the Apriori Algorithm [3, 11].

This methodology results in the extraction of association rules, which are cached into the profile-repository for future reference. IPIAs are also represented in the Information Processing layer in figure 1.

3.2.4 Supplier Profile Identification Agents

A *Supplier Profile Identification Agent* (SPIA) is responsible for creating and providing a supplier-specific profile. The identification of that profile is performed in a similar manner to the one implemented for the CPIA. The suppliers' profiles are used by a SPIA in order for it to choose the most appropriate suppliers from a list of available ones, by reviewing its value towards the company. Any agent described by this agent-type is part of the Information Processing layer, illustrated in figure 1.

3.2.5 Recommendation Agents

A *Recommendation Agent* (RA) is responsible for gathering all profiles created by agents in the Information Processing layer, and producing the final recommendation. This agent employs a rule-based reasoning mechanism, implemented with the Java Expert System Shell (JESS) [2]. Any RA is an expert agent, which uses customizable rules, reflecting company's policies. These rules represent fixed business policies created by managers in the form of non-compiled rule documents. In this way business rules cannot only apply to raw legacy data, but also to generated profiles. This agent-type implements the Decision Making layer in figure 1.

3.2.6 The Customer-Order Agents

A *Customer-Order Agent* (COA) provides the main GUI to the operator of our application. This agent implements the Graphical User Interface layer of figure 1 and enables the operator to transparently interoperate with the ERP for providing input information and requesting visualization services. One capability of a COA is to show information of all available products to the user. Through the GUI provided by a COA, a user can select the products ordered by the customer, and enter information about the desired payment terms, the selected backorder policy and the type of product transportation costs. When the user completes all required information, COA transfers the customers' order and their preferences to the RA, from which the decision process begins.

In figure 3 and 4 two snapshots of the main GUI are shown. The customer is a company that sells toys and requests an order about three different products to be completed. After processing user's input, our system informs

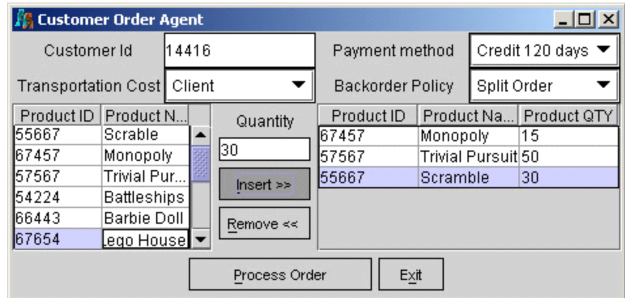

Figure 3: Order Input GUI.

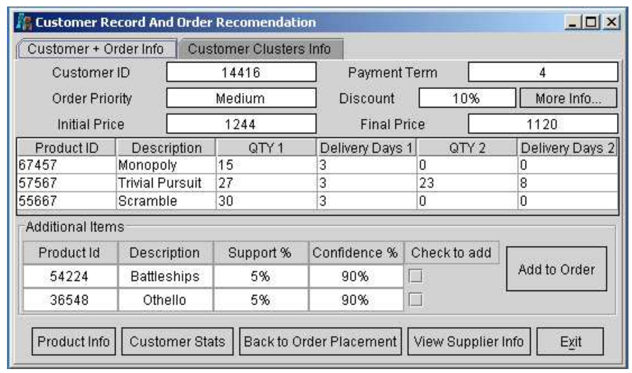

Figure 4: Customer Profile and Recommendation GUI.

the customer about how the company will satisfy the request and also finds out that two additional items are related to those already ordered by the customer. Another purpose of the GUI is to provide an explanatory unit to the MAS operator. By clicking on the appropriate GUI components, the user can gain information explaining the recommendation and providing the required steps to be executed, such as the procurement of necessary supplies.

4. INFORMATION PROCESSING TECHNIQUES

Three agent types that perform information processing on data from legacy resources, CPIA, IPIA and SPIA, exploit several known AI techniques in order to obtain customer, supplier and product profiles, which are used for generating suitable recommendations. To be more specific, let us present the core techniques used by each one of them.

The CPIA, uses pattern recognition clustering techniques and fuzzy logic inference for the generation and the characterization of the profiles. A customer's profile consists of customer priority and discount due to past behaviour (negative or positive). The profile of suppliers is identified by SPIA, using the same procedure as CPIA. However, the supplier's profile only consists of supplier's credibility, which is an indicative criterion about suppliers' reliability against the company. All generated profiles are fine-tuned on a constant time basis.

