Scalable Computing: Practice and Experience
Volume 8, Number 1, pp. 101–114. http://www.scpe.org
ISSN 1895-1767

A MULTI-AGENT INFRASTRUCTURE FOR ENHANCING ERP SYSTEM INTELLIGENCE

ANDREAS L. SYMEONIDIS[†‡*], KYRIAKOS C. CHATZIDIMITRIOU[†], DIONYSIOS KEHAGIAS[‡], AND PERICLES A. MITKAS[†‡]

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. In this paper 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 architecture and development details of the developed framework, and demonstrate its application on a real test case.

Key words. ERP systems, Data Mining, Soft Computing, Multi-Agent Systems, Adaptive Decision Making

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 [16].

Despite the support current ERP systems provide on process coordination and data organization, most of them – especially legacy systems – lack advanced Decision-Support (DS) capabilities, resulting therefore in decreased company competitiveness. In addition, from a functionality perspective, most ERP systems are limited 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 [25]. 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 [7] 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 [3] 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 by Holsapple and Senna [14]. The most significant, among others, are:

1. Enhancement of the decision maker's ability to process knowledge.
2. Improvement of reliability of the decision support processes.
3. Provision of evidence in support of a decision.
4. Improvement or sustainability of organizational competitiveness.
5. Reduction of effort and time associated with decision-making, and
6. 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 [5, 31]. This is why alternative technologies, such as Data Mining and Agent Technology 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 [30] can lead to less development effort, while the cooperation primitives that Agent Technology adopts point to MAS as the best choice for addressing complex tasks in systems that require synergy of multiple entities. Moreover, DM has repeatedly been

*Corresponding author
†Electrical and Computer Engineering Dept., Aristotle University of Thessaloniki, GR541 24, Thessaloniki, GREECE
‡Intelligent Systems and Software Engineering Laboratory, Informatics and Telematics Institute - CERTH, GR570 01,Thessaloniki, GREECE, {asymeon,diok}@iti.gr, kyrxa@ee.auth.gr, mitkas@auth.gr

used for Market Trend Analysis, User Segmentation, and Forecasting. Knowledge derived from the application of DM techniques on existing ERP historical data can provide managers with useful information, which may enhance their decision-making capabilities.

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. [24] have exploited DM techniques for Customer Relationship Management (CRM), while Choy et. al. [4, 5] 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 [22], and for the identification and maintenance of oversights and malfunctions inside the ERP systems [15].

Elaborating on previous work, we have integrated AT and DM advantages into a versatile and adaptive multi-agent system that acts as an add-on to established ERP systems. Our approach employs Soft Computing, DM, Expert Systems, standard Supply Chain Management (SCM) and AT primitives, in order to provide intelligent recommendations on customer, supplier, and inventory issues. The system is designated to assist not only the managers of a company – "Managing by wire" approach [12] –, but also the lower-level, distributed decision makers – "Cowboys" approach [18]. Our framework utilizes the vast amount of corporate data stored inside ERP systems to produce knowledge, by applying data mining techniques on them. The extracted knowledge is diffused to all interested parties via the multi-agent architecture, while domain knowledge and business rules are incorporated into the system by the use of rule-based agents. It merges the, already proven capabilities of data mining with the advantages of multi-agent systems in terms of autonomy and flexibility, and therefore promises a great likelihood of success.

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. Section 3 illustrates the basic functional operations of IPRA, an already developed add-on in a real enterprise environment. Finally, Section 4 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 processing. The order processing 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. These recommendations are not designed to substitute the human operator, rather to aid him/her and the company to increase profit and efficiently manage customer orders and company supplies.

2.1. IRF Architecture. The general IRF architecture is illustrated in Figure 2.1. The IRF agents belong to one of six different agent types ($Q_1 - Q_6$) and are listed in Table 2.1. The main characteristics and the functionality of each type are discussed in the following paragraphs.

Table 2.1

The IRF agent types and their functionality

Agent type	Name	Functionality
Q_1	COA – Customer Order Agent	GUI agent
Q_2	RA – Recommendation Agent	Organization & Decision Making agent
Q_3	CPIA – Customer Profile Identification Agent	Knowledge Extraction agent
Q_4	SPIA – Supplier Profile Identification Agent	Knowledge Extraction agent
Q_5	IPIA – Inventory Profile Identification Agent	Knowledge Extraction agent
Q_6	ERPA – Enterprise Resource Planning Agent	Interface agent

2.1.1. Customer Order Agent type (COA). COA is an interface agent that may operate at the distribution points, or at the telephone center of an enterprise. COA enables the system operator to: a) transfer information into and out of the system, b) input order details into the system, and c) justify, by means of visualization tools, the proposed recommendations. When an order arrives into the system, COA provides the

