AN INTELLIGENT RECOMMENDATION FRAMEWORK

FOR ERP SYSTEMS

Andreas L. Symeonidis[a,b,*], Kyriakos C. Chatzidimitriou[a], Dionisis Kehagias[a,b]
and
Pericles A. Mitkas[a,b]
[a]Electrical and Computer Engineering Dept.,
Aristotle University of Thessaloniki,
[b]Intelligent Systems and Software Engineering Laboratory,
Informatics and Telematics Institute/CERTH,
57001, Thessaloniki, Greece

Andreas L. Symeonidis	Kyriakos C. Chatzidimitriou	Dionisis D. Kehagias	Pericles A. Mitkas
Tel. +30-310-99-6399	Tel. +30-310-99-6349	Tel. +30-310-99-6349	Tel. +30-310-99-6390
Fax +30-310-99-6398	Fax +30-310-99-6398	Fax +30-310-99-6396	Fax +30-310-99-6398
email: asymeon@ee.auth.gr	email: kyrxa@ee.auth.gr	email: diok@iti.gr	email: mitkas@eng.auth.gr

ABSTRACT
Enterprise Resource Planning systems efficiently administer all tasks concerning real-time planning and manufacturing, material procurement and inventory monitoring, customer and supplier management. Nevertheless, the incorporation of domain knowledge and the application of adaptive decision making into such systems require extreme customization with a cost that becomes unaffordable, especially in the case of SMEs. We present an alternative approach for incorporating adaptive business intelligence into the company's backbone. We have designed and developed a highly reconfigurable, adaptive, cost efficient multi-agent framework that acts as an add-on to ERP software, employing Data Mining and Soft Computing techniques in order to provide intelligent recommendations on customer, supplier and inventory management. In this paper, we present the architectural details of the developed framework.

KEY WORDS
ERP systems, Data Mining, Multi-Agent Systems

1. Introduction

Enterprise Resource Planning (ERP) systems are business management tools that automate and integrate all company facets, including real-time planning, manufacturing, sales, and marketing. These processes produce large amounts of enterprise data that are, in turn, used by managers and employees to handle all sorts of business tasks such as inventory control, order tracking, customer service, financing and human resources [1].

Despite the support current ERP systems provide on process coordination and data organization, most of them – especially legacy ones – lack advanced *Decision-Support* (DS) capabilities, resulting therefore in decreased company competitiveness. In addition, from a functionality perspective, most ERP systems are deprecated to mere *transactional IT systems*, capable of acquiring, processing and communicating raw or unsophisticated processed data on the company's past and present supply chain operations [2]. In order to optimize business processes in the tactical supply chain management level, the need for *analytical IT systems* that will work in close cooperation with the already installed ERP systems has already been identified, and DS-enabled systems stand out as the most successful gateway towards the development of more efficient and more profitable solutions. Probing even further, Davenport [3] suggests that decision-making capabilities should act as an extension of the human ability to process knowledge and proposes the unification of knowledge management systems with the classical transaction-based systems, while Carlsson and Turban [4] claim that the integration of smart add-on modules to the already established ERP systems could make standard software more effective and productive for the end-users.

The benefits of incorporating such sophisticated DS-enabled systems inside the company's IT infrastructure are analyzed in [5]. The most significant, among others, are:

1. Provision of evidence in support of a decision,
2. Improvement or sustainability of organizational competitiveness,
3. Augmentation of the decision makers' abilities to tackle large-scale, complex problems.

Within the context of *Small and Medium sized Enterprises* (SMEs) however, applying analytical and mathematical methods as the means for optimization of the supply chain management tasks is highly impractical, being both money- and time-consuming [6]. This is why alternative technologies, such as Data Mining (DM) and Agent Technology (AT) have already been employed, in order to provide efficient DS-enabled solutions. The increased flexibility of multi-agent applications, which provide multiple-loci of control [7] can lead to less development effort, while the application of DM techniques on existing ERP historical data can provide managers with information *non-trivial, implicit, previously unknown and potentially useful* [8], which can be used as an advisor to their decision-making capabilities. Our approach employs *Soft Computing (SC), DM, Expert Systems (ES),* standard *Supply Chain Management (SCM)* and *AT primitives*, in order to provide intelligent recommendations on customer, supplier and inventory issues. It is addressed not only to the managers of a company - *"Managing by wire"* approach [9] -, but also the lower-level, distributed decision makers - *"Cowboys"* approach [10].

