



iSOCO
intelligent software for
the networked economy

Experto universitario para desarrollo de sistemas para comercio electrónico

Apuntes del módulo *Negociación on-line
automatizada: negociaciones y subastas
electrónicas*

Electronic Negotiations and Auctions

v1.0

Authors
Jesús Cerquides
Maite López-Sánchez
Antonio Reyes Moro
Juan A. Rodríguez-Aguilar

Tabla de contenido

1	INTRODUCTION.....	2
2	ELECTRONIC NEGOTIATIONS AND STRATEGIC SOURCING	2
2.1	THE SOURCING PROCESS	2
2.2	COMPONENTS OF STRATEGIC SOURCING SOLUTIONS.....	5
2.3	E-SOURCING BENEFITS	6
3	QUOTES: A NEGOTIATION TOOL FOR STRATEGIC SOURCING	8
3.1	REQUEST FOR QUOTATION (RFQ) CREATION.....	8
3.1.1	<i>Template selection</i>	<i>10</i>
3.1.2	<i>RFQ value setting.....</i>	<i>11</i>
3.2	PROFILES OF PRODUCTS AND SERVICES	12
3.2.1	<i>Preferences and capabilities.....</i>	<i>12</i>
3.2.2	<i>Business and bidding rules.....</i>	<i>13</i>
3.3	NEGOTIATION STAGES	15
3.3.1	<i>Provider selection: smart matching algorithm.....</i>	<i>15</i>
3.3.2	<i>Automatic provider response via optimal generation of offers.....</i>	<i>16</i>
3.3.3	<i>Negotiation phase</i>	<i>17</i>
3.3.4	<i>Auctions</i>	<i>18</i>
3.3.5	<i>Decision support modules.....</i>	<i>20</i>
4	PRESENT AND FUTURE OF E-SOURCING.....	26

1 Introduction

The sourcing process of multiple goods or services usually involves complex negotiations (via telephone, fax, etc) that include discussion of product features as well as quality, service and availability issues. Currently, this is a high-cost process due to the scarce use of tools that streamline this communication and assist purchasing managers' decision-making. With the advent of Internet-based technologies, it becomes feasible the idea of an affordable tool that enables to maintain an assisted, fluid, on-line dialog at virtually no cost and wherever your providers are. Consequently, several commercial systems to support on-line negotiations become available. However there is still a need that these systems incorporate effective decision support techniques.

Traditionally, the core of the sourcing process comprises the following tasks: Request for Quotation (RFQ) elaboration, provider selection for RFQ delivery; best candidate offer analysis, negotiation through offer-counteroffer interaction or reverse auction, and selection of best offers. Recently, several commercial systems targeted at supporting such processes have become available. However, to the best of our knowledge not a single system can claim to address the full complexity of on-line negotiations. Most of them merely incorporate single-item, price-quantity reverse auctions mechanisms. Others only offer basic negotiation capabilities that are usually reduced to a demand-offer matching tool. In general terms, there is a lack of decision support functionalities (decision making in sourcing can involve a few hundred offerings each of which is described by several dozen attributes). Finally, there is a lack of technology support for computationally complex negotiation paradigms, which inhibit the application of interesting models such as combinatorial reverse auctions [8].

The aim of this chapter is twofold. On the one hand, the subprocesses and issues characterising the sourcing process are fully explained so as to identify the challenges to be faced by sourcing applications. On the other hand, this chapter presents *Quotes* [15] as iSOCO's e-solution for strategic sourcing that incorporates Artificial Intelligence (AI) based techniques that successfully address previous limitations within a single and coherent framework. In this manner, *Quotes* shall ideally serve us to illustrate technological solutions that enhance and empower the traditional sourcing process.

The chapter is organised in three sections. Section 2 describes the traditional sourcing process in order to identify the issues to be tackled by sourcing applications. Furthermore the benefits derived from the adoption of sourcing applications are also identified. Next, Section 3 thoroughly dissects the features of *Quotes* as a sample of advanced sourcing application. Finally, Section 4 presents a critical view of current sourcing applications in the market and elaborates on the expected characteristics of the next generation of sourcing applications.

2 Electronic negotiations and strategic sourcing

2.1 The sourcing process

Businesses' goal is to sell goods and/or services to other businesses and/or consumers. In order to produce these goods and services, materials or services such as computers, travel, and temporary staffing must be purchased from other parties in a business supply chain. Hence purchasing is a capital business activity. We identify two main costs involved in this activity: the

direct cost of the material or service purchased and the process cost (transaction costs, which include the personnel and overhead required to search for, compare, negotiate and source various goods and services).

The sourcing process comprehends the identification, evaluation, negotiation, and configuration of products and services to ensure that a company can establish the most efficient global value chain. The sourcing process ranges from simple sourcing of indirect goods and services¹ to more complex sourcing of direct materials and services². On the one hand, the sourcing of direct materials is based on conformance to quality, current and future provider capacity, security stocks, design specifications, on-time performance and total cost to the buyer, including manufacturability and logistics. On the other hand, the sourcing of indirect materials/services is based on price, service and convenience. Notice that in both cases the buying decision is not solely based on price but on a wide variety of criteria. Figure 2-1 attempts at summarising the key criteria customers use to make a buying decision.

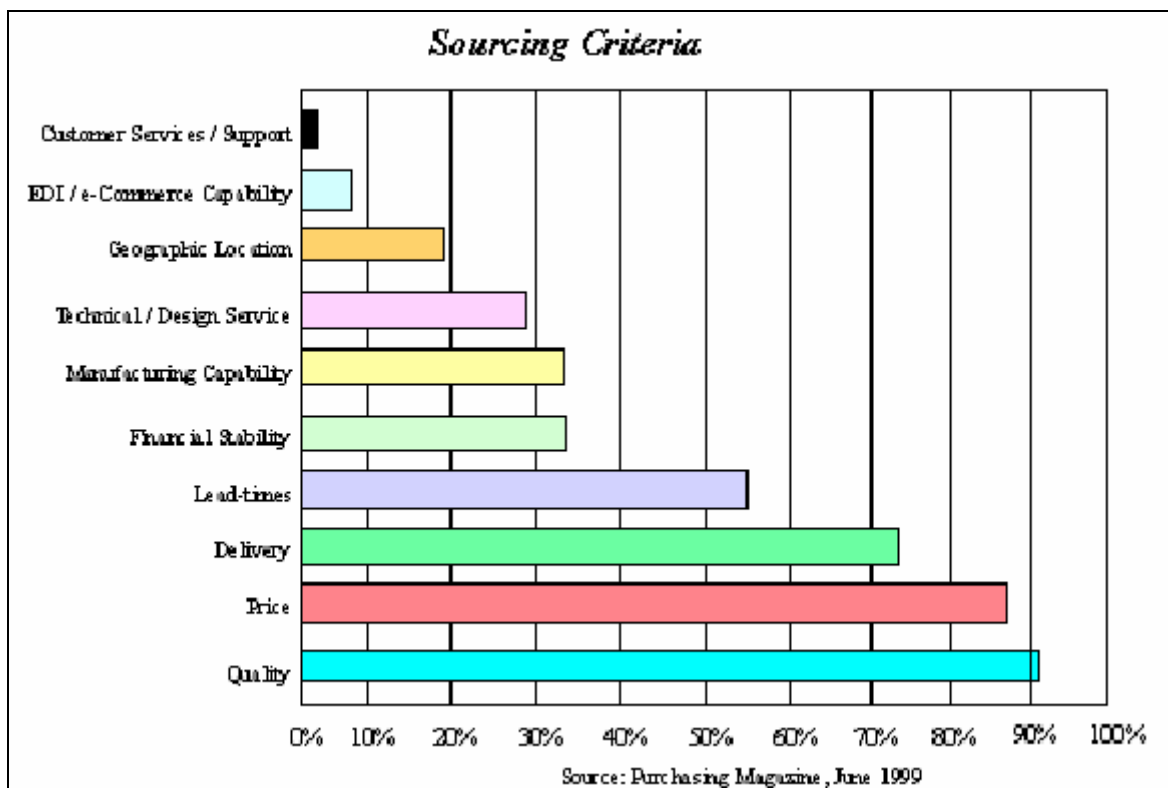


Figure 2-1. Key criteria customers use to make a buying decision

Traditionally the sourcing process comes through several stages. First comes the provider search and analysis. The next step is provider selection via competitive bidding or negotiation. Once a contract is awarded, order releases against that blanket contract are calculated by an MRP³ system and typically sent (for large companies) via EDI⁴ to trading partners. Lastly, the

¹ Indirect items such as travel, office supplies and information technology equipment are not acquired with the aim of manufacturing them but with the purpose of helping execute the mainline business activity.

² Manufacturers acquire direct materials (raw materials and components) to add value through some manufacturing or assembly process so that the final product can be sold to customers.

³ MRP stands for Manufacturing Resource Planning

purchasing data is fed in along with logistics and production information into a supply chain planning/optimisation engine to fine-tune schedules for improved efficiency. The sourcing process can be further broken down as depicted in Figure 2-2.

