

A COST BENEFIT EVALUATION SERVER FOR SUPPORTING GENERAL  
DECISION-MAKING

By

FAHONG YU

A THESIS PRESENTED TO THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2001

Copyright 2001

by

Fahong Yu

## ACKNOWLEDGMENTS

I am most grateful to my committee chair, Dr. Stanley Su. Certainly without his support and guidance, I would not have been able to carry out this research. I especially thank Dr. Herman Lam, a member of my committee, for his creative and unique perspective on issues concerning many aspects of my research. I also appreciate the help of another committee member, Dr. Joachim Hammer, who was willing to give me his time and offer sound guidance.

My sincere appreciation goes to the Ph.D. student Mr. Youzhong Liu, who provided valuable comments and suggestions throughout the process of developing this thesis and helped me to significantly improve the quality of the work.

I also owe many thanks to Ms. Sharon for her friendly help and service. I would also like to express my thanks to Mr. Yuan Shi, Ms. Xiaoli Liu, Ms. Jie Meng, Miss Maggie, and to all the members of the Database Systems R&D Center, University of Florida.

## TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS .....	iii
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
ABSTRACT.....	viii
1 INTRODUCTION .....	1
2 COST BENEFIT DECISION MODEL (CBDM).....	8
2.1. Elementary Criteria.....	11
2.2. Aggregation Structure.....	13
2.3. Cost-Benefit Analysis .....	15
3 COST BENEFIT EVALUATION SERVER.....	17
3.1 Knowledge Specification .....	17
3.1.1 Preference and Cost Specification .....	17
3.1.2 Preference Aggregation Structure.....	20
3.2 CBES Evaluation Modes .....	21
3.2.1 EXHAUSTIVE Mode.....	21
3.2.2 BEST Mode .....	23
3.2.3 WORST Mode .....	27
3.2.4 APPROXIMATE Mode.....	27
4 IMPLEMENTATION OF CBES.....	29
4.1 CBES Service Selection.....	30
4.2 Build-Time Knowledge Specification .....	31
4.2.1 Preference Specification .....	31
4.2.2 Cost Specification .....	36
4.2.3 Preference Aggregation .....	38
4.2.4 Other Evaluation Parameters .....	41
4.2.5 The Representation of Knowledge Specification .....	42
4.3 Runtime CBES Evaluation .....	43
4.3.1 Business Specification .....	44

4.3.2 Evaluation Mode Selection .....	45
4.3.3 Cost-Benefit Evaluation and Evaluated Result.....	45
5 CASE STUDY .....	49
5.1. Supplier Selection .....	49
5.2. Automated Negotiation.....	50
6 SUMMARY AND FUTURE WORK .....	54
6.1 Summary .....	54
6.2 Future Work .....	55
LIST OF REFERENCES .....	56
BIOGRAPHICAL SKETCH .....	60

## LIST OF TABLES

<u>Table</u>	<u>Page</u>
2.1 Aggregate Function Spectrum .....	15
4.1 Conversions between the Slider and Aggregation Function.....	39
5.2 Elementary Scoring Function on <i>Price</i> .....	52
5.2 Elementary Scoring Function on <i>Delivery_Day</i> .....	52

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
2.1 The Parameter Tree of a Product Specification .....	9
2.2 Cost-Benefit Decision Model and Evaluation Process .....	10
2.3. The Aggregation Structure of <i>Computer</i> .....	14
2.4 Region of Acceptable Solution (ROAS).....	16
4.1 Cost Benefit Evaluation Server.....	29
4.2 CBES Service Selection.....	31
4.3 Preference Specification .....	32
4.4 Interpolation: a) Interpolation of Elementary Criteria .....	34
4.4 Interpolation: b) Adjustment of Interpolation.....	35
4.5 Definition of Class <i>ValueScorePair</i> .....	36
4.6 Definition of Class <i>CBMAttribute</i> .....	36
4.7 Cost Specification .....	37
4.8 Preference Aggregation .....	38
4.9 Definition of Class <i>CBMnode</i> .....	40
4.10 Preference and Cost Threshold Specification.....	42
4.11 Definition of <i>CBMInfo</i> .....	43
4.12 Business Specification .....	44
4.13 Evaluation Mode Selection.....	45
4.14 Definition of Class <i>EvaluatedConstraint</i> .....	48

Abstract of Thesis Presented to the Graduate School  
of the University of Florida in Partial Fulfillment of the  
Requirements for the Degree of Master of Science

A COST BENEFIT EVALUATION SERVER FOR SUPPORTING GENERAL  
DECISION-MAKING

By

Fahong Yu

December 2001

Chairman: Stanley Y. W. Su

Major Department: Computer and Information Science and Engineering

A business company often faces a decision situation in which the costs and benefits of some competing business specifications, such as business offers, product specifications, negotiation proposals, supplier capability specifications, need to be evaluated in order to select the best offer or the desirable ones. In e-business, there is a need to automate the cost-benefit evaluation process to support decision-making. This thesis presents a general-purpose cost-benefit evaluation server, which is implemented based on a quantitative cost-benefit decision model. In this model, a business specification is defined in the form of a hierarchical structure of attributes, each of which can have either a single value, an enumeration of alternative values, or a range of values. The cost of a specification is the sum of a base cost and additional costs associated with some attribute-value or attribute-value-range pairs given in the specification. The benefit of a specification is calculated based on the preference scores assigned to possible values and value ranges, and an aggregation structure containing weights and aggregation

functions specified by a decision-maker. The calculated cost and the global preference score are used in a cost-benefit evaluation process to derive a final cost-benefit indicator for each alternative value combination given in the specification. The cost-benefit evaluation server (CBES) is implemented in Java, Java Swing, and Java Applets/Servlets. It consists of a set of build-time tools and a run-time evaluation engine. The build-time tools provide a set of web-based interfaces to assist the decision-maker to capture the cost and preference information. Based on this information, a run-time evaluation engine evaluates the alternative value combinations given in a business specification and produces their corresponding cost-benefit indicators as the output. The server supports four different modes of evaluation to meet different needs of users. The design and implementation of the build-time tools and the run-time evaluation engine are presented in this thesis. A business scenario, which involves supplier selection and automated negotiation, is given to illustrate the application of the server and its evaluation modes.

## CHAPTER 1 INTRODUCTION

With the advent of the Internet, collaborative e-business has been attracting more and more attention from both academia and industry [PHI00, LIU01]. Various technologies are being developed to support collaborative e-business, such as customer relationship management [SIE01], supply chain management [I201, IND01], eMarketplaces [ARI01, COM01], and automated negotiation and auction systems [BEA96, HUA00, KUM98, SU01, TSV97, WUR98].

In e-business, a business company or person often faces a decision situation in which a cost-benefit analysis on a business specification needs to be performed before a proper decision can be made or an appropriate action can be taken. By “business specification”, we mean any business document which 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 inter-relationship 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.

To perform cost-benefit evaluation in the context of e-business, we need:

- A structured way to present a business specification so that it can be more easily processed by computers
- A quantitative cost-benefit decision model to calculate costs and to capture the preferences of decision makers based on which cost-benefit evaluation of business specifications can be performed
- A cost-benefit evaluation server capable of evaluating multiple business specifications concurrently
- A scalable Internet-based information infrastructure to support collaborative e-business

The current evaluation tools for evaluating programs, alternatives, and products are based on the cost-effectiveness analysis (CEA) and cost-benefit analysis (CBA) models. These two models diverge in their treatment of the consequences, or the benefits, of the programs and its alternatives [WER98]. The major concern of CEA is to determine which of the available alternatives is the least expensive way to produce the same benefit. Whereas, CBA deals with the comparative analysis of alternatives in terms of both the costs and consequences (or benefits) [WER98] and has become the most

common and popular evaluation technique used to evaluate programs and products. Since Eckstein [ECK58] 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 [ELI93, GOL96, WYL83, STE84, WER98], financial decision for business [COS01, AVI01, IND01], government management and administration [CLA94, GAO86, SSA90, DEP92, DEP93], and environmental damage assessment [HAN95, CFE01]. 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 [DRU87, CLA94, ELI93, WER98, SU87]. 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 outweighs 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 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) [CLA94, GAO86, SSA90]. This model became economically feasible in the early 1970s and is now extended for many different purposes, such as the confirmation of continuing eligibility for a benefit and data quality audit [CLA94]. However, the cost-benefit evaluation process in this model is time-consuming and the valuation of benefit is based on the approximate estimation.

In regard to the valuation of benefits 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 worth that is measured by the discounted value over time [DEP92, CLA94, WER98]. The net benefit is the summation of all the comprehensive benefits, including outcomes that are monetary, quantitative, qualitative [CFE01], and/or the estimation of the future monetary benefits, if applicable [WYL83, STE84, WER98, COS01]. In these models, the ranges and types of costs and benefits could vary widely according to the study design and the intervention of interest. The costs used to evaluate the program could include the total value of resources directly and indirectly consumed by the program under evaluation as well as other alternative programs being investigated. 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 cost-benefit analysis in Avicom [AVI01] is a comprehensive suite of integration service. In this service, all related information systems are fully integrated in order to communicate to each other and exchange data. However, because each and every company has its unique set of applications, system integration has to be implemented as a predefined solution. The evaluation system in Indent [IND01] can deal with complicated business specifications with multiple alternatives, but either the evaluation system is predefined like that in Avicom or the evaluation is by simulation, thus lacking flexibility. In both models, users cannot directly interact with the cost-benefit evaluation tools. The cost-benefit decision model (CBDM) [SU87], which is based on the concept of logic scoring of preferences, 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 1, 2 or 3 weeks. In a traditional cost-benefit evaluation system, a pre-processor 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 thesis, we describe a general-purpose cost-benefit evaluation server (CBES), which is implemented based on an extended quantitative cost-benefit decision model (CBDM) [SU87], 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 the CBES at the same time and in different modes of evaluation. The CBES provides a set of build-time specification tools and a run-time 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 modes. 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. Second, the model allows more elaborate structures to be specified by decision-makers for aggregating the preference scores assigned to attribute-value or attribute-value-range pairs to derive the global preference score. Third, the cost-benefit evaluation 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 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. Su, et al. [SU00a, SU01] 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 reported in this thesis is one of the replicable servers of the ISEE infrastructure. The design of this server was done with the assistance of a Ph.D. student, Mr. Youzhong Liu. My main contribution is in the implementation of CBES.

