

IBM Research Report

ECCO - Automated Evaluation of Configurable Offerings

Martin Bichler, Jayant R. Kalagnanam, Ho Soo Lee

IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598



Research Division

Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

ECCO - Automated Evaluation of Configurable Offerings

Martin Bichler, Jayant R. Kalagnanam, Ho Soo Lee
IBM T.J. Watson Research Center
P. O. Box 218, Yorktown Heights, NY 10598, USA
{bichler, jayant, leehs}@us.ibm.com

Abstract

Up until now e-commerce is primarily focused on the trading of simple goods and services, where products can be described either by price alone or as a set of attributes. There is little support for trading complex, configurable products, such as computer systems, automobiles, insurances and services in general. Companies want to communicate offers including complex rules and business policies, and they need decision support to evaluate these offers. This paper describes ECCO, a decision analysis methodology to evaluate configurable offers. Configurable offers allow multiple options for each attribute and they include rules on how to combine the various attribute values. These offers exhibit some combinatorial aspects. With only 10 attributes and 5 possible attribute values for each of them, one can generate almost 10 million possible configurations. ECCO helps a user to select the best out of all possible configurations based on a user's preferences. We draw on mathematical programming and conventional decision analysis techniques. We also introduce CPML, an XML schema for describing configurable offers in a compact way, and outline possible application scenarios, as well as a concrete implementation of the methodology in an e-marketplace environment.

Keywords: Configurable product, bid evaluation, decision analysis, integer programming, XML/EDI

1 Introduction

Decision support for the evaluation of bids and offerings is a core functionality of today's electronic commerce systems. Decision support is essential for the evaluation of complex, multi-attribute bids in a Request for Quotation (RFQ) as well as multi-attribute auctions and negotiation support tools (Bichler 2001). For the evaluation of bids, while the price of a product could be important, so could be its delivery time and various other qualitative attributes.

Commercial bid analysis products of Frictionless Commerce¹ and Perfect.com² are based on traditional decision analysis techniques, which have been actively studied in Multi-Attribute Decision Analysis (MADM), an area of Operations Research. The most wide-spread techniques in the field are Multi-Attribute Utility Theory (MAUT) (Keeny and Raiffa 1993), Simple Multi-Attribute Rating Technique (SMART) (Edwards 1977) and the Analytic Hierarchy Process (AHP) (Saaty 1980), all of them implemented in several software applications. The essence of all these widely used decision aids is breaking complicated decisions down into small pieces that can be dealt with individually and then recombined in an additive manner. The buyer's preferences are defined as a weighted utility or value function, which is used to score and rank the bids. The key difference among the various methods is the way the scores on individual attributes and their weights are assessed.

Recent interest in electronic markets and electronic procurement systems have revived and accelerated interest in decision analysis techniques. Several negotiation support systems use decision analysis to evaluate offers in a bilateral bargaining situation (Kersten 1998). Multi-attribute auctions are an approach to support procurement negotiations on multiple attributes, not just the price of a product (Bichler et al. 1999). Other approaches have tried to enhance existing decision analysis techniques. For example, methods from fuzzy decision analysis use fuzzy sets for weights and individual scoring functions and fuzzy operators for the aggregation of those fuzzy sets (Ribeiro 1996). ABSolute is a Java-based decision analysis framework, which provides advanced visualization capabilities based on parallel coordinates to help the user navigate the information space (Bichler et al. 2001a). Q-Eval (Iyengar et al. 2001) and WORA

¹ <http://www.frictionless.com>

² <http://www.perfect.com>

(Bichler et al. 2001a) are both methodologies to help the user assess the weights of individual attributes based on pairwise comparisons of alternatives. A broad survey of decision analysis techniques can be found in (Olson 1995).

The assumption in all of these approaches is that alternatives (or bids) are described as *set of attribute value pairs*. In practice, however, offerings are more than simple attribute value pairs. Suppliers have complicated rules and policies associated with their offerings. For example, many offers are specified as configurations, where each attribute can take a number of different attribute values. These offers specify pricing rules for different attribute values, constraints on the combination of attribute values as well as volume discounts. Most services such as insurances or transportation can be considered as *configurable offerings*. Another example is personal computers. Vendors such as IBM, Dell, and Compaq, describe their offerings along 20 to 30 attributes on their Web site, and each of these attributes can take multiple values. This way, customers can configure and order a PC tailored to their individual needs.

Lets assume a PC has only three attributes, namely processor speed, hard disk size, operating system, and price. A supplier could specify that there are three processors available {850MHz, 950 MHz, and 1GHz}, as well as two sizes of hard disks {10GB and 15GB}. The base configuration (850MHz, 10GB) prices for \$1000.--. A configuration with a 1GHz computer is \$100.-- more, and one with a 15GB hard disk costs an additional \$200.--. Of course, there can be many more rules, which need to be considered. For example, a certain operating system might require a minimal processor speed to function properly, and the supplier might specify a linear decreasing volume discount.