Going briefly through related work, we see that DM and MAS have been used separately for efficient enterprise management and decision support. Rygielski et. al. [11] have exploited DM techniques for Customer Relationship Management (CRM), while Choy et. al. [6] have used a hybrid machine learning methodology for performing Supplier Relationship Management (SRM). On the other hand, MAS integrated with ERP systems have been used for production planning [12], and for the identification and maintenance of oversights and malfunctions inside the ERP systems [13].

The rest of the paper is organized as follows. Section 2 presents the extensive Recommendation Framework in detail and describes the functional characteristics of the different types of agents that comprise it. Finally, Section 3 summarizes the work presented, and concludes this paper.

2. The Intelligent Recommendation Framework

The arrival of a new customer order designates the initialization of the Intelligent Recommendation Framework (IRF) operation. All customer order preferences are, at first, gathered by the system operator via a front-end agent and are then transferred to the backbone (order) agents for further processing. These agents are of different types, each one related to a specific entity of the supply chain (company, customers, suppliers, products), and manage entity-specific data. In order to establish connectivity to the ERP system's database and access ERP data, another agent has also been implemented. By the use of DM techniques, all related entities' profiles are constructed for the recommendation procedure to be based on. When all processes are finalized, the front-end agent returns to the operator the intelligent recommendations produced by the framework, along with an explanatory memo.

2.1 IRF Architecture

The MAS add-on is composed of 6 different agent-types, working in close cooperation with each other. The agent types along with the human agents and other objects are depicted in Figure 1.

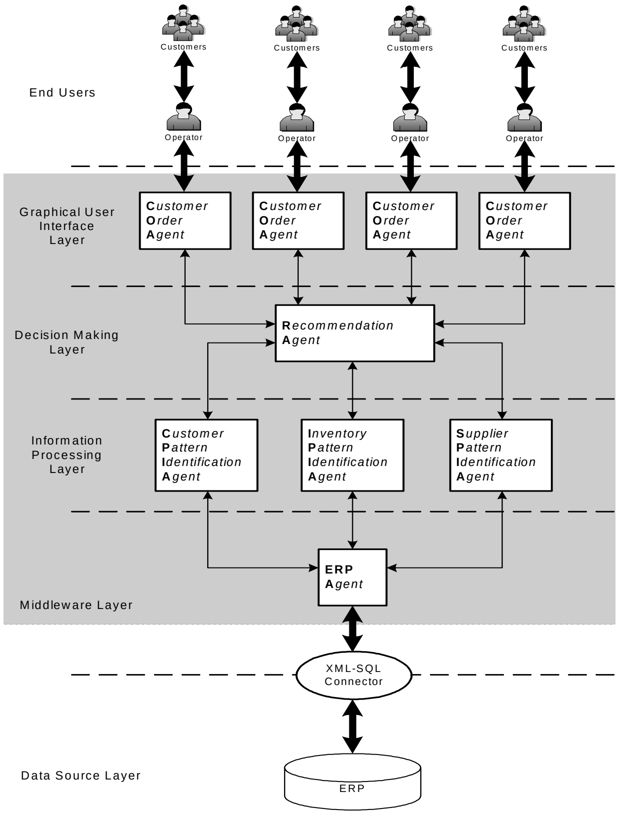

Figure 1. IRF Functional diagram

2.1.1. Customer Order Agent type (COA)

COA is a Graphical User Interface (GUI) agent that could be found at the distributed stores, or at the telephone center of an enterprise. COA enables the system operator (either manager or lower-level employee) to: (a) transfer information in and out the rest of the system, (b) input order preferences into the system and (c) explain by means of visualization the proposed recommendations. When an order comes into the system, COA provides the human agent with basic functionalities for inserting information on customer id, ordered products along with their corresponding quantities, payment terms (cash, check, credit etc.), backorder policies and, finally, the party (client or company) that will undertake transportation costs. COA also offers an explanatory unit for the final recommendation.