The remaining portion of this paper is organized as follows: In Chapter 2, the cost-benefit decision model is presented. In Chapter 3, the build-time knowledge setup tools and the four run-time evaluation modes and the algorithms for implementing them are presented. The implementation of CBES is discussed in Chapter 4. In Chapter 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 conclusion and future work are presented in Chapter 6.

## CHAPTER 2 COST BENEFIT DECISION MODEL (CBDM)

In an earlier work [SU87], a cost-benefit decision model was introduced 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 (LSP)” method introduced by Dujmovic [DUJ75]. The cost-benefit decision model (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 Figure 2.1). The attributes and the structure of a parameter tree form a “template”, which is commonly accepted by business companies that conduct a particular type of business. Their meanings are mutually understood by the sender and receiver of a business specification. In a cost-benefit evaluation, the attributes of a parameter tree are separated into a preference tree and a cost tree. The former contains those attributes and values to which CBES can assign some preference scores based on the preferences pre-specified by a decision-maker at build-time, and the latter contains those attributes and values to which costs in monetary terms can be assigned. The attributes of these two trees may overlap. A preference analysis model is used to evaluate the preference tree, and a cost analysis model is used to evaluate the cost tree. The results of these two evaluations are then combined in a cost-preference analysis to derive an overall evaluation indicator (see Figure 2.2).

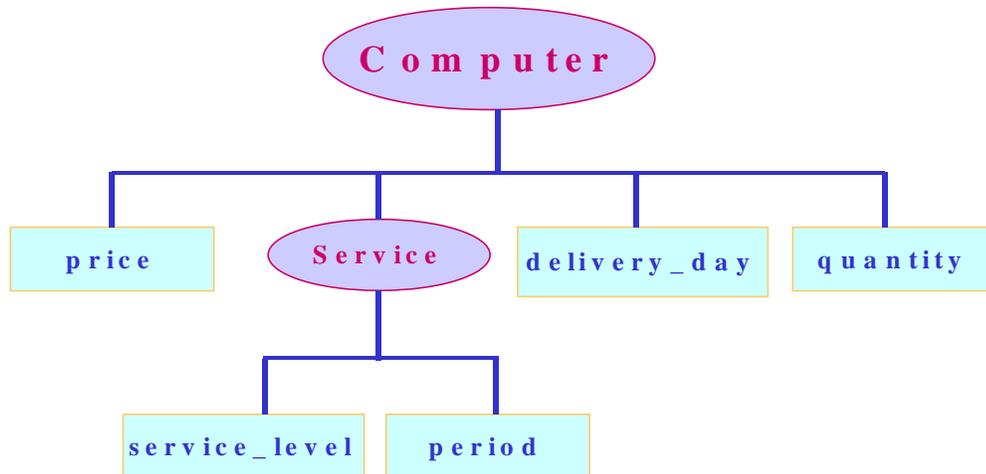


Figure 2.1 The Parameter Tree of a Product Specification

Figure 2.1 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-hour service) and *period* (3 years 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).

The cost tree is a sub-tree 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 \$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-hour service and the service period is 3 years, an additional cost of \$199 is added to the base cost to derive an aggregated cost.

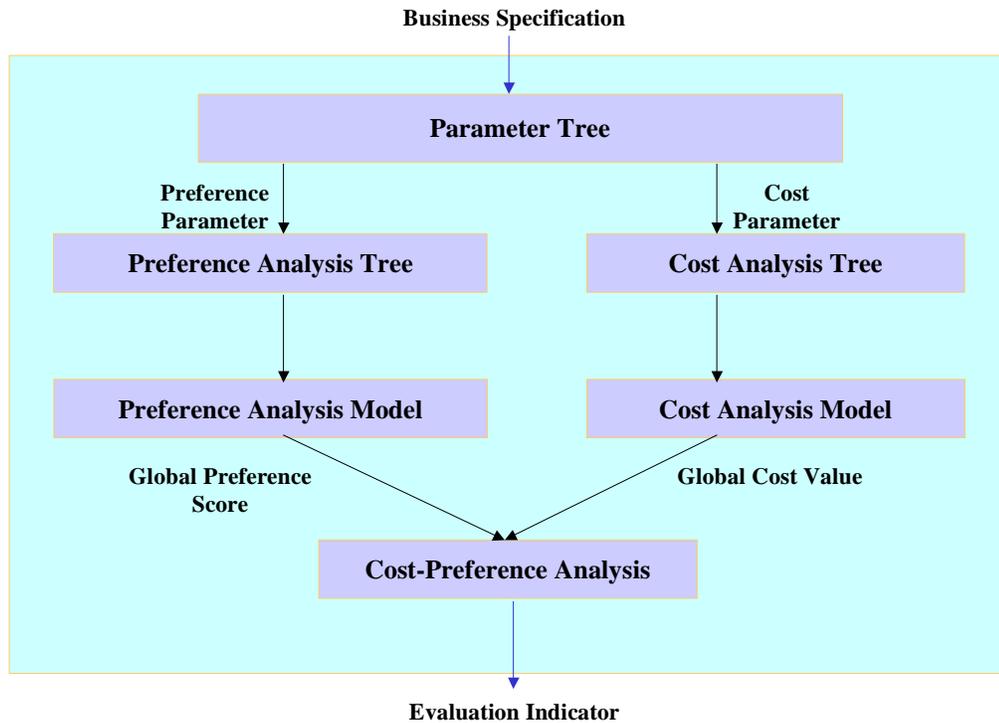


Figure 2.2 Cost-Benefit Decision Model and Evaluation Process

The preference tree is also a sub-tree of the parameter tree. It contains those attributes and values given in the parameter tree to which a decision-maker has pre-defined 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 pre-specified 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 upon by the issuers and receivers of that type of business specifications. Each template contains a pre-defined 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 between [0, 100]. The integer expresses the percentage of satisfaction the decision-maker has with a specific attribute-value or attribute-value-range pair. The purpose of elementary criteria is to transform

attribute values to preference scores within a scale of (0,100). These preference scores are used in an aggregation of preference during preference analysis.

There are several ways of formulating elementary criteria. 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-hour” service. The above example shows three elementary criteria defined for three discrete values of *service\_level*. For the attribute of the RANGE type, such as *delivery\_day*, a decision maker may specify 100 if the number of days 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 delivery time is greater than 11 days. The general form for this type of specification is as follows:

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

or, in the above example,  $\text{ENUM } \{\text{RANGE } [1,1], \text{RANGE } [2,5], \text{RANGE } [6,10], \text{RANGE } [11, \text{MAX}]\}$ . In a RANGE specification, CBES uses *MAX* and *MIN* to represent the maximum and minimum value of an attribute, respectively. 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,  $\text{RANGE}[1, 1]$  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. This is particularly important when the value set of an attribute cannot be enumerated. For

instance, *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  $E(\textit{deductible}) = \textit{Integer}((1 - \textit{deductible} / 1000) * 100)$ . If the possible values of *deductible* are in the range of [0, 1000]. The above function says that the buyer is fully satisfied with 0 *deductible* and completely unsatisfied if the deductible is \$1000. If the *deductible* is between 0 and \$1000, the preference score is calculated by evaluating the above function.

## 2.2. Aggregation Structure

The aggregation structure of the preference tree is also pre-defined 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, to generate a global preference score. An aggregation structure specifies how different subsets of the related attributes should be aggregated, what aggregation functions should be applied for these substructures, and what weights should be assigned to attributes to indicate their relative importance. It allows complex logical relationships among preference attributes to be unambiguously expressed.

In each aggregation substructure, some attributes are required and others are optional. The required attributes describe the features that a specification must exhibit. If a required attribute were not given a value, the preference rating of the substructure that this attribute is in would be greatly reduced. The optional attributes describe the nonessential features of a specification. The presence or absence of this kind of features would minimally impacts the overall preference rating of the aggregation substructure. Figure 2.3 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 substructures are then aggregated again and the result is then aggregated with *price*.

