A Cost–Benefit Evaluation Server for decision support in e-business

Youzhong Liu*, Fahong Yu, Stanley Y.W. Su, Herman Lam

Database Systems R&D Center, University of Florida, Gainesville, FL 32611, USA

Accepted 3 July 2002

Abstract

Business organizations are often faced with decision situations in which the costs and benefits of some competing business specifications such as business offers, product specifications, or negotiation proposals need to be evaluated in order to select the best or desirable ones. In e-business, there is a need to automate the cost–benefit evaluation process to support decision making. This paper presents a general-purpose Cost–Benefit Evaluation Server (CBES) and its underlying Cost–Benefit Decision Model (CBDM), which models benefits in terms of costs and logical scoring and aggregation of preferences associated with products and services. The Server provides build-time tools for users to specify preference and cost information and a run-time engine to perform cost–benefit evaluations. A business scenario involving supplier selection and automated negotiation is given to illustrate the application of the Server and its four evaluation modes.

Keywords: Cost–benefit evaluation; e-Business; Decision support

1. Introduction

With the advent of the Internet, collaborative e-business has been attracting more and more attention from both academia and industry [22,24]. Various technologies are being developed to support collaborative e-business, such as customer relationship management [27], supply chain management [19,20], eMarketplaces [1,5], and automated negotiation and auction systems [3,18,21,32].

In e-business, a business company or person often faces a decision situation in which a cost–benefit analysis (CBA) of a business specification needs to be performed before an appropriate decision can be made or a proper action can be taken. By “business specification”, we mean any business document that specifies terms and conditions of a business transaction, offer, or proposal. For example, in supplier selection, a buyer often needs to evaluate a number of supplier specifications that describe different suppliers’ capabilities in order to choose the best one from a list of candidate suppliers. In business negotiation, a company needs to evaluate a negotiation proposal of another company in order to determine the cost and benefit of an offer. In the decision to purchase, a company may receive a number of responses from different vendors after issuing a “request-for-quote”. They need to be evaluated to determine their costs and benefits.
A business specification may contain many terms and conditions. Their relative importance to an evaluator and their interrelationship will have to be taken into consideration in a cost–benefit evaluation. Also, a business specification may have different costs and benefits to different evaluators. Subjectivity is unavoidable in cost–benefit evaluation and, for that matter, in decision making. However, it is important to have a quantitative way to calculate the costs and to evaluate the benefits of the terms and conditions given in a specification so that an overall cost–benefit indicator (or value) can be assigned to it and be compared with the overall cost–benefit indicators of other competing specifications. A decision made based on the result of a quantitative evaluation and selection is traceable and justifiable because it can be shown how the best overall cost–benefit value is quantitatively derived based on the subjective opinion of the evaluator.

In order for cost–benefit evaluation to be useful in the context of e-business, we need (1) a structured way to present a business specification so that it can be more easily processed by computers, (2) a quantitative Cost–Benefit Decision Model (CBDM) to calculate costs and to capture the preferences of decision makers based on which cost–benefit evaluation of business specifications can be performed, (3) a Cost–Benefit Evaluation Server (CBES) capable of evaluating multiple business specifications concurrently, and (4) a scalable Internet-based information infrastructure to support collaborative e-business.

CBA is the comparative analysis of alternatives in terms of their costs and consequences [33]. It is one of the most common and popular evaluation techniques used to evaluate programs and products. Since Eckstein [12] first deployed CBA techniques for benefit estimations using market information, the scope of cost–benefit analysis has been extended to other application areas, such as health care-related economic evaluation [13,15,29,33], financial decision for business [2,7,19], government management and administration [4,8,9,14,28], and environmental damage assessment [6,17]. Researchers have introduced different models and approaches with different assumptions and values in the evaluation of costs and benefits to improve the service of CBA [4,10,13,30,33].

In these models, both costs and benefits are measured in monetary terms. Cost–benefit analysis is used to determine whether the benefit of a specification measured in dollars outweigh its cost and thus justify the allocation of resources to that specification. The cost–benefit ratio and the net benefit are commonly used as evaluation indices. The evaluation of business specifications in government management and administration model is based on the computer-supported process in which personal data records relating to many people are compared in order to identify cases of interest (or benefit) [4,14,28]. In regard to the valuation of benefit in some other models, the benefit is evaluated according to either what individuals would be willing to pay for the benefit or an individual’s value that is measured by the discounted value over time [4,8,33]. The net benefit is the summation of all the comprehensive benefits, including outcomes that are monetary, quantitative, qualitative [6], and/or the estimation of future monetary benefits, if applicable [7,29,33]. Obviously, these evaluation systems are designed for specific purposes and cannot be generically applied. Moreover, the evaluation is too simple to handle specifications with hierarchical structures. The evaluation service implemented in AVICOM [2] and Indent [19] can deal with complicated business specifications with multiple alternatives, but either the evaluation system is predefined or the evaluation is by simulation, thus lacking flexibility. Also, users cannot directly interact with the cost–benefit evaluation tools. The CBDM [30], which is based on the concept of logical scoring of preferences (LSPs), provides a comprehensive cost analysis and an elaborate analysis of benefits expressed in terms of the decision maker’s preferences. However, as a general model, it only allows a single value to be specified for each attribute in a business specification.