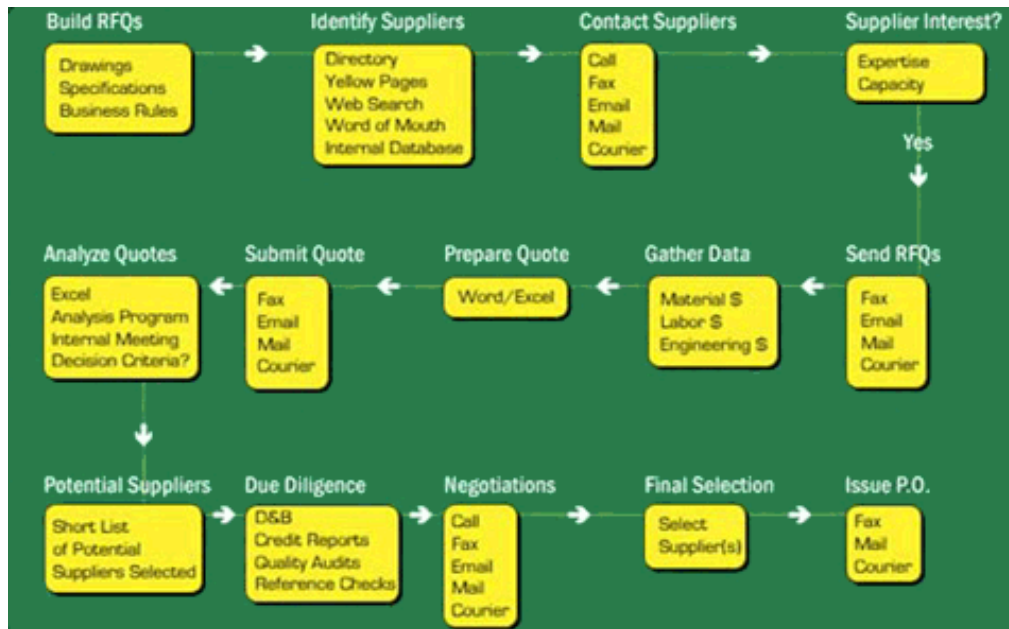


Figure 2-2. Traditional Process for Sourcing

The traditional sourcing cycle can be approximately broken down into the following time-weighted components: identifying appropriate providers (52% of cycle), managing/communicating preferred provider list (5%), RFQ development (8%), RFQ response/receipt (5%), screening/sorting proposals (20%), and contract negotiation (10%).

The goal of sourcing is simple: reduce the cost of the finished product or service while keeping quality concerns in the forefront. It is not only about price. However, the rewards are disproportionately great: even a minuscule 2 percent cost reduction during the sourcing cycle can lead to a whopping 14 percent reduction in the end cost of a new product or service. Hence companies should concentrate on deploying efficient strategic sourcing to drive down costs. Reality says though that so far they are not succeeding. Thus approximately 75 percent of companies rate their ability to affect or reduce the cost or quality of a product or service after the initial sourcing process as only fair or poor. This fact is motivated by the inefficiency of the legacy process and the lack of decision support. On the one hand, the strategic buying aspect for both direct and indirect materials is still very much a manual process. Most companies still utilise a convoluted mix of phone calls, faxes, e-mails, and snail mail when communicating and exchanging information with providers. Not surprisingly 80% of professional buyers' time is devoted to administrative tasks, saving a scant 20% of their time for strategic analysis. Furthermore, on the other hand, their strategic thinking does not accustom to be assisted by decision support tools.

⁴ EDI stands for Electronic Data Interchange

2.2 Components of strategic sourcing solutions

Strategic sourcing is a process whereby businesses select providers in a calculated manner. Selection decisions under strategic buying are based on factors such as a provider's new product development capabilities, the ability to share information electronically, or the means for a component to differentiate the final product. With strategic sourcing, businesses analyse and decide on providers based on the strategic impact of that provider on the overall business (supply chain) instead of simply awarding each supply contract to the provider with the lowest bid. Strategic sourcing applications are the software programs that provide critical decision-support functionality to purchasing departments and decision-makers within the supply chain. These applications facilitate the determination of providers, the configuration of new products and services, how to allocate contracts and volume among providers, how to manage supply risk, and how to optimise the provider base.

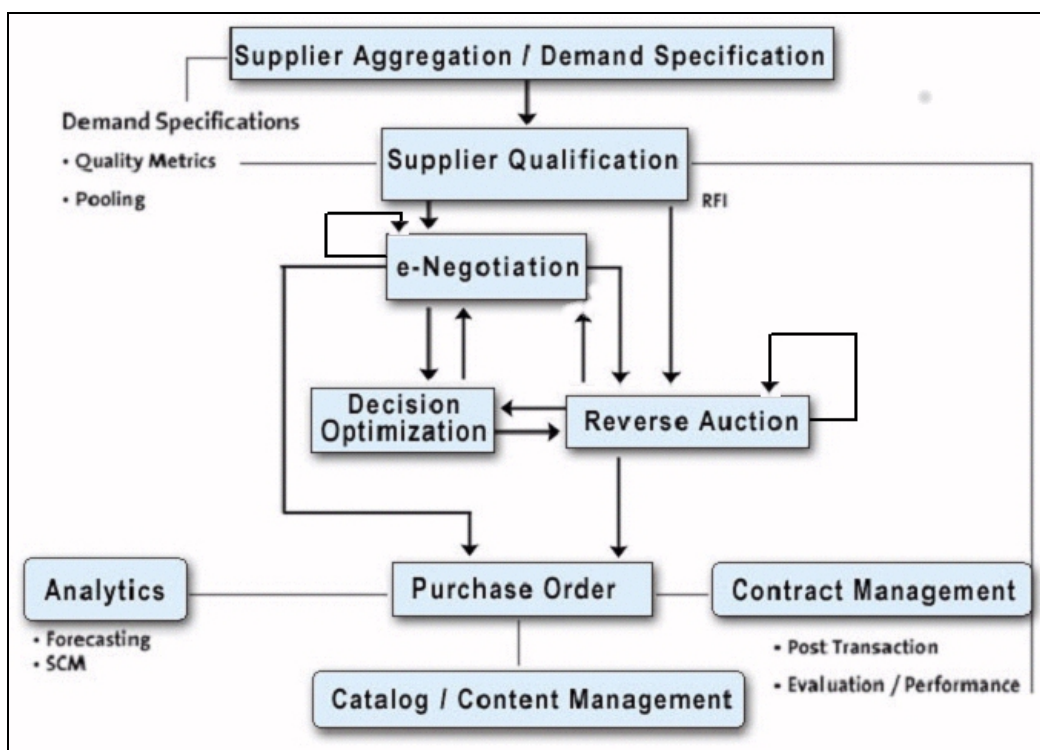


Figure 2-3. Components of a complete strategic sourcing solution

Figure 2-3 depicts a model of the components that we regard as necessary to build a complete strategic sourcing solution that captures and enhances the traditional sourcing model:

- *Supplier qualification (homologation)*. It allows to carry out the request for information (RFI) process, allowing buyers to issue RFIs so as to assess and qualify providers' capacities, capabilities and additional performance criteria. The goal of this module is to allow buyers to evaluate providers prior to invitations for negotiations.
- *E-Negotiation engine*. It is targeted at providing the necessary support for both buyers and providers to streamline and automate the negotiation processes that traditionally have been conducted over phone, fax and value-added networks (VANs).

- q *Reverse auction engine.* It allows buyers to move from one-to-many haggling scenarios in which each provider attempts at reaching an individual agreement with the buyer without a transparent sense of competition (since the actions of the rest of providers are unknown) to more competitive scenarios in which providers are gathered together in the very same trading arena. Thus, the buyer is relieved from making an effort to reach win-win agreements with his providers, putting the burden of determining the features of the agreement in the market itself. Likewise e-negotiations a buyer might decide to iterate over several auctions and, in the more general case, interleave negotiations and auctions at will. Ideally a strategic sourcing solution should support such functionality so that the sourcing practices of a wide range of industries can be captured. Notice too that along with the e-negotiation engine, this engine is the core of any strategic sourcing application.
- q *Decision optimisation support.* Module devoted to evaluating offers, providers and RFQs based on buyers' and providers' preferences and weightings. It also assists buyers and providers in suggesting bids that bring them closer to a deal. Furthermore, it should ideally help determine the best combination of offers at both negotiation and auction time.
- q *Analytics module.* Group of applications that transform sourcing data into measurable data that leads to benefits. Spend analysis and sensitivity analysis are common analytical features.
- q *Catalog/content management.* Strategic sourcing applications are expected to provide some form of catalog retrieval and functionality, particularly to store negotiated products and services along with templates that serve to create requests for new products and/or services.
- q *Contract management* refers to the process governing the life cycle of trade agreements between business entities after negotiation and execution of the purchase order. Contract management tools typically offer, as major functionalities, standardise contract creation and maintenance, faster contract approval and signing with automated workflow, global contract visibility, and risk management through reporting across clauses and terms.

2.3 E-sourcing benefits

E-sourcing applications can help all along the cycle and deliver impressive benefits along four major directions: shortening in cycle times, unit price reductions, time-to-market cycle reductions, and process cost reductions. Although there is an apparent agreement on the identification of these major benefits, a large number of varying figures from a variety of sources have been offered to shape them in figures. Figure 2-4 offers some ranges of figures intended to quantify e-sourcing benefits. The depicted summary stems from a collection of market reports focusing on e-sourcing technologies.