2.1.2. Recommendation Agent type (RA)

RA is responsible for gathering the profiles of the related to the current order entities. By requesting from the

Information Processing Layer agents (CPIA, SPIA and IPIA) the corresponding profiles, and by taking into account concurrency issues, RA diminishes the cycle-time of the recommendation process. RA is a rule-based agent and static business rules can be incorporated into it, by writing the latter into a document that RA reads during its execution phase. In this way, business rules can be changed on-the-fly, without the need of recompiling, or even restarting the application. RA is responsible for the final formatting of the recommendation that is forwarded to COA.

2.1.3. Customer Profile Identification Agent Type (CPIA)

CPIA is designed to identify customer profiles, according to historical data found in the ERP system's records. The customer profile identification process can be described at a glance as: Initially, managers and application developers produce a model for generating the profiles of customers. They select the appropriate customer attributes that can be mapped from the data residing in the ERP database; these are the attributes that are considered as valuable assets for reasoning on customer value. Then, they decide on the desired classification of customers, e.g. added-value to the company, discount due to past transactions etc. CPIA, by the use of clustering techniques, analyzes customer profiles periodically, and stores the outcome of this analysis into a profile repository for posterior retrieval. When a CPIA is asked to provide the profile of a customer, the current attributes of the specific customer are requested from the ERP database and are matched with those inside the profile repository, resulting into the identification of the group the specific customer belongs.

2.1.4. Supplier Pattern Identification Agent Type (SPIA)

SPIA is responsible for identifying supplier profiles according to their historical records found in the ERP database. In a similar to CPIA manner, managers identify the valuable attributes for generating the supplier profiles. Such profiles can be consisted of value of a supplier for the company and his/her credibility. In the case there is a pool of suppliers responsible for procuring an item requested in the current processed order, SPIA as requested by RA, can retrieve all the current records of the suppliers, match them with the classified ones found inside the profile repository and return all corresponding supplier profiles, characterized on the classification attribute (e.g. credibility). Then RA can select the most appropriate one, and recommend it to the human operator of the system. Figure 2 illustrates the workflow of the SPIA, once it is instantiated. In this workflow all tasks can be detected, as described earlier in this section. The IPIA and CPIA workflows are similar to the one shown in Figure 2.

2.1.5. Inventory Profile Identification Agent Type (IPIA)

IPIA is responsible for identifying product profiles. These consist of raw data found inside the ERP (i.e. product price, related store, remaining quantities), unsophisticated processed data (for example, statistical data on product demand) and intelligent recommendations on products (such as related products that the customer may be willing to purchase). Besides the directly derived data, IPIA is responsible for identifying buying patterns Market Basket Analysis (MBA) by the use of Association Rule Extraction techniques.