The business specifications that all the existing cost–benefit evaluation systems take as input can only have a single value for each of its attributes. None of them accepts a specification that contains an attribute having an enumeration of discrete values (ENUM type) or a range of values (RANGE type). Consequently, an evaluation system has to be invoked many times to evaluate all the business specifications one by one. For example, a company wants to buy a product and the purchase specification contains, among other attributes and values, the following attributes: {quantity={RANGE[800, 900]}; delivery_day = ENUM{7, 14, 21}}. In this case, the company would accept any
quantity between 800 and 900, and the number of days for the delivery can be either 1, 2, or 3 weeks. In a traditional cost–benefit evaluation system, a preprocessor has to be used to take all the value combinations and feed each combination as a different specification into the evaluation tool. This is obviously inefficient.

In this paper, we describe a general-purpose CBES, which is implemented based on an extended quantitative CBDM [30], to support e-business. The extended CBDM allows all types of business specifications to be defined in terms of a structure of attributes, each of which can have a range of continuous values or an enumeration of discrete values. Thus, a business specification may specify a number of value combinations, each of which represents an alternative business offer or proposal. These value combinations can be evaluated by CBES at the same time and in different modes of evaluation. CBES provides a set of build-time specification tools and a runtime evaluation engine. The build-time tools provide a set of Web-based interfaces to assist a decision maker to specify his/her preference scoring and aggregation methods, based on which the run-time evaluation engine performs cost and benefit (or preference) analysis on a given business specification. The run-time evaluation engine works in four different modes: EXHAUSTIVE, BEST, WORST, and APPROXIMATE. They have different time complexities in evaluation and can be applied in different decision situations.

The main differences between our CBDM and CBES and the existing models and systems are as follows. First, our model captures the subjective preferences of decision makers (i.e., individuals or business organizations) with respect to different attributes and values presented in a business specification, and our system provides the user-friendly GUI tools to capture the information. Second, our model and system allow more elaborate structures to be specified by decision makers for aggregating the preference scores (PS) assigned to attribute–value or attribute–value–range pairs to derive the global preference score (GPS). Third, our system supports a wide spectrum of preference aggregation functions to fit different decision situations. Fourth, the input business specification can contain attributes of RANGE and ENUM types instead of limiting each attribute to have a single value, and the evaluation system is able to calculate preference scores for both discrete values and value ranges. Last, but not the least, CBES offers several different modes of cost–benefit evaluation to meet the different users’ needs.

The R&D work on CBES is a part of a larger research effort in building an information infrastructure for supporting collaborative e-business. In Refs. [31,32], Su et al. present an information infrastructure to support Internet-based Scalable E-business Enterprises (ISEE). The ISEE infrastructure consists of a network of ISEE Hubs. Each ISEE Hub contains a number of servers. Each server provides a number of e-services to support collaborative e-business. Business companies register with these distributed ISEE Hubs and make use of their e-services to conduct e-business collaboratively. The Cost–Benefit Evaluation Server presented in this paper is one of the replicable servers of the ISEE infrastructure.

This paper is organized as follows. In Section 2, the Cost–Benefit Decision Model is presented. In Section 3, the four run-time evaluation modes and the algorithms for implementing them are presented. The implementation of CBES is discussed in Section 4. In Section 5, an e-business scenario involving supplier selection and automated negotiation is used to illustrate the application of CBES and its evaluation modes. Finally, the key features of CBES are summarized in Section 6.

2. Cost–Benefit Decision Model

Su et al. [30] introduced a Cost–Benefit Decision Model to evaluate, compare, and select alternative DBMS products, each of which has a multiplicity of features. The benefit analysis part of the model is based on the “logical scoring of preference” method introduced by Dujmovic in Ref. [11]. The CBDM presented here is an extension of our earlier work by allowing attributes in a business specification to have RANGE and ENUM types of attributes. In this model, a business specification is represented by a hierarchical structure of attributes and values, which is called a parameter tree (see Fig. 1). The attributes and the structure of a parameter tree form a “template”, which is commonly accepted and understood by business companies that conduct a particular type of business.

In a parameter tree, some attributes are required and others are optional. Fig. 1 illustrates a parameter
tree, which represents a product specification given by a computer vendor. Among other attributes of a computer, which are not shown in the figure, it contains price, delivery_day, quantity, and Service. Service is a substructure containing its own attributes: service_level (morning-only, daytime-only, or 24-h service) and period (3 or 5 years). It is optional, meaning a buyer does not have to take the service contract. (Note: The values associated with the attributes are not shown in the figure.)

In a cost–benefit evaluation, the attributes of a parameter tree are separated into a preference tree and a cost tree, as shown in Fig. 2. The cost tree is a subtree of the parameter tree. It contains those attributes and values that have additional costs associated with them. These additional costs are to be added to the base cost, which is given as price in this particular example. For example, assume that the base cost for a computer is US$999. The cost tree in this example contains the root node and the substructure Service. If a service contract is requested and the service_level requested is 24-h service and the service period is 3 years, an additional cost of US$199 is added to the base cost to derive an aggregated cost (AC).

The preference tree is also a subtree of the parameter tree. It contains those attributes and values given in the parameter tree to which a decision maker has predefined a set of “elementary criteria” to express the percentages of satisfaction in the range of [0, 100] that he/she has with some possible values or value ranges for these attributes. The cost tree and the preference tree may have some overlapping attributes and values. These attributes and values are the ones with both preference scores and cost values associated with them. Based on the preference information provided by the decision maker, CBES assigns preference scores to the attribute–value or attribute–value–range pairs given in the preference tree. These preference scores are then aggregated based on an aggregation tree prespecified by the decision maker to derive a global preference score. The aggregated cost and the global preference score are then used in a
cost–benefit analysis to obtain an overall cost–benefit indicator for the input business specification. We shall explain the different types of elementary criteria in Section 2.1, the aggregation structure in Section 2.2, and the cost–benefit analysis in Section 2.3.