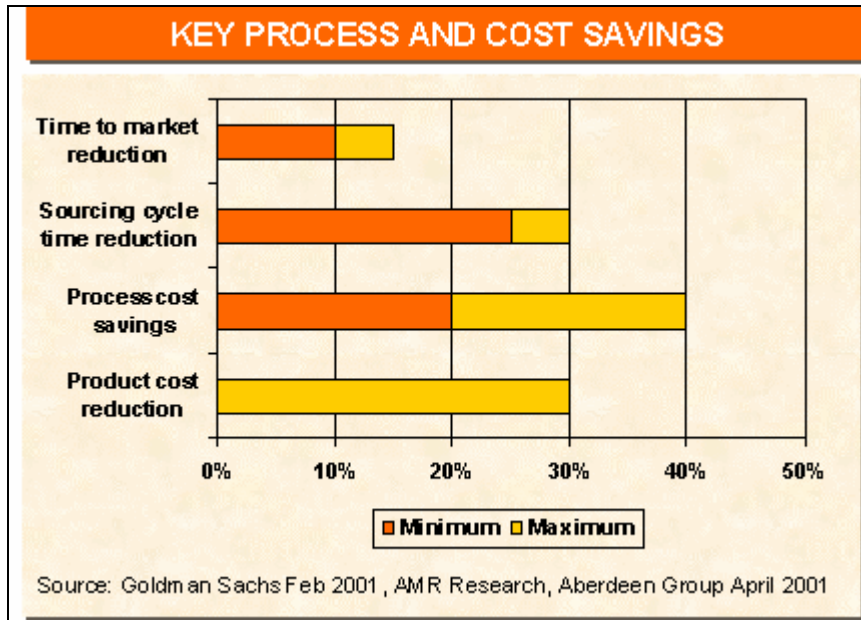


Figure 2-4 e-Sourcing technologies' benefits

So far it seems that we have been primarily concerned about buying parties as the unique beneficiaries from the deployment and exploitation of e-sourcing applications. Nonetheless there are additional parties that largely benefit from these technologies too. First, providers may also enormously benefit from e-sourcing technologies. For this purpose, e-sourcing applications must provide them with decision support and business automation facilities that help them streamline the cycle and readily differentiate and present their most advantageous offers. Second, electronic marketplaces can offer sourcing applications as a differentiating value-added service. In this manner, not only large companies can start benefiting from e-sourcing technologies as early adopters, but also SMEs⁵ that may not afford their acquisition and alternatively opt for renting the service to some marketplace acting as ASP⁶.

Table 2-1 summarises the expected benefits of e-sourcing technologies per role.

⁵ SME stands for Small and Medium Enterprise.

⁶ ASP stands for Application Service Provider.

E-SOURCING BENEFITS	
Role	Benefits
Buyer	<ul style="list-style-type: none"> □ Efficiently manage the identification, evaluation, negotiation, and configuration of the buying process of products and services. □ Ensure the establishment of the most efficient global value chain based on total cost.
Provider	<ul style="list-style-type: none"> □ Ability to offer the buyer competitive advantage, be it in price, demand schedule flexibility, advantageous warranty terms, higher quality, or other specifications, terms or conditions. □ Ability to quickly decide which clients to supply and which not. □ Increase competitiveness by holding simultaneous negotiations with multiple buyers and automating their bidding through business rules.
Marketplace	<ul style="list-style-type: none"> □ Increased market share. By offering an easily implementable strategic sourcing solution to its clients a marketplace can quickly differentiate itself in a crowded marketplace. □ Additional revenue stream. The strategic sourcing solution offered to a marketplace's clients can generate an additional revenue stream.

Table 2-1 Summary of e-sourcing benefits for involved parties

3 Quotes: a negotiation tool for strategic sourcing

This section introduces *Quotes* [15], iSOCO's e-solution for strategic sourcing. Although section 2.2 identified all the components ideally composing a complete sourcing solution, notice that *Quotes* focuses on and realises the components related to the negotiation process occurring after provider qualification and prior to the sending of purchases orders (see Figure 2-3).

3.1 Request for quotation (RFQ) creation

Quotes supports multi-attribute, multi-line RFQs, enabling the creation of multiple types of RFQs (commodity, catalogue, BOM or group by). Furthermore it provides the expressiveness needed to cope with multi-criteria negotiation procedures. Typically a buyer creates an RFQ by sequentially adding lines. Each line specifies a product, be it either a good or service. Figure 3-2 shows an *RFQ* composed of several lines, a so-called multi-line RFQ. A paradigmatic example of multi-line RFQ occurs at industrial settings such as illustrated in Figure 3-1. The production plan outlined by some company's ERP⁷ or SCM⁸ application comes in the shape of a list of items to be produced along with the parts required to comply with the plan, the so-called bill of material. The bill of material is the basis for the buyer to start out multiple sourcing events, each one devoted to the procurement of the parts of each one of the items whose production has been forecast.

⁷ ERP stands for Enterprise Resource Planning.

⁸ SCM stands for Supply Chain Management.

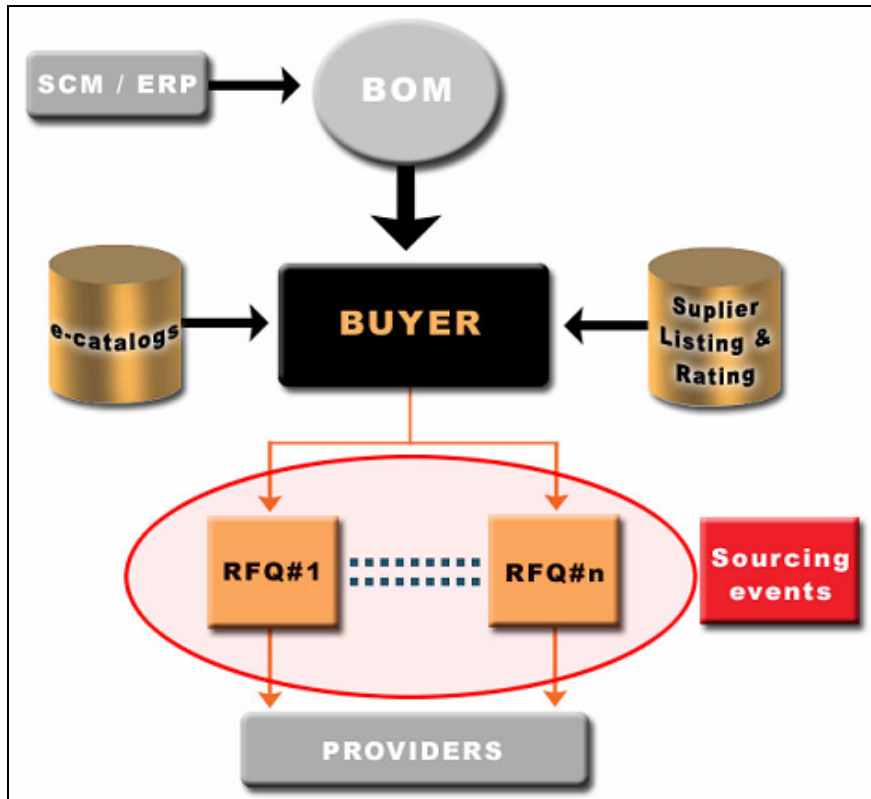


Figure 3-1. Information flow for a BOM (bill of material) RFQ

The screenshot shows a web browser window titled 'Toni [New RFQ] - Microsoft Internet Explorer'. The browser address bar shows 'Address'. The page content includes a navigation menu with 'Home', 'What's Quotes?', 'Help', and 'About us'. The main content area is titled 'Buyer Side' and contains a form for creating a new RFQ. The form includes a text input field for 'RFQ Name' and a table of items with columns for 'Template', 'Description', and 'Reserve Score'. Each row in the table has an 'Edit' link. At the bottom of the form, there is an 'Add item' button and an 'RFQ OK' button.

Template	Description	Reserve Score	
Memory SIMM	Memory SIMMs	30.0	Edit
Soldados	Cooling Fan	25.0	Edit
Soldados	9 pin connectors	0.0	Edit
Alicatados	Lithium bateries	70.0	Edit

Figure 3-2. Example of a multi-line RFQ

The process of including an item (line) is composed of two steps: template selection and RFQ value setting. Next we explain both subprocesses in detail.

3.1.1 Template selection

The process starts when some buyer creates an RFQ line by selecting a product from a list of product templates. Product templates are previously created by the system administrator and consist of a list of attributes describing the product (Figure 3-3 shows a typical product template). Each attribute makes reference to a physical characteristic or negotiable condition or term. Each of these attributes is defined in terms of a name and its domain. The name is used as an identifier and provides semantics to the template if a meaningful string is associated. The domain specifies the universe of possible values that an attribute can take. There are five different types of attributes, namely:

- *Any Number*. For attributes that can take any numerical value. This type is recommended for attributes whose values cannot be constrained.
- *Range of Numbers*. This type is intended for attributes whose possible values belong to a numerical range. For example, PH can only take values between 0 and 14. This type is always associated to a maximum and a minimum value definition.
- *Set of Labels*. Non-numerical values can be of type 'set of labels' when there is an associated list of predefined textual values and there is no order among them. For instance, a list of colour names.
- *Ordered Set of Labels*. This type of attribute can be used for quantitative measures. In this case there needs to be an order for associated values. As an example we can think of quality levels such as 'regular', 'extra', 'luxury', etc.
- *Free Text*. Finally, a free text attribute is added so that the buyer has the opportunity to add information that is not included in the template.

Domain specification also includes the measurement units that are associated to numerical attribute values. Hence Quotes enhances each attribute specification with a variety of associated fields that provide both functional flexibility and semantic enrichment.