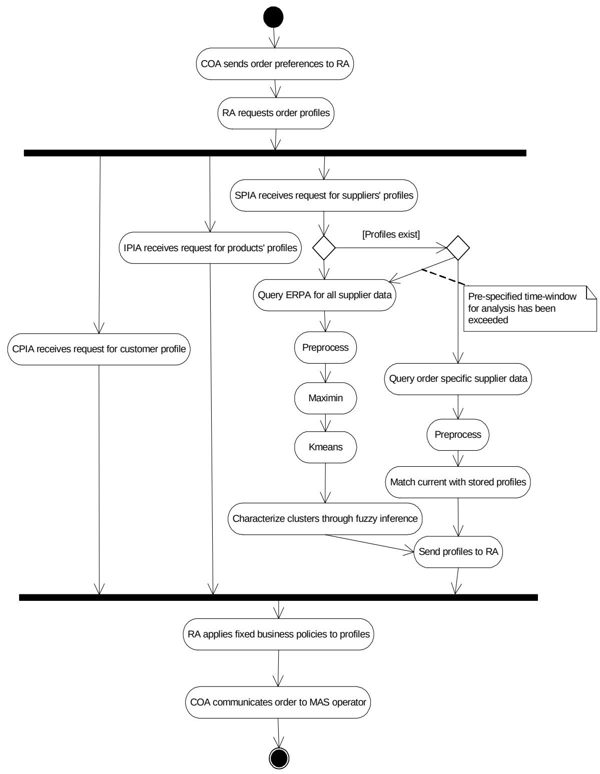

Figure 2. The workflow of SPIA

2.1.6. Enterprise Resource Planning Agent Type (ERPA)

ERP Agents provide the middleware between the MAS application and the ERP system. These agents can be resembled to transducers [14], because they are responsible for transforming data from heterogeneous applications into agent comprehensible message formats. ERPA handles all queries posted by CPIAs, IPIAs, and SPIAs by connecting to the ERP database and fetching all the required data. It works in close cooperation with an XML connector used for sending XML-SQL queries to the ERP and receiving data in XML format. ERPAs are the only agent types that need to be configured properly, in order to meet the connection requirements of different ERP systems.

2.1.7. Technologies adopted

IRF has been developed with the use of Agent Academy (AA) [15], a platform for developing MAS architectures and for enhancing their functionality and intelligence through the use of DM techniques (Figure 3). All the agents are developed over the Java Agent Development Framework (JADE) [16], which conforms to the FIPA specifications, while the required ontologies have been developed through the Agent Factory module of AA. Data mining has been performed on ERP data that are imported to AA in XML format, and are forwarded to the Data Miner (DMM) of AA, a DM suite that expands the Waikato Environment for Knowledge Analysis (WEKA) tool [17].

The extracted knowledge structures are represented in PMML (Predictive Model Markup Language), a language that efficiently describes clustering, classification and association rule knowledge models. The resulting knowledge has been incorporated into the agents by the use of the Agent Training Module (ATM) of AA. All necessary data files (ERP data, agent behavior data, knowledge structures, agent ontologies) are stored into AA's main database, the Agent Use Repository (AUR). Agents can be periodically recalled for retraining, since appropriate agent tracking tools have been incorporated into Agent Academy, in order to monitor agent activity after their deployment.

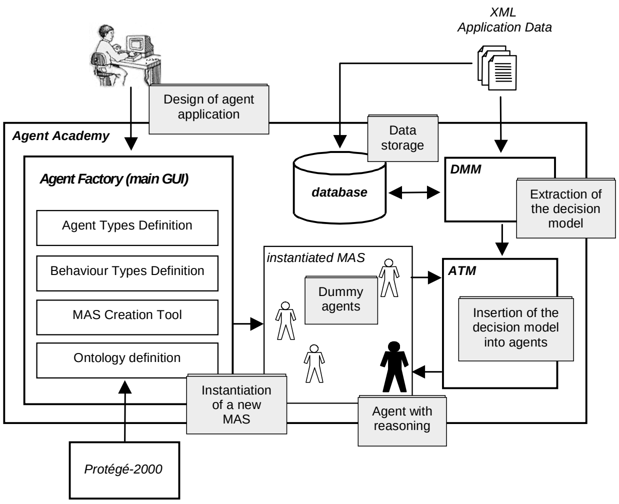

Figure 3. The Agent Academy platform

2.2 Embedded Intelligence

2.2.1 Benchmarking customer and suppliers

In order to perform customer and supplier segregation, CPIA and SPIA use a hybrid approach that employs DM and SC methodologies. Clustering techniques and fuzzy inferencing are adopted, in order to decide on customer and supplier "quality". Initially, the human experts select the attributes on which the profile extraction procedures will be based on. These attributes can either be socio-demographic, managerial or financial data, deterministic or probabilistic. We represent the deterministic attributes, which are directly extracted from the ERP database by ERPA, as D_i, $i=0..n$, where n is the cardinality of the selected deterministic attributes. On the other hand, we represent the average (AVG) and standard deviation values (STD) of probabilistic variables, which are calculated by ERPA, as AVG_j and STD_j, $j=0..m$, where m is the cardinality of the selected probabilistic attributes P_j.