2.1. Elementary criteria

In order for CBES to evaluate an input business specification automatically, the evaluation has to be based on some cost information and decision maker’s preferences related to the specified product, proposal, or any other type of business specification given as the input. We assume that, for each type of business specification (e.g., a negotiation proposal for a specific product, a capability specification of manufacturers of a specific product, etc.), a template has been defined and commonly agreed by the issuers and receivers of that type of business specifications. Each template contains a predefined set of attributes having some meaningful values. The meanings of these attributes and values are commonly understood by issuers and receivers (i.e., a common ontology). The cost information associated with each template and its possible attributes and values may be given in a business specification by its issuer and/or can be accessed from some information source (e.g., a database) accessible to the receiver. The preference information associated with each type of business specification needs to be provided by the decision maker, who may receive a specification of that type. In CBDM, the preferences of a decision maker are defined in terms of a set of elementary criteria. An elementary criterion is a mapping from an attribute–value or attribute–value–range pair to an integer in the range \([0, 100]\). The integer indicates the percentage of satisfaction the decision maker has with a specific attribute–value or attribute–range pair. Some examples to illustrate different ways of specifying elementary criteria are given below. For example, if \(service\_level\) is an attribute of the ENUM type in the template of a product specification, say, a computer, the decision maker may assign a preference score of 60 if the value of \(service\_level\) is “morning-only”, 80 if it is “daytime-only”, and 100 if there is a “24-h” service. The above example shows three elementary criteria defined for three discrete values of \(service\_level\). For some other attribute of the RANGE type, such as \(delivery\_day\), a decision maker may specify 100 if the number of day to deliver the product is 1, 80 if the number of days is in the range of \([2, 5]\), 70 for the range \([6, 10]\), and 0 if the number is greater than 10. The general form for this type of specification follows.

\[
\text{ENUM}\{\text{RANGE}[v_1, v_2], \ldots, \text{RANGE}[v_n, v_{n+1}]\}
\]

or, in the above example, \(\text{ENUM}\{\text{RANGE}[1, 1], \text{RANGE}[2, 5], \text{RANGE}[6,10], \text{RANGE}[11, \text{MAX}]\}\). Here, \(\text{MAX}\) represents the maximum value of an attribute. If the preference score is a constant, every value in the range or enumeration shares the same preference score. For simplicity, if the lower bound and the upper bound of a RANGE specification are the same, we allow the RANGE specification to be simplified as a single value. For example, RANGE\([1, 1]\) can be represented as 1. If an ENUM has only one element, the keyword ENUM and the brackets can be deleted. For example, \(\text{ENUM}\{\text{RANGE}[1200, 1300]\}\) can be simplified as \(\text{RANGE}[1200, 1300]\).

Another way to define an elementary criterion is by using a function. For example, \(\text{deductible}\) is an attribute in an insurance contract specification. The degrees of satisfaction of an insurance buyer can be specified by a function such as \(E(\text{deductible}) = \text{Integer}((1 - \text{deductible}/1000)\times100)\). If the possible values of \(\text{deductible}\) are in the range \([0, 1000]\). The above function says that the buyer is fully satisfied with 0 \(\text{deductible}\) and completely unsatisfied if the deductible is US$1000. The preference scores for other deductible values are in between 0 and 100 calculated by the function.

2.2. Aggregation structure

The aggregation structure of the preference tree is also predefined by a decision maker based on the template of a business specification type. It is used to aggregate the preference scores, which are assigned to the attributes and values given in a business specification by CBES at run-time based on the elementary criteria provided by the decision maker. The aggregation structure specifies how different subsets of the related attributes should be aggregated, what weights should be assigned to attributes to indicate their relative importance, and what aggregation functions should be applied for these substructures, respectively. It allows complex logical relationships and relative
weights of preference attributes to be unambiguously expressed and applied to produce the global preference score for the entire tree. Fig. 3 shows an example aggregation structure of some attributes associated with a computer product specification. The service-level and period, quantity, and delivery-day are aggregated separately. The two aggregated preferences are then aggregated again and the result is then aggregated with price.

The elementary preference scores (EPS) assigned to attribute–value or attribute–value–range pairs (not shown in the figure) are weighted based on the weights assigned to the links as shown in the figure. For example, the weights assigned to the preference scores of service-level and period are 50% and 50%, respectively. The scores and weights are used by an aggregation function (Maximum) to produce an aggregated preference score, which is then weighted by the weight shown in the next level. A spectrum of “preference aggregation functions” can be used to generate aggregated values for the substructures at different levels. These functions are derived from a weighted power mean shown below as Eq. (1) [11] by varying the value of \( r \), where \( e_i \) are the EPSs and \( w_i \) are the weights.

\[
E = \left( w_1 e_1^r + w_2 e_2^r + \ldots + w_n e_n^r \right)^{1/r}
\]

By varying the value of \( r \), a spectrum of aggregation functions can be generated, including functions such as min, max, weighted arithmetic mean, weighted harmonic mean, weighted geometric mean, weighted square mean, etc. Some commonly used functions are given in Table 1.