Administrator tool

RFQ Template

Template Name

Attributes

Name	Type	Values	Units
<input type="text" value="Precio unitario"/>	Range of numbers	<input type="text" value="0.0"/> <input type="text" value="300.0"/>	<input type="text" value="Euros"/>
<input type="text" value="Cantidad"/>	Range of numbers	<input type="text" value="1.0"/> <input type="text" value="100.0"/>	<input type="text" value="Unidades"/>
<input type="text" value="Origen madera"/>	Ordered set of labels	1. <input type="text" value="Pino"/> 2. <input type="text" value="Haya"/> Add value	
<input type="text" value="Calidad acabado"/>	Ordered set of labels	1. <input type="text" value="Normal"/> 2. <input type="text" value="Buena"/> 3. <input type="text" value="Lujo"/> Add value	
<input type="text" value="Plazo de entrega"/>	Range of numbers	<input type="text" value="1.0"/> <input type="text" value="30.0"/>	<input type="text" value="Dias"/>

[New Attribute](#)

[Template OK](#)

Figure 3-3. A product template definition

3.1.2 RFQ value setting

Once a product template is selected, the buyer must specify desired values for each attribute. Obviously, the way values are set depends on the attribute type, but in general, we can stand that the buyer can do it so by either defining a single value or a range of values. Additionally, he can also express his preferences among possible values within a set. Figure 3-4 shows the interface available for buyers at this aim.

In this manner, for an *Any Number* attribute a buyer can specify its desired value as either a single number or as a range of numbers that satisfies their needs. In the last case, three values are required: a minimum, a maximum, and a preference slope indicating which values are most preferred within the range. The slope can take on three different values:

- *Flat*. When all values within the range are equally preferred.
- *More Is Better (MIB)* indicates that higher values are more preferred than lower ones.
- *Less Is Better (LIB)*. On the contrary to MIB, the buyer chooses an LIB slope when smaller values are preferred.

The same kind of single / range value specification can be used for the *Range of Numbers* and *Ordered Set of Labels* types. The only difference is that attribute values must belong to the domain defined by the corresponding attribute template. As an example, consider specifying that we are prepared to accept any value for attribute *quality* that is at least *standard* but taking into account that the more quality the better.

As to the *Set of Labels* type, the buyer is enabled to choose one value (label) or a subset of values (Notice though that since no order is associated, there is no need for defining any range or slope).

Finally, a buyer assigns some value to a *Free Text* type by filling out a text field.

Buyer Side

Select a template and fill Item's attributes and preferences

New RFQ

RFQ Name maiteMemory

Add Item in RFQ

Template: Memory SIMM

Item Name:

Reserve Score:

Attributes:

Name:	Value Type:	Value:	Units:	Must Have:	Importance:	Reveal:
Price	Interval	[0.0 ... 0.0 10000.0 ...20000.0] \$ Slope: Any is good		<input type="checkbox"/>	● ● ● ● ● ●	<input type="checkbox"/>
Quantity	Single	10.0	pcs	<input type="checkbox"/>	● ● ● ● ● ●	<input type="checkbox"/>
Size	Single	<input type="radio"/> 64 Mb <input checked="" type="radio"/> 128 Mb <input type="radio"/> 256 Mb		<input checked="" type="checkbox"/>	● ● ● ● ● ●	<input type="checkbox"/>
Speed	Single	<input type="radio"/> 100 Mhz <input checked="" type="radio"/> 133 Mhz		<input checked="" type="checkbox"/>	● ● ● ● ● ●	<input type="checkbox"/>
Package	Set of labels	<input checked="" type="checkbox"/> O.E.M. <input checked="" type="checkbox"/> Branded		<input type="checkbox"/>	● ● ● ● ● ●	<input type="checkbox"/>

Item OK

Figure 3-4. RFQ value specification

The buyer has also the capability of defining the importance that each attribute is expected to have during the selection phase and the negotiation process. Basically, he assigns a value ranging from *don't care (irrelevant)* to *extremely important* (there are five different values). Additionally, if it is compulsory that offers satisfy the requested values, the user must then tick the *must-have* checkbox. The way Quotes exploits preferences and importance values is latter explained.

Finally, the buyer assigns a reservation score, a threshold value, to the RFQ. Thus offers whose percentage of matching with the RFQ fall below the reservation score are filtered out (subsection 3.3.5.1 fully explains scoring how offers are scored and thus matched to RFQs). That is, depending on the matching score value, Quotes automatically rejects offers that unsatisfactorily match buyers' requirements.

3.2 Profiles of products and services

3.2.1 Preferences and capabilities

While buyers need to specify their product requirements in terms of negotiable attributes, providers can analogously do the same regarding their product capabilities and their proposals preferences.

Thus each provider is capable of declaring thoroughly detailed lists of goods or services he is able to provide, the so-called production profiles. Figure 3-5 shows a production profile for the product described in the template shown in Figure 3-3.

Product filter : ▼ Conditions & preferences ▼ Business Rules

Filter Name:

Attribute Name:	Allowed values :	Listen Broader:	Preferred Type:	Preferred Values:	Importance:
Precio unitario	Min: <input type="text" value="20.0"/> Max: <input type="text" value="300.0"/>	<input checked="" type="checkbox"/>	Interval ▼	[0.0 ... <input type="text" value="100.0"/> <input type="text" value="300.0"/> ... 300.0] Slope: <input type="text" value="More is better"/> ▼	<input checked="" type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/>
Cantidad	Min: <input type="text" value="1.0"/> Max: <input type="text" value="40.0"/>	<input checked="" type="checkbox"/>	Interval ▼	[1.0 ... <input type="text" value="10.0"/> <input type="text" value="25.0"/> ... 100.0] Slope: <input type="text" value="Any is good"/> ▼	<input checked="" type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/>
Origen madera	<input checked="" type="checkbox"/> Pino <input checked="" type="checkbox"/> Haya	<input checked="" type="checkbox"/>	<input type="text" value=""/>		<input checked="" type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>
Calidad acabado	Min: <input type="text" value="Buena"/> Max: <input type="text" value="Lujo"/>	<input checked="" type="checkbox"/>	Single ▼	<input type="radio"/> Normal <input type="radio"/> Buena <input checked="" type="radio"/> Lujo	<input checked="" type="radio"/> <input checked="" type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/>
Plazo de entrega	Min: <input type="text" value="10.0"/> Max: <input type="text" value="3200.0"/>	<input checked="" type="checkbox"/>	Interval ▼	[1.0 ... <input type="text" value="20.0"/> <input type="text" value="30.0"/> ... 30.0] Slope: <input type="text" value="Any is good"/> ▼	<input checked="" type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>

Go

Figure 3-5. A sample of production profile

As discussed above, Quotes describes a product in terms of its negotiable attributes. Consequently, three concepts need to be specified for each attribute: the provider production capabilities, his selling preferences and, once again, a measure of importance. The production capabilities determine which product demands the provider can actually accept. For example, suppose that some provider needs a minimum of five days to deliver his goods or that he can only provide a maximum cable thickness. Selling preferences allow a provider to state which requests he may favour. He might be interested in quickly identify requests for large volumes or for a specific product model, while being less interested in requests for discontinued products.

3.2.2 Business and bidding rules

Beyond declaring attribute capabilities and preferences, Quotes allows each provider to declare his own business rules in the form of bidding rules.

Bidding Rules

Rules:

if	Size	=	64 Mb	then	fix	Price	by/to	5000.0
if	Size	=	128 Mb	then	fix	Price	by/to	8000.0
if	Size	=	256 Mb	then	fix	Price	by/to	15000.0
if	Speed	=	133 Mhz	then	+%	Price	by/to	30.0
if	Quantity	#	[11.0 ... 100000]	then	-%	Price	by/to	10.0

[New Rule](#)

[Go](#)

Figure 3-6. Bidding rule example

A bidding rule is an *if-then* rule that checks and changes the value of one or several attributes. Examples of rules include discount per volume, additional charges for express delivery, no delivery charge when a minimum price is offered, etc. Figure 3-6 shows an example for memory SIMM modules. The general syntax to define bidding rules is shown in Figure 3-7.

```

RULE: if CONDITION then ACTION;
CONDITION: attribute_name1 [=, !=] value1 |
           attribute_name1 [#,!#] (min_value, max_value) |
           (CONDITION) |
           CONDITION [and, or] CONDITION;
ACTION: attribute_name2 [fix, +%, -%] value2
```

Figure 3-7. Bidding rule syntax.

Operations for checks and changes can be selected among a list of available operations that depends on the chosen attribute type:

- *Condition operators* are: =, !=, #, !#. Equality (=) and inequality (!=) operators can be chosen for all types of attributes (both numerical and sets of labels) and require a value to be compared with. On the contrary, membership (#) and non-membership (!#) operators only apply to types that allow to specify intervals (that is, *any number, range of numbers, and ordered sets of labels*).
- *Action operators* are functions to be applied to the attribute that has been chosen to be affected. These are: assignment (fix), which can be applied to all attribute types, and percentage increment (+%) and decrement (-%), which are only applied to numerical types.

Notice that the free text type is not supported here because it is a natural language value whose automatic treatment goes beyond the scope of this application. However, the manual setting of a Boolean value associated to free text types specifies if the provider meets the requested attribute value or not.