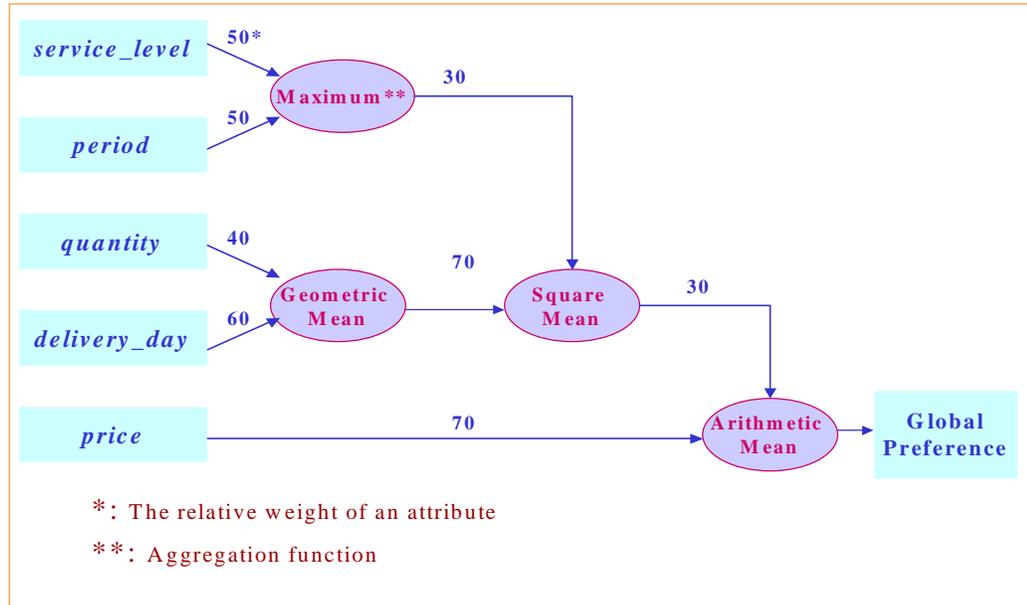


Figure 2.3. The Aggregation Structure of *Computer*

The attribute preference scores (APS) assigned to attribute-value or attribute-value-range pairs are weighted based on the weights shown in the figure and aggregated into a global preference rating using a spectrum of “preference aggregation functions”. These functions are derived from a weighted power mean shown below as Equation 2.1[DUJ75], where  $e_i$  are the APSs and  $w_i$  are the weights:

$$E = (w_1 \cdot e_1^r + w_2 \cdot e_2^r + \dots + w_n \cdot e_n^r)^{1/r} \quad (2.1)$$

By varying the value of  $r$ , a spectrum of aggregation functions can be generated, including functions such as *min*, *max*, *weighted arithmetic mean*, etc. The logical conjunction operation corresponds to the minimum function  $\min(e_1, \dots, e_n)$  and the logical disjunction operation corresponds to  $\max(e_1, \dots, e_n)$ . Some commonly used functions are given in Table 2.1.

Table 2.1 Aggregate Function Spectrum

Minimum	$E = \min(e_1, e_2, \dots, e_n)$	$r = -\infty$
Weighted harmonic mean	$E = 1 / (w_1/e_1 + w_2/e_2 + \dots + w_n/e_n)$	$r = -1$
Weighted geometric mean	$E = (e_1)^{w_1} \cdot (e_2)^{w_2} \dots$	$r = 0$
Weighted arithmetic mean	$E = w_1 * e_1 + w_2 * e_2 + \dots + w_n * e_n$	$r = 1$
Weighted square mean	$E = \sqrt{w_1 * e_1^2 + w_2 * e_2^2 + \dots}$	$r = 2$
Maximum	$E = \max(e_1, e_2, \dots, e_n)$	$r = +\infty$

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 Maximum aggregate function 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 Minimum aggregation function will take the minimum score among all the preference scores as the global preference. In the above example, if the Minimum aggregation function is used and a client is only 10% satisfied with the speed of the CPU, the 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 to integrate the global preference score (GPS) and the aggregated cost (AC) to get an overall evaluation indicator on the input specification. The evaluation indicator is the reflection of the overall goodness or desirability of the evaluated business specification. It is represented by the ratio of GPS and AC subject to some threshold constraints: a preference threshold ( $E_0^*$ ), a cost threshold ( $C_0^*$ ), and a preference cost ratio threshold ( $R_0^*$ ), which are specified by the

decision-maker. As shown in Figure 2.4, 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.

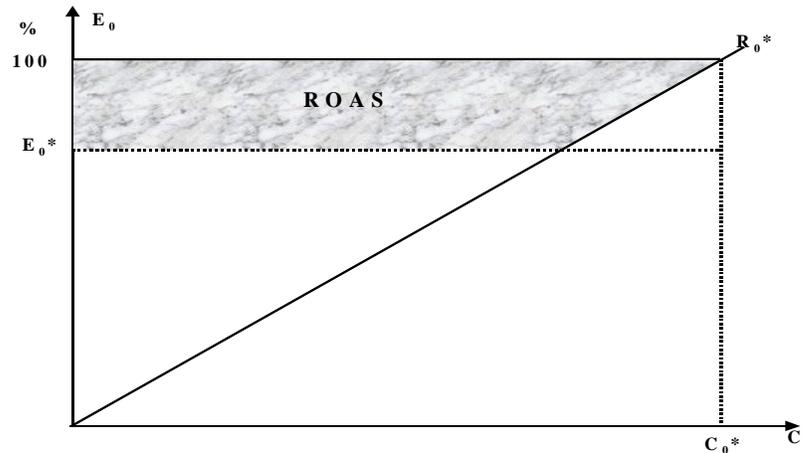


Figure 2.4 Region of Acceptable Solution (ROAS)

With different specifications of  $E_0^*$ ,  $C_0^*$ , and  $R_0^*$ , the ROAS may differ from case to case. In CBES, the following four approaches are used to derive the final evaluation indicator for an input specification:

1. If  $E_0^*$ ,  $C_0^*$ , and  $R_0^*$  are all specified, but one of the given thresholds is not satisfied, the indicator is set to 0. If all the thresholds specified by the user are satisfied, the ratio of the calculated preference score and the aggregated cost is returned as the evaluation indicator.
2. If  $E_0^*$ ,  $C_0^*$ , and  $R_0^*$  are all not specified, no threshold is assumed for the corresponding parameter. The final indicator is set as the ratio of the calculated preference score and the aggregated cost.
3. If the global preference score is zero, the final indicator is set to zero.
4. If the aggregated cost is zero, the global preference score is used as the final indicator.

## CHAPTER 3 COST BENEFIT EVALUATION SERVER

The cost-benefit evaluation server (CBES) is a key component of the ISEE information infrastructure to support decision-making in collaborative e-business. It is designed and developed based on the extended cost-benefit decision model (CBDMM) and includes a set of build-time knowledge specification tools (Section 3.1) and a run-time evaluation engine (Section 3.2).

### 3.1 Knowledge Specification

At build-time, CBES allows a decision-maker (or user) to formulate her/his elementary criteria for some possible values of the attributes given in a template, specify the cost information, and define the cost-benefit evaluation requirements. All these information are used for the run-time evaluation, comparison, and selection of business alternatives.

#### 3.1.1 Preference and Cost Specification

In CBES, the base type of each attribute can be either *Number* (int, float, and double) or *String*. The value of an attribute can be a single value or a range of values and the preference score assigned to a particular value or value range can be an integer ranging from 0 to 100 or a function that defines the possible values. If the base type of attribute is *String*, the preference score can only be an integer.

##### 1. Value Specification

CBES allows the user to assign preference scores to single values or value ranges of an attribute whose base type is *Number*. The specified values or value ranges should not have overlaps else their preference scores cannot be determined. This is particularly important when the input specification contains ranges of values. For example, assuming the value range [10, 15] is already specified by the user with a preference score of 80, the user should not be allowed to specify another value range [14, 20) or [8, 12] with a different preference score. Otherwise, the overlapping values will have different preference scores. If the user violates the above non-overlapping rule, the user has to remove one of them.

## 2. Preference Score Specification

Preference score indicates the users' degree of satisfaction with a specific attribute value or value range. In CBES, the user can either assign an integer value in the range of 0 to 100 or provide a function for computing the preference scores for different attribute values. To keep the consistency for all attributes, CBES requires that the function be defined as a polynomial. If a single value is given for an attribute in an input specification and the preference scores of that attribute are defined by a function, the function is evaluated to derive the preference score for that input attribute value. For example, if the preference scores of *quantity* are computed based on the function:  $E(\text{preference score}) = 2 * \text{quantity}^2 + \text{quantity} + 5$ , and the value given in an input specification for *quantity* is 3, then the preference score for this particular value would be 26. If the value of an attribute in the input specification is a value range and the preference scores of that attribute are defined by a function, then an integration operation is performed on the preference scores of these values in that range at runt-

time. The integrated value is the preference score of that range (see Chapter 4 for details). Obviously, for the *String* type of attributes, it is not meaningful to use a function to compute the preference score for a given string value.

### 3. Interpolation

It is not realistic to expect or require the user to specify the preference scores for all-possible values or value ranges of an attribute. To improve the functionality, CBES supports interpolation to derive an analytic expression based on the preference scores assigned by the user to some selected values and value ranges. The purpose of interpolation is to find the coefficients of the unique polynomial  $G(x)$  of degree  $\leq n-1$  that goes through the given  $n$  points  $(x_i, y_i)$ .

There are many methods for performing polynomial interpolations [PRE92]. The Lagrange's interpolation is very straightforward but has a very high time complexity,  $O(n^3)$ . The Newtonian interpolation is an improvement of Lagrange's method and the time complexity is reduced to  $O(n^2)$ . Polynomial interpolation in CBES is implemented based on the Newtonian interpolation [PRE92]. The basic idea of this algorithm is:

Suppose we already have  $n$  points  $(x_n, y_n)$  and an interpolating polynomial  $G(x)$  such that  $G(x) = y_i$ , for  $1 \leq i \leq n$  and we want to add just one more point  $(x_{n+1}, y_{n+1})$  to  $G(x)$ . Namely,

let  $G_{j-1}(x)$  interpolate  $j-1$  points  $(x_k, y_k)$ ,  $1 \leq k < j$ , so that

$G_{j-1}(x_k) = y_k$  and  $G_{j-1}(x) = (x - x_1) \dots (x - x_{j-1})$ , then

$$G_j(x) = [y_j - G_{j-1}(x_{j-1})] * D_{j-1}(x) / D_{j-1}(x_j) + G_{j-1}(x). \quad 3.1$$

Here,  $D(x)$  represents the interpolating polynomial. To achieve this  $n-1$  degree polynomial, the simple solution is to apply the formula (3.1)  $n$  times. The total time complexity is  $O(n^2)$ .

Interpolation is not applicable to all situations. It is meaningful to perform interpolation on the given specification only when the value of an attribute is continuous. It does not make sense to perform interpolation for an attribute whose base type is *String* and its values are discrete. Also, when a function is used in an elementary criterion specification, it is not meaningful to perform interpolation.

Interpolation can also be used in the calculation of costs. The extra costs for the attributes that have costs associated with them can be specified in a similar way as the preference specification. But the extra cost for a specific value or a range of values could be a float. In practice, only a few attributes can have extra charges.