<table>
<thead>
<tr>
<th>Aggregate function spectrum</th>
<th>( E = \min(e_1, e_2, \ldots, e_n) )</th>
<th>( r = -\infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted harmonic mean</td>
<td>( E = \frac{1}{w_1 e_1 + w_2 e_2 + \ldots + w_n e_n} )</td>
<td>( r = -1 )</td>
</tr>
<tr>
<td>Weighted geometric mean</td>
<td>( E = (e_1)^{w_1} (e_2)^{w_2} \ldots )</td>
<td>( r = 0 )</td>
</tr>
<tr>
<td>Weighted arithmetic mean</td>
<td>( E = w_1 e_1 + w_2 e_2 + \ldots + w_n e_n )</td>
<td>( r = 1 )</td>
</tr>
<tr>
<td>Weighted square mean</td>
<td>( E = \sqrt{w_1 e_1^2 + w_2 e_2^2 + \ldots} )</td>
<td>( r = 2 )</td>
</tr>
<tr>
<td>Maximum</td>
<td>( E = \max(e_1, e_2, \ldots, e_n) )</td>
<td>( r = +\infty )</td>
</tr>
</tbody>
</table>

These alternative aggregation functions represent different degrees of conjunction and disjunction of data conditions. They can be selected by the decision maker to suit different decision situations. For example, let us assume that CPU speed is one of several attributes given in a specification and the aggregate function, Maximum, is used. If the preference score assigned for a value of this attribute is 90, then the global preference score is 90 even if the preference scores of the other attributes and values are below 90. At the other end of the spectrum, the aggregation function, Minimum, will take the minimum score among all the preference scores as the global preference. In the above example, if the aggregation function, Minimum, is used and a client is only 10% satisfied with the speed of the CPU, the

![Fig. 3. The aggregation structure of computer.](image-url)
global score would be 10 even though he/she may be totally satisfied with all the other attributes and their values.

2.3. Cost–benefit analysis

Cost–benefit analysis is the process that integrates the global preference score and the aggregated cost to get an overall evaluation indicator on the input specification. We use the ratio of GPS and AC as the evaluation indicator, subject to some threshold constraints. A preference threshold (\(E_0^*\)), a cost threshold (\(C_0^*\)), and a preference cost ratio threshold (\(R_0^*\)) are specified by the decision maker. As shown in Fig. 4, the Region Of Acceptable Solution (ROAS) is an area \(a\), such that for any point \(p \in a\), we have \(E_p \geq E_0^*\), \(C_p \leq C_0^*\) and \((E_p/C_p) \geq R_0^*\), where \(E_p\) and \(C_p\) are the preference score and the cost value for \(p\), respectively. If \(E_0^*\), \(C_0^*\) or \(R_0^*\) is not specified, no threshold is assumed for the corresponding parameter. If the preference and cost ratio, GPS/AC, is a point in ROAS, the value is returned as the final cost–benefit indicator. Otherwise, a zero value is returned.

3. Cost–Benefit Evaluation Server

The input to CBES is a business specification, which is a specification of a business object being evaluated. RANGE and ENUM can be used in the specification to allow enumerated values or ranges of values to be associated with some attributes. In effect, a specification can have a number of alternative value combinations, each of which is a set of attribute–value or attribute–value–range pairs. These value combinations can be evaluated by CBES together following the steps shown in Algorithm 1. The evaluation result is a set of value combinations and their corresponding cost–benefit indicators. CBES supports four modes of evaluations, which are explained in Sections 3.1–3.4, respectively.

Algorithm 1 (Cost–Benefit Evaluation).

1. Calculates the Preference Score of Attribute (PSA) and the Additional Cost of Attribute (ACA) for each attribute by evaluating the input attribute values against the elementary criteria prespecified by the decision maker and by accessing cost information.
2. Analyzes the aggregation structure that corresponds to the preference tree and the cost tree to get the global preference score and the aggregated cost.
3. Applies cost-preference analysis to get a final cost-preference indicator.

3.1. EXHAUSTIVE mode

In the EXHAUSTIVE mode, CBES evaluates the value combinations of the input attributes and values
by following the steps of Algorithm 1. For example, the specification of a product (e.g., computers) may contain the following attributes and value specifications:

- Price = RANGE(1200, 1300], base type: int;
- Delivery_day = ENUM{6, 9, 12}, base type: int;
- Quantity = RANGE(800, 1000], base type: int;

Here, parentheses are used to represent exclusive ranges, and square brackets are used for inclusive ranges. Price = RANGE(1200, 1300] means that all prices in the range greater than 1200 but less or equal to 1300.

The above specification has 100*3*200 = 60,000 value combinations. In a traditional cost–benefit analysis, these combinations will have to be evaluated one by one. In our case, we can take advantage of the range and enumeration specifications given in the elementary criteria to reduce the number of combinations. For example, if the elementary criteria for attribute price specify the preferences scores for the following two ranges: RANGE(1200, 1250] and RANGE(1250, 1300], then the input specification of RANGE(1200,1300] can be partitioned into these two ranges instead of 100 different values. Thus, the total number of combinations is reduced to 2*3*200 = 1200. The same thing can be done for the range specification of quantity to further reduce the number of combinations.

For each combination, CBES analyzes the aggregation and cost structures and derives a final cost–benefit indicator. The resulting set of value-combination and indicator pairs can be ordered by the numeric values of the indicators and returned to the requester.

Before discussing the time complexity of this mode of evaluation, we define the cardinality of an attribute a first. The input specification is compared with the elementary criteria (including both preference and cost specifications) to identify the number of RANGE and ENUM specifications for attribute a. We call this number the cardinality of attribute a. For example, in the above example, the input price is RANGE(1200, 1300]; the preference specification involves RANGE(1200, 1250] and RANGE(1250, 1300]. If the cost specification involves RANGE(1200, 1220], RANGE(1220, 1260], and RANGE(1260, 1300], then by combining all the ranges together, we have RANGE(1200, 1220], RANGE(1220, 1250], RANGE(1250, 1260], and RANGE(1260, 1300]. The cardinality is 4. The time complexity to get the cardinality is linear to the cardinality given the ranges are sorted.

