A DECISION SUPPORT MODEL FOR COST AND ACTIVITY-BASED PERFORMANCE MEASUREMENT IN STEEL CONSTRUCTION

By

AIYIN JIANG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2005

UMI Number: 3177984

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This dissertation is dedicated to my family.

ACKNOWLEDGMENTS

It is impossible to adequately thank my long-time advisor and mentor, Dr. R. Raymond Issa, for his tremendous guidance, considerable patience and encouragement needed for me to proceed through the study and complete this dissertation. His rich experience in the construction management area, his thoughtful insights, and his great sense of humor have made my Ph.D. studies very rewarding.

Dr. Joseph Geunes deserves many thanks. His lecture on supply chain design and management introduced me to numerical analysis applications in the supply chain management area. I also like thanking Dr. Ian Flood and Dr. Robert Cox for their discussions, suggestions, and encouragement during the development of this dissertation. It is a great honor to have them serve on my committee.

I would like to thank Dr. William O'Brien for his past helpful suggestions. I must express my sincere gratitude particularly to Mark Danso-Amoako for his help in interviewing the steel fabricators, and contractors.

I would also like to thank my family for their endless love, support, encouragement, and understanding in dealing with all of the challenges that I have faced.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A DECISION SUPPORT MODEL FOR COST AND ACTIVITY-BASED
PERFORMANCE MEASUREMENT IN STEEL CONSTRUCTION

By

Aiyin Jiang

May 2005

Chair: R. Raymond Issa
Major Department: Design, Construction, and Planning

Structural steel construction procedure requires special technologies and skills and it is usually undertaken by specialty contractors. The principal actors involved in the structural steel construction service include fabricators, steel mills, engineer designers, specialty shops, and erectors. The fabricator is an organizer and provider of this construction service. A fabricator usually selects its material suppliers and subcontractors based on lowest quotes or familiarity for convenience. However, uncertainties exist in the upstream material supplies and construction services. One supplier may offer the lowest quote with uncertain delivery that may incur additional costs for the fabricator. Therefore, suppliers and subcontractors selection should be based on several criteria such as pricing structure, delivery, product quality, and service. Frequently, these criteria involve trade-offs. In addition, the importance of each criterion varies from one job to the next and is complicated further by the fact that some criteria are quantitative while others are qualitative. Thus a method is needed that can adjust for the decision maker's attitude

toward the importance of each criterion and incorporates both qualitative and quantitative factors.

The goal of this study is to provide a decision-making tool for the industry practitioners to conveniently evaluate and select their material suppliers and subcontractors, forming an efficient and effective project team or a supply chain.

The objective of this study is to develop a supplier and subcontractor selection decision support model which synthesizes the quantitative and qualitative performance criteria in selecting a construction supply chain or suppliers and subcontractors, and assesses the tangible and intangible performance criteria through tracing the activities between the inter-organizations. The study applies the proposed selection decision model to structural steel construction services.

By using the model, structural steel fabricators will be able to select the appropriate supply chain and project partners and is provided non-cost-based performance data as a point for negotiation with suppliers and subcontractors. It also allows them to forecast future supplier performance based on the relevant historical data and it provides data for improving the management process by limiting non-valuable activities.

Although there are models available to solve selection problems for the manufacturing industries, a model that addresses both quantitative and qualitative performance criteria for selecting a subcontractor/supplier in the construction industry does not currently exist. The study disentangles the complicacy of subcontractor and supplier selection by using an activity-based performance and costing model and validating it through a case study.