In order for the information techniques used by CPIA and SPIA, to be implemented, the following steps are executed:

1. Clustering is performed on the financial records of cus-

tomers and suppliers, by firstly using the Maximin algorithm [14], which provides a good estimation of the number of clusters. Maximin's outcome provides input into the K-means algorithm [14, 15]. After that, clustering is performed on the financial datasets acquired by the ERP agents for customers and suppliers. The dataset contains multiple variables, which are either deterministic or random. These variables include, among others, *Account Balance, Credit Limit, Turnover, Geographic Location* etc. Each variable has a preferred tendency denoting whether the company is benefited by its increase or not. As a final step the clusters are inserted into the profile-repository.

2. (a) In order to characterize the clusters generated in the previous step, in terms of priority and discount, the use of an adaptive fuzzy logic inference engine is adopted. The attributes of each cluster-center are used as input variables of the fuzzy engine.

   (b) Random variables (RV) are treated separately, according to their variation coefficient which is defined as the ratio of standard deviation towards the average of the variable. Random variables are used only when (1) is satisfied:

$$\left(\frac{STD(RV)}{AVG(RV)}\right)^2 < \xi \tag{1}$$

   (c) Since all inputs have three fuzzy variables (LOW, MEDIUM, HIGH), 3^ν Fuzzy Rules (FRs) are produced, where ν denotes the number of inputs in the fuzzy engine. FRs are of the following type:

   if X1 is LZ1(k) and X2 is LX2(k) and .. and Xn is LXn(k) Then U is LU(k), K = 1,.., m

   Triangular membership functions are adopted for all inputs and outputs, whereas maximum defuzzification is used for crisping the FRs. All inputs are given a Corresponding Value (CV) ranging from -1 to 1, according to the preferred tendency. The output value calculated for each FR is given by equation (2).

$$OV = \sum_{i=1..n} CV \tag{2}$$

   where n is the number of inputs. The OV-Fuzzy Values (FV) lookup table is illustrated in table 1.

3. From this point forward each time a RA asks either CPIA or SPIA about a specific customer or supplier profile, a dataset for this specific entity is fetched by the ERP agent and matched with a characterized cluster center using the space distance metric. The previously characterized cluster is then send back to RA.

The final recommendation is produced by the RA. In general, RA decides whether the order should be split or not. The splitting policy is initiated each time the customer requests to do so and the company stock availability cannot satisfy order needs. In case that the requested quantities are matched by the company's inventory stock, the order is fulfilled immediately.

In addition to that, the RA produces an estimation on the delivery time of the ordered products and another one on the requested quantity out of the available supplies. A more detailed description of the AI algorithms used, can be found in [21].

Table 1: The OV-FV lookup table for output discount. For example VERYLOW equals 0% discount while VERYHIGH equals 30% discount.

Output Value	Fuzzy Value
-6,-7	VERYLOW
-4,-5	LOW
-2,-3	MIDLOW
-1,0,1	MEDIUM
2,3	MIDHIGH
4,5	HIGH
6,7	VERYHIGH

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a multi-agent application about customer management in an enterprise environment. In our system we involved several information agents, data acquisition agents and a rule-based reasoning agent, acting in collaboration for generating recommendations about how a new customer-order should be executed. All information agents receive data from a legacy ERP and manipulate the extracted information using a combination of known AI techniques.

In order to test the efficiency of our system, we used real data from a company with approximately 3,100 customers and 500 suppliers. This company's installed ERP system contains data about 14,500 customer orders executed in the past two-year period. The existing legacy database stores over 150 Mbytes of information. After having information agents process available data, we managed to reduce the size of valuable information to 6 Mbytes. Another significant point, noticed during our experiments was that, based on the aforementioned amount of data, we received a recommendation in between 30 to 60 seconds, depending on the cardinality of ordered products and whether there was a stock-out or not.

Regarding the clustering process of customers and suppliers, the characterization of the cluster-centers, as well as the Market Basket Analysis, led to quite efficient results. For a specific company, CPIA has identified five major customer clusters, the same number as SPIA has identified for the suppliers. As a sample, the results for customers are summarized in table 2, while the results from the Market Basket Analysis were: 25 generated rules for 2% support and 90 % confidence, and 10 generated for 4% support and 90% confidence.

Table 2: The resulting customer clusters and the corresponding Discount and Priority values.

CenterID	Population%	Discount%	Priority
0	0.002	20	High
1	10.150	10	Medium
2	46.600	15	Medium
3	22.240	10	Medium
4	20.830	5	Low

An ERP or legacy system is a costly investment, and the process of replacing such a system is a matter out of question. With our system we provided some added-value features to the existing ERP system, without rebuilding or reengineering the installed system. Our approach has shown that it is able to achieve a significant reduction of work effort for human users, since the previously hard-coded business rules, can now be autonomously adapted. On the other hand, we managed to filter and manipulate data from legacy resources, resulting in reduction of information overload. Even though we do not consider obligatory for a company employee who handles new orders, to adopt the same decision as the one that our system recommends, she/he can obtain a clear view of the customers' and suppliers' value, and information about what products should be recommended to the customer, satisfying the company's sales policies. The system presented in this paper designed to be highly reusable and customizable. In this respect, it can be applied to different ERP systems without enormous development efforts. It can also be adapted to match the different company's policies in terms of order handling, by adequately regulating those parameters, which describe the rules of the RA. Thus for each different ERP that our system is applied on, a different set of rule-parameters is incorporated by the generic model of the RA.