Each customer/supplier is thus represented by a tuple:

$$<D_0,\dots D_n, AVG_0, STD_0,\dots, AVG_m, STD_m)>,\ i=0..n,\ j=0..m,\ i+j>0..$$

Since real-world databases contain missing, unknown and erroneous data, ERPA preprocesses data prior to sending the corresponding datasets to the Information Processing Layer Agents. Indicative preprocessing tasks are tuple omission and filling of missing values.

After the datasets have been preprocessed by ERPA, they are forwarded to CPIA and SPIA. Clustering is performed in order to separate customers/suppliers into distinct groups. This way K disjoint customer/supplier clusters are created.

In order to decide on customer/supplier clusters' added-value, CPIA and SPIA employ an Adaptive Fuzzy Logic Inference Engine (AFLIE), which characterizes the already created clusters with respect to an outcome defined by company managers, i.e. supplier credibility. Domain knowledge is incorporated into AFLIE, providing to IRF the capability of characterization.

The attributes of the resulting clusters are the inputs of AFLIE, and may have positive (➚) or negative (➘) preferred tendencies, according to their beneficiary or harmful impact on company revenue. Once domain knowledge is introduced to AFLIE in the form of preferred tendencies and desired outputs, the attributes are fuzzified according to Table 1. A [LOW, MEDIUM, HIGH] value range and triangular membership functions have been adopted for the inputs and outputs of the system, whereas maximum defuzzification is used for crisping the produced fuzzy rules.

The probabilistic variables are handled in an adaptive way, and are used as inputs only when Chebyshev's inequality (1) is satisfied:

$$P\{|P_j - AVG_j| \geq \varepsilon\} \leq \frac{(STD_j)^2}{\varepsilon^2}, \text{ for any } \varepsilon>0 \qquad (1)$$

Eq. (1) ensures the concentration of probabilistic variables near their mean value, in the interval $(AVG_j - \varepsilon, AVG_j + \varepsilon)$. No attributes with high distribution are taken as inputs to the final inference procedure, avoiding therefore decision polarization.

The formulation of the inputs (3 fuzzy values: [*LOW, MEDIUM, HIGH*]) leads to 3^v *Fuzzy Rules* (FR), where v is the number of AFLIE inputs. FRs are of type:

If X_1 is $LX_1(k)$ and X_2 is $LX_2(k)$ and...and X_n is $LX_n(k)$

Then Y is LY(l), k=1..3, l=1..q,

where X_i are the system inputs, $LX_i(k)$ the corresponding fuzzy values, Y is the system output, $LY(q)$ the corresponding fuzzy value and q is the cardinality of the fuzzy values of the output.

All inputs are assigned a *Corresponding Value* (*CV*), ranging from -1 to 1, according to their company benefit criterion (Table 1). The *Output Value* (*OV*) of Y is then calculated for each FR as:

$$OV = \sum_{i=1..n+m} w_i \cdot CV_i \tag{2}$$

where w_i is the weight of importance ($0 \le w_i \le 1$) of the i^{th} input attribute.

The *OVs* are mapped to *Fuzzy Values* (*FV*), according to the degree of discrimination of the output decision variables. By categorizing the range of the output into q fuzzy values, the $OV \rightarrow FV$ mapping is based on the following formula:

$$FV(OV) = RND\left[(OV \cdot [\frac{2(n+m)}{q}]) \right] \tag{3}$$

where $RND[x]$ is the well-known approximation to-the-closest-integer function. The *FV* values, which vary from 1 to q, are mapped to the corresponding output fuzzy values (i.e. MEDIUM for x=3, MEDIUM_HIGH for x=4 etc).

Table 1. Fuzzy variable definition and Interestingness of dataset attributes

Variable		**Fuzzy Tuple**
Input	**Preferred Tendency**	
D_i	↗	<D_i, [LOW, MEDIUM, HIGH], [D_{i1}, D_{i2}], Triangular>
D_i	↘	<D_i, [LOW, MEDIUM, HIGH], [D_{i1}, D_{i2}], Triangular>
AVG_j	↗	<AVG_j, [LOW, MEDIUM, HIGH], [AVG_{j1}, AVG_{j2}], Triangular>
AVG_j	↘	<AVG_j, [LOW, MEDIUM, HIGH], [AVG_{j1}, AVG_{j2}], Triangular>
Output	**Value Range**	
Y	Varies from Y_1 to Y_2 with a step of x	<Y, [#($Y_2 - Y_1$)/x Incremental Fuzzy Values], [Y_1, Y_2], Triangular>

After all clusters have been characterized, the corresponding *OVs*, along with the cluster centers are stored inside a profile repository for posterior retrieval. This process signals the end of the training phase of CPIA and SPIA.