CHAPTER 1
INTRODUCTION

Since the material, *steel*, became available at the beginning of the 19th century, it has had a fundamental effect on the whole evolution of architecture (Shulitz et al. 2000). Steel is a preferred choice for structural material as a result of its desirable properties that include high strength, uniformity, fracture toughness, ductility and elasticity. The cost of manufacturing steel is very high and makes the product more expensive than most other structural materials on the basis of price per pound. However this high cost is offset by the high strength-per-unit-weight of steel that exceeds all traditional structural material with the exception of aluminum. As a result of this property, steel is one of the least expensive materials in terms of cost per unit area of structural framing (Cooper and Chen, 1985). The structural steel construction procedure requires special technologies and skills. It is usually undertaken by specialty contractors. The quote that structural steel fabricators submit to the general contractor is an important factor considered in the general contractor's evaluation of bid proposals. Producing accurate quotes and delivering excellent construction services are critical for the fabricators to stay in business. As for the general contractor, structural steel construction cost, time, and quality seriously affects the overall performance of a project. There are many uncertainties involved in the use of the structural steel from material procurement, design, fabrication, to erection. A steel fabrication contractor has to organize its raw material suppliers and specialty partners to provide good construction services to the general contractors and to meet the requirements of construction projects. Accordingly, making the right decision in selecting its suppliers and sub-subcontractors to make sure that the upstream supply chain of products and services is not interrupted is critical for a steel fabricator.

Principal Actors in Structural Steel Construction Procurement/Services

The principal actors involved in processing steel from iron ore and scrap metal to the incorporation of the structural steel in the completed project are shown in Figure 1-1. The primary path of processing is from the mills to the erection site. This path can be taken directly or through warehouses, specialty shops and fabrication shops depending on the nature of the project.

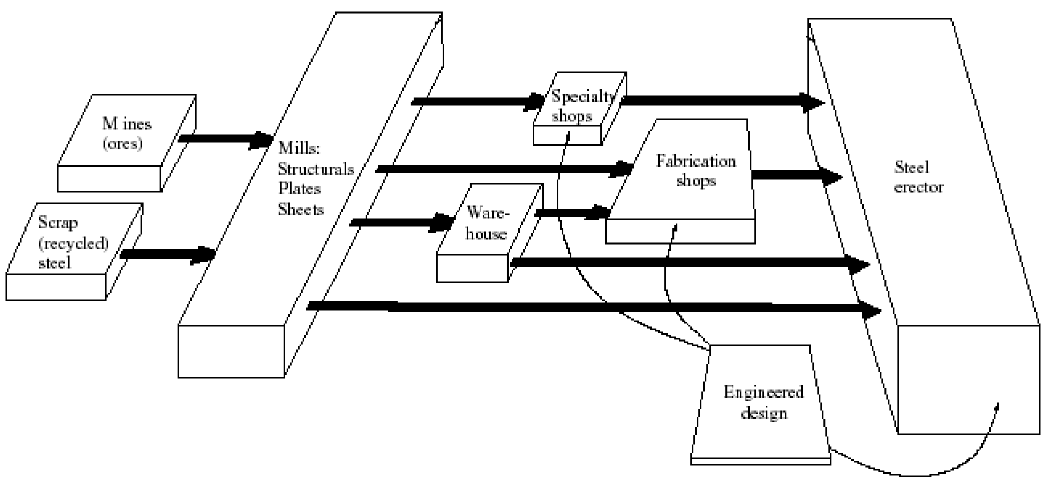


Figure 1-1. Principal Actors in Structural Steel Construction Services (Adapted from Cooper and Chen (1985)).

Steel Mills

The main sources of materials for steel mills are "scrap metal" or mined iron ore, coke and limestone. The raw materials are crushed, washed and fed into a blast furnace. After a series of chemical processes at high temperatures, molten steel is produced and this is then cast into ingots. Depending on the final intended use of the steel, the ingots are processed through special mills to produce *blooms*, *billets* or *slabs* (Marcus 1981). At this stage the structural steel members are ready to be delivered to fabricators, warehouses, specialty shops or the construction job site depending on the conditions of the project under consideration.

Fabricators

Of all the parties involved in steel construction, the fabricator has the most complex job which requires very exact and high precision processes (Engel 1988). The fabricator processes mill-delivered steel into forms ready to be assembled on site by bending, punching, welding and drilling. Depending on the type and conditions of contract, size of project and organizational structure of the fabricator, there may be slight variations to the processes.