Our future plans include the expansion of the MAS to be capable of handling distributed ERP systems by exploiting agent mobility. Moreover, new data mining techniques will be studied to enable even more efficient decision-making, which is expected to result in more precise recommendations.

6. ACKNOWLEDGMENTS

Work presented in this paper illustrates a large part of the development effort about the implementation of a test case application, in the context of the research and development project "Agent Academy: A Data Mining Framework for Training Intelligent Agents" [1]. This project is partially funded by the European Commission, under the IST initiative (contract number IST-2000-31050). Authors would like to thank all members of the Agent Academy consortium and especially ALTEC, for their valuable support in the development of the presented test case application.

7. REFERENCES

[1] Agent academy. http://agentacademy.iti.gr/.

[2] Java expert system shell (jess). http://herzberg.ca.sandia.gov/jess/.

[3] A. Amir, R. Feldman, and R. Kashi. A new versatile method for association generation. *Information Systems*, 22(6-7):333–347, 1997.

[4] M. Barbucceanu, R. Teigen, and M. Fox. Agent based design and simulation of supply chain systems. In *Proceedings of WET-ICE '97*, pages 36–42. IEEE Computer Society Press, 1997.

[5] F. Bellifemine, G. Caire, T. Trucco, and G. Rimassa. *JADE Programmer's Guide*. available at: http://sharon.cselt.it/, 2001.

[6] F. Bellifemine, A. Poggi, G. Rimassa, and P. Turci. An object-oriented framework to realize agent systems. In *Proceedings of WOA 2000 Workshop*, pages 52–57, 2000.

[7] K. Choy, W. Lee, and V. Lo. Development of a case based intelligent customer-supplier relationship management system. *Expert Systems with Applications*, 23(3):281–297, 2002.

[8] Foundation for Intelligent Physical Agents, available at: http://www.fipa.org/specs/fipa00021/. *FIPA Developer's Guide*, 2001.

[9] Foundation for Intelligent Physical Agents, available at: http://www.fipa.org/specs/fipa00008/. *FIPA SL Content Language Specification*, 2002.

[10] M. R. Genesereth and S. Ketchpel. Software agents. *Communications of the ACM*, 37(7):48–53, 1994.

[11] J. Han and M. Kamber. *Data Mining Concepts and Techniques*. Morgan Kaufmann, Burnaby, 2001.

[12] P. Koutsakas and A. Koumpis. Devising, best practices for customisation of a multi-agent production planning technology. In *Third International Workshop on Industrial Applications of Holonic and Multi-Agent Systems*, pages 1–13. DEXA, 2002.

[13] K. Liu, M. Fox, P. Apers, M. Klein, A. Cheng, R. Stamper, S. Chattopadhyay, and T. Greene. Enterprise information systems: Issues, challenges and viewpoints. In J. B. Filipe, editor, *Enterprise Information Systems*, pages 1–13. Kluwer, 2000.

[14] C. Looney. *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists*. Oxford University Press, 1997.

[15] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, 1967.

[16] Y. Peng, T. Finin, Y. Labrou, B. Chu, J. Long, W. Tolone, and A. Boughannam. A multi agent system for enterprise integration. *Applied Artificial Intelligence*, 13(1-2):39–63, 1999.

[17] C. Rygielski, J. Wnag, and D. Yen. Data mining techniques for customer relationship management. *Technology in Society*, 24(4):483–502, 2002.

[18] J. Shapiro. Bottom-up vs. top-down approaches to supply chain modeling. In S. Tayur, R. Ganeshan, and M. Magazine, editors, *Quantitative Models for Supply Chain Management*. Kluwer, 1999.

[19] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi. *Designing and Managing the Supply Chain: Concepts, Strategies, and Cases*. McGraw Hill, Illinois, 2000.

[20] J. Swaminathan, S. Smith, and N. Sadeh-Koniecpol. Modeling supply chain dynamics: A multiagent approach. *Decision Sciences*, April 1997.

[21] A. L. Symeonidis, D. Kehagias, and P. A. Mitkas. Intelligent policy recommendations on enterprise resource planning by the use of agent technology and data mining techniques. *Expert Systems with Applications*, 25(4):589–602, 2003.

[22] G. Wagner. The agent-object-relationship meta-model: Towards a unified view of state and behaviour. *Information Systems*, 28(5):475–504, 2003.