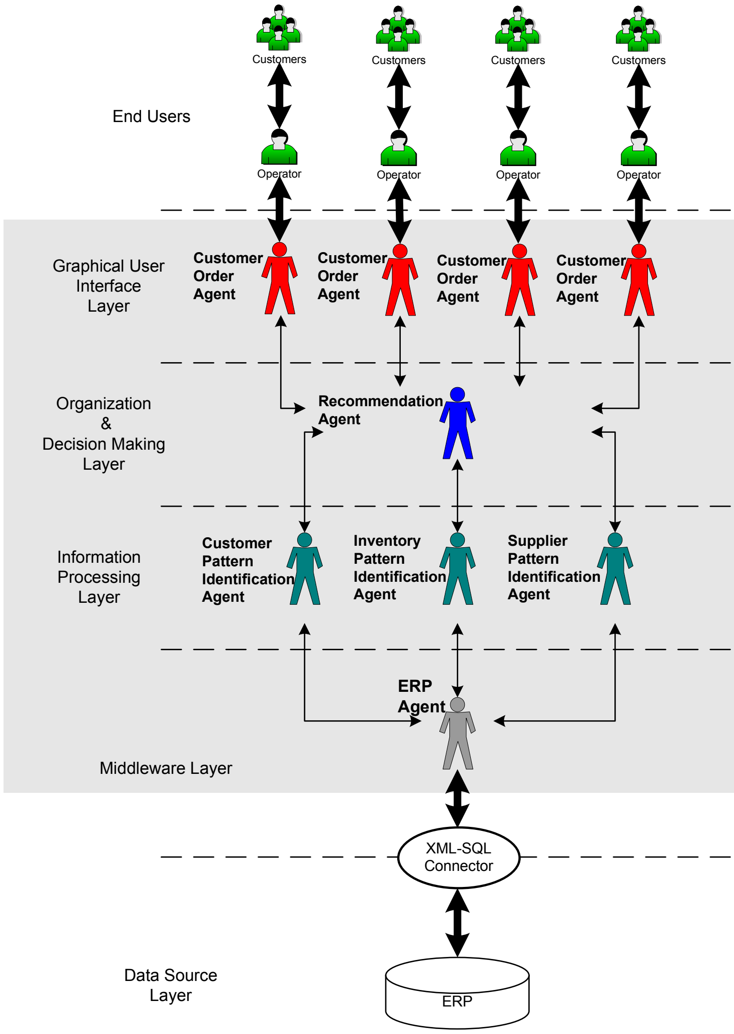

Fig. 2.1. *The IRF architectural diagram*

human agent with basic functionalities for inserting information on the customer, the order details (products and their corresponding quantities), payment terms (cash, check, credit etc.), backorder policies and, finally, the party (client or company) responsible for transportation costs. COA also encompasses a unit that displays information in various forms to explain and justify the recommendations issued by the RA.

2.1.2. Recommendation Agent type (RA). The RA is responsible for gathering the profiles of the entities involved in the current order and for issuing recommendations. By distributing the profile requests to the appropriate *Information Processing Layer* agents (CPIA, SPIA and IPIA - each one of them operating on its own control thread), and by exercising concurrency control, this agent diminishes the cycle-time of the recommendation process. RA is a rule-based agent implemented using the Java Expert System Shell (JESS) [9]. Static and dynamic business rules can be incorporated into the RA. The latter must be written into a document that the agent can read during its execution phase. In this way, business rules can be modified on-the-fly, without the need of recompiling, or even restarting the application.

2.1.3. Customer Profile Identification Agent Type (CPIA). CPIA is designed to identify customer profiles, utilizing the historical data maintained in the ERP system. The process can be described as follows: 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 instrumental for reasoning on customer value. Then, they decide on the desired classification of customers, i.e., 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 against those in the profile repository, resulting into the identification of the group the specific customer belongs to. During the development phase, one or more CPIA agents can be instantiated, and the distinction of CPIAs into training and recommendation ones, results to quicker response times when learning and inference procedures overlap.

2.1.4. Supplier Pattern Identification Agent Type (SPIA). SPIA is responsible for identifying supplier profiles according to the historical records found in the ERP database. In a similar to CPIA manner, managers identify the key attributes for determining a supplier's value to the company and their credibility. SPIA then generates supplier profiles and updates them periodically. For every requested item in the current order, the RA identifies one or more potential suppliers and requests their profiles from the SPIA. SPIA has to retrieve the current records of all the suppliers, identify for each one the best match in the profile repository, and return the corresponding profiles to the RA. Then RA can select the most appropriate supplier combination (according to its rule engine), and recommend it to the human operator. SPIA is also responsible for fetching to RA information about a specific supplier, such as statistical data on lead-times, quantities to be procured etc.

2.1.5. Inventory Profile Identification Agent Type (IPIA). IPIA is responsible for identifying product profiles. Product profiles comprise raw data from the ERP database (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). Once more, managers and application developers have to identify the company priorities and map the profile to the data maintained by the ERP. Besides the directly–derived data, IPIA is responsible for identifying buying patterns. Market basket analysis can be performed with the help of association rule extraction techniques. Since this process is, in general, time-consuming, two or more IPIAs can be instantiated to separate the recommendation from the learning procedure.

2.1.6. Enterprise Resource Planning Agent Type (ERPA). ERPAs provide the middleware between the MAS application and the ERP system. These agents behave like transducers [11], because they are responsible for transforming data from heterogeneous applications into message formats that agents can comprehend. An ERPA handles all queries posted by CPIAs, IPIAs, and SPIAs by connecting to the ERP database and fetching all the requested data. It works in close cooperation with an XML connector which relays XML-SQL queries to the ERP and receives data in XML format. ERPA is the only IRF agent type that needs 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) [20, 27] a platform for developing MAS architectures and for enhancing their functionality and intelligence through the use of DM techniques. All the agents are developed over the Java Agent Development Framework (JADE) ([2], which conforms to the FIPA specifications [28], while the required ontologies have been developed through the Agent Factory module (AF) 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 (DM) of AA, a DM suite that expands the Waikato Environment for Knowledge Analysis (WEKA) tool [29].