### 3.1.2 Preference Aggregation Structure

CBES allows flexible aggregations of preferences. The aggregation structure in CBES is a hierarchical or tree-based structure that may contain a number of substructures. This structure is used to calculate the global preference score for an input specification at run-time. Three pieces of information are necessary to construct each substructure of the aggregation structure: some related attributes, the aggregation function, and the relative weight of each attribute. The weights, attributes, and aggregation function used to form a substructure reflect the decision-maker's feeling about the relative importance of these attributes and how the preference scores shown as the leaves of the substructure should be aggregated to produce an aggregated score for the substructure. The aggregated score becomes one of the leaf nodes of another substructure at the next higher level. Thus, CBES uses a bottom-up strategy to construct the aggregation structure.

### 3.2 CBES Evaluation Modes

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-rang pairs. These value combinations can be evaluated by CBES together following the steps shown in Algorithm 3.1. 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 3.2.1 – 3.2.4, respectively.

#### **Algorithm 3.1 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 pre-specified 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.

#### 3.2.1 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, (1200, 1300] represents a range from 1200 (exclusive) to 1300 (inclusive).

The above specification has  $100 * 3 * 200 = 60000$  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  $i^{\text{th}}$  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.

### 3.2.2 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* is equal to 3” has the highest preference score (e.g., 80), based on the pre-defined 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 3.1]** 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]: 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 Equation 2.1 shown in Section 2.2 are fixed. Equation 2.1 can be rewritten as

$E = (w_1 \cdot e_1^r + C)^{1/r}$ , where  $C = (w_2 \cdot e_2^r + \dots + w_n \cdot e_n^r)$  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

$$e_{1a}^r + C > e_{1b}^r + C$$

$$\left(e_{1a}^r + C\right)^{\frac{1}{r}} > \left(e_{1b}^r + C\right)^{\frac{1}{r}}$$

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

$$e_{1a}^r + C < e_{1b}^r + C$$

$$\left(e_{1a}^r + c\right)^{\frac{1}{r}} > \left(e_{1b}^r + c\right)^{\frac{1}{r}}$$

If  $r = 0$ , as shown in [SU87], the weighted power mean shown in Equation 1 can be rewritten as

$$\left(e_1\right)^{w_1} * \left(e_2\right)^{w_2} * \dots * \left(e_n\right)^{w_n}$$

Since  $\left(e_{1a}\right)^{w_1} > \left(e_{1b}\right)^{w_1}$ , we still have  $\left(e_{1a}^r + c\right)^{\frac{1}{r}} > \left(e_{1b}^r + c\right)^{\frac{1}{r}}$ , i.e.,  $E_a > E_b$ .

**[Lemma 3.2]** 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 3.1 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 3.1]** 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 3.2, 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 3.1, Algorithm 3.2 shown below is used for the preference evaluation in the BEST mode.

**Algorithm 3.2 Best Mode Preference Evaluation**

1. The input values of each attribute are evaluated against the elementary criteria pre-specified by the decision-maker to determine their preference scores. For each attribute, the attribute-value or attribute-value-range pair that has 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.

### 3.2.3 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.

### 3.2.4 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 pre-defined 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  $\frac{\int_a^b f(x)}{b-a}$ , 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.

## CHAPTER 4 IMPLEMENTATION OF CBES

The cost-benefit evaluation server (CBES) is implemented using Java, Java Swing, and Java Applets/Servlets. The build-time tools are implemented as a set of web-based applets and the runtime evaluation engine is implemented as a RMI server (see Figure 4.1).

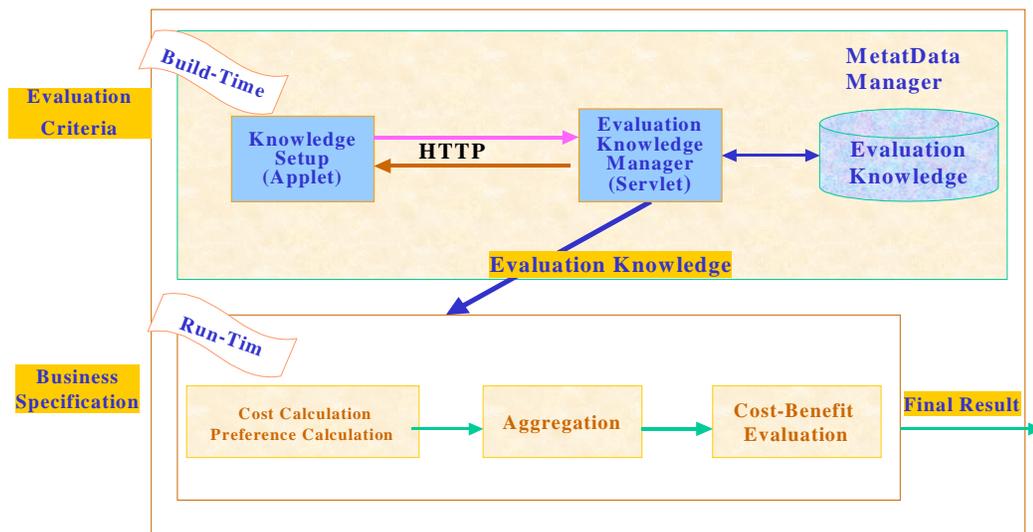


Figure 4.1 Cost Benefit Evaluation Server

The build-time tools assist a decision-maker to provide the following information:

- Preference specification
- Cost specification
- Preference aggregation
- Other parameters, such as the thresholds for cost-benefit analysis, and the evaluation mode

The run-time engine evaluates an input specification against the information specified at build-time.

In this Chapter, we will briefly introduce the CBES registration in Section 4.1. The build-time knowledge setup process, which includes preference and cost specification, preference aggregation, and evaluation requirements, is presented in Section 4.2. Section 4.3 shows the implementation of the CBES run-time engine.

#### 4.1 CBES Service Selection

As we stated in Chapter 2, 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. We assume that templates associated with those business specifications of interest to a decision-maker have been pre-defined and stored in a database. CBES is implemented as an independent RMI server and can be invoked by application systems or by users through interactive interfaces. For security, it requires each user to have his/her unique username and password to access the services of CBES (Figure 4.2). After logged on CBES, a user first selects a template from a set of pre-defined templates. The applet which interacts with the user would open a HTTP connection to the servlet URL and the meta-data information related to the selected template including the schema name, class name, attribute name, attribute base type, and other related information is retrieved from the *MetaData Manager*. The *MetaData Manager* is a RMI server used to store all the templates of different business types. The retrieved template will be used by the user to enter the preference and cost information.



Figure 4.2 CBES Service Selection

#### 4.2 Build-Time Knowledge Specification

CBES provides four graphic user interfaces (GUIs) embedded in a *TabPane* to assist the user to specify the preference and cost information (i.e., knowledge specification).

##### 4.2.1 Preference Specification

Preference specification in CBES is a build-time process, in which a user uses a GUI tool provided by CBES to define a set of elementary criteria to specify the percentages of satisfaction for some possible values of the attributes given in a template. Figure 4.3 shows the user interface for preference specification. For each attribute defined in the template, the user can give a preference score for each possible value or

value range. This process will produce a set of values (or value ranges) and their corresponding preference scores for each attribute. The same thing is done for all the attributes. There are three components in preference specification, the attribute list, input attribute information, and confirmation panel.

Attribute	Value	SCORE
delivery_day	[7, 15]	50
	[1, 2]	85
	[2, 8]	75

Figure 4.3 Preference Specification

The attribute-list panel simply displays the attribute names and their corresponding base types defined in the selected template. Users can either select an attribute and specify the preference information for it, or view the information of those attributes that have been specified.

The panel of the input attribute information allows the user to assign preference scores to some values or value ranges of each selected attribute, which are displayed on

the left part of the panel. As described in Chapter 3, the value could be a string, a single value, or a range of values. The preference score could be any integer number from 0 to 100 or a function of an attribute value. By selecting different buttons, users can modify, add, and remove any input directly from this panel. The *OK* button is the confirmation of the acceptance of the values and their corresponding scores of the selected attribute. Once the preference specification is done for all the attributes of a selected template, the user needs to confirm whether or not to accept the current specification by selecting the proper button. The *Help* will give the detail about how to specify the preference information.

Theoretically, the above preference specification scheme works. However, it is unrealistic to require a user to specify preference scores of all the 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* button shown in Figure 4.3 can be used to indicate that the user accepts the current input and would also like to perform an interpolation for this attribute based on the input values and their preference scores specified by the user. An applet window will display the interpolation result if the *View Interpolation* button is selected. The implementation of interpolation is based on the formula 3.2 using the Newtonian algorithm (see Chapter 3 for details). The existing interpolation algorithms [PRE92] 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. But 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. However, this way to perform the interpolation does not produce an accurate result. Some adjustments need to be made. For example, in Figure 4.4(a),  $R1_a$  and  $R1_b$ ,  $R4_a$  and  $R4_b$  are used to represent ranges R1 and R4, respectively. Assume that 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 Figure 4.4(a). The adjusted result is shown in Figure 4.4(b), which accurately represents the fact that all the values in the range R1 have the same preference score, and so as those in R4.