3.3 Negotiation stages

This section aims to describe the main processes that occur when a newly created RFQ is launched by *Quotes*. Figure 3-8 shows the main stages. After some buyer submits an RFQ, potential providers are automatically identified and made aware of the RFQ. On the providers' side, offers are automatically built as responses to received RFQs by firing pre-defined bidding rules. Thereafter the buyer can conduct simultaneous one-to-one negotiations as part of the one-to-many negotiation process. This negotiation phase may end with success (that is, the buyer winds up accepting some offer(s)) or may be used as an initial selection of providers that are invited to participate in a reverse auction.

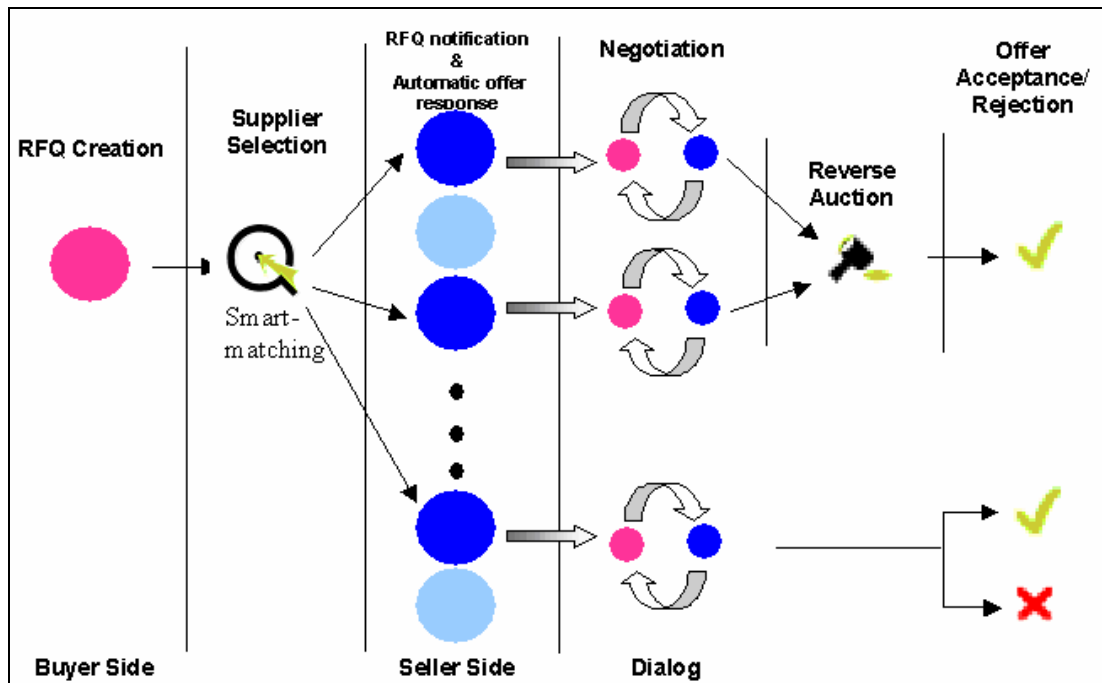


Figure 3-8. *Quotes' e-sourcing process description*

3.3.1 Provider selection: smart matching algorithm

Provider selection consists of two filtering steps. It has been already described how buyers and providers have interfaces that allow them to describe their necessities and capabilities based on common product templates. The first filtering process is a straightforward approach that solely requires identifying those providers that offer products/services specified with the same template than the product the buyer is requesting.

For every identified provider offering the required product, the second filtering process focuses on attribute values. Internally, this common language provided by templates allows to match values over the same domain. As mentioned above, when a buyer specifies its requirements by assigning values to each attribute in a template, these values can be single values, sets of values or intervals. Those values are then internally fuzzified via the generation of associated fuzzy functions. Then fuzzy values are matched to the values described by the providers and, if the resulting membership degree is not null for any attribute, then the corresponding provider receives the RFQ from the buyer.

3.3.2 Automatic provider response via optimal generation of offers

In this section, we describe the module of *Quotes* that is in charge of generating indicative offers on behalf of providers. Such process takes place immediately after provider selection. Its objective is to build a complete offer (where all attributes have been given a value). Moreover, the algorithm pursues to build the *best* complete offer in terms of either buyer preferences or provider's or both. For the sake of clarity, this section assumes an offer to have a single item (that is, it is only offering one product or service), so that 'offer' is used instead of 'offer item'. For the general case of offering more than one product, the process described here will be repeated for each item.

The implemented algorithm is a variation of the well-known hill-climbing random neighborhood search procedure. We start with an incomplete offer O as the candidate solution and then enter into an iteration phase. At each iteration, we generate a neighboring offer O' of O and we apply the business rules (which have been already explained in subsection 3.2.2). If the new solution O' is better than the candidate offer O we accept O' as the current solution. The process continues until the termination criterion (no improvement for the last k iterations) is reached.

In order to explain completely the optimisation procedures, the following definitions are in place.

Definition 1: Space of solutions.

We define an offer as a tuple O of the form $\langle o_1, o_2, \dots, o_n \rangle$ where $o_i \in (A_i^P \cap A_i^B) \cup \emptyset$ being:

- n the number of negotiable attributes.
- A_i^P the set of values for *ith*-attribute of the item produced by the provider.
- A_i^B the set of values for *ith*-attribute of the item asked by the buyer.
- The symbol \emptyset denotes *unassigned* value.

In other words, *an offer* O gives a value for each requested attribute (i.e., o_i is the value offered for attribute A_i), provided such value is within both provider's capabilities and buyer's acceptable values.

An offer s is *complete* if $o_i \neq \emptyset \forall i \in [1..n]$. Additionally, we say that an offer O is *more complete* than O' if O has the same or less *unassigned* values than O' .

The evaluation function defines the objective of the optimization process in terms of determining if O' is better than O . In our case the objective is twofold: obtain a complete offer that optimizes a target function.

Definition 2: Objective function.

The objective function $C: O \rightarrow \Re$ is defined as

$$C(O) = w^P \cdot \sqrt{S^P(O)} + w^B \cdot \sqrt{S^B(O)}$$

where

- $S^P(O): O \rightarrow [0..100]$ is the scoring function based on the preferences of the provider (see subsection 3.3.5.1 for details about scoring functions).
- $S^B(O): O \rightarrow [0..100]$ it is the same scoring function than the previous one except for that it is based on the buyer's preferences.
- w^P and w^B are weight factors.

Such evaluation function tries to favour high scorings while penalising big differences between the buyer's and the provider's revenues. In other words, it prefers a 50 - 50 rather than an 80 - 20 thus seeking win-win situations (and this is desirable because if a provider generates an offer that best suites his/her preferences without taking into account the buyer's ones, it will be likely

rejected by the buyer). Finally, w^P and w^B can be used to tune the objective function (for example, to focus more on the buyer's preferences).

Optimisation algorithm

Informally, we say that an offer O' is *better* than an offer O if O' is more *complete* than O and $C(O') > C(O)$. Consequently, we only favour solutions that are closer to be complete and that improve the existing candidate solution. Figure 3-9 shows the pseudocode for the optimisation algorithm.

```

1.    $O = \langle \emptyset, \emptyset, \dots, \emptyset \rangle$ 
2.   While termination_criterion not reached
      2.1 randomly select an attribute in the condition of a rule.
      2.2 randomly obtain a new value for the chosen attribute.
      2.3 obtain  $O'$  as the result of executing the set of bidding
          rules over  $\langle o_1, \dots, o_i', \dots, o_n \rangle$ 
      2.4 if  $O' > O$  then  $O = O'$ 
      end_while
3.   If  $O$  is complete return with success
      Else return with failure

```

Figure 3-9. Optimisation offer algorithm

If the process ends with success, a complete *offer* is obtained and Quotes automatically submits it as an indicative offer. Assuming $w^P = w^B = 1$ the offer is likely to be close to a win-win agreement and thus reducing the number of buyer-provider interactions. However the negotiation process may progress further since the submitted offer is set as indicative.

3.3.3 Negotiation phase

So far potential providers have been notified and even some of them have already submitted automatically-generated indicative offers. The process now enters into a negotiation phase. Negotiation is conducted through multiple structured dialogs that are performed in parallel. Each dialog is established between the buyer and a single provider and it is ruled by a negotiation protocol. Buyer actions can be: offer acceptance, offer rejection, counter-offer submission, and request for firm (offer). Provider actions are limited to the submission of either firm or indicative offers. The following sequence illustrates a typical negotiation.

1. The buyer submits an RFQ asking for *service1* and *service2*.
2. Quotes identifies a potential provider and automatically constructs two indicative offers on his behalf based on his bidding rules: *offer1* for *service1* and *offer2* for *service2*.
3. The buyer evaluates *offer1* and submits a counter-offer asking for lowering the price.
4. The provider responds with an extension of *offer1* so that it also includes an offer for *service2*. In other words, he is accepting a price reduction provided that the buyer acquires both *service1* and *service2*.
5. The buyer evaluates the modified *offer1*, agrees with it and requests a firm offer.
6. The provider responds with a firm offer.
7. The buyer accepts the offer and so the negotiation successfully finalises.

Provider Side My negotiations About my business

This page shows you the specific offer values for this item.