The extracted knowledge structures are represented in PMML (Predictive Model Markup Language), a language that efficiently describes clustering, classification and association rule knowledge models [6]. 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.

2.2. Installation and Runtime Workflows. Once a company chooses to add IRF to its already operating ERP system, a few important steps have to be performed. The installation procedure of the IRF is shown in Figure 2.2.

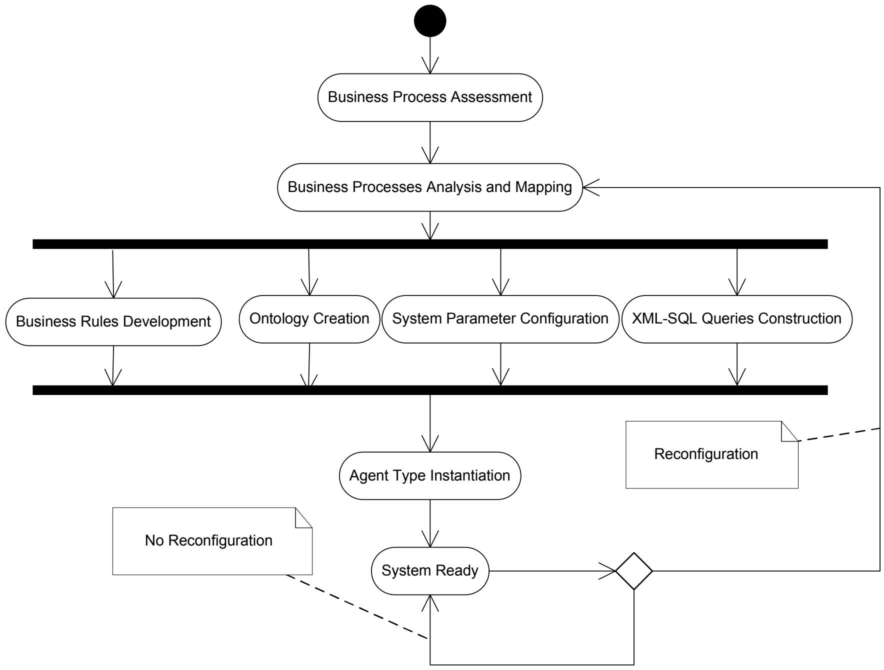

Fig. 2.2. *Installing IRF on top of an existing ERP*

At first, the company's business process expert, along with the IRF application developers have to make a detailed analysis and assessment of the current customer order, inventory and products procurement processes. The results are mapped to the recommendation process of the add-on and the relevant datasets are delineated in the ERP.

After modeling the recommendation procedure according to the needs of the company, parallel activities for producing required documents and templates for the configuration of the MAS application follow. Fixed business rules incorporating company policy are transformed to expert system rules, XML-SQL queries are built and stored in the XML documents repository, ontologies (in RDFS format) are developed for the messages exchanged and for the decision on the workflow of the agents, agent types instantiation requirements are defined (at different workstations and cardinalities) and other additional parameters are configured (i.e., simple retraining time-thresholds, parameters for the data-mining algorithms, such as support and confidence for market basket analysis etc).

Once bootstrapped, reconfiguration of the system parameters is quite easy, since all related parameters are documents that can be conveniently re-engineered. Figure 2.3 illustrates the workflow of the SPIA, where all the tasks described earlier in this section, can be detected. In case IRF needs to be modified due to a change in the company processes, the reconfiguration path must be traversed. The IPIA and CPIA workflows are similar and, thus, they are omitted.

2.3. System Intelligence.

2.3.1. Benchmarking customer and suppliers. In order to perform customer and supplier segregation, CPIA and SPIA use a hybrid approach that combines data mining and soft computing 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 Det_i, $i = 1, ...n$, 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, $j = 1..m$, where m is the cardinality of the selected probabilistic attributes P_j.