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

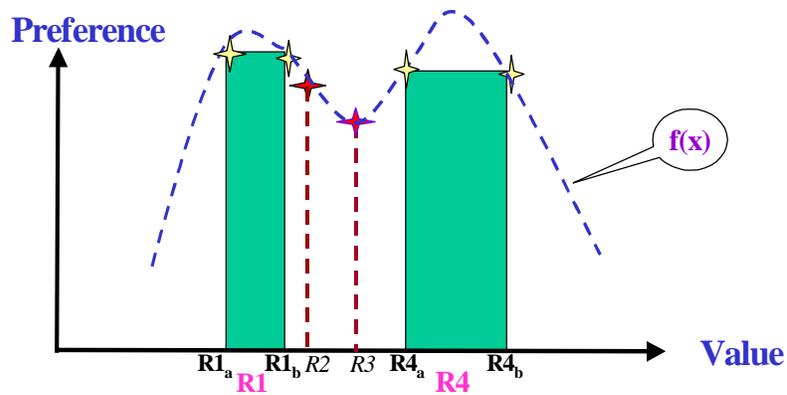


Figure 4.4 Interpolation: a) Interpolation of Elementary Criteria

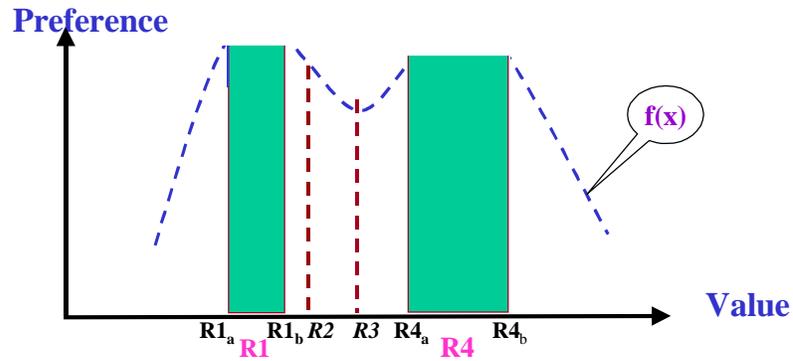


Figure 4.4 Interpolation: b) Adjustment of Interpolation

To improve the reliability, CBES will reject any illegal input. Whenever an error is detected, an applet window will be popped up and the user is required to fix the corresponding error. The common errors in preference specification include:

- Preference score is greater than 100 or less than 0.
- Input value is not a number when the base type of the attribute under specification is *Number*.
- The values of an attribute are overlapped.
- The function does not match with the predefined format when a function is specified instead of a preference score.

In CBES, each value (or value range) and its corresponding preference score are represented by an object class *ValueScorePair* (see Figure 4.5), which consists of the value type (i.e., string, single value, or value range), score type (i.e. single value or a function), value, and preference score. The *functionScore* is used to store the exponent-coefficient pairs when the preference score is a function. The preference specification of an attribute is represented by an object class *CBMAttribute*, which stores the preference specification, cost specification (see Section 4.2.2 for details), and other related

information, such as the interpolation of preferences and costs. The definition of the object class *CBMAttribute* is given in Figure 4.6.

```
class ValueScorePair {
    int valueType;           //single value or range
    Object low_value;       //single value (String) or the lower bound of a range
    Object high_value;      //the higher bound of a range
    boolean range_Start;    // inclusive '[' or exclusive '('
    boolean range_End;
    int scoreType;          //single value or function for score or cost
    float preferenceScore;  //used for the preference score or extra cost
    Vector functionScore;   //a function: each element in Vector is
                           //an object of exponent-coefficient pair.
}
```

Figure 4.5 Definition of Class *ValueScorePair*

```
class CBMAttribute {
    String name;            //attribute name.
    String baseType;        //attribute base type: Number or String
    Vector valueScorePair;  //value-Score Pairs of the attribute represented
                           //in object 'ValueScorePair'
    Vector valueCostPair;   //value-Cost Pairs of the attribute represented
                           // in object 'ValueScorePair'
    Vector interpolated_func; //interpolated function for preference
    Vector interpolated_cost; //interpolated function for extra cost
}
```

Figure 4.6 Definition of Class *CBMAttribute*

#### 4.2.2 Cost Specification

Cost specification is also a build-time process, in which the user identifies those attributes and values of a selected template that have costs associated with them and assign 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 \$999 and the standard memory configuration for the PC is 64M. If memory size is 96M or 128M, an additional \$20 or \$40 has to be added to the base cost, respectively. As shown in Figure 4.7, CBES provides a GUI tool to allow the user to specify the extra costs associated with some attribute-value (or attribute-value-range) pairs. Needless to say, the sender of a business specification may provide the cost information (e.g., the sender is the seller of a product).

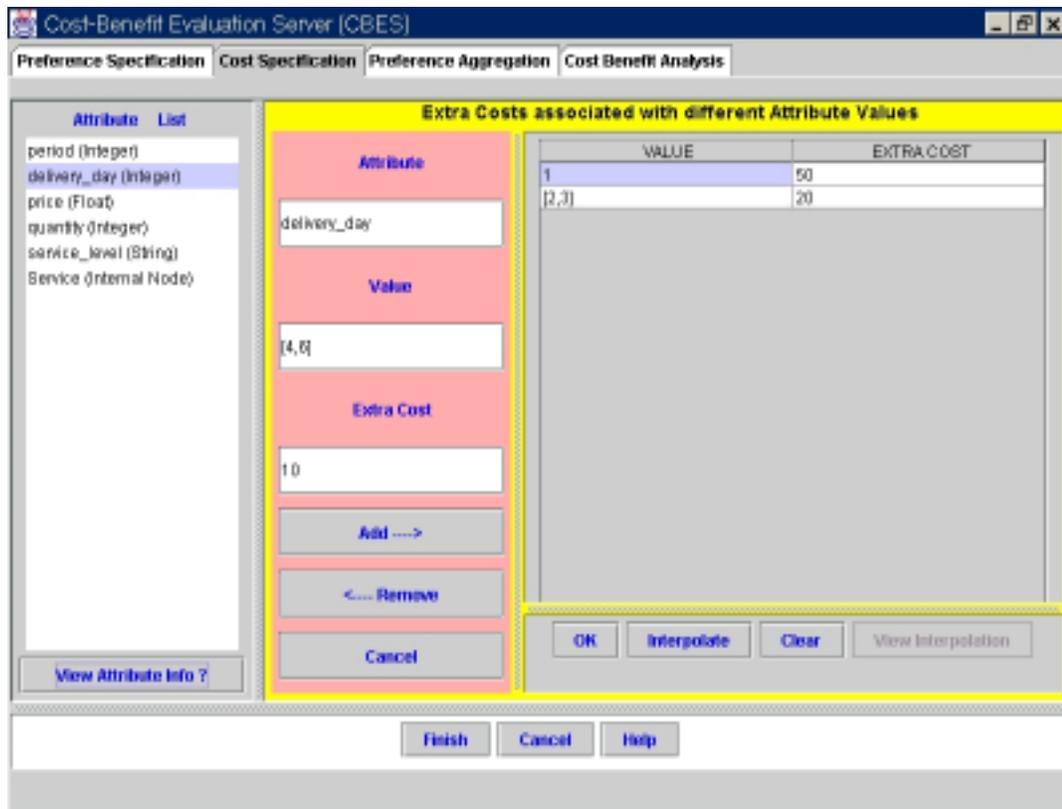


Figure 4.7 Cost Specification

Similar to preference specification, the additional cost could be defined as a function of an attribute value. Interpolation can also be used to derive the additional cost trend for the unspecified attribute values. We note here that only some attributes of a

template may have additional costs. They become part of the cost tree discussed before. The representation of each additional cost and the cost specification of each attribute are the same as those of the preference specification.

#### 4.2.3 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 Figure 4.8, which is composed of four panels. The user can 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.

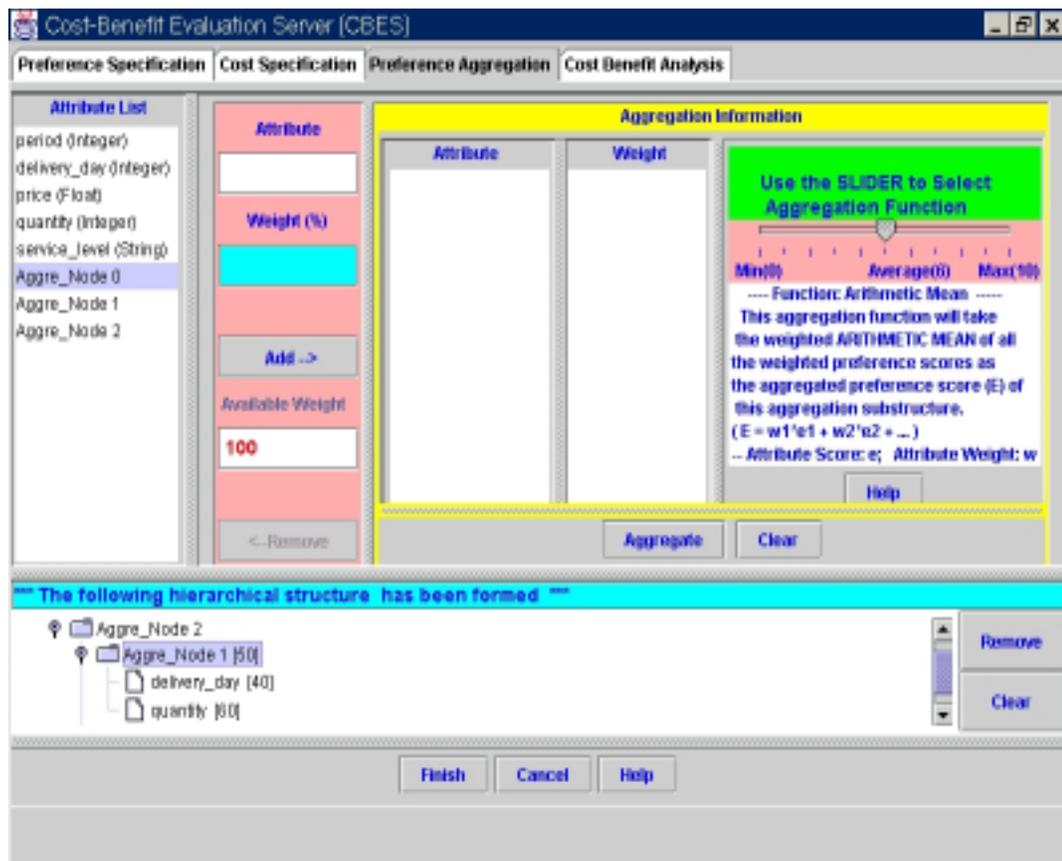


Figure 4.8 Preference Aggregation

The attribute-list panel is for the user to select the related attributes for aggregation. The aggregated substructures are numerated consecutively and automatically added at the end of the attribute list for further aggregations, such as the *Aggre\_Node 0 and Aggre\_Node1* shown in Figure 4.8. The weight-assignment panel (the middle part of Figure 4.8) is designed for the user to assign the relative weight to the selected attribute. The attributes and their corresponding weights of each aggregation are displayed in the *Aggregation Information* panel (the right panel). As a reminder, a textbox is used to display the available weight for each aggregation. If the total weight is greater than 100, a warning window will be popped out. In case the user ignores the warning, CBES automatically normalizes the relative weights of attributes to maintain 100 as the total weight of each aggregation substructure.