Edit Item Offer

Template: Mesa de oficina (cod-2333)
RFQ Name: Pedido edificio A
Buyer name: cmanual

Offer details

Attributes	RFQ Request	Last Indicative Offer	Last Counter Offer	Offer New Offer	
Precio unitario	[400.0 ... 700.0] 700.0		500.0	400 [0.0 ... 1000.0]	<input checked="" type="checkbox"/> Final
Cantidad	20.0	20.0	20.0	20.0 [4.0 ... 50.0]	<input type="checkbox"/> Final
Origen madera	Roble Haya Pino	Roble	Roble	Roble	<input type="checkbox"/> Final
Calidad acabado	[Buena ... Lujo] Normal		Buena	Buena	<input type="checkbox"/> Final
Plazo de entrega	[1.0 ... 30.0]	30.0	20.0	20.0 [10.0 ... 30.0]	<input type="checkbox"/> Final

Go
Cancel

Figure 3-1. Offer edition for RFQ item

Notice that the buyer holds one-to-one dialogs with providers. Quotes allows that multiple dialogs take place in parallel between the buyer and the providers competing for some RFQ. And at the same time, each provider is allowed to simultaneously hold several dialogs with the buyer whenever each dialog corresponds to a different offer.

3.3.4 Auctions

Eventually buyers may prefer to employ the negotiation phase to better qualify providers. Through offer and counter-offer dialog rounds, a buyer can finally identify a set of providers who are able to approximately provide the requested product's features. At this point, the buyer may opt for launching a multi-attribute combinatorial reverse auction with the objective of lowering the price while increasing quality of service. Only selected providers will be invited to participate in a buyer-customised auction event. For this purpose, auctions in Quotes include several parameters[7] (see Figure 3-10):

- When to clear the auction (by buyer, when a specific time is reached, no bids have been received for a specified time).
- Tie-breaking rule (random, older bid overrides, newer bid overrides).
- What information is revealed to bidders during bidding concerning contenders' identities (none, nickname, full identity).
- What information is revealed to bidders during bidding (highest bid, all bids, none).
- Whether or not bid retraction is allowed before winners are determined.
- Whether or not bid retraction is allowed after winners are determined.
- Maximum number of auction extensions and time per extension

Buyer Side

Here you must specify the auction parameters
Current time: Tue Apr 23 11:48:30 CEST 2002

Parameter	Value
Clearing rule	Clearing by buyer
Begin date [dd/mm/yyyy] & time [hh:mm]	23/04/2002 11:48
End date [dd/mm/yyyy] & time [hh:mm]	23/04/2002 11:48
Bidders Information Revelation	None
Bids Revelation	Best bid
Ranking Revelation	Enabled
Voyeur providers	Not allowed
Maximum number of extensions	0
Extension time (minutes)	0
Extension detection time (minutes)	0

Auction Settings OK

Figure 3-10. Auction parameters' setting

Quotes' combinatorial reverse auctions allow providers to directly bid for bundles of items. They are convenient for providers that have non-additive values for bundles of items. Furthermore, they allow buyers to express complementarities over the requested items to avoid the risk of obtaining incomplete bundles. Notice also that providers are allowed to place multiple bids for bundles of items. Figure 3-11 and Figure 3-12 depict how buyers and providers visualise the very same auction process.

Buyer Side

Here you have the RFQ auction state with each line details
Current time: Mon Apr 22 18:18:00 CEST 2002

Auction ON

Automatic page refresh every 10 seconds
RFQ Name: Subasta 4
Begin date: Mon Apr 22 18:18:00 CEST 2002

Lines

1	Template	Line name	Status			
	422	Linea 1				
ID	Provider	Bid Item Ranking	Bid Type	Total Lot Price	Reception Time	Award
5388	Maria Alomar 2	1	Single	480.00€	Mon Apr 22 18:14:15	
5388 [Best]	Maria Alomar	2	Composed	570.00€	Mon Apr 22 18:48:05	

2	Template	Line name	Status			
	424	Linea 2				
ID	Provider	Bid Item Ranking	Bid Type	Total Lot Price	Reception Time	Award
5388 [Best]	Maria Alomar	1	Composed	570.00€	Mon Apr 22 18:48:05	

[Cancel](#) [Accept](#)

[Close window](#)

Figure 3-11. Buyer façade of an ongoing auction

Provider Side

My negotiations [About my business](#)

Here you have the RFQ auction state with each line details
Current time: Mon Apr 22 18:39:47 CEST 2002

Auction ON

Automatic page refresh every 10 seconds
RFQ Name: Subasta 4
Begin date: Mon Apr 22 18:18:00 CEST 2002

Items

1	Template	Line name	Status			
	422	Linea 1				
ID	Provider	Bid Item Ranking	Lot Price	Reception Time	Bid Type	Status
5388 [Best]	Maria Alomar	2	570.00 €	Mon Apr 22 18:48:05	Composed	

2	Template	Line name	Status			
	424	Linea 2				
ID	Provider	Bid Item Ranking	Lot Price	Reception Time	Bid Type	Status
5388 [Best]	Maria Alomar	1	570.00 €	Mon Apr 22 18:48:05	Composed	

[Create New Offer](#) [Close window](#)

Figure 3-12. Provider façade of an ongoing auction

3.3.5 Decision support modules

Quotes provides with the necessary tools to help users manage the complex sourcing mechanisms involved in multi-line, multi-attribute RFQs. This section describes two decision

support systems. One is key to negotiation and scores RFQs and Offers. The other is a decision support system for auctions: the combinatorial reverse auction solver.

From the point of view of decision support, we have identified two main processes where the application of Artificial Intelligence (AI) techniques can help the user in the decision making process. These processes have been studied and implemented in *Quotes* with satisfactory results. Summarized below are the main ideas of both.

1. *Multi-parameter scoring algorithm.* Based on the importance that the buyer gives to each parameter of an item in a RFQ (price, quality, delivery time, etc.) and his flexibility to accept offers beyond his preferences, a *fuzzy* offer-matching algorithm scores each offer and ranks it accordingly. This helps the user to easily identify both interesting offers as well as non-competing ones. By defining a reserve score, users can tell the system to automatically reject offers that are clearly a bad choice.
2. *Computation of the optimal bid set for combinatorial auctions.* Given a set of offers for a multi-item RFQ, an AI search algorithm obtains the subsets of offers that optimizes a given criteria such as: minimization of price, start/finish date, maximization of product quality, etc. iSOCO applies leading edge algorithms that compete with the current state-of-the art techniques being studied in academia.

3.3.5.1 Fuzzy matching

Quotes provides both buyers and providers with a fuzzy matching module that allow them to score negotiation messages they receive based on their own preferences. In this manner, a buyer can order incoming offers from different providers in the same way that a provider can order incoming RFQs and counteroffers from different buyers. This is specially useful when dealing with many messages because the more interesting is a message the earlier it should be identified and answered. And the sense of interest is extracted from the preferences both buyers and providers specify.

Most commercial bid selection tools are based on Multi attribute utility theory (MAUT) [2]. We extend these techniques by incorporating *fuzzy* functions (see [4] for further reference).

3.3.5.1.1 Fuzzy functions

As we have previously seen, both buyers and providers define their preferences. On the one hand, buyers specify their preferences when assigning values to the RFQ item attributes (see subsection 3.1.2). And on the other hand, providers specify their preferences when defining preferred values in their product profiles (see subsection 3.2.1). Internally, these preferences are represented as fuzzy functions.

A special feature of fuzzy logic [9] is its ability to handle the concept of relative truth of one proposition “x is P” through the specification of a membership function that represents the predicate P. In our case, we can see a preference is a predicate, and the degree of truth of the proposition as the degree of the preference satisfaction of an offered value x. In this manner, if for example, we consider a domain of four quality values ‘low’, ‘medium’, ‘high’, and ‘luxurious’, and if a buyer asks for ‘luxurious’ quality and a provider offers ‘high’ quality, we cannot say the provider satisfies completely the buyer’s preference, but, since ‘high’ is close to ‘luxurious’, satisfaction should neither be zero.

In Quotes, this degree of satisfaction is computed by means of a fuzzy function (also known as membership function of the fuzzy set defined by predicate P), which is defined for each preference over each item attribute. This section shows how these functions are defined for different types of preferences.

There are some considerations these functions must satisfy:

- Preferences define the fuzzy function, whose domain is defined by the attribute template: its type and possible values (see subsection 3.2).
- Satisfaction values belong to the interval [0,1]. When an offered value x coincides with a preference, satisfaction is 1. Otherwise, it will take decreasing values (down to 0) as x goes further away the preference inside the domain. This implies satisfaction must behave asymptotically when the domain is not limited (that is, for *Any Number* template attribute type).
- Whenever possible, satisfaction must behave symmetrically. Therefore, if for example there is a preferred value of 4, both 3 and 5 offered values should have the same satisfaction value.
- Multiple values in a preference mean they are different preferred options (i.e., they are OR combined). Thus, all values in a Flat preference interval (or preference set) take the maximum satisfaction.
 $\forall x \in \text{Preference interval or Preference Set, Satisfaction}(x) = 1$
- When preferences are specified through intervals with a non-Flat slope, satisfaction values range from 0.5 to 1.
 $\forall x \in \text{Preference MIB or LIB interval, Satisfaction}(x) \geq 0.5$
- Outside non-Flat intervals, those values 'close enough' to the preferred side (i.e., maximum value in a MIB interval or minimum value in a LIB interval) would also take satisfaction values inside [0.5, 1]. By 'close enough' we consider values within a 10% of the interval length. The rest of values always take satisfaction values under 0.5.
 $\forall x \notin \text{MIB or LIB interval} \cup 10\%(\text{interval length}), \text{Satisfaction}(x) \leq 0.5$
- Symmetry does not apply for non-Flat preference intervals. In this manner, right side values outside a MIB (More is better) interval should decrease slower than left side values. And the same applies for LIB (Less Is Better) intervals: small values outside the interval should have higher satisfaction values than higher values.
- In case an offer contains several offered values (x_1, \dots, x_m) for an attribute, the satisfaction of the offer for this attribute must correspond to the best offered value satisfaction:
 $\text{Satisfaction}(x_1, \dots, x_m) = \max(\text{Satisfaction}(x_1), \dots, \text{Satisfaction}(x_m))$
 Thus, for example, if the preference is 4 and the offer has both 3 and 4 values, the returning satisfaction must be 1.