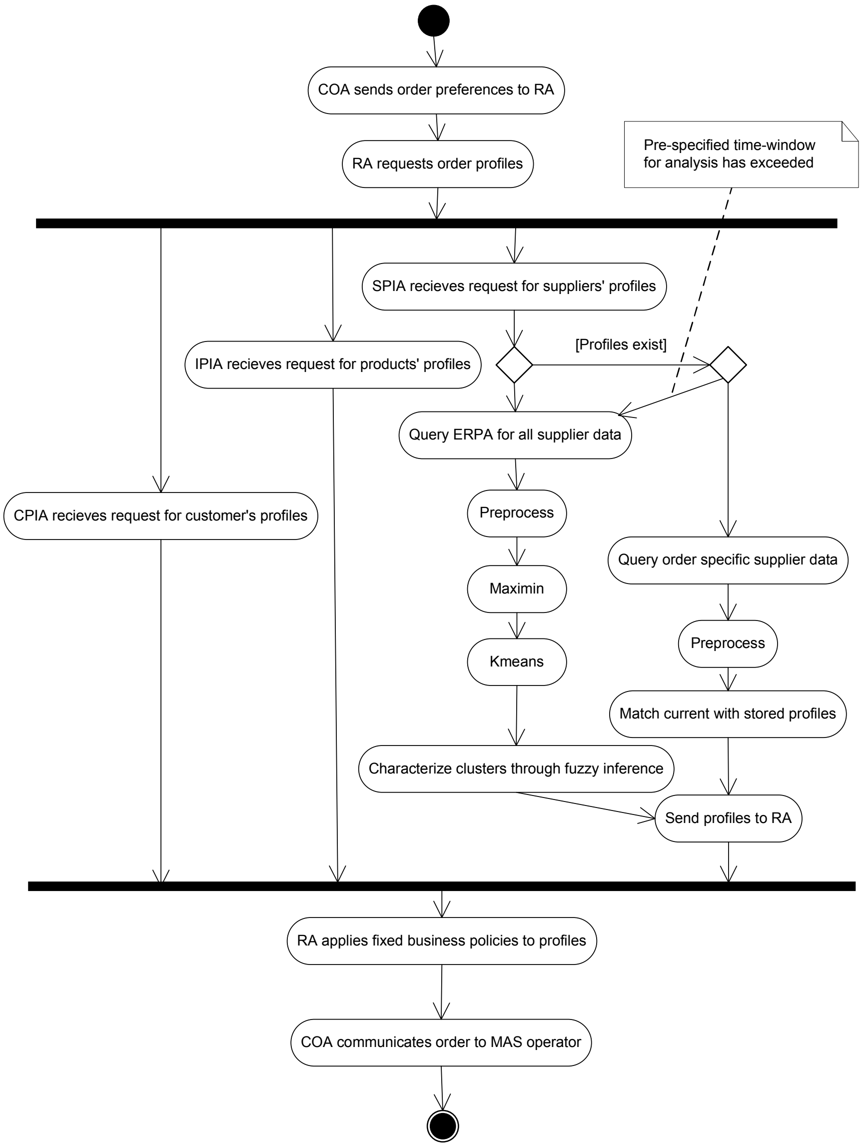

Fig. 2.3. *The Workflow of SPIA*

Each customer/supplier is thus represented by a tuple:

$$< Det_1, ..., Det_n, AVG_1, STD_1, ..., AVG_m, STD_m) > \tag{2.1}$$

where $i = 1..n$, $j = 1..m$, $i + j > 0$. Since real-world databases contain missing, unknown and erroneous data [13], ERPA preprocesses data prior to sending the corresponding datasets to the Information Processing Layer Agents. Typical 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. The Maximin algorithm [17] is used to provide the number of the centers K that are formulated by the application of the K-means algorithm [19]. 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 [8], providing to IRF the capability of characterization.

The attributes of the resulting clusters are the inputs to AFLIE and they may have positive ($\nearrow$) or negative ($\searrow$) preferred tendencies, depending on 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 2.2.

Table 2.2
Fuzzy variable definition and Interestingness of dataset attributes

Variable		**Fuzzy Tuple**
Input	**Preferred Tendency**	
Det_i	$\nearrow$	$\langle Det_i, [LOW, MEDIUM, HIGH], [Det_{i_1}, Det_{i_2}], Triangular\rangle$
Det_i	$\searrow$	$\langle Det_i, [LOW, MEDIUM, HIGH], [Det_{i_1}, Det_{i_2}], Triangular\rangle$
AVG_j	$\nearrow$	$\langle AVGj, [LOW, MEDIUM, HIGH], [AVG_{j_1}, AVG_{j_2}], Triangular\rangle$
AVG_j	$\searrow$	$\langle AVGj, [LOW, MEDIUM, HIGH], [AVG_{j_1}, AVG_{j_2}], Triangular\rangle$
Output	**Value Range**	
Y	Varies from Y_1 to Y_2 with a step of x	$\langle Y, [\#(Y_2 - Y_1)/x$ Incremental Fuzzy Values$], [Y_1, Y_2], Triangular\rangle$

The probabilistic variables are handled in an adaptive way and are used as inputs only when Chebyshev's inequality (Eq. 2.2) is satisfied [21]:

$$P\{|P_j - AVG_j|\epsilon\} \leq \frac{(STD_j)^2}{\epsilon^2}, \text{for any } \epsilon > 0 \tag{2.2}$$

Eq. 2.2 ensures the concentration of probabilistic variables near their mean value, in the interval $(AVG_j - \epsilon, AVG_j + \epsilon)$. No attributes with high distribution are taken as inputs to the final inference procedure, avoiding therefore decision polarization.

The formulation of the inputs (3 values: $[LOW, MEDIUM, HIGH]$) leads to 3^ν *Fuzzy Rules* (FR), where ν 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 q is the cardinality of the fuzzy values of the output. Triangular membership functions are adopted for all the inputs and outputs, whereas maximum defuzzification is used for crisping the FRs.