These types of configurations exhibit some combinatorial aspects. With only 10 attributes and 5 possible attribute values for each of them, there are already *9.7 million* possible configurations. Clearly, finding the best configuration among all these is not an easy task. One possibility is to enumerate all possible configurations. This approach has considerable disadvantages. First of all, a large amount of bids has to be generated and communicated to the customer. Not only is it computationally expensive to enumerate all possible configurations, it is also expensive to transfer and evaluate the huge amount of data. Second, all these configurations have to be priced consistently. Suppliers often restrict the number of configurations they offer and communicate only a selection. As a result, buyers might not find the best configuration and choose the offer of another supplier, i.e. it might lead to inefficient outcomes.

Traditional bid evaluation tools can only handle individual configurations described as attribute-value pairs. There is a clear need to provide bid evaluation tools that operate on the rules of a complex offering, and there is a need for more expressive languages to communicate these rules in a machine-readable way. In section 2, we will introduce a language to flexibly describe configurable goods and services. The respective XML schema enables suppliers to express pricing rules for different configurations, volume discounts, as well as constraints among attribute values. Based on this language we describe the ECCO (Evaluation of Complex, Configurable Offers) bid evaluation methodology in section 3, which allows buyers to extract the best configurations based on information about their preferences. Section 4 will describe an existing implementation and endowment of the method, and finally, section 6 will conclude with a discussion and comparison to other approaches.

2 Compact Description of Configurable Offers

A prerequisite to evaluate configurable offers is a language, i.e. number of message formats to express these offers in a compact way. In a first step we will describe the necessary constructs to express configurable offers in a mathematical notation. Based on this we will introduce CPML, an XML schema to describe configurable offers and communicate them as EDI messages over the Internet.

2.1 Functional Specification of a Configurable Offer

A compact format defines the total price of a configurable product as a function of the possible attribute values x_j of a product, j , and the quantity q_j purchased³. The vector a provides the parameters of a certain functional form. In general this can be any nonlinear function:

$$p_j = p_j(q_j, \mathbf{x}_j, \mathbf{a}) \quad (1)$$

³ We use lower-case letters to denote scalars, lower-case bold letters to denote vectors, and upper-case letters to denote matrices.

Typical forms for such a function could be based on classic functional forms such as the Cobb-Douglas function

$$f(x_1, x_2) = x_1^{a_1} x_2^{a_2} \quad (2)$$

where x_i are two attributes (one of them could be quantity q) and a_i the parameters. In practice it is convenient to use purely additive functions to specify the pricing rules of an offering. Assuming additivity of all the attributes the total price p_j for a particular product j , which is denoted in (1), can be written as:

$$p_j(q_j, \mathbf{x}_j) = q_j * p_{bj}(q_j) + q_j * \sum f_{ij}(x_{ij}) \quad (3)$$

where q_j is quantity, $p_{bj}(q_j)$ is the base price per item as a function of quantity (i.e. specifies a volume discount), and $f_{ij}(x_{ij})$ is a functional specification for the impact of particular attribute values on the base price of a product. The individual functions $p_{bj}(q_j)$ and $f_{ij}(x_{ij})$ can in general be nonlinear, but are often approximated using linear or discrete functions. Examples would be discrete functions, which specify price markups for different types of CPUs (850 MHz, 900 MHz, 1GHz), as well as continuous functions, which specify the impact of decreasing lead time on price.

An important requirement is the ability to specify rules that allow the supplier to express certain constraints on the configurations. For example, certain values of attribute x_1 can only be combined with certain values of x_2 , e.g. a certain type of CPU can only be used with a certain motherboard. Also, certain combinations of attribute values might lead to a price discount or markup. For example, a buyer, who chooses the premium configuration, i.e. the best possible attribute values for all attributes, might get a discount. We will call these constraints “*add on rules*”. In order to express these types of constraints we can use propositional logic operators (and \wedge , or \vee , implies \Rightarrow , equivalent \Leftrightarrow , not \neg) (Russell and Norvig 1995). A rule can then be expressed in terms of logical operations between attribute values (see section 3.3). Propositional logic has shown to be sufficient for our needs. Although, predicate logic provides for greater expressiveness including quantifiers, most rules we found in real-world applications could easily be expressed using propositional logic operators.

2.2 CPML – An XML Schema for Configurable Products

In business-to-business relationships, it is essential to have product data and offers available in machine-readable format. Buyer organizations want to be able to import product data into their ERP (enterprise resource planning) systems, in order to make this data available to employees internally, to compare product specifications of different suppliers, or to reuse product and price data for various planning, engineering or accounting purposes.

Many suppliers make their catalog data available in various formats, such as ANSI X12 832, Ariba cXML⁴, Catalog Interchange Format (CIF), Commerce One xCBL⁵, Requisite eCX 2.0⁶, and RosettaNet PIP 2A1⁷. Based on the various standards and classification schemes many e-marketplace providers as well as vendors of e-procurement systems provide tools and services for catalog aggregation. The promise is that buyers can perform searches across multiple suppliers, getting up-to-the-minute pricing, inventory, and sourcing information. In fact, the management of catalog data is the key feature of most e-procurement solutions. Today’s catalog standards focus mostly on parts catalogs, where the product is described as a set of attributes and eventually a price. An example is CIF, which is a simple comma delimited text format (see Figure 1).