Table 4.1 Conversions between the Slider and Aggregation Function

<b>Aggregation Function</b>	<b>Value in Slider</b>
<i>Minimum</i>	0
<i>Harmonic Mean</i>	4
<i>Geometric Mean</i>	5
<i>Arithmetic Mean</i>	6
<i>Square Mean</i>	7
<i>Maximum</i>	10
Others	1, 2, 3, 8, 9

To improve the usability, a slider is provided as the selection tool for the user to choose the desired aggregation function from a spectrum of aggregation functions (the

right panel of Figure 4.8). Thus, the user does not have to know or understand the mathematics of these functions. Table 4.1 shows the conversions between the scales of the slider and the actual functions. The selected scale is used to set the r-value of the weighted power mean (equation 2.1) 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 including the aggregated substructures, the attribute names, and their relative weights are displayed in a hierarchical structure, which provides a convenient way for the user to view the constructed aggregation structure (see the lower part of Figure 4.8). CBES also allows the user to dynamically remove any attribute or substructure from the aggregation structure. When a substructure is removed, all the attributes (or nodes) attached to it will be removed as well. The aggregation structure will be automatically restructured. Using the same GUI tool, different users can construct different aggregation structures for the same template of a business specification. To help the user understand the meanings of aggregation functions, a separate window is provided to explain the aggregation functions.

```
class CBMnode {
    String nodeName;           //substructure Name
    Vector childList;          //list of the aggregated attributes
    Vector childWeight;        // list of the relative weights of attributes
    int aggregationFunction;   // aggregation function
}
```

Figure 4.9 Definition of Class *CBMnode*

The aggregation structure or its substructure is represented by an object class *CBMnode*, which includes the substructure name, the list of attributes aggregated, the list of the relative weights of attributes, and the aggregation function. *CBMnode* is defined in Figure 4.9.

#### 4.2.4 Other Evaluation Parameters

The GUI shown in Figure 4.10 is used for 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.

The preference score threshold is used to eliminate those value combinations whose global preference scores are less than the preference threshold. Its maximum value is 100. The default value is 0.

The cost threshold is the maximum acceptable cost specified by the user and is used by CBES to compare the aggregated costs derived for all the value combinations given in a business specification to eliminate those that are out of the acceptable cost limit. If the user does not specify it, all the costs associated with the value combinations are regarded as acceptable to the user.

The preference/cost ratio threshold is another index used for evaluating the value combinations. It ranges from 0 to 1. If not specified, CBES will take 0 as the default of this threshold.

During the run-time evaluation, the above thresholds are compared with the global preference score and the aggregated cost. If one of them is violated, the evaluation indicator will be set to 0.

The screenshot displays the 'Cost-Benefit Evaluation Server (CBES)' application window. It features four tabs: 'Preference Specification', 'Cost Specification', 'Preference Aggregation', and 'Cost Benefit Analysis'. The 'Cost Benefit Analysis' tab is active, showing two main panels:

- Cost-Benefit Evaluation Thresholds:** A red-shaded panel with four input fields:
  - Preference Threshold (E<sup>\*</sup>): 35
  - Cost Threshold (C<sup>\*</sup>): 1500
  - Ratio Threshold (E<sup>\*</sup>C<sup>\*</sup>)(%): 15
  - Base Cost: 1200
- Parameter Threshold Information:** A white panel with a text area explaining the process of cost-benefit analysis and the role of thresholds. It includes a sub-section titled '--- Preference Threshold ---' which states: 'It is the lower bound of the global preference score of a business specification. It is used to...'

At the bottom of the window, there are 'Finish' and 'Cancel' buttons.

Figure 4.10 Preference and Cost Threshold Specification

#### 4.2.5 The Representation of Knowledge Specification

The specified knowledge, which consists of the preference and cost specifications, the aggregation structure, and the evaluation requirements, is represented as an object class *CBMInfo* (Figure 4.11). The *clientID* and *product* are used to access the database. The build-time applet (*Knowledge Setup*) interacts with the *Evaluation Knowledge Manager* (servlets) by sending a POST method to pass the *CBMInfo* to the *MetaData Manager* (Figure 4.1). The information in *CBMInfo* is retrieved and used to evaluate the business specification at run-time.

```

Class CBMInfo{
    String  clientID;      //client Name
    String  product;      // product Name
    int     cost_threshold; // the maximum acceptable cost
    int     score_threshold; // the minimum acceptable score
    int     ratio_threshold; // the minimum acceptable ratio
    int     basalcost;    // the base cost of the product
    CBMnode root;        // the aggregation structure
    Hashtable attributes; // preference and cost information
}

```

Figure 4.11 Definition of *CBMInfo*

### 4.3 Runtime CBES 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 for activating the engine in one of 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 graphical interface is provided by CBES to allow the user to define a business specification on an applet, select the CBES mode, and evaluate the specification through a servlet. The run-time CBES's applet performs a two-way communication with a 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 applet, which displays the result to the user.

#### 4.3.1 Business Specification

CBES allows a user to directly define a business specification through the web-based interface (Figure 4.12). Based on the selected template, the related meta-data information is queried from the *MetaData Manager* through the HTTP connection and displayed in a panel for selection (the left panel of Figure 4.12). Each attribute can have multiple values which can be a string, a single value or a value range. All values or value ranges specified for each attribute are numerated and considered as an enumeration. To avoid the inconsistency, all values and value ranges within an attribute should not overlap. In effect, a specification may have multiple value combinations, which will be evaluated by CBES together.