Engineered Design

The actual design of connections between framing members is usually left to the engineers. The design of such connections can be somewhat complex. The engineered design isolates all connections that are not standard connections or are not fully detailed on the structural engineer's drawings. Connection details are submitted to the structural engineer for approval before shop detailing has substantially started. The engineers or detailers draw a picture for each piece of steel to be fabricated showing the specific work to be performed. The drawings show general configuration, hole locations, plates, connection angles, bolts, weld sizes, and so on.

Erectors

After the structural steel is fabricated to the required engineered sizes in the fabricator's shop, it must be transported to the site and then erected to form the designed project. Erection can be undertaken by the fabricators' erection crew or more typically by a specialist erection company to whom the job is subcontracted. If subcontracted the specialist erection firm has the responsibility of tying in its schedule to both that of the fabricator and that of the general contractor.

Specialty Shops

These are firms that specialize in producing complex and intricate geometric shapes and specialized treatments such as special bending, cambering and galvanizing. In order to reduce the fabricator's labor cost by self-performing such jobs, they are usually subcontracted to these firms.

Uncertainties exist at every level in the structural steel fabrication and erection process. Upstream uncertainty can be late deliveries by suppliers or poor quality of the incoming materials and parts. Looking downstream, uncertainties take the form of unforeseen demand variability, which in turn creates problems in planning, scheduling, and control that jeopardize delivery performance. Greater uncertainties in the structural steel material flow affect adversely delivery performance.

Uncertainties in Structural Steel Construction Procurement

Usually, structural steel fabricators award their supplier and subcontractor work based on the pricing or existing business relationships. The purchasing departments of steel fabricators prepare the order for each of the structural steel items after being awarded a contract from the general contractor. These orders are placed on the basis of the constraints of cost and construction schedule. The decision on where to place the order will vary on the basis of the quantity or tonnage of steel, and the types and sizes of material. Three basic options for purchasing structural steel are commonly used as:

- Rolling mill stock
- Service center (warehouse) stock
- Fabricators' stock

The rolling mills usually require a minimum order or a minimum item quantity of one “bundle.” All mills have rolling schedules showing how often they produce the various shapes. The cycles for each series vary on the basis of demand and the ability to produce and sell stock. For example, most heavy sections are rolled every 4 to 6 weeks. Some of the light beam sections are only rolled every three months. Such restrictions may mean that for a mill order for a project of any size, it may take 8 to 9 weeks for the fabricator to receive all of the necessary steel for the initial sequence. It is critical that openings are available in the next scheduled rolling, or that appropriate mill stock is available. If the rolling is fully booked, the schedule will increase by the amount of time to the next cycle with an available rolling. Fabricators may also have a problem when ordering from a mill that only rolls angles and bars, such mills are relatively specialized facilities that do not produce the full range of sizes. This often necessitates some juggling of orders. Service centers offer an alternative to mill purchases, especially for small projects and rush deliveries, however at a higher price.

The actual assembly of all the material required for a given contract is a responsibility of the fabricator. Material may have to be assembled from a number of sources, such as rolling mills, warehouses, and the fabricators’ own stock. The long material procurement time and various material supply sources aggregate to the structural steel fabrication time and cost. These uncertainties affect the structural steel erection process on the project site, and even further affect the whole construction project performance. When a supplier fails to meet delivery, quality, and price requirements, additional costs are incurred by the purchasing organization to correct these deficiencies. These excess costs, both direct and overhead, have an immediate impact on the firm’s available resources. Waste of people, equipment, and time, which all cost money, adversely affects the firm's competitive position. Therefore choosing the right suppliers and subcontractors as project partners is critical for the success of a project.