<pre> DATA 942888711,100,,"Ball-point Pen",1213376,1.95,EA,,,, 942888711,101,,"No. 2 Pencil",1213377,1.50,DZN,,,, 942888711,102,,"Rubber Eraser",1213472,0.25,PK,,,, 942888711,103,,"Stapler, Standard",1237461,2.95,BX,,,, ENDOFDATA </pre>
--

Figure 1: CIF Example

⁴ <http://www.cxml.org/>

⁵ <http://www.xcbl.org/>

⁶ <http://www.ecx-xml.org/>

⁷ <http://www.rosettanet.org>

Currently, there is no support for the description of configurable offerings in a machine-readable way. We have defined CPML, an XML schema, which can be used to describe configurable products. The schema defines messages for a product request, for conventional multi-attribute offers, as well as configurable offers.

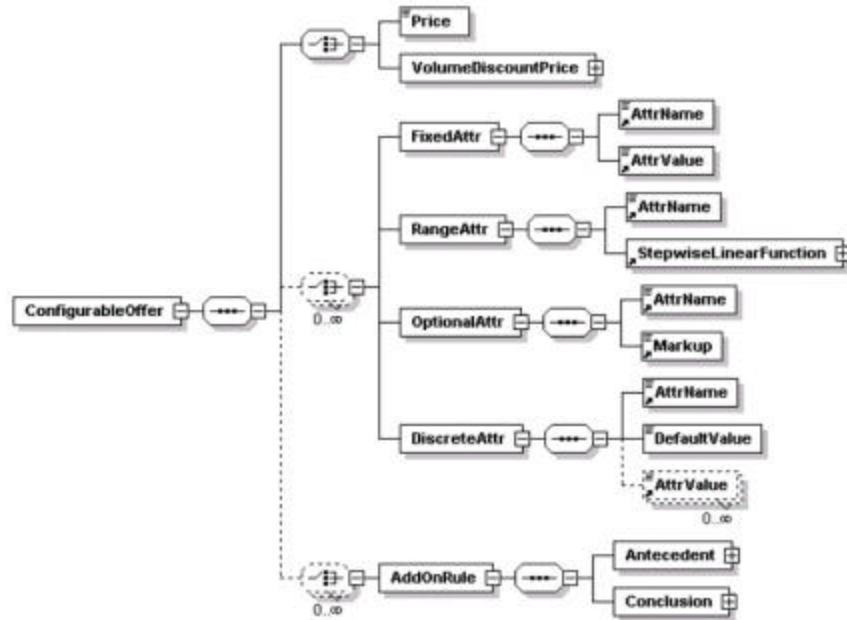


Figure 2: Main Elements of CPML Configurable Offer

Figure 2 shows the main elements of a CPML configurable offer element (TAG: ConfigurableOffer) in a tree structure. A configurable offer defines either a single price for a base configuration and quantity, or a volume discount price (TAG: VolumeDiscountPrice), which is a function of quantity. The offer can have multiple attributes, which are either fixed attributes (TAG: FixedAttr), range attributes (TAG: RangeAttr), optional attributes (TAG: OptionalAttr), or discrete attributes (TAG: DiscreteAttr). Fixed attributes are described as simple attribute-value pairs, similar to the ones one can find in traditional catalog standards. Range attributes allow the specification of a certain range of values and their impact on price. For example, one can specify lead time as an attribute ranging from 7 to 30 days. The faster the good needs to be delivered, the higher the markup on price will be. Optional attributes specify dichotomous features. For example, a PC might be delivered with a certain modem or without. Discrete attributes allow for a larger number of allowed attribute values. With discrete attributes a buyer can specify a number of different modems to choose from, as well as a price markup for each of them.

In addition a configurable offer can specify so called add on rules (TAG: AddOnRules). Add on rules have the form of a logical implication with an antecedent and a conclusion. The antecedent can be any logical proposition including conjunctions, disjunctions, negations as well as atomic propositions. The conclusion defines in addition to this a Markup tag, which specifies a certain discount or markup. This way, suppliers can specify a certain discount as a result of certain attribute values.

A configurable offer is one of a sequence of messages and a response to a particular request, which specifies a buyer's basic requirements. The tags defined in CPML can be used standalone or they can also be embedded to enrich existing catalog standards. In the first version of this XML schema we have tried to make it easy to understand and use, but at the same time flexible enough to cover the broad range of business rules, one can find in practice. Using the CPML tags we were able to describe configurable offers in a wide variety of industries ranging from logistics to insurances and electronic components.