All inputs are assigned a *Corresponding Value* (CV), ranging from -1 to 1, according to their company benefit criterion (Table 2.2). The *Output Value* (OV) of Y is then calculated for each FR as:

$$OV = \sum_{i=1..n+m} w_i \cdot CV_i \tag{2.3}$$

where w_i is the weight of importance ($0 \leq w_i \leq 1$) 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 $OV \longrightarrow FV$ mapping is based on the following formula:

$$FV(OV) = RND\left[OV \cdot [\frac{2(n+m)}{q}]\right] \tag{2.4}$$

where $RND(x)$ is the rounding function of x to the closest integer (i. e., $MEDIUM$ for $x = 3$, $MEDIUM\_HIGH$ for $x = 4$ etc).

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 against the profiles stored inside the profile repository, by the use of the Assigned Cluster (AC) criterion. AC is a closeness-to-cluster-center function, given by the following equation:

$$AC = \min_{i=1..k}\{\sqrt{\sum_{i=1}^{n+m}(c_i - xc_{ji})}\} \tag{2.5}$$

where k is the number of clusters, n the number of attributes, c_i is the i^{th} attribute value of the cluster center vector $c = (c_1, c_2, ..., c_n)$, and xc_{ij} the i^{th} attribute value of the j^{th} current vector $xc_j = (xc_{j1}, xc_{j2}, ..., xc_{jn})$. The winning cluster along with its OV is returned to RA.

2.3.2. IPIA products profile. The IPIA plays 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 Apriori algorithm ([1, 10]. The association rules extracted are stored inside the profile repository for later retrieval.

Special attention should be drawn to the fact that the transactions included into the dataset to be mined may span several different customer order periods. XML-SQL queries can be adapted to perform data mining either to the whole dataset or the datasets of specific periods. Thus, IPIA is highly adaptable, both for companies in the general merchandize domain, but also for companies that sell seasonal goods (for example toys). The recommendations of IPIA, as well as the information concerning stock availability and price, are sent to the RA.

2.3.3. RA Intelligence. As mentioned earlier, 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 $\langle If \ldots Then \ldots\rangle$ statements,
2. Rules describing mathematical formulas, and
3. Rules providing solutions to search problems and constraint satisfaction problems.

An example is provided below for each one of these rule types:

Example 1: Simple Rules

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):

1. IF ($TotalOrderRevenue >= 100$) AND (**CustomerValue** $= LOW$) THEN $TotalDiscount+ = 5\%$;
2. IF (**CustomerValue** $= LOW$) THEN $TotalDiscount- = 5\%$;
3. IF ($ProductType = ChristmasProducts$) AND ($TotalQuantity >= 100$) THEN **ProductDiscount**$+ = 10\%$;
4. IF (**RecommendedProductsPurchased** $= True$) THEN $ProductDiscount+ = 5\%$;

Example 2: Mathematical Formulas

(a) Re-order/Order-up-to-level metric sS

The re-order/order-up-to-level-point metric (sS) provides efficient inventory management for either no-fixed cost orders or fixed cost orders [16]. In the case of no-fixed cost orders (where $s = S$), the reorder point is calculated as:

$$sS = AVGD \cdot AVGL + z \cdot \sqrt{AVGL \cdot STDD^2 + AVGD^2 \cdot STDL^2} \tag{2.6}$$

where z is a constant chosen from statistical tables to ensure the satisfaction of a pre-specified value for the company's service level. Table 2.3 illustrates the value of z in correlation with the desired service level. In most legacy ERP systems such attributes have to be provided by users and cannot be derived automatically.

Table 2.3
Service Level and corresponding z Value

Service Level	90%	91%	92%	93%	94%	95%	96%	97%	98%	99%	99.9%
z	1.29	1.34	1.41	1.48	1.56	1.65	1.75	1.88	2.05	2.33	3.08

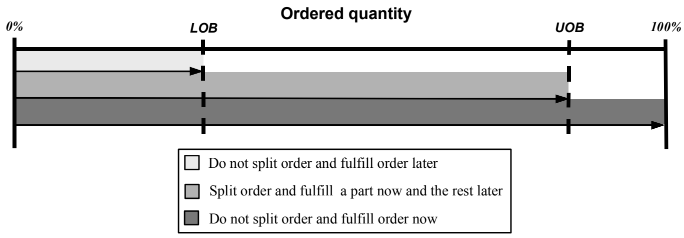

Fig. 2.4. *RA order splitting policy*

(b) Splitting Policy

A splitting policy is applied when company stock availability cannot satisfy order needs. Upon arrival of a new order, the quantity of ordered items and available stock are cross-checked. If the requested quantities are available, the order is fulfilled immediately. Otherwise, the final supplying policy that the RA recommends is set according to the schema illustrated in Figure 2.4.

The LOB and UOB thresholds depend on the estimated customer value. In case we choose to incorporate product discount and customer priority into our splitting policy (for example, customers that enjoy better discount and have a higher priority to have a lower LOB and an higher UOB), we may adjust LOB and UOB according to the following equations:

$$LOB = \alpha_l \cdot \exp[-(b_{pl}\hat{p} + b_{dl}\hat{d})] \tag{2.7}$$

$$UOB = \alpha_u \cdot \exp(b_{pu}\hat{p} + b_{du}\hat{d}) \tag{2.8}$$

where $\hat{p}$ is the priority normalized factor, $\hat{d}$ is the discount normalized factor, while the weighting factors $\langle \alpha_l, b_{pl}, b_{dl}, \alpha_u, b_{pu}, b_{du} \rangle$ are estimated in order to satisfy minimal requirements on LOB and UOB range.