However, the supplier selection process encompasses different functions such as purchasing, quality, production, etc. within the company. It is a multi-criteria problem, encompassing many tangible and intangible factors. Frequently, these evaluation criteria involve trade-offs. One supplier may offer inexpensive parts of slightly below average quality, while another supplier may offer higher quality parts, with uncertain delivery. In addition, the importance of each criterion varies from one another and is complicated further by the fact that some criteria are quantitative while others are qualitative. Thus a method is needed that can adjust for the decision maker's attitude toward the importance of each criterion and incorporates both qualitative and quantitative factors. This research focuses on the supplier and subcontractor selection decision support in the structural steel procurement procedures. It treats the supplier selection criteria, potential alternatives, and activities involved in procurement procedures in a hierarchical manner. The hierarchical structure decomposes and synthesizes these factors to find the right suppliers and subcontractors for a specific project.

This research applies the selection decision model to structural steel construction services. By using the model structural steel fabricators will be able to select their appropriate supply chain and project partners and they are also provided non-cost-based performance data as a point for negotiation with suppliers and subcontractors. This research also provides data for improving the management process by limiting non-valuable activities. Models exist to solve some of the selection problems for the manufacturing industries. However, a model that addresses both quantitative and qualitative performance criteria for selecting a subcontractor/supplier in the construction industry does not currently exist. The research disentangled the complications encountered in the selection process based on an activity-based performance and costing model and applies it to the case of structural steel services in order to facilitate the decision-making process.

Organization of This Study

This Chapter introduces the necessities of the study. Chapter 2 will present a literature review of existing selection criteria and models/methods such as Analytical Hierarchy Process, Activity-based Costing, and Total Cost of Ownership. Based on the results of the literature review, Chapter 3 presents the research objectives. Chapter 4 discusses the objectives, problems, significance, and limitations of this research. The activity-based performance and costing model is explained in details in the Chapter 5. A case study in structural steel fabrication is conducted and the proposed model is applied and described in the Chapter 6. The model involves a lot of data analysis and calculations and the system components, structure, and the underlying database model are introduced in Chapter 7. Chapter 8 gives the conclusions of this research and suggests future areas of research.

CHAPTER 2
RESEARCH METHODOLOGY

Construction draws on a wide variety of established subjects, including natural sciences, social sciences, engineering and management, and applies them to its particular context and requirements. Only by use of appropriate methodologies and methods of research, applied with rigor, can the body of knowledge for construction be established and advanced with confidence (Fellows and Liu, 1997).

This research was divided into several major phases such as identifying the problems in selecting the right subcontractors/suppliers to form an effective and efficient supply chain, setting the research aim, a literature review, establishing research objectives, proposing activity-based performance and costing decision making-model, conducting case studies and data analysis, computerization of the proposed decision-making model, and conclusions and recommendations. Figure 2-1 illustrates the research methodology using a flow chart.

Literature Review

This stage of the research is to collect the information done by other researchers in selection decision-making and supply chain design problems. The following are the categories of literature review:

- Selection problem in supply chain management.
- Performance measurements in supply chain management.
- Performance management in the manufacturing industries and the construction industry.

- Methods and models about selection decision-making and supply chain design problems.