In real time, when a new order comes into the system, RA requests the corresponding customer profile and the profiles of the suppliers that are related to the ordered products. CPIA and SPIA request, in turn, the attributes of these entities from ERPA, and match them with the profiles stored inside the profile repository, by the use of the *Assigned Cluster* (*AC*) criterion, a closeness-to-cluster-centre function (Euklidean distance). The winning cluster along with its *OV* is returned to RA.

2.2.2 IPIA products profile

The IPIA has a dual role in the system:

1. It fetches information on price, stock, statistical data about demand faced by the ordered products, and
2. It provides recommendations on additional items to buy, based on association rule extraction techniques.

In order to provide adaptive recommendations on ordering habits, IPIA incorporates knowledge extracted by the use of association rule extraction techniques. The Apriori algorithm is used [18] and the rules extracted are stored inside the profile repository for later retrieval. The recommendations of IPIA, as well as the information concerning stock availability and price, are sent to the RA.

2.2.3 Intelligence of RA

As earlier prescribed, RA is an expert agent that incorporates fixed business policies applied to customers, inventories and suppliers. These rules are related, not only to raw data retrieved from the ERP database and order preferences provided by customers, but also to the extracted knowledge provided by the Information Processing agents. There are three distinct rule types that RA can realize:

1. Simple <*If … Then …*> statements,
2. Rules describing mathematical formulas, and
3. Rules providing solutions to search problems and constraint satisfaction problems.

<u>Example 1 - Simple Rules:</u> Additional discounts or burdens to the total price of an order can be implemented by the use of simple rules (knowledge extracted is denoted in bold):

```
(a) IF (TotalOrderRevenue >= 1000 €) AND
       (CustomerValue = HIGH)
       THEN TotalDiscount += 5%;

(b) IF (RecommendedProductsPurchased = True)
       THEN ProductDiscount +=5%;
```

<u>Example 2 - Mathematical Formulas:</u> The re-order/order-up-to-level-point metric (sS) provides efficient inventory management for either no-fixed cost orders or fixed order costs orders [1]. In the case of no-fixed cost orders (where *s=S),* the reorder point is calculated as:

$$\begin{aligned} sS &= AVGD \cdot AVGL + \\ &+ z \cdot \sqrt{AVGL \cdot STDD^2 + AVGD^2 \cdot STDL^2} \end{aligned} \tag{4}$$

<u>Example 3 - Problem Searching:</u> Problems that require applying heuristics and satisfying constraints: In the case of Supplier and quantity selection decisions, RA can, based on raw data from the ERP and on knowledge provided by SPIA, provide solutions to problems like the selection of the most appropriate supplier with respect to his/her added-value, location to the depleted storage, or

the identification and application of an established contract.

3. Conclusion

An ERP system, although undoubtedly beneficiary is a costly investment and the process of replacing, updating or adding customization modules to it is unaffordable, especially for SMEs. Through our approach we attempt to overcome the already mentioned deficiencies of non DS-enabled ERP systems, by incorporating versatile and adaptable knowledge inside companies' CRM/SRM solutions in a low-cost, yet effective manner. IRF establishes an efficient, quick and easy way of providing intelligent recommendations to the incoming requests for quotes a customer makes, therefore providing a number of enhancements in an integrated way. Recommendations are autonomously adapted, without having an impact at IRF run-time performance. The implementation of IRF through the AA platform dramatically decreases development costs and efforts, while the framework architecture ensures reusability and easy re-configurability, with respect to the – in each case – underlying ERP. Table 3 summarizes the key enhancements provided by the integration of IRF with ERP systems.

Table 3. Specific enhancements provided by the use of IRF over ERP systems.