If available stock is below $LOB\%$ of the ordered quantity, the entire order is put on hold until the company is supplied with adequate quantities of the ordered item. When item availability falls within the $[LOB-UOB]\%$ range of the ordered quantity, the order is split. All available stock is immediately delivered to the customer, whereas the rest is ordered from the appropriate suppliers. Finally, in case the available stock exceeds $UOB\%$ of the ordered quantity, the order is immediately preprocessed and the remaining order percentage is ignored.

Example 3: Problem Searching

(a) Problems that require heuristics application and/or constraint satisfaction

Based on raw data from the ERP and on knowledge provided by SPIA, Recommendation Agents can yield solutions to problems like the selection of the most appropriate supplier with respect to their added-value, proximity to the depleted company store, or the identification and application of an established contract.

(b) Enhanced Customer Relationship Management

Using the knowledge obtained by customer clustering, RA can implement a variety of targeted discount strategies in the form of crisp rules. Thus, the company has additional flexibility in its efforts to retain valuable customers and entice new ones with attractive offers [23].

Table 2.4
IPRA inputs and outputs

CPIA		**SPIA**		**IPIA**	**RA**
Input	**Preferred Tendency**	**Input**	**Preferred Tendency**	**Input**	**Input**
Account balance	↘	Account balance	↘	Stock Availability	Ordered Quantity
Credit Limit	↗	Credit Limit	↗	Item price	Stock Availability
Turnover	↗	Turnover	↗	Supplier ids	Re-order metric
Average Order Periodicity	↘	Average Order Completion	↘	Average Item Turnover (AIT) for the last two years	Supplier Geographic Location
Standard deviation of Order	-	Standard deviation of Order	-	Monthly Standard Deviation of	Lower Order Break-point
Average Order Income	↗	Average Payment Terms	↘		Upper Order Break-point
Standard deviation of Order Income	-	Standard deviation of Payment Terms	-		Customer Geographic Location
Average Payment Terms	↘	Supplier Geographic Location	↘		
Standard deviation of Payment Terms	-				
Customer Geographic Location	↘				
IPRA Outputs					
Output	**Value Range**	**Output**	**Value Range**	**Output**	**Output**
DISCOUNT	Varies from 0 – 30%, using a step of 5%	CREDIBILITY	Ranging from 0 – 1, using a step based on the number of supplier clusters	PROPOSED ORDER ITEMS	SPLITTING POLICY
PRIORITY	Varies from 0 – 3, using a step of 1				ADDITIONAL DISCOUNT
					CUSTOMER STATISTICS

3. An IRF Demonstrator. In order to demonstrate the efficiency of IRF, we have developed IPRA [26], an Intelligent Recommender module that employs the methodology presented in Chapter 5. The system was integrated into the IT environment of a large retailer in the Greek market, hosting an ERP system with a sufficiently large data repository. IPRA was slightly customized to facilitate access to the existing Oracle™ database.

Our system proved itself capable of managing over 25.000 transaction records, resulting in the extraction of truly "smart" suggestions. The CPIA and the SPIA performed clustering of over 8.000 customers (D_{IQ_3} dataset) and 500 suppliers (D_{IQ_4} dataset), respectively, while IPIA performed association rule extraction on

14125 customer transactions (D_{IQ_5} dataset).

All the attributes used by the Information Processing agents as inputs for DM, their corresponding preferred tendency, the inputs of the RA JESS engine, as well as the outputs of the IPRA system and their value range, are listed in Table 2.4.

The Information Processing agents of IPRA, in order to provide RA with valid customer and supplier clusters, as well as interesting additional order items, performed DM on the relevant datasets. For the specific company, CPIA and SPIA have identified each five major clusters representing an equal number of customer and supplier groups, respectively. Resulting customer (supplier) clusters, as well as the discount and priority (credibility), calculated by the CPIA (SPIA) Fuzzy Inference Engine for each cluster, are illustrated in Table 3.1 and Table 3.2.

Table 3.1
The resulting customer clusters and the corresponding Discount and Priority values

Center ID	**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

Table 3.2
The resulting supplier clusters and the corresponding Supplier Value towards the company

Center ID	**Population (%)**	**Value**
0	15.203	Low
1	10.112	Medium
2	25.646	Low
3	34.521	Medium
4	13.518	High

Table 3.3
The generated association rules with the predefined support and confidence thresholds.

Generated Rules	Support	Confidence
25	2%	90%
10	4%	90%

IPIA, on the other hand, has extracted a number of association rules from the records of previous orders, as shown in Table 3.3.