The time complexity of the evaluation is $O((\prod_{i=1}^{n} S_i)(n + N_a))$, where $n$ is the number of attributes, $S_i$ is the cardinality of the $i$th attribute; $N_a$ is the number of aggregation functions in the aggregation structure. This is because the evaluation process has to be repeated $\prod_{i=1}^{n} S_i$ times and, in each time, all the attribute–value or attribute–value–range pairs and aggregation functions in the aggregation structure have to be evaluated.

This mode is useful for evaluating, for example, a negotiation proposal in which alternative acceptable values for some attributes are specified and the evaluator of the proposal is interested in ranking all the possible value combinations based on their global cost–benefit indicators.

3.2. BEST mode

Unlike the EXHAUSTIVE mode in which all the value combinations of the input specification are evaluated, the BEST mode evaluated only the best combination. Only the attribute value or value range that has the highest preference score is selected for evaluation. For example, if the attribute delivery_day is of type ENUM{3, 5, 7, 9} and “delivery_day equal to 3” has the highest preference score (e.g., 80), based on the predefined elementary criteria for this attribute, CBES will select 3 as the value of this attribute and drop the others. The preference score 80 is assigned to this attribute. If multiple attribute–value or attribute–value–range pairs have the same highest reference score, they will all be selected.

We prove below that the value combination or combinations with the highest global preference score will be returned as a result.

Lemma 1. The higher the $e$ is, the higher the preference score $E$ for a specification $p$, where $e$ is the preference score for an attribute $a$ in $p$ under evaluation, $E$ is the aggregated preference score of the business specification being evaluated.
Proof. Without the loss of generality, we consider the contribution of \( e_1 \) to \( E \), where \( e_1 \) is the preference score of the first attribute. We assume that the other elements in Eq. (1) shown in Section 2.2 are fixed. Eq. (1) can be rewritten as \( E = (w_1 e_1 + C)^{1/r} \), where \( C = (w_2 e_2^r + \ldots + w_n e_n^r) \) is a constant. We shall compare \( E_a \) and \( E_b \) in three cases, \( r > 0 \), \( r < 0 \), and \( r = 0 \), where \( E_a \) and \( E_b \) are the aggregated preference scores derived from two preference scores \( e_{1a} \) and \( e_{1b} \) assigned to two different values of the first attribute, and \( e_{1a} > e_{1b} \).

If \( r > 0 \), \( e^r \) and \( e^{1/r} \) increases monotonically for different values of \( r \). If \( e > 0 \), we have \( E_a > E_b \) because

\[
E_a = (w_1 e_{1a} + C)^{1/r} > (w_1 e_{1b} + C)^{1/r} = E_b
\]

If \( r < 0 \), \( e^r \) and \( e^{1/r} \) decreases monotonically for different values of \( r \). If \( e > 0 \), we have \( E_a > E_b \) because

\[
E_a = (w_1 e_{1a} + C)^{1/r} < (w_1 e_{1b} + C)^{1/r} = E_b
\]

If \( r = 0 \), as shown in Ref. [30], the weighted power mean shown in Eq. (1) can be rewritten as

\[
(w_1 e_{1a})^{w_1} (w_2 e_{1b})^{w_2} \ldots (w_n e_{1b})^{w_n}
\]

Since \( (e_{1a})^{w_1} > (e_{1b})^{w_1} \), we still have \( (e_{1a} + C)^{1/r} > (e_{1b} + C)^{1/r} \), i.e., \( E_a > E_b \).

Lemma 2. If the preference score of every attribute–value pair of value combination 1 is equal to or greater than that of value combination 2, the aggregated preference score of value combination 1 is equal to or greater than that of 2.

Proof. By applying Lemma 1 to each and every attribute, it is obvious that the aggregated preference score of value combination 1 is greater than or equal to that of value combination 2.

Theorem 1. The value combination in a business specification that has the highest aggregated preference score is the one that has the highest preference score for each and every one of its attribute–value or attribute–value–range pair.

Proof. Let us consider all the value combinations of a specification. By using Lemma 2, we know that the aggregated preference score of the combination with the highest preference score for each of its attribute–value or attribute–value–range pairs is equal to or greater than the aggregated preference scores of all the other value combinations.

Based on Theorem 1, Algorithm 2 shown below is used for the preference evaluation in the BEST mode.

Algorithm 2 (BEST Mode Preference Evaluation).

1. The input values of each attribute are evaluated against the elementary criteria prespecified by the decision maker to determine their preference scores. For each attribute, the attribute–value or attribute–value–range pair with the highest preference score is selected. This process will produce a set of attribute–value or attribute–value–range pairs with their highest preference scores. The generated set is the best value combination.

2. Analyze the aggregation structure based on the identified value combination to derive its global preference score.

3. Return the global preference score and the best value combination.

We note here that it is possible that more than one value combination can have the same global preference score. In that case, they will be identified and returned.

The time complexity of this algorithm is \( O(\sum_{i=1}^{n} T_i + N_a) \), where \( T_i \) is the number of alternative values or value ranges associated with attribute \( i \) given in the input specification. \( N_a \) is the number of the aggregation functions specified in the preference structure. In Step 1 of the algorithm, each enumerated value or value range given in the input specification has to be evaluated. In Step 2, the aggregation structure has to be traversed.