3 Bid Evaluation for Configurable Offers

The basic purpose of CPML is to communicate pricing rules about configurable offers in a compact way. This data can then be used to evaluate and compare offerings from various suppliers. As already mentioned in the introduction, traditional decision analysis methods cannot be used to evaluate configurable offerings with complex pricing rules. The ECCO bid evaluation methodology is a new algorithm to evaluate and select the best out of the multiple possible configurations of a configurable offer. It helps the user to score and compare the configurable offers of multiple suppliers. ECCO operates directly on the rules and functional specifications of a configurable offer without the need to enumerate all possible configurations. This way, we avoid the computationally expensive process of generating, transmitting and evaluating huge amounts of individual configurations by just communicating the rules for a single offer. ECCO consists of two consecutive steps, namely preference elicitation and bid evaluation in which we deploy mathematical programming techniques. The treatment of logical configuration rules and a buyer's business rules requires additional consideration and is treated in a separate subsection.

3.1 Preference Elicitation

An important step for the evaluation of offers in general is the elicitation of a buyer's preferences. These preferences have to be captured in some kind of utility or value model, which is the basic means for the evaluation of offers. Traditional decision analysis literature such as MAUT (Clemen 1996) or conjoint analysis (Hu 1998) have used linear or "additive" utility functions. This approach assumes that there are no interaction effects among the attributes and all attributes are mutually "preferential independent". An attribute x is said to be preferentially independent of y if preferences for specific outcomes of x do not depend on the level of attribute y (Olson 1995). If this assumption is found to be inappropriate, one has to model these interaction effects in a non-linear utility function. However, for most real-world applications, an additive utility function is used as a good rough-cut approximation.

In the case of an additive scoring function $U(\mathbf{x})$ the buyer evaluates each relevant attribute x_i through a scoring function $u_i(x_i)$. The overall utility $U(\mathbf{x})$ for a product \mathbf{x} is the sum of all individual scorings of the attributes in vector \mathbf{x} . It is convenient to scale U and each of the single-attribute scoring functions u_i from zero to one, and to weigh the individual attributes. All weights w_i with $(0 < i < n)$ sum up to 1. That is, for an attribute vector \mathbf{x} that has values $x_1 \dots x_n$ and a scoring function that has weights $w_1 \dots w_n$ on the n relevant attributes, the overall utility for a bid is given by

$$U(\mathbf{x}) = \sum_i w_i u_i(x_{ij}) \quad (4)$$

Elicitation of weights is of major importance for the utility model. Lots of previous literature has focused on the assessment of appropriate weights, which reflect the user's true preferences. For now, we assume the weights as given.

3.2 Bid Evaluation

The bid evaluation can be modeled as an integer programming (IP) problem. We pick the individual configuration that maximizes the user's utility, given a certain budget constraint, add on rules and a number of additional business rules. The utility function in (5) is adapted accordingly to accommodate for multiple sourcing, i.e. multiple winning configurations, and then used as an objective function in an integer program. The configurable offerings are turned into inequalities and used as side constraint in the optimization.

$$Max_{x,q} U(\mathbf{x}, q) = \sum_{j \in M} \sum_{i \in N} (w_i u_i(x_{ij})) \quad (5)$$

s.t.

$$\sum q_j \geq d \quad \forall j \in M \quad (6)$$

$$p_j = p_j(q_j, \mathbf{x}_j) \quad \forall j \in M \quad (7)$$

$$\sum p_j \leq B \quad \forall j \in M \quad (8)$$

The goal is to maximize the objective function (6) under the constraint (7) that the overall quantity $\sum q_j$ provided is equal or greater than the demand d . Constraint (8) is provided by the functional form of the configurable offers as described in (3). Constraint (9) specifies that the sum of the winning prices is lower or equal to the budget constraint B . If we restrict the demand to a quantity of 1, the optimization finds the

best configuration among all offers. This is also true for single sourcing, where quantity is fixed and a buyer wants to buy the entire demand from a single supplier. In order to get a ranking of the n best configurations, the optimization has to be repeated n times. In each run we add an additional constraint that restricts the currently best configuration from optimality. This way we get a ranking of the n best configurations. For each configuration we calculate the utility to indicate to the user, how well the configuration satisfies her preferences.

3.3 Treatment of Buyer's Business Rules

An important set of rules, which have to be considered for the selection of configurations can be derived from a buyer's business rules. These business rules are concerned with the number of winning suppliers, some constraints on the homogeneity of the different supplies, or the allocation per supplier. It is important that these business rules are considered in the selection process, although most of them are only relevant in the case of multiple sourcing, where the entire demand is purchased from more than one supplier. Some of these side constraints might arise from:

- The minimum and maximum allocation per supplier, $z_j q_{j,min} \leq q_j \leq z_j q_{j,max}$, where z_j is a counting variable which is 1 if supplier j is chosen.
- Minimum and maximum number of winning suppliers can be modeled by $S_{min} \leq \sum_j z_j \leq S_{max}$
- If there exist requirements regarding selection of minority suppliers then we can ensure at least one (or k) minority suppliers by adding a constraint, $\sum_{j \in M} z_j \geq K$ where M is the set of minority suppliers.
- Often piecewise linear volume discounts are offered by suppliers based on the total spend. We can model this by introducing indicator variables y_{jk} for each offering j and for each spend level k . We replace constraint (9) by the following set of constraints:

$$p_j = \sum_k I_{jk} p_{jk} \quad (9)$$

$$y_{jk} P_{min,jk} \leq p_{jk} \leq y_{jk} P_{max,jk} \quad (10)$$

$$p_{jk} = p_j(q_j, x_j) \quad (11)$$

$$\sum_k y_{jk} \leq 1 \text{ where } y_{jk} \in \{0, 1\} \quad (12)$$

- If goods are purchased from multiple suppliers, suppliers often want to make sure that certain qualitative attributes are the same throughout the entire quantity purchased. This requires the introduction of additional constraints, which ensure homogeneity of the purchase.

Note that the I_{jk} denotes the discount levels at different spend levels. In an initial step we assume linear or stepwise linear functions for all $u_i(x_{ij})$, $u_p(p_j)$, $f_i(x_i)$. This way, the optimization problem can be solved as a mixed integer program (MIP).

3.4 Treatment of Logical Configuration and Discount Rules

The basic formulation of a configurable offer in CPML is a functional form, which specifies, how different attribute values impact the base price of a product. It is essential for suppliers to express also configuration rules, which define constraints on the combination of attribute values or discounts and markups based on some combination of attribute values (see section 2.1). For instance, a *configuration rule* may include compatibility restrictions, saying attribute value x_{12} cannot be connected to attribute value x_{23} , or requirements like attribute value x_{12} and x_{32} needs attribute value x_{23} . In CPML we provide the possibility to express these rules as logical implications. The proposition

$$x_{23} \Rightarrow \neg x_{31} \quad (13)$$

describes the configuration rule that if x_{23} is true, then x_{31} is false. For example, if a certain motherboard is selected by the user, then she is restricted from using a certain type of CPU. Logical implication (\Rightarrow) allows, however, that if any other kind of motherboard is selected, the particular CPU may or may not be chosen by the user. Another type of rules, which is often found in practice are so called *discount rules*. For example,

$$x_{12} \wedge x_{31} \Leftrightarrow p' \quad (14)$$

where x_{ij} describes particular attribute values and p^{\sim} describes a certain discount. The discount is only given, if and only if $x_{12} \wedge x_{31}$ is true. If $x_{12} \wedge x_{31}$ is false, then there is no discount. Therefore, we use the equivalence operator (\Leftrightarrow).

For the evaluation of a configurable offering, these add on rules have to be considered in the IP formulation. Several logic-based modeling frameworks for IP have been proposed, investigating both propositional logic and predicate logic. In order to obtain an equivalent mathematical representation for any propositional logic expression, one must first consider basic logical operators to determine how each can be transformed into an equivalent representation in the form of an equation or inequality. Raman and Grossman (Raman and Grossmann 1991) specify a table with basic transformations, which can then be used to convert general logical expressions into an equivalent mathematical representation. Some of them are described in Table 1.

Logical relation	Pure logical expression	Representation as linear inequalities
Logical “OR”	$P_1 \vee P_2 \vee \dots \vee P_n$	$x_1 + x_2 \dots + x_n \geq 1$
Logical “AND”	$P_1 \wedge P_2 \wedge \dots \wedge P_n$	$x_1 \geq 1; x_2 \geq 1; \dots; x_n \geq 1$
Implication (\Rightarrow)	$\neg P_1 \vee P_2$	$1 - x_1 + x_2 \geq 1$
Equivalence (\Leftrightarrow)	$(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1)$	$x_1 - x_2 \leq 0; x_2 - x_1 \leq 0$

Table 1: Representation of logical relations with linear inequalities

A common approach to convert a general logical expression into inequalities is to first transform it in its equivalent conjunctive normal form (CNF) representation. CNF involves the application of pure logical operations (and \wedge , or \vee , not \neg), and is a conjunction of clauses. A clause is defined as a set of basic literals separated by \vee -operators, such as

$$(P_1 \vee P_2) \wedge (P_3 \vee P_4) \quad (15)$$

CNF can easily be expressed as a set of linear inequality constraints, as shown in Table 1. We have chosen this approach to transform the configuration and discount rules in CPML into appropriate constraints in our IP formulation described in (6) – (9). Formulas (16) to (21) show how the proposition in (16) can be translated into inequalities.

$$x_{12} \wedge x_{23} \Leftrightarrow p^{\sim} \quad (16)$$

$$\neg(x_{12} \wedge x_{23}) \vee p^{\sim} \wedge (\neg p^{\sim} \vee (x_{12} \wedge x_{23})) \quad (17)$$

$$(\neg x_{12} \vee \neg x_{23} \vee p^{\sim}) \wedge (\neg p^{\sim} \vee x_{12}) \wedge x_{23} \quad (18)$$

$$x_{12} + x_{23} - p^{\sim} \leq 1 \quad (19)$$

$$x_{12} - p^{\sim} \geq 0 \quad (20)$$

$$x_{23} \geq 1 \quad (21)$$

In (17) the equivalence operator has been transformed into a proposition with pure logic operators. Using DeMorgan’s Theorem the negation operator of the first term in brackets is moved inwards, so that we get CNF in (18). Finally, in (19) to (21) CNF has been translated into inequalities, which can be added to the integer programming formulation. A difficulty in this particular case is that the discount variable p^{\sim} has not been considered in the previous IP formulation. Therefore, we have to introduce an additional binary variable in our model, which indicates the discount if the rule takes effect. The size of the discount has to be considered in the product description in (3).