Keeping these considerations in mind, we present some examples of different fuzzy functions for different types:

Example 1: Domain attribute type: range of numbers, preference type: single.

Next Figure 3-13 shows the corresponding fuzzy function for a preference value of 10 considering as template domain a [-20, 20] range of numbers. The function is computed so that satisfaction is 1 at 10 and it decreases down to 0 at the further extreme of the domain interval (which in this case is -20). Since symmetry must be respected, the same decreasing slope is applied to values at the right side of the preferred value. Nevertheless, since the maximum value for the domain is 20, this value takes a satisfaction value of 0.66.

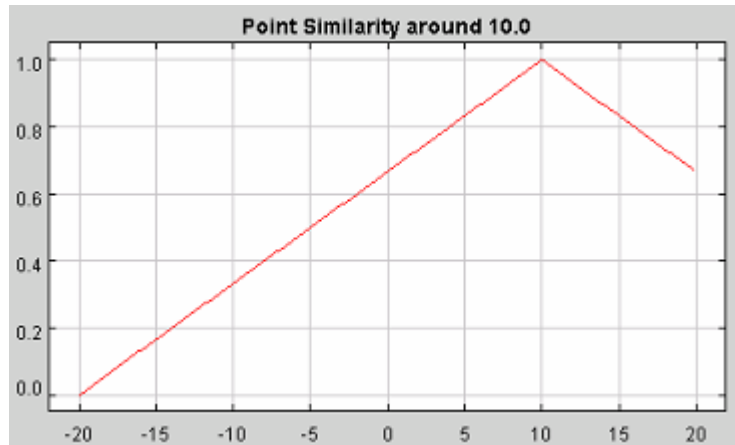


Figure 3-13: Fuzzy function for a single preference inside a range of numbers domain.

Example 2: Domain attribute type: ordered set of labels, preference type: flat interval.

This second example corresponds to the case the domain has been defined as an ordered set of labels: QA, QB, ..., QI and the preference has been set as a flat interval [QC, QE]. In this case, labels inside the interval have satisfaction equal to 1, and as in the previous example, satisfaction values decrease for labels down to 0 at QI (the further label). Again, labels on the left of the preferred interval decrease with the same slope than the right side, so QA gets a satisfaction of 0.5.

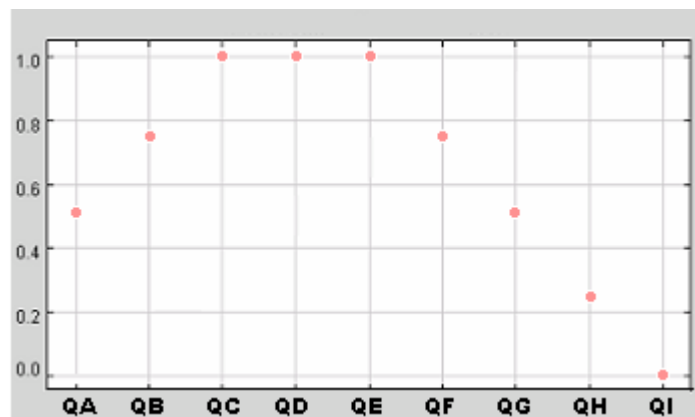


Figure 3-14: Fuzzy function for a flat interval preference and ordered set of labels domain

Example 3: Domain attribute type: any number, preference type: MIB interval.

Last example corresponds to a MIB (more is better) interval preference ([-20,20]) considering a non-restricted numerical domain (that is, any number is allowed). Satisfaction values for this domain are computed using a 4-piece formula (see Figure 3-15):

- (1) $\forall x \in [-20, 20], \text{ Satisfaction}(x) = 0.5 + \frac{x + 20}{80}$
- (2) $\forall x \in [20, 24], \text{ Satisfaction}(x) = 1 - \frac{x - 20}{8}$

$$(3) \quad \forall x > 24, \text{ Satisfaction}(x) = e^{-\frac{(x-\text{mean})^2}{2 \cdot \text{variance}}} \quad \text{where:}$$

$$\text{variance} = \text{spread} \cdot (\text{Length}([-20, 20] \cup [20, 24]))^2 \cdot \rho$$

$$\text{mean} = 24, \text{ spread} = 2, \rho = 2$$

$$(4) \quad \forall x < -20, \text{ Satisfaction}(x) = e^{-\frac{(x-\text{mean})^2}{2 \cdot \text{variance}}} \quad \text{where:}$$

$$\text{variance} = \text{spread} \cdot (\text{Length}([-20, 20] \cup [20, 24]))^2 \cdot \rho$$

$$\text{mean} = -20, \text{ spread} = 2, \rho = 1$$

First part (1) corresponds to the satisfaction inside the preference interval, and increases linearly from 0.5 at the minimum up to 1 at the maximum. Second piece (2) corresponds to the 10% additional interval, which is 4 units long, and decreases from 1 at the maximum of the preference interval (i.e., 20) down to 0.5 at the maximum of the additional 10% interval (i.e., 24).

Third and fourth pieces define satisfaction degrees for the domain values outside the previous intervals. Since the domain is not restricted, we use probabilistic functions that decrease asymptotically on the x axis. Mean values are set to be the preference interval point that is closer to the described side, assuming the interval to be the union of both defined preference and additional 10% intervals. In this manner, for the right side (3) mean is the maximum of the additional 10% interval (i.e., 24) and for the left side (4) it is the minimum of the defined preference interval (i.e., -20).

Regarding variance values, they are assigned as the product of a spread value, a ρ value, and the square value of the length of the interval. Spread is fixed to 2 for both sides, but since symmetry is not desired for a MIB interval, ρ takes different values for each side. More concretely, ρ is set to 2 on the right side (3) and to 1 on the left side (4), so that variance in the side closer to the more preferred values is twice the variance of the other side.

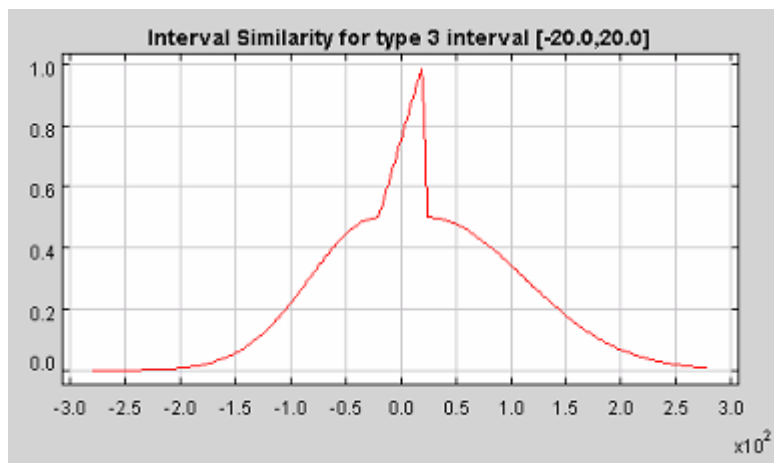


Figure 3-15: fuzzy function for a MIB interval preference and any number domain.

3.3.5.1.2 Offer evaluation

An RFQ specifies buyer's preferences. Quotes scores Providers' offers for an RFQ so that the buyer can have an ordered list of offers, having first the offers that best satisfies his/her necessities. On the other hand, an offer is a set of lines or items describing offered products or services (which must have been previously asked by the buyer in the RFQ the offer is

answering to). Each item is described through a set of attributes. Therefore, to score an offer in relation to an RFQ, it is necessary to compute satisfaction values at the level of attributes and propagate it to line level and up to offer level.

In a more formal manner, having an offer O composed by m items O^j $j=1..m$, we describe an offer item j as the n -tuple of the form $\langle o_1^j, o_2^j, \dots, o_n^j \rangle$ where o_i^j represents the i th attribute value of item j .

In order to compute attribute scoring, Quotes creates the fuzzy functions previously seen (see subsection 3.3.5.1.1), where attributes at the RFQ items represent preferences and chosen templates specify domain. Offer values, which are always single, are then used to compute satisfaction degrees:

$$(5) \quad \text{scoring}(o_i^j) = \text{Satisfaction}(o_i^j)$$

Attribute satisfaction degrees are then weighted to obtain offer item scoring. Since preferences are RFQ values, weights w_i^j are taken to be the importance the buyer assigned to an attribute i in the j -th RFQ item (see subsection 3.1.2).

$$\text{scoring}(O^j) = \sum_{\text{attribute } i=1}^n \frac{\text{scoring}(o_i^j) \cdot w_i^j}{n}$$

Finally, we compute offer scoring as a weighted combination of item scorings. Currently, weights w^j are set to 1.

$$S^B(O) = 100 \cdot \text{scoring}(O) = 100 \cdot \sum_{\text{item } j=1}^m \frac{\text{scoring}(O^j) \cdot w^j}{m}$$

Scoring values are translated from $[0,1]$ to $[0,100]$ range in order to fit the definition of buyer scoring function $S^B(O): O \rightarrow [0..100]$ assumed by Quotes.