Comparing with the time complexity of the EXHAUSTIVE mode, it is easy to see that this algorithm is much more efficient. The BEST mode is useful, for example, for selecting the best choice out of the alternative choices given in a product specification. It should be noted that this mode of evaluation does not take the cost of a value combination into
consideration. The best choice selected may not have the best cost–benefit indicator if the cost is considered. If the cost is to be considered in this mode of evaluation, one can treat the cost as one of the preference attributes in the evaluation.

3.3. WORST mode

The WORST mode is the converse of the BEST mode. It identifies the worst value combination given in an input business specification. The proof and the algorithm for this mode are similar to the BEST mode, and its time complexity is the same as the BEST mode. This mode is useful for eliminating the worst offer.

3.4. APPROXIMATE mode

In this mode, if an attribute of the input business specification is of RANGE or ENUM type, CBES will first evaluate the preference score for each value or value range based on the predefined elementary criteria. Then, the preference scores for all the values or value ranges of an attribute are aggregated, for example, by taking the average of these scores (or other aggregation method). The aggregated value is used as the preference score of that attribute. If an attribute is of type RANGE and the elementary criterion specified by a decision maker is a function in the range \([a, b]\), integration will be used to calculate the preference score for that range. The preference score for that range is \(\frac{\int_a^b f(x)dx}{b-a}\), where \(f(x)\) is the function given as the elementary criteria for the range \([a, b]\). The time complexity of this mode is the same as the best mode since the only work is to evaluate each enumerated value or value range in the input specification and traverse the aggregation structure.

This mode of evaluation is useful, for example, to a buyer who received a large number of offers from different suppliers. The buyer wants to have a rough estimate of the preference score for each offer and select some of them for more detailed evaluations.

4. CBES implementation

The CBES has been implemented using Java, Java Swing, and Java Applet/Servlet. CBES consists of a set of build-time tools and a runtime evaluation engine (see Fig. 5). The build-time tools provide a set of Web-based interfaces to assist a decision maker to specify his/her/its preferences and cost information (i.e., knowledge setup), which are used for the run-time evaluation, comparison, and selection of business alternatives. The run-time engine evaluates an input specification against the information specified at build-time. Several graphical user interfaces (GUIs) embedded in a TabPane can be used by a decision maker to provide the preference specification, the cost specification, the preference

![Fig. 5. Components of CBES.](image-url)
aggregation, and other parameters, such as the thresholds for cost–benefit analysis, and the evaluation mode.

4.1. Preference specification

As we stated before, for each type of business specification (e.g., a negotiation proposal for a specific product, a capability specification of a manufacturer of a specific product, etc.), a template containing a structure of attributes can be defined and is known to both issuers and receivers of a business specification. Preference specification in CBES is a build-time process, in which a decision maker uses a GUI tool to define a set of elementary criteria to specify the percentages of satisfaction for some possible values of the attributes given in a template. Fig. 6 shows the user interface for preference specification. A user first selects a template from a set of predefined templates. He/she then specifies some values and their corresponding preference scores for each attribute of the template.

It is unrealistic to require a user to specify preference scores of all possible values of an attribute. To improve the usability of CBES, the Preference Specification Tool is able to use interpolation to derive an analytic expression from those values (or value ranges) and their corresponding preference scores given by the user. The interpolation is implemented using the Newtonian algorithm.

The existing interpolation algorithm [25] can be applied directly if the value of each attribute is a single value. However, we need to tailor the existing algorithms to perform interpolation on elementary criteria with RANGE specifications. A trivial solution is to perform interpolation with all the points in a value range. However, this may lead to a high-degree polynomial, which will be too complicated for interpolation and polynomial evaluation. In our implementation, we use the lower bound and the upper bound of a range to represent the range. For instance, if the user assigns a preference score 40 to the range [2, 4] of delivery_day, the pairs (2, 40) and (4, 40) are used for the interpolation. The interpolation is performed based on the information provided by R1_a, R1_b, R2, R3, R4_a, and R4_b. The interpolation result is shown as the dotted line in the upper part of Fig. 7. The adjusted result is shown in the bottom part of Fig. 7, which accurately represents the fact that all the values in the range R1 have the same preference score, as well as those of R4.

Fig. 6. Preference specification.
In order to give the user a visual view of the result of an interpolation, the result is displayed in a separate applet window. The user can accept or reject the interpolation result.

We note here that interpolation is applicable only to attributes whose values are continuous but not to attributes of the String type with discrete values.

4.2. Cost specification

Cost specification is also a build-time process, in which the user through a GUI tool identifies those attributes and values of a selected template that have costs associated with them and assigns costs to them. Usually, a business specification (e.g., a product specification) has a base cost and the values of some of its attributes may require some additional costs. For example, the base cost of a PC may be US$999 and the standard memory configuration for the PC is 64 M. If memory size is 96 or 128 M, an additional US$20 or US$40 has to be added to the base cost, respectively. Similar to preference specification, the additional cost could be a function of an attribute value. Interpolation can also be used to derive the additional cost trend for the unspecified attribute values.

4.3. Preference aggregation

CBES allows a user to construct his/her own aggregation structure for the template he/she selected. The GUI for preference aggregation is shown in Fig. 8. It allows the user to select a subset of related attributes given in the template, assign relative weights to these attributes, and select the proper...
aggregation function to form a substructure of the aggregation structure. The weights are normalized to make the total weight of each aggregation substructure to be 100. This process is repeated on the remaining attributes until the aggregation structure is constructed.