4 ECCO Applications

ECCO has been implemented in Java. It parses CPML documents, transforms them in the appropriate optimization model and solves the model using `lp_solve`, an open source linear programming package⁸. The

⁸ An LGPL version of this package can be downloaded at ftp://ftp.es.ele.tue.nl/pub/lp_solve

following section describes integration of ECCO in eHub a marketplace infrastructure based on WebService standards, as well as potential other application areas of the method.

4.1 eHub – A Sample Implementation

Transportation services are one example for configurable offerings. Shipping companies such as Fedex, UPS, Airborne, or DHL describe their shipping services along a variety of attributes. The weight of a certain package, together with a particular origin and destination specification determine the base price for a shipment. Then, a customer is provided with a variety of options, which determines the overall price of the shipment. Above all, the delivery time impacts the base price. Delivery time typically takes values such as “next day, 8am”, “next day, 3pm”, or “3rd day, 3pm”, which specifies until when a package will be delivered. Besides, customers have multiple options for pickup (after hours, on Saturday, on Sunday), for delivery confirmation (signature, adult signature), or return policies (1 pickup attempt - 3 pickup attempts). Also, there are optional fees for cash on delivery, or the transportation of hazardous materials. EHub is a tool developed at IBM Research to help suppliers aggregate information from multiple WebServices. We have implemented ECCO as a decision analysis tool in this context to help users evaluate the offerings of different logistics WebServices.

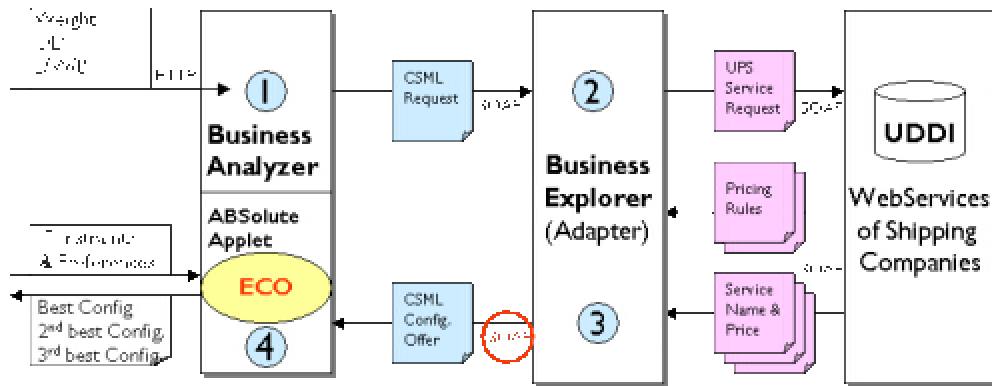


Figure 3: eHub Service Request and Architecture

Figure 3 provides an overall picture of an eHub service request. eHub consists of two basic components, Business Analyzer and Business Explorer. Business Analyzer provides a number of analysis features needed in a Web service environment. Business Explorer basically provides adapters for the specific Web services of different companies. This is necessary, since the responses of the various WebServices are mostly not standardized and use a different ontology to describe their services. So far eHub has been used as a demo application to gather the base prices of shipping services from participating companies. The pricing rules of the companies for the various additional options have not been considered, which makes a thorough comparison fairly difficult. We have extended the architecture and integrated ECCO. In a first step, a user enters the weight, the origin and destination (OD), and the dimensions of a product to trigger a service request in Business Analyzer (1). Business Analyzer uses an industry specific request template and sends a service request via the SOAP protocol to the Business Explorer WebService (2). The Web services of different companies are not standardized. Therefore, Business Explorer acts as an adapter, translates the service request into a company specific request, and converts the responses and company specific pricing rules into a CPML configurable offer (3).

ECCO has been implemented as a component of the ABSolute decision analysis framework. ABSolute provides a Java based framework of components for various decision support features, as well as advanced visualization (Bichler et al. 2001b). Upon initialization of the ABSolute applet, the ECCO component sends a SOAP request to the Business Explorer to dynamically download the CPML offers for a particular service request. ECCO parses the XML documents and reads the pricing rules into an internal object representation. When the applet is downloaded, users can specify their preferences for the various options. Based on this input, ECCO generates a list of the individual configurations, which best specify the user’s preferences. In a demo scenario with four offers, ECCO detected already more than eight thousand possible configurations. After the user specifies his preferences, ECCO returns a list of service configurations, which best satisfy her utility.