	IRF + ERP	**Legacy ERPs**
Fixed Business Rules 1	**Yes /** Provided as rule documents changed on the fly.	**Yes /** Incorporated into the source code by the ERP vendor.
Fixed Business Rules 2	Applied to data + knowledge	Applied only to data
MBA	**Yes**	**No**
Recommendation Procedure	Automatically generated	Through reports
Inventory Management	Automatically adapted	Manually if applicable
Decision cycle-time	**Small**	**High**
Cust. / Sup. Intelligent Evaluation	**Yes**	**No** (Unless special modules incorporated)
Information Overload	**Low**	**High** (Through reports)
Cost of enhancement	**Low** (Use of AA platform)	**High** (Customization/third party DS COTS)

Further research work will be focused on automated negotiation strategies for B2B commerce, for enhancing IRF. Moreover we will try to optimize the performance of our inductive recommendation procedure, by simulating a variety of supply-chain scenarios.

4. Acknowledgements

Work presented here has been partially supported by the European Commission through the IST initiative (IST project No 2000-31050).

References:

[1] S. D. Levi, P. Kaminsky & S. E. Levi, Designing and managing the supply chain. (McGraw-Hill, Illinois, 2000).
[2] J. Shapiro, Bottom-up vs. top-down approaches to supply chain modeling , in: S. Tayur, R. Ganeshan & M. Magazine, eds., Quantitative models for supply chain management, (Kluwer, 1999), 737-759.
[3] T. H. Davenport, The future of enterprise system-enabled organizations, Information Systems Frontiers, 2(2), 2000, 163-180.
[4] C. Carlsson & E. Turban, DSS: directions for the next decade, Decision Support Systems 33, 2002, 105-110.
[5] C. W. Holsapple & M. P. Sena, ERP plans and decision-support benefits, Decision Support Systems, article in press, 2003.
[6] K. L. Choy, W. B. Lee & V. Lo, Development of a case based intelligent customer-supplier relationship management system, Expert Systems with Applications, 23(3), 2002, 281-297.
[7] M. Wooldridge, Intelligent Agents, in: Multiagent Systems, G. Weiss, ed., (The MIT Press, 1999).
[8] U. Fayyad, G. Piatetsky-Shapiro & P. Smyth, Knowledge Discovery and Data Mining: Towards a unifying framework, Proceedings of The Second International Conference on Knowledge Discovery and Data Mining, (Portland, 1996), 82-88.
[9] S. H. Haeckel & R. Nolan, Managing by wire, Harvard Business Review, September – October, 1994, 122-132.
[10] T. W. Malone, Inventing the organizations of the twentieth first century: control, empowerment and information technology, in: S. P. Bradley & R. L. Nolan, eds., Sense and Respond: Capturing Value in the Network Era, (Harvard Business School Press, Boston MA, 1998), 263-284.
[11] C. Rygielski, J. C. Wang & D.C. Yen, Data mining techniques for customer relationship management, Technology in Society, 24(4), 2002, 483-502.
[12] O. B. Kwon & J.J. Lee, A multi agent intelligent system for efficient ERP maintenance, Expert System with Applications, 21, 2001, 191-202.
[13] Y. Peng, T. Finin, Y. Labrou, B. Chu, J. Long, W. Tolone & A. Boughannam, A multi agent system for enterprise integration, Applied Artificial Intelligence, 13(1-2), 1999, 39-63.
[14] M. R. Genesereth & S. Ketchpel, Software agents, Communications of the ACM, 37(7), 1994, 48-53.
[15] Agent Academy Consortium, the, The Agent Academy Project, http://agentacademy.iti.gr (2004).
[16] F. Bellifemine, A. Poggi & G. Rimassa, Developing multi-agent systems with JADE, in: the Seventh International Workshop on Agent Theories, Architectures, and Languages (ATAL-2000, Boston MA), http://jade.cselt.it.
[17] I. H. Witten & E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, (Morgan Kaufmann, New Zealand, 1999).
[18] A. Amir, R. Feldman & R. Kashi, A new and versatile method for association generation,. Information Systems, 22 (6-7), 1997, 333-347.