To improve the usability, a “Slider” is provided as the selection tool for a user to choose the desired aggregation function from a spectrum of aggregation functions. Thus, the user does not have to know or understand the mathematics of these functions. The selected scale is used to set the \( r \) value of the weighted power mean (Eq. (1)) to determine the proper aggregation function. The “Arithmetic Mean” function is set as the default. At the end of the aggregation specification process, the resulting aggregation structure is displayed in a hierarchical manner, which provides a convenient way for the user to view the constructed aggregation structure.

4.4. Other evaluation parameters

Another GUI is used by the user to specify the additional parameters needed for cost–benefit evaluation: the base cost, the preference score threshold, the cost value threshold, and the preference/cost ratio threshold. These values are used for the cost–benefit analysis as explained in Section 2.3.

4.5. Runtime evaluation

The CBES’s run-time engine is an RMI server, which takes requests from the clients and returns the evaluation results to them. Three parameters are required in all the four evaluation modes: the input business specification, the evaluation mode selected by a client, and the client’s user ID. The user ID is used to retrieve the preference and cost information provided by the user at the build-time.

This engine can be invoked by application systems or by users through interactive interfaces. In the first case, the calling applications run as RMI clients to interact with the run-time engine. In the latter case, to improve the flexibility and usability, a tool is provided by CBES to allow the user to define a business specification on the Web, select the CBES mode, and evaluate the specification through a Java servlet. Once the specification is defined and submitted, the servlet acts as an RMI client and communicates with CBES’s evaluation engine to get the evaluation result and return it to the client.

5. An application scenario

In this section, we shall use a typical business scenario to show how the different modes of CBES can be used to support decision making in e-business. The scenario is that Company A is interested in purchasing a number of laptops. It sends out a request-for-quote to 100 companies, which conduct business over the Internet. Assume that 50 companies responded to the request and return their quotes. Company A would use a supplier selection server to select a small number of suppliers that have made reasonable offers. Assume five suppliers are selected for further business contact and negotiation. Both the supplier selection and the follow-up negotiations can make use of the services of CBES.

5.1. Supplier selection

In this scenario, Company A can use the BEST mode to evaluate the offer of each of the 50 responding suppliers to find the best value combination in that offer and its final cost–benefit indicator. By comparing the final indicators of the 50 offers, the top five suppliers can be selected. Alternatively, Company A can use the APPROXIMATE mode to get a rough idea about the relative “goodness” of the 50 offers. Different from the BEST mode, all values, not just the best value, of each attribute in a specification are considered. In the following, we consider the use of the APPROXIMATE mode for supplier selection.

Suppose Suppliers S1 and S2 are among the ones that responded to the request-for-quote and have the following attributes and values in their quotes:

```
Supplier S1: {\{price=\text{RANGE}[1600, 1800], delivery\textunderscore day=\text{ENUM}\{7, 9, \text{RANGE}[13, 15], 17\}\}}
Supplier S2: {\{price=\text{RANGE}[1500, 1900], delivery\textunderscore day=\text{ENUM}\{14, 21\}\}}
```

CBES would use Company A’s preregistered preference information (Table 2) to determine the prefer-
ence scores for the above values and value ranges and derive the global preference scores as follows:

Supplier S1: \(\text{PS} (\text{price}) = 95\), \(\text{PS} (\text{delivery}_\text{day}) = \text{AVERAGE}(90, 80, 70, 60) = 75\).
\(\text{GPS} = 95 \times 0.6 + 75 \times 0.4 = 93\).

Supplier S2: \(\text{PS} (\text{price}) = (300 \times 95 + 100 \times 80)/400 = 91.25\), and \(\text{PS} (\text{delivery}_\text{day}) = \text{AVERAGE}(70, 60) = 65\).
\(\text{GPS} = 91.25 \times 0.6 + 65 \times 0.4 = 80.75\).

In the above calculation, we assume that the weights assigned to the two attributes are 60% and 40%, respectively. The same calculation can be done for the other 48 suppliers.

### 5.2. Automated negotiation

Automated negotiation is a value-added e-service of the ISEE infrastructure. As documented in Refs. [16,18,32], a Negotiation Server is replicated and installed in multiple ISEE Hubs. A pair of Negotiation Servers would conduct an automated bilateral negotiation on behalf of two companies. In the automated negotiation process, negotiation proposals and counterproposals are transmitted between two servers either until both servers come to an agreement or one of the servers terminates the process unilaterally. A negotiation proposal or counterproposal is a business specification, which may contain alternative values and value ranges that are proposed by a server on behalf of a negotiation party. They are evaluated against the acceptable values or value ranges of the other negotiation party that receives the proposal or counterproposal. The acceptable values and value ranges of both parties are specified in terms of constraints and preregistered with their respective Negotiation Servers. The evaluation of a proposal against the preregistered constraints is a constraint satisfaction processing problem and is handled by a Constraint Satisfaction Processor (CSP) [18], which is a component of the Negotiation Server. In the evaluation, if a constraint violation is found, a rule-based concession scheme is used to automatically change some value in the received proposal and send the modified proposal as a counterproposal to the other party.

In our scenario, assume that the five best suppliers have been selected in the supplier selection process. A negotiation process would take place with each of the five suppliers. Each negotiation proposal that is passed between Company A’s Negotiation Server and the Negotiation Server of each supplier may contain multiple value combinations which specify alternative offers. After the rule-based concession process takes place, there may still be multiple value combinations that satisfy the constraints of the receiver. In this case, CBES can be used to evaluate them and select the best offer to accept. The BEST and EXHAUSTIVE modes are suitable for this purpose. If there are no additional costs introduced by different attributes and values, the BEST mode can be used to select the best offer efficiently; however, if additional costs are involved, the EXHAUSTIVE mode can be used to evaluate and rank all the offers and select the one with the best cost–benefit indicator.