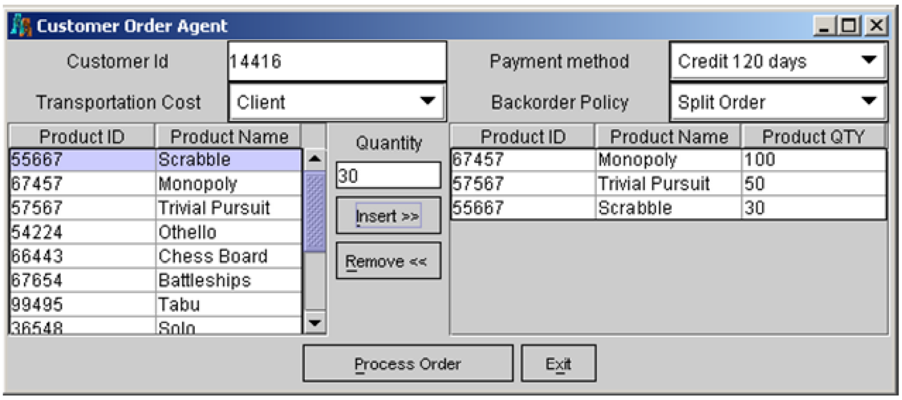

Fig. 3.1. *GUI of Customer Order Agent with information on the new order*

As already mentioned, upon receiving an order, the human agent collects all the necessary information, in order to provide IPRA with input. Data collected are handled by COA, the GUI agent of the system. An instance of the GUI is illustrated in Figure 3.1.

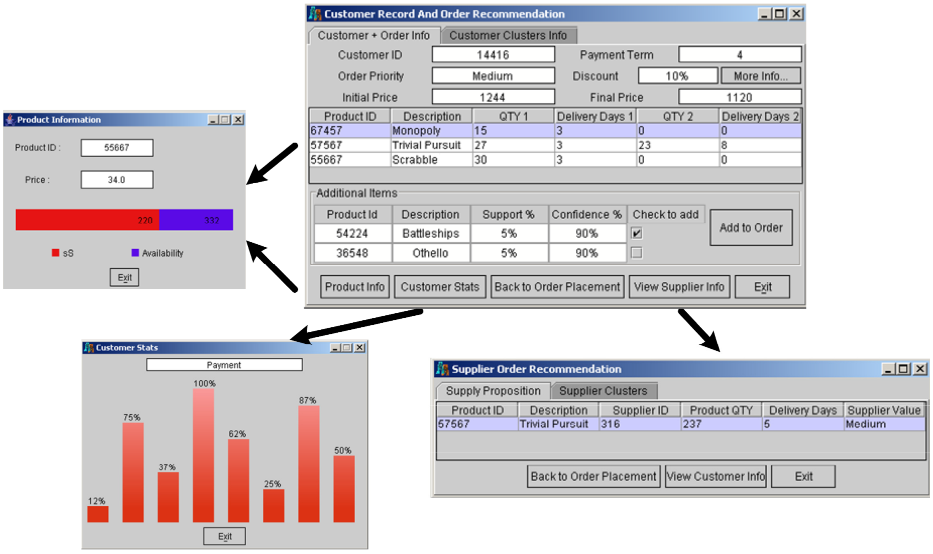

Fig. 3.2. *The final IPRA Recommendation*

All information on items and quantities to be ordered, backorder policy, payment method, and transportation costs are given as input to IPRA. When the order process is initialized, COA forwards to the CPIA, SPIA, IPIA and RA respectively the already collected information. CPIA checks on the cluster the client falls into, SPIA decides on the best supplier, (according to his/her added-value), in case an order has to be placed to satisfy customer demand, IPIA proposes additional items for the customer to order, and all these decisions are passed on to the RA, which decides on the splitting policy, (if needed) and on additional discount.

Figure 3.2 illustrates the final recommendation created. Detailed information on the order and its products, customer suggested priority and discount, customer clusters, supplier suggested value and supplier clusters, additional order items, suggested order policy and statistics, are at the disposal of the human agent, to evaluate and realize the transaction at the maximal benefit of the company.

4. Conclusions. An ERP system, although indispensable, constitutes a costly investment and the process of updating business rules or adding customization modules to it is often unaffordable, especially for SMEs. The IRF methodology aspires to overcome the already mentioned deficiencies of non DS-enabled ERP systems, in a low-cost yet efficient manner. Knowledge residing in a company's ERP can be identified and dynamically incorporated into versatile and adaptable CRM/SRM solutions. IRF integrates a number of enhancements into a convenient package and establishes an expedient vehicle for providing intelligent recommendations to incoming customer orders and requests for quotes. Recommendations are independently and perpetually adapted, without an adverse impact on IRF run-time performance. IRF architecture ensures reusability and re-configurability, with respect to the underlying ERP. Table 3.4 summarizes the key enhancements provided by the augmentation of ERP systems with the IRF module.

REFERENCES

[1] A. Amir, R. Feldman, and R. Kashi, *A new and versatile method for association generation*, Information Systems, 22 (1999), pp. 333–347.

[2] F. Bellifemine, A. Poggi, and R. Rimassa, *Developing multi-agent systems with JADE*, Lecture Notes in Computer Science, 1986 (2001), pp. 89–101.

[3] C. Carlsson and E. Turban, *Dss: directions for the next decade*, Decision Support Systems, 33 (2002), pp. 105–110.

[4] K. L. Choy, B. Lee, and V. Lo, *Design of an intelligent supplier relationship management system: a hybrid case based neural network approach*, Expert Systems with Applications, 24 (2003), pp. 225–237.