Response ID	Carrier	Utility	Price	Type of Transport	Arrival Date	Return	Picking Characteristics
1	FedEx	32	44	Air	Two Day, 8AM	None	Normal
2	FedEx	32	46	Air	Two Day, 8AM	None	Normal
3	FedEx	21	46	Air	Two Day, 8AM	None	Normal
4	FedEx	15	34	Air	Two Day, 8AM	Call Tag	Normal
5	FedEx	15	20	Air	Two Day, 8AM	None	Normal
6	FedEx	15	31	Air	Two Day, 8AM	Print and Mail	Normal
7	FedEx	15	26	Air	Two Day, 8AM	Call Tag	Normal
8	FedEx	14	38	Air	Two Day, 8AM	None	After Hours
9	FedEx	14	32	Air	Two Day, 8AM	Print and Mail	Normal
10	FedEx	11	25	Air	Two Day, 8AM	None	Normal
11	DHL	21	26	Air	Next Day, 3PM	None	Normal
12	DHL	38	39	Air	Next Day, 3PM	Call Tag	Normal
13	DHL	38	38	Air	Next Day, 3PM	None	Normal
14	DHL	38	38	Air	Next Day, 3PM	None	Normal
15	DHL	38	36	Air	Next Day, 3PM	Print and Mail	Normal
16	DHL	38	42	Air	Next Day, 3PM	Call Tag	Normal
17	DHL	38	42	Air	Next Day, 3PM	Call Tag	Normal
18	DHL	28	29	Air	Next Day, 3PM	None	Normal
19	DHL	38	41	Air	Next Day, 3PM	None	Normal
20	DHL	38	40	Air	Next Day, 3PM	None	Normal
21	Airborne	38	45	Ground	Two Day, 3PM	None	Normal
22	Airborne	38	44	Ground	Two Day, 3PM	None	Normal
23	Airborne	38	42	Ground	Two Day, 3PM	None	Normal
24	Airborne	28	40	Ground	Two Day, 3PM	Print and Mail	Normal
25	Airborne	38	45	Ground	Two Day, 3PM	None	Normal
26	Airborne	38	44	Ground	Two Day, 3PM	None	Normal
27	Airborne	38	43	Ground	Two Day, 3PM	None	Normal
28	Airborne	38	44	Ground	Two Day, 3PM	1 Pickup, 1 Parcel, 1 Parcel	Normal

Figure 4: Example of Result Set

4.2 Other Applications

Evaluation of configurable products can be used in a variety of applications in both the business-to-business and the business-to-consumer field. One such application, which is similar to the eHub example, is *catalog aggregation*. In business-to-business e-commerce, buying organizations are interested in getting product data about configurable products and services in a machine-readable format. They want to make information about the various products available in their ERP systems, and also compare the product descriptions of different suppliers. Pricing information is highly competitive information. So in certain environments vendors will restrict the catalog data to non-price information, where buyers are just informed about possible product configurations. In other environments vendors will be willing to provide complete offerings including the pricing rules. In the context of a catalog aggregation process, the buyer generates a query, which is sent to one or more catalog services. The catalog service automatically generates and sends back a CPML document with descriptions of configurable products, which adhere to the constraints in the catalog query. Based on the collected data, the user can specify additional information about preferences and constraints and select the best possible configuration using the ECCO decision analysis tool.

Another application of ECCO are *tenders* and *reverse auctions*. There have been many approaches to support negotiations on complex goods. Probably, the most wide-spread approach to automate these negotiations is tenders, where a buyer specifies a request for a certain product and suppliers respond with multi-attribute bids. A more structured approach is multi-attribute reverse auctions. Here, a buyer specifies his preferences for the various attributes of a product in form of a scoring function. The auctioneer publishes information about the buyer's preferences and suppliers compete on multiple attributes, not only price. Typically, conventional decision analysis techniques are used to determine the winner of these auctions. Although, multi-attribute auctions suggest one mechanism to support multilateral negotiations on complex goods with multiple attributes, they do not allow for negotiations on configurable products. ECCO can be used as a winner determination mechanism in reverse auctions on configurable products.

5 Discussion

Until now, most electronic commerce systems have focused on trading simple products and services. Only recently, researchers have focused on complex products and services. ViMP is a system to provide negotiation support for complex products such as insurances. It provides a variety of matching services,

which are implemented in a trader component. This component compares the stated requirements of a customer with the features of the products as well as the restrictions of the provider (Field and Hoffner 1998). Multi-attribute auctions have already been mentioned as a negotiation protocol to support negotiations on complex multi-attribute goods. Several negotiation support systems try to support multi-attribute bargaining situations (Kersten 1998). These approaches are limited to simple multi-attribute bids, and do not tackle the specific requirements of complex, configurable products.