We consider the BEST mode below:

If the following proposal is sent by a Negotiation Server to CBES for evaluation: \{\text{price}=[1730, 1850], \text{delivery}_\text{day}=[7, 12]\}. Based on the elementary score for each parameter given by the user (Table 2), the best preference score is obtained when price is in the range \([1730, 1800]\) and \text{delivery}_\text{day} = 7.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value range</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>[2000, MAX)</td>
<td>0</td>
</tr>
<tr>
<td>Price</td>
<td>[1900, 2000)</td>
<td>60</td>
</tr>
<tr>
<td>Price</td>
<td>[1800, 1900)</td>
<td>80</td>
</tr>
<tr>
<td>Price</td>
<td>(MIN, 1800)</td>
<td>95</td>
</tr>
<tr>
<td>\text{delivery}_\text{day}</td>
<td>(14, MAX)</td>
<td>60</td>
</tr>
<tr>
<td>\text{delivery}_\text{day}</td>
<td>(10, 14]</td>
<td>70</td>
</tr>
<tr>
<td>\text{delivery}_\text{day}</td>
<td>(7, 10]</td>
<td>80</td>
</tr>
<tr>
<td>\text{delivery}_\text{day}</td>
<td>(MIN, 7]</td>
<td>90</td>
</tr>
</tbody>
</table>
If the EXHAUSTIVE mode is used for the evaluation, we will still get the same result since there is no additional cost specified for the given attributes.

6. Summary

The design and implementation of a general-purpose Cost–Benefit Evaluation Server for supporting decision making in e-business has been presented. We have extended a Cost–Benefit Decision Model used in our earlier work to allow attributes of a business specification to contain enumerated values or value ranges. It also allows the user to assign preference scores to values and value ranges. Thus, CBES can efficiently evaluate a business specification that contains a number of value combinations. CBES provides a number of build-time GUI tools for capturing preference and cost information from different decision makers and a run-time engine to evaluate business specifications based on the subjective preference scoring methods and aggregation functions selected by different decision makers. In the ISEE infrastructure, multiple CBESs can be installed in multiple ISEE Hubs. Each business company can register and maintain its preference and cost information in its own CBES, thus keeping the information private. CBES has the following features. First, the extended Cost–Benefit Decision Model captures the subjective preferences of decision makers with respect to different attributes and values presented in business specifications. Second, the model allows very elaborate aggregation structures to be specified by decision makers for aggregating the preference scores assigned to individual attribute–value or attribute–value–range pairs. Third, the system supports a wide spectrum of preference aggregation functions to fit different decision situations. Fourth, the input business specification can contain attributes of RANGE and ENUM types instead of limiting each attribute to have a single value, and the evaluation system is able to calculate preference scores for both discrete values and value ranges and to derive evaluation functions by interpolation. Last, but not least, CBES offers several modes of operations to meet the different cost–benefit evaluation needs of the users.

In this work, we assume that templates can be predefined for different types of business specifications, thus standardizing the ontology used in business communication and evaluation. We believe that this is a reasonable assumption because several groups in the industry, such as the Open Application Group [23] and RosettaNet [26], have been developing standard terms and structures for specifying business object documents.

Acknowledgements

This project is supported by the National Science Foundation under grant no. EIA-0075284.

References


Dr. Youzhong Liu is a Software Architect in Verizon Communications. He received his PhD degree in Electrical and Computer Engineering from the University of Florida in 2001. His research interests include business rule management, knowledge sharing and management, business intelligence, and legacy system integration.

Dr. Stanley Y.W. Su received his MSc and PhD degrees in Computer Science from the University of Wisconsin, Madison in 1965 and 1968, respectively. He is a Distinguished Professor of the Department of Computer and Information Science and Engineering, and the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL. He is the Director of the Database Systems Research and Development Center. He served as an editor of IEEE’s Transactions on Software Engineering (1981–1988), an area editor of the Journal of Parallel and Distributed Computing (1983–1989), an editor of IEEE’s Transactions on Knowledge and Data Engineering journal (1989–1995), a member of the Board of Trustees (1990–1999) and Treasurer of the Executive Committee of the Very Large Data Base Endowment (1991–1999), and an Editor-in-Chief of the International Journal on Very Large Data Bases (1993–2001). He currently serves as an editor of the International Journal on Computer Languages, an editor of the Information Sciences journal, and an editor of the Asian Journal of Business and Information Systems. He is an IEEE Fellow for his contributions to parallel database systems and knowledge base management systems in support of integrated manufacturing.

Dr. Fahong Yu received his MSc degree in Computer and Information Science and Engineering and his PhD degree in Zoology from the University of Florida in 2001 and 2002, respectively. He specializes in cost–benefit evaluation.
Dr. Herman Lam is an Associate Professor at the University of Florida in the Department of Electrical and Computer Engineering and the Database Systems R&D Center. He has over 20 years of research and development experience in the area of database management and specialized in the areas of object-oriented systems, extensible database management systems, and knowledge base management systems. Currently, his work is focused on Web services, distributed objects, and interoperability of Web services and distributed objects through event-trigger-rule processing (ETR) technology. Dr. Lam has authored or coauthored over 60 refereed journal and conference articles and one textbook.