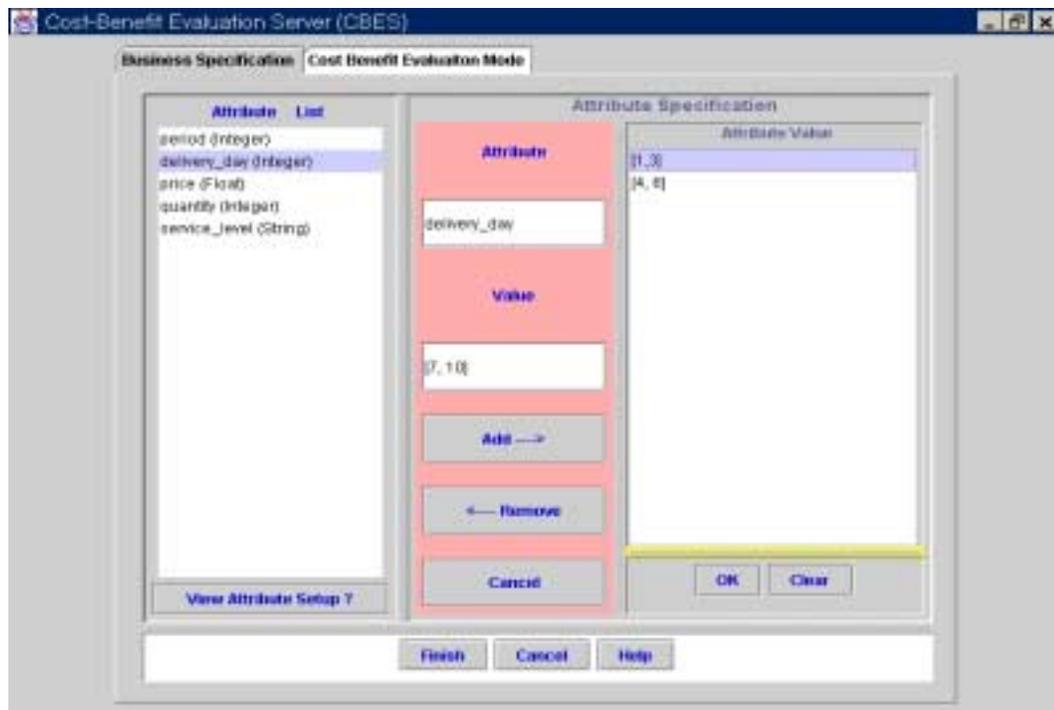


Figure 4.12 Business Specification

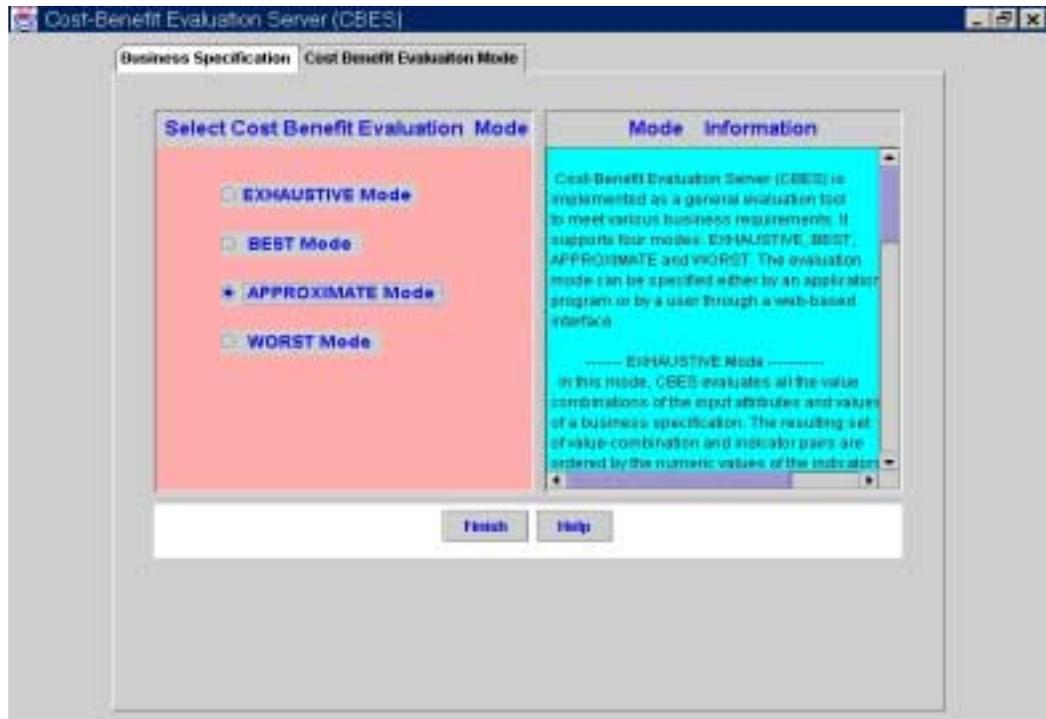


Figure 4.13 Evaluation Mode Selection

#### 4.3.2 Evaluation Mode Selection

CBES is implemented as a general evaluation tool to meet various business requirements. It supports four modes: EXHAUSTIVE, BEST, APPROXIMATE and WORST. The evaluation mode can be specified either by an application program or by a user through a web-based interface (Figure 4.13). In the latter case, the APPROXIMATE mode is set as the default. The details of the evaluation modes are described on the right part of Figure 4.13.

#### 4.3.3 Cost-Benefit Evaluation and Evaluated Result

Once a business specification is submitted, CBES communicates with the *MetaData Manager* to retrieve the corresponding preference and cost information of the selected template that has been specified by the decision-maker at build-time. This

information is used to derive the global preference score and the aggregated cost using Algorithm 3.1 and Algorithm 3.2, respectively.

The technique to calculate the preference score of a value or a value range is very straightforward. When the elementary criteria of the attribute being evaluated are a set of value-score or value-range-score pairs, CBES simply checks whether or not the input value matches or overlaps with any value or value range of the elementary criteria. If it does, the corresponding score is set as the preference score of this value or value range. Otherwise, the preference score of this particular value is set to zero. However, integration has to be applied when the input value of an attribute is a value range and the elementary criterion of this attribute is defined as a function. Using integration, the result is more accurate and reasonable than that obtained by a simple calculation based on the arithmetic mean method. The implementation of integration is based on the following formula:

$$\frac{\int_a^b f(x)}{b-a} \quad (4.1)$$

Where  $f(x)$  is the function given as the elementary criteria for the range  $[a, b]$ . In the formula 4.1, the numerator is evaluated based on the following equation:

$$f(x) = h [1/2f_1 + f_2 + f_3 + \dots + f_{N-1} + 1/2f_N] + O((b-a)^3 f''/N^2)$$

Here,  $f(x)$  is the evaluated value. The interval  $(a, b)$  is denoted as  $x_0, x_1, \dots, x_N$ , which are spaced apart by a constant  $h$ ,  $N$  stands for the number of steps over which  $f(x)$  is evaluated, and  $(f'')$  is the value of  $f(x)$ 's second derivative somewhere in the interval of integration. The term  $O((b-a)^3 f''/N^2)$  represents the estimated error. The above

technique is also applicable for the calculation of the additional cost of a value or a range of values by accessing the cost information.

Because CBES supports different evaluation modes, the way to calculate the preference score and the additional cost of an attribute differs from mode to mode. In the EXHAUSTIVE mode, the above technique can be directly used to calculate the attribute preference score and extra cost because the input value of each attribute is either a value or a range of values. In the BEST mode, when an attribute has multiple values or value ranges, CBES will first evaluate the preference score for each value or value range. Then the highest preference score is assigned to this attribute. While in the WORST mode, the lowest preference score is selected. In both modes, only the preference scores are considered because it is not always true that the attribute value having the highest (or lowest) preference score also has the minimal (or maximal) extra cost. In the APPROXIMATE mode, for an attribute with multiple values or value ranges, all these values will be evaluated first. The arithmetic mean of all the evaluated preference scores or costs is selected as the preference score or the extra cost of this attribute.

The calculation of the global preference score actually is a recursive process. With the derived attribute information, CBES evaluates the aggregation structure from bottom up (i.e., leaves to the root). Each aggregation substructure is evaluated based on the aggregation function, the preference scores, and the relative weights of all the related attributes given in that substructure. The preference score of the root is the global preference score of the input specification. The aggregated cost of the specification is the sum of the base cost and the extra costs associated with some attributes and their values. The derived global preference score and the aggregated cost are then compared with the

evaluation thresholds specified by the user at build-time. If any threshold is violated, the evaluation indicator will be 0. Any specification within the ROAS is acceptable and its evaluation result is returned to the requestor.

```
class EvaluatedConstraint{
    String userID;           //user name
    String productID;       //evaluated business specification
    Hashtable evaluatedAttrList; //list of the evaluated attributes
    int cost;               // total cost
    int mode;               // evaluation mode
    int finalResult;        //evaluation indicator
}
```

Figure 4.14 Definition of Class *EvaluatedConstraint*

An object class, *EvaluatedConstraint*, which is defined in terms of the evaluation indicator, a list of the evaluated attributes, and other related information, such as userID, productID, is specified as the return object (see Figure 4.14). Because CBES provides different APIs for activating CBES in different modes, the contents of the returned object can be different. For example, in the APPROXIMATE mode, the *evaluatedAttrList* is empty. In the BEST and WORST modes, the *evaluatedAttrList* may contain multiple value combinations. The *evaluatedAttrList* contains the ranked value combinations when the evaluation mode is the EXHAUSTIVE mode.

## CHAPTER 5 CASE STUDY

In this Chapter, 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 one hundred companies, which conduct business over the Internet. Assume that fifty companies responded to the request and return their quotes. Since negotiation can be a time-consuming process, it is unrealistic for Company A to negotiate with all these fifty suppliers. 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 the above scenario, Company A can use the BEST mode to evaluate each supplier's offer to find the best value combination in that offer and its final cost-benefit indicator. By comparing the final indicators of the fifty 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 fifty 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 the ones that responded to the request-for-quote and have the following attributes and values in their quotes:

Supplier S1: {  $\{price = \text{RANGE}[1600, 1800], delivery\_day = \text{ENUM}\{7, 9, \text{RANGE}[13, 15], 17\}\}$  }

Supplier S2: {  $\{price = \text{RANGE}[1500, 1900], delivery\_day = \text{ENUM}\{14, 21\}\}$  }

CBES would use Company A's pre-registered preference information to determine the preference scores (PS) for the above values and value ranges and derive the global preference scores (GPS) as follows:

Supplier S1:  $PS(price) = 95$ ,  $PS(delivery\_day) = \text{AVERAGE}(90, 80, 70, 60) = 75$ .

$$GPS = 95 * 0.6 + 75 * 0.4 = 87.$$

Supplier S2:  $PS(price) = (300*95 + 100*80)/400 = 91.25$ , and

$$PS(delivery\_day) = \text{AVERAGE}(70, 60) = 65.$$

$$GPS = 91.25 * 0.6 + 65 * 0.4 = 80.75.$$

In the above calculation, we assume that the weights assigned to the two attributes are 60% and 40% respectively. Since Supplier S1 has the better global preference score, it is selected as the supplier with which to conduct business.

## 5.2. Automated Negotiation

Automated negotiation is a value-added e-service of the ISEE infrastructure. As documented [SU00b, HAM00, HUA00, SU01], a negotiation server is replicated and installed in multiple ISEE hubs. A pair of negotiation servers would conduct an automated bi-lateral 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 pre-registered with their respective negotiation servers. The evaluation of a proposal against the pre-registered constraints is a constraint satisfaction processing problem and is handled by a constraint satisfaction processor (CSP) [HUA00], 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 counter-proposal 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:  $\{price = [1730, 1850], delivery\_day = \{7, 12\}\}$ . Based on the elementary score for each parameter given by the user (Tables 5.1 and 5.2), the best preference score is obtained when price is in the range  $[1730, 1800]$  and the  $delivery\_day$  is 7.

Table 5.2 Elementary Scoring Function on *Price*

<i>price</i>	Score
[2000, MAX)	0
[1900, 2000)	60
[1800, 1900)	80
(MIN, 1800)	95

Table 5.2 Elementary Scoring Function on *Delivery Day*

<i>delivery_day</i>	Score
(14, MAX)	60
(10, 14]	70
(7, 10]	80
(MIN, 7]	90

If arithmetic mean is used to aggregate the preference scores of these two parameters and the weight for *price* and *delivery\_day* are 60% and 40% respectively, the global preference score is

$$\begin{aligned}\text{GPS} &= \text{PS}(\text{price}) * 0.6 + \text{PS}(\text{delivery\_day}) * 0.4 \\ &= 95 * 0.6 + 90 * 0.4 = 93.\end{aligned}$$

The best preference score 93 together with the associated proposal  $\{\text{price}=\text{RANGE}[1730, 1800], \text{delivery\_day}=7\}$  is returned to the Negotiation Server.

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.