Table 3.4
IRF enhancements to ERPs

	IRF + ERP	Legacy ERPs
Static Business Rules	**Yes** Provided as rule documents changed on the fly.	**Yes** Hard-coded by the ERP vendor.
Dynamic Business Rules	Applied to data + knowledge	Applied only to data
Market Basket Analysis	**Yes** Added online to the recommendation procedure	**No** (Unless external MBA is performed)
Recommendation Procedure	Automatically generated	Through reports
Inventory Management	Thresholds automatically adapted	Thresholds inserted manually if applicable (Unless SCM module incorporated)
Decision cycle-time	**Short** (Not related to database size)	**Long** (Related to database size)
Distributed Decision Making	**Yes** Recommendations can be used by lower level personnel	**No**
Adaptability	**High**	**Low**
Autonomy	**Yes**	**No**
Customers Intelligent Evaluation	**Yes**	**No** (Unless CRM module incorporated)
Suppliers Intelligent Evaluation	**Yes**	**No** (Unless SRM module incorporated)
Information Overload Reduction	**High**	**Small** (Through reports)
Cost of enhancement	**Low** (Use of AA platform)	**High** (Customization/third party DS COTS)

[5] K. L. Choy, W. B. Lee, and V. Lo, *Development of a case based intelligent customer-supplier relationship management system*, Expert Systems with Applications, 23 (2002), pp. 281–297.
[6] Data Mining Group, *Predictive model markup language specifications (pmml), ver. 2.0.*, tech. report, The DMG Consortium, 2001.
[7] T. H. Davenport, *The future of enterprise system-enabled organizations*, Information Systems Frontiers, 2 (2000), pp. 163–180.
[8] A. A. Freitas, *On rule interestingness measures*, Knowledge-Based Systems, 12 (1999), pp. 309–315.
[9] E. J. Friedman-Hill, *Jess, The Java Expert System Shell*, Sandia National Laboratories, Livermore, CA, USA, 1998.
[10] V. Ganti, J. Gehrke, and R. Ramakrishnan, *Mining very large databases*, Computer, 32 (1999), pp. 38–45.
[11] M. R. Genesereth and S. Ketchpel, *Software agents*, Communications of the ACM, 37 (1994), pp. 48–53.
[12] S. H. Haeckel and R. Nolan, *Managing by wire*, Harvard Business Review, October 1994, pp. 122–132.
[13] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2001.
[14] C. W. Holsapple and M. P. Sena, *Erp plans and decision-support benefits*, Decision Support Systems, (2004).
[15] O. B. Kwon and J. J. Lee, *A multi agent intelligent system for efficient erp maintenance*, Expert Systems with Applications, 21 (2001), pp. 191–202.
[16] S. D. Levi, P. Kaminsky, and S. E. Levi, *Designing and managing the supply chain*, McGraw-Hill, 2000.
[17] C. G. Looney, *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists*, Oxford University Press, 1997.
[18] T. W. Malone, *Inventing the organizations of the twentieth first century: control, empowerment and information technology*, in Harvard Business Review Sept/Oct 1998, Harvard Business School Press, 1998, pp. 263–284.
[19] J. B. McQueen, *Some methods of classification and analysis of multivariate observations*, in Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, L. M. L. Cam and J. Neyman, eds., 1967, pp. 281–297.
[20] P. A. Mitkas, D. Kehagias, A. L. Symeonidis, and I. Athanasiadis, *A framework for constructing multi-agent applications and training intelligent agents*, in Proceedings of the 4th International Workshop on Agent-Oriented Software Engineering, Springer-Verlag, 2003, pp. 1–16.

[21] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, EDITION:2nd; McGraw-Hill Book Company; New York, NY, 1984.
[22] Y. Peng, T. Finin, Y. Labrou, B. Chu, W. Tolone, and A. Boughannam, *A multi agent system for enterprise integration*, Applied Artificial Intelligence, 13 (1999), pp. 39–63.
[23] R. T. Rust, V. A. Zeithaml, and K. Lemon, *Driving customer Equity: How customer lifetime value is reshaping corporate strategy*, The Free Press, 2000.
[24] C. Rygielsky, J. C. Wang, and D. C. Yen, *Data mining techniques for customer relationship management*, Technology in Society, 24 (2002), pp. 483–502.
[25] J. Shapiro, *Bottom-up vs. top-down approaches to supply chain modeling*, Kluwer, 1999, pp. 737–759.
[26] 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 (2003), pp. 589–602.
[27] A. L. Symeonidis, P. A. Mitkas, and D. Kehagias, *Mining patterns and rules for improving agent intelligence through an integrated multi-agent platform*, in Proceedings of the 6th IASTED International Conference on Artificial Intelligence and Soft Computing, 2002.
[28] The FIPA Foundations, *Fipa-sl specifications, 2000, fipa sl content language specification*, tech. report, The FIPA Consortium, March 2000.
[29] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufman, 2000.
[30] M. Wooldridge, *Intelligent agents*, in Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, G. Weiss, ed., The MIT Press, Cambridge, MA, USA, 1999, ch. 1, pp. 27–78.
[31] J. H. Worley, G. R. Castillo, L. Geneste, and B. Grabot, *Adding decision support to workflow systems by reusable standard software components*, Computers in Industry, 49 (2002), pp. 123–140.

Edited by: Marcin Paprzycki, Niranjan Suri
Received: October 1, 2006
Accepted: December 10, 2006