A number of newer approaches try to capture more complex business rules in order to further automate electronic commerce. Various kinds of knowledge representation such as propositional logic and first-order logic are investigated for the exchange of offers and counter offers. For example, Iwaihara (2000) introduces so called dynamic constraints to model complex negotiation conditions, and a query language to select these constrained offerings from a database. Reeves et al. (2000) present an approach where offers in a negotiation are expressed in first-order logic. Although, this approach offers greater flexibility, the description of configurable offers becomes more difficult for businesses, and bid evaluation is unclear. In contrast, the goal of CPML was to provide a set of constructs which is on the one hand powerful enough to express real-world offerings, but also easy to understand for businesses. CPML uses functional specifications combined with logical propositions to specify and communicate the required pricing and configuration rules.

During the past couple of years, a number of companies such as i2, Selectica, Trilogy, Siebel and Ilog have developed online product configuration systems. These systems have a number of similarities to ECCO. They allow a user to configure a certain product with multiple options on the Web based on some pre-specified rules. They are also able to enumerate a list of possible configurations based on constraints specified by the user and rules specified by the seller. Commercial product configurators are typically based on rule-based systems or constraint programming solvers. The ECCO approach enhances commercial product configuration systems in two ways. First of all, in addition to the configuration rules and the user's hard constraints ECCO takes information about a user's preferences into account. In other words, it combines mathematical programming with traditional preference elicitation and decision analysis algorithms to rank and score the resulting configurations, and therefore makes the selection process much easier for the user. This is important, since the number of resulting configurations might still be huge. Second, the mathematical programming approach allows continuous attributes, whereas rule-based systems can only handle attributes with discrete values. This way, we can also deal with volume discounts specified by the suppliers. Finally, with CPML we specify a message format to communicate configurable offerings, and therefore enable applications such as catalog aggregation or reverse auctions on configurable products.

6 Acknowledgements

The authors would like to thank Juhnyoung Lee and Sudhir Verma for their many contributions to the project, in particular the user interface design and implementation. We would also like to thank Liang-Jie Zhang and Henry Chang for the integration of ECCO into eHub infrastructure.

7 References

- Bichler, M. (2001). *The Future of eMarkets: Multi-Dimensional Market Mechanisms*, Cambridge University Press, Cambridge, UK.
- Bichler, M., Kaukal, M., and Segev, A. "Multi-attribute auctions for electronic procurement." *First IBM IAC Workshop on Internet Based Negotiation Technologies*, Yorktown Heights, NY, USA.
- Bichler, M., Lee, J., Kim, C. H., and Lee, H. S. "Design and Implementation of an Intelligent Decision Analysis System for ESourcing." *International Conference on Artificial Intelligence 2001*, Las Vegas, NV.
- Bichler, M., Lee, J., Lee, H. S., and Chung, J.-Y. "ABSolute: An Intelligent Decision Making Framework for E-Sourcing." *3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, San Jose, CA.
- Clemen, R. T. (1996). *Making Hard Decisions: An Introduction to Decision Analysis*, Wadsworth Publishing Co., Belmont, CA.
- Edwards, W. (1977). "How to use multiattribute utility measurement for social decisionmaking." *IEEE Transactions on Systems, Man, and Cybernetics SMC*, 7(5), 326-340.

- Field, S., and Hoffner, Y. (1998). "VIMP - A Virtual Market Place for Insurance Products." *Journal of Electronic Markets*, 8(4).
- Hu, C. (1998). "The Basics of Conjoint Analysis."
- Iwaihara, M. "Supporting Dynamic Constraints for Commerce Negotiations." *Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS)*, Milpitas, CA, pp. 167-181.
- Iyengar, V. S., Lee, J., and Campbell, M. (2001). "Q-Eval: Evaluating Multiple Attribute Items Using Queries.", IBM Research, New York.
- Keeny, R. L., and Raiffa, H. (1993). *Decision Making with Multiple Objectives: Preferences and Value Tradeoffs*, Cambridge University Press, Cambridge, UK.
- Kersten, G. E. "Negotiation Support Systems and Negotiating Agents." *Modeles et Systemes Multi-Agents pour la Gestion de l'Environnement et des Territoires*, Clermont-Ferrand, France, 307-316.
- Olson, D. L. (1995). *Decision Aids for Selection Problems*, Springer, New York, et al.
- Raman, R., and Grossmann, I. E. (1991). "Relation between MILP modelling and logical inference for process synthesis." *Computers and chemical engineering*, 15(2), 73-84.
- Reeves, D., Grosz, B., and Wellman, M. "Automated Negotiations from Declarative Contract Descriptions." *AAAI-2000 Workshop on Knowledge-Based Electronic Markets*, Austin, TX.
- Ribeiro, R. A. (1996). "Fuzzy multiple attribute decision making: A review and new preference elicitation techniques." *Fuzzy Sets and Systems*, 78, 155-181.
- Russell, S. J., and Norvig, P. (1995). *Artificial Intelligence - A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, USA.
- Saaty, T. L. (1980). *The Analytic Hierarchy Process*, McGraw Hill, New York, USA.