## CHAPTER 6 SUMMARY AND FUTURE WORK

### 6.1 Summary

In this thesis, we have described the design and implementation of a general-purpose cost benefit evaluation server for supporting decision-making in e-business. 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. Our work is different from the existing work in the following ways:

- The extended cost-benefit decision model (CBDM) captures the subjective preferences of decision-makers with respect to different attributes and values presented in business specifications.
- CBDM 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.
- CBES supports a wide spectrum of preference aggregation functions to fit different decision situations.
- The input business specification can contain attributes of RANGE and ENUM types instead of limiting each attribute to have a single value so that the

evaluation system is able to calculate preference scores for both discrete values and value ranges more efficiently.

- An interpolation tool is built to derive evaluation functions based on the limited information given by the user instead of requiring the user to specify a preference score for every possible value.
- CBES offers four modes of operations to meet the different cost-benefit evaluation needs of the users.

## 6.2 Future Work

In this work, we assume that templates can be pre-defined 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 [OAG01], and RosettaNet [ROS01] have been developing standard terms and structures for specifying business object documents. Some possible improvements of CBES are listed below:

- Currently, CBES only evaluates business object documents that are specified either by a set of predefined object classes (Java classes) or by the user through a GUI. CBES can be extended to take XML documents as input specifications.
- At present, decision makers provide CBES with preference and cost information directly through GUIs. This information, i.e., elementary criteria, aggregation structure, costs, etc., could as well be specified in XML documents and be transmitted to CBES over the Internet for its run-time use.

## LIST OF REFERENCES

- [ARI01] Ariba Inc., Ariba marketplace.  
[http://www.ariba.com/solutions/ariba\\_product.cfm?solutionid=7&pid=404am](http://www.ariba.com/solutions/ariba_product.cfm?solutionid=7&pid=404am). 2001.
- [AVI01] AVICOM Technowhow, Inc., Business models. <http://www.avicomsys.com>, 2001.
- [BEA96] Beam, C., Segev, A. and Shanthikumar, J., Electronic negotiation through Internet-based auctions. Technical Report, University of California at Berkeley, 1996.
- [CFE01] CF Economic Consulting, Cost-Benefit and impact analysis.  
<http://www.netheaven.com/~faugere/netconsl/fropr.htm>. 2001.
- [CLA94] Clarke, R. A., Computer matching by government agencies: The failure of cost/benefit analysis as a control mechanism.  
<http://www.anu.edu.au/people/Roger.Clarke/DV /MatchCBA.html>, 1994.
- [COM01] Commonce One, Inc., Commerce one e-business solutions.  
<http://www.commerceone.com/solutions/>, 2001.
- [COS01] The Costbenefit Company <sup>TM</sup>, Cost\$Benefit analysis tool.  
<http://www.costbenefit.com/index.htm>, 2001.
- [DEP92] Department of Finance, Introduction to cost-benefit analysis for program managers. Department of Finance, Canberra, 1992.
- [DEP93] Department of Finance, Value for your IT dollar: Guidelines for cost-benefit analysis of information technology proposal. Department of Finance, Canberra, 1993.
- [DRU87] Drummond, M., Stoddart, G. and Torrance, G., Methods for the economic evaluation of health care programmes. Oxford Medical Publications, New York, 1987.
- [DUJ75] Dujmovic, J. J., Extended continuous logic and the theory of complex criteria. Journal of Belgrade, Series on Mathematics and Physics, 537:197-216, 1975.

- [ECK58] Eckstein, O., Water resource development: The economics of project evaluation. Harvard University Press, Cambridge, MA, 1958.
- [ELI93] Elixhauser, A., Luce, B. R., Tylor, W. R. and Reblando, J., Health care cost-benefit and cost effectiveness analysis (CBA/CWA) from 1979 to 1990: A bibliography. Medical Care, 31(7): JS1-JS11, 1993.
- [GAO86] General Accounting Office (GAO), Computer matching: Assessing its costs and benefits. General Accounting Office, GAO/PEMD-87-2, 1986.
- [GOL96] Gold, M. R., Siegel, J. E., Russell, L. B. and Weinstein, M. C., Cost-effectiveness in health and medicine. Oxford University Press, New York, 1996.
- [HAM00] Hammer, J., Huang, C., and Huang, Y., The IDEAL approach to Internet-based negotiation for e-business. In Proceedings of the Sixteenth International Conference on Data Engineering, San Diego, CA, pp. 666, 2000.
- [HAN95] Hanley, N. and Spash, C. L., Cost benefit analysis and the environment. Edward Elgar Publishing Limited, Aldershot, GB, 1995.
- [HUA00] Huang, C., A web-based negotiation server for supporting electronic commerce. Ph.D. Dissertation, Department of Computer and Information Science and Engineering, University of Florida, 2000.
- [I201] i2 Technologies, Inc., Supply chain management, <http://www.i2.com>, 2001.
- [IND01] Indent Innovative Technologies, Inc., Cost-benefit analysis of supply chain alternatives through simulation. <http://www.indent.org/projects.htm>, 2001.
- [KUM98] Kumar, M. and Feldman, S., Business negotiation on the Internet. IBM Institute for Advanced Commerce (IAC) Report, <http://www.ibm.com/iac/reports-technical/reports-bus-neg-internet.html>, 1998.
- [LIU01] Liu, Y., Pluempitiwiriyaewej C., Yuan, S., Su, S.Y.W. and Lam, H. A rule warehouse system for knowledge sharing and business collaboration. Technical Report, UF CISE TR01-006, <http://www.cise.ufl.edu/tech-reports/tech-reports/tr01-abstracts.shtml>, University of Florida, 2001.
- [OAG01] Open Applications Group, Open applications group integration specification. [ftp://ftp.openapplications.org/openapplications.org/oagis/rls71/oagis\\_release\\_7.1\\_pdf.zip](ftp://ftp.openapplications.org/openapplications.org/oagis/rls71/oagis_release_7.1_pdf.zip), 2001.
- [PHI00] Phillips, C. and Meeker, M., The B2B Internet report collaborative commerce. Morgan Stanley Dean Witter, New York, 2000.

- [PRE92] Press, W. H., Teukolsky, S.A., Vetterling, W. T. and Flannery, B P., Numerical recipes in C: The art of scientific computing. 2<sup>nd</sup> Cambridge University of Press, New York, 1992.
- [ROS01] RosettaNet, RosettaNet partner interface processes. <http://www.rosettanet.org/rosettanet/Rooms/DisplayPages/LayoutInitial?Container=com.webridge.entity.Entity%5B0ID%5B279B86B8022CD411841F00C04F689339%5D%5D>, 2001.
- [SIE01] Siebel Systems, eBusiness architecture. <http://www.siebel.com/products-solutions/architecture.html>, 2001.
- [SSA90] Social Security of Administration, Guide for cost/benefit analysis of SSA computer matches. Office of the Chief Financial Officer, Office of Program and Integrity Reviews, Social Security Administration, Washington, DC, March, 1990.
- [STE84] Stein, J. A., Swisher, J. D., Hu, T. W. and McDonnell, N., Cost-effectiveness evaluation of a channel one program. Journal of Drug Education, 14(3):251-269, 1984.
- [SU87] Su, S. Y. W., Dujmovic, J., Batory, D. S., Navathe, S. B. and Elnichi, R., A cost-benefit decision model: Analysis, comparison, and selection of data management System. ACM Transactions on Database Systems, 12(3): 472-520, 1987.
- [SU00a] Su, S. Y. W. and Lam, H., Iknet: Scalable infrastructure for achieving internet-based knowledge network. In Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, L'Aquila, Italy, 2000.
- [SU00b] Su, S. Y. W., Huang, C. and Hammer, J., A replicable web-based negotiation server for e-commerce. Proceedings of the Hawaii International Conference on Systems Sciences, Minitack on "Evolution of Business-to-business Electronic Commerce," Maui, Hawaii, January 4-7, 2000.
- [SU01] Su, S. Y. W., Huang, C. and Hammer, J., An internet-based negotiation server for e-commerce. To appear in the VLDB Journal, 2001.
- [TSV97] Tsvetovatyy, M., Gini, M., Mobasher, B. and Wieckowski, Z., MAGMA: An agent-based virtual market for electronic commerce. Journal of Applied Artificial Intelligence, special issue of Intelligent Agents, 11(6): 501-524, 1997.

- [WER98] Werthamer, L. and Chatterji, P., Preventive intervention cost effectiveness and cost benefit (literature review). <http://www.nida.nih.gov/HSR/dapre/WerthamerPreventive.htm>, 1998.
- [WUR98] Wurman, P., Wellman, M. and Walsh, W., The Michigan internet auctionBot: A configurable auction server for human software agents. Proceedings of the Second International Conference on Autonomous Agents, pp. 301-308, Minneapolis, MN, May 1998.
- [WYL83] Wylie, W. E., Cost-benefit analysis of a school health education program: One method. *Journal of School Health*, 53(6): 371-373, 1983.

## BIOGRAPHICAL SKETCH

Fahong Yu was born in Lanzhou, Gansu province, China, and finished grade school in this city. Fahong graduated from Mingqing High School in 1979 and completed his teaching certification program from a professional school in 1981. By 1982 Fahong became a high school teacher in Mingqing County, Gansu province.

In 1984, the Northwest Normal University accepted him as an undergraduate student and he received his B.A. in 1988. From 1989 to 1992, as a graduate student, Fahong studied the “Evolution of Primates” in Kunming Institute of Zoology (KIZ), the Chinese Academy Science, with Professor Yanzhang Peng. In May 1992, he graduated with an M.Sc. in biology from KIZ. Then he continued his research in biology as an assistant researcher in KIZ.

In 1996, Fahong left KIZ to pursue his doctoral studies at the University of Florida, US. In the summer of 1999, when he became a Ph.D. candidate in zoology, he was also accepted as a graduate student at the Department of Computer and Information Science and Engineering (CISE). In 2000, he started his research with Dr. Stanley Su. In December 2001, he received his Master of Science from the Computer and Information Science and Engineering Department of the University of Florida.