3.3.5.1.3 RFQ Evaluation

Similarly to offer scoring functionality, Quotes provides providers with an RFQ scoring computation that prioritizes buyers' requests. Again, fuzzy functions are generated for each attribute, taking domain as defined by templates and preferences as provider's preferred values (see subsection 3.2.1).

The main difference is in the computation of the (5) formula due to that attribute values in RFQ items are not restricted to be single, so that value intervals or sets must be evaluated against these fuzzy functions. In this case, Quotes assumes attribute scoring to be the best satisfaction degree of all values x defined in the attribute value range rfq_i^j .

$$\text{scoring}(rfq_i^j) = \text{MAX}_{\forall x \in rfq_i^j} (\text{Satisfaction}(x))$$

3.3.5.2 Winner determination in combinatorial auctions

Allowing providers to bid on combinations of products of an *RFQ* has the interesting feature of enhancing economic/service efficiency (providers would offer price discounts or/and better service if they obtain all the business) [5]. However, the determination of the optimal winning combination is a complex problem that, excluding very small instances, cannot be solved manually with common data analysis tools. This has recently attracted some research [1][6]. *Quotes* provides with an optimisation module to cope with this situation. The core of this module is a branch & bound [3] systematic global search algorithm. The buyer decides the target attribute (overall score, price, quality, etc) and the optimisation criterion (minimise/maximise). *Quotes* returns a collection of offers which, in case of being accepted, would optimise the desired target.

Not only is the determination of the optimal winning combination required at auction time, but also the buyer may need it when holding one-to-many negotiations with providers as described in section 3.3.3. In this case, the buyer may call the same optimisation module employed for auctions (see Figure 3-16) in order to obtain the set of optimal offers and thus discover which providers are more promising to continue negotiations with.

Buyer Side

Expand each item to view its related negotiations

RFQ View

RFQ Name Pedido edificio A

Items

Template	Description	Status	
347	Las mesas edificio A		Expand
346	Las sillas edificio A		Expand

Optimal Bid Set Start auction

Buyer Side

You can create a new RFQ by adding several items

Optimal Set of Offers

RFQ Name Pedido edificio A

Offers

Negotiation ID	Provider	Oferted Items	State	Score
4214	pmanual2	Las mesas edificio A		
4212	pmanual1	Las sillas edificio A		

Target Attribute: Plazo de entrega Criterion: Minimize Recalculate Optimal Set

Figure 3-16. Optimal bid set computation

4 Present and future of e-sourcing

E-sourcing is an established part of the business landscape. This year alone, \$270 billion worth of corporate spending will be channeled through e-sourcing in the US, and the marketplace

itself is said to be worth some \$1.15 billion to software vendors [16]. However, there is still an enormous challenge confronting users who want to get the maximum value out of e-sourcing. Many companies have realised this value on a transactional level but failed to see it through to the bottom line. The early value of e-sourcing tools has been tactical rather than strategic, as 88 percent of all adopters got into e-sourcing primarily to negotiate price reductions. Adopters picked up quick cost-cutting benefits and ROI⁹ from core RFx and reverse auction tools. These benefits included a 14.3 percent reduction in goods and services costs, a 50 percent shortening of cycle times, a 60 percent reduction in sourcing administration costs, and a 10 to 15 percent reduction in time-to-market.

The bad news, for many adopters, is that these well-documented savings have not been converted to the bottom line. Reasons for low conversion rates include a lack of savings implementation strategies, an inability to effectively communicate negotiated terms to the enterprise, and insufficient integration between e-sourcing and order-execution systems. The solution is to go beyond RFx/reverse auctions and implement a workable e-business strategy that incorporates provider performance measurement, sourcing collaboration, contract management, process- and commodity-specific RFx templates, and related tools that can convert e-sourcing savings into bottom-line savings.

Furthermore, good e-sourcing is not just about buying software but also making the right cultural and operational changes. Some global businesses have made an effort to tie together normally isolated departments, like purchasing and logistics, in the hope that the newly integrated business units and processes will result in better conversion.

By industry, the highest penetration is in chemicals and pharmaceuticals (24 percent) and automobile manufacturing (20 percent), whereas retail, wholesale and distribution, metals and metal products, finance, banking, and accounting are at 4 percent each. Across all industries, e-sourcing adopters are using the technology primarily for MRO goods (71 percent), followed by standard parts (67 percent) and raw materials (57 percent).

As to e-sourcing applications in the market, although we are witnessing their continuous, tightly competitive progress we can still identify two major, several commonly unsatisfactorily solved issues that prevent them from supporting effective strategic sourcing, namely:

- *Capability to support sourcing processes for varying industries and businesses.* Since sourcing processes are highly dependent on each business case (because of each industry's particularities and individual businesses' practices) it is extremely complex to capture in a single product all processes and negotiation requirements of a general-purpose solution. Thus e-sourcing solutions are far away from product maturity. Along this direction, perhaps the major drawback of market solutions is the lack of support for creating customised sourcing events as compositions of negotiation processes. For instance, a sourcing event might be composed of a pre-defined sequence of auctions or an interleave of auctions and negotiations.
- *Decision support tools for strategic thinking.* Strategic sourcing is founded on the availability of powerful decision support tools. Nonetheless current vendors' solutions are lacking as to this matter. As an example, most tools share the commonality of not providing support for determining the winner in combinatorial negotiations and auctions. The unavailability of such support poses an intricate, combinatorial problem to professional buyers that leads them to either relinquish or opt for alternative, and less

⁹ ROI stands for *Return of Investment*

efficient, non-combinatorial protocols. In general, vendors are expected to make headway in the development of value-added decision support tools.

In this chapter we have tried to exemplify how the sourcing process can be highly automated, allowing companies to achieve enormous benefits: cost savings, processing time reduction, less time-to-market and more time left to strategy. We have presented iSOCO's contribution along this direction by dissecting *Quotes*, an Internet-enabled sourcing solution capable of streamlining the sourcing process. *Quotes*' main strengths can be summarised as follows:

- *Quotes* allows goods and services to be represented and managed with all their attributes, overcoming rigid and unreal price-discovering approaches.
- *Quotes* provides a powerful negotiation framework based on the composition of structured negotiation protocols and flexible reverse auctions.
- *Quotes* provides with the necessary tools to help users manage the complex sourcing mechanisms involved in multi-line, multi-attribute RFQ.

Most importantly, the fundamental contribution of *Quotes* lies on the incorporation of highly valuable decision-making support functionalities targeted at spurring the transition of sourcing processes from transactional to strategy-centered.

Finally, there is an important aspect affecting the successful exploitation of e-sourcing applications. Auction design matters. Although for the sake of generality we should state instead that *negotiation design matters*. Indeed there are many parameters involved in the design of negotiation scenarios. Thus we must keep in mind that careless, faulty designs of such scenarios may eventually lead to terribly catastrophic outcomes as reported through several case studies in [14].

References

- [1] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceeding of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI'99*, 548–553, August 1999.
- [2] R. L. Keeny and H. Raiffa, *Decision Making with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge, UK: Cambridge University Press, 1993.
- [3] R.Korf. Artificial intelligence search algorithms, *CRC Handbook of Algorithms and Theory of Computation*, M.J. Atallah (Ed.), CRC Press, Boca Raton, FL, 1998, pp. 36-1 to 36-20.
- [4] R. A. Ribeiro, "Fuzzy multiple attribute decision making: A review and new preference elicitation tech-niques," *Fuzzy Sets and Systems*, 78; 155-181,1996.
- [5] M.H. Rothkopt, A. Pekec, and R.M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131-1147, 1995.
- [6] T. W. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *Proceeding of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI'99*, pages 542–547, August1999.
- [7] Sandholm, T. eMediator: A Next Generation Electronic Commerce Server. International Conference on Autonomous Agents (AGENTS), Barcelona, Spain, June 3-8.
- [8] Sven de Vries and Rakesh Vohra. Combinatorial Auctions: A survey. January 12th, 2001.

- [9] Lluís Godó and Ramón López de Mántaras. "Fuzzy Logic" in Encyclopedia of Computer Science and Technology. Marcel Dekker, Inc. Pennsylvania 1993. Vol 29, pp 211-229.
- [10] Ariba Enterprise Sourcing feature description available on the web at http://www.ariba.com/solutions/product_features.cfm?solutionid=6
- [11] Commerce One Source features description available on the web at http://www.commerceone.com/solutions/products/source_features.html
- [12] MetalSite: the auto-referenced Internet's premier site for buying and selling metals at <http://www.metalsite.com/>.
- [13] Aberdeen Group. Making E-Sourcing Strategic: From Tactical Technology to Core Business Strategy, 2002.
- [14] McAfee, R. Preston. Auction Design for the Real World. Notes from a seminar, 1995.
- [15] Antonio Reyes Moro, Juan A. Rodríguez-Aguilar, Jesus Cerquides, Maite López-Sánchez, David Gutiérrez-Magallanes. Quotes: a Negotiation Tool for Industrial Procurement. Fourth International Conference on Enterprise Information Systems (ICEIS 2002), April 4-6, Ciudad Real, Spain, 2002.
- [16] Stephens Inc. Internet Supply Chain Team. Strategic Sourcing: Applications to Turn Direct Materials Procurement into Competitive Advantage. Industry Report, January 30, 2001.