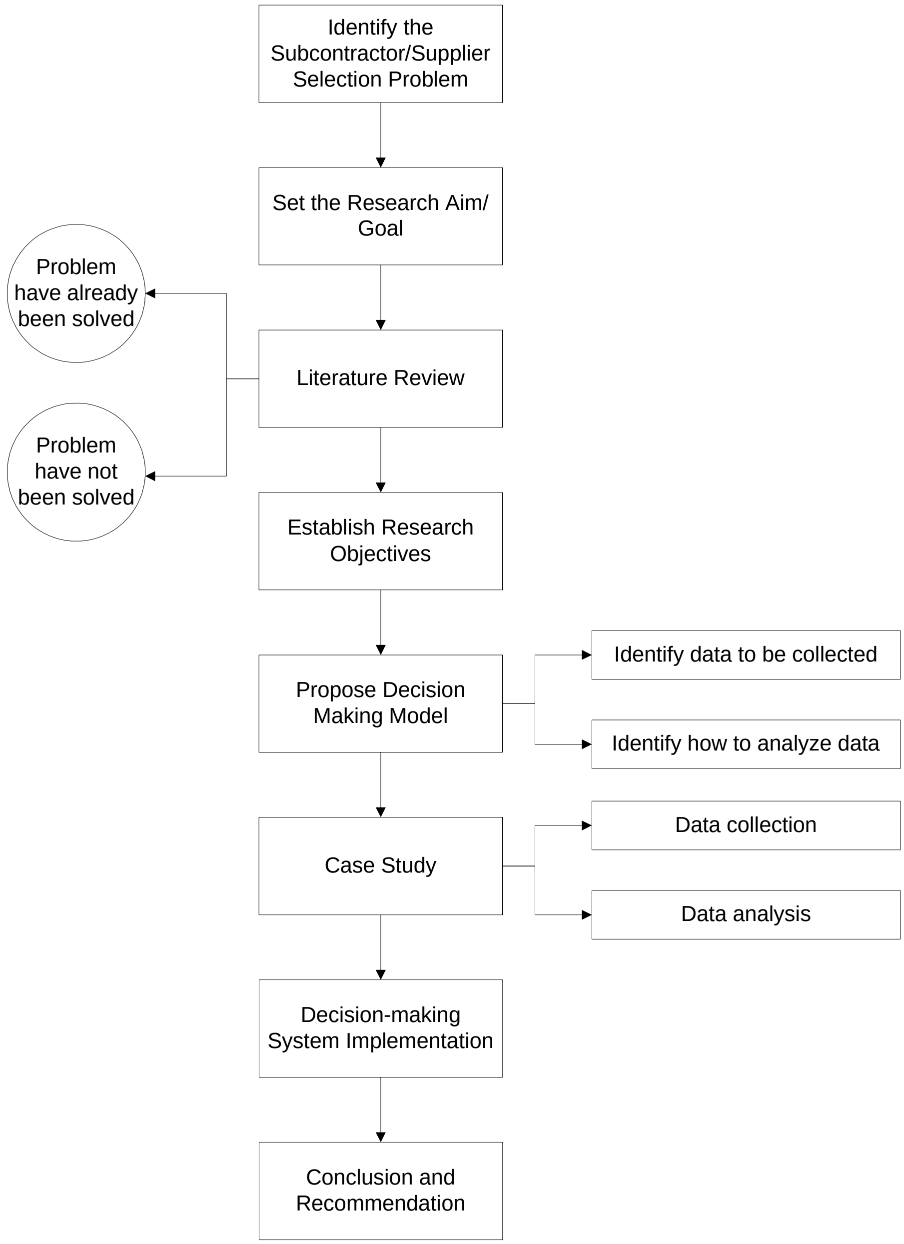


Figure 2-1. Research Methodology Flow Chart

The tasks of this stage are through the literature review to

- Categorize supply chain performance measures

- Compare existing decision-making models of supply chain management

Research Objectives

The literature review stage identifies which problems have been solved and which problems have not been solved. Based on the research aims set and literature review, research objectives are established and research significances and contribution are identified.

Propose Decision-making Model

In this stage, a hierarchically structured decision-making model is proposed to solve the selection problems in subcontractors and suppliers in order to form an efficient and effective supply chain in the construction services. The model considers cost impact and non-cost performance impact on the selection problems. It measures the overall performance of the subcontractors and suppliers based on the activities. Basically it treats the supplier selection criteria, potential alternatives, and activities involved in procurement procedures in a hierarchical manner. The hierarchical structure decomposes and synthesizes these factors to find the best suppliers and subcontractors for a project.

Case Studies

Case studies encourage in-depth investigation of particular instances within the research subject (Fellows and Liu, 1997). The case study stage is to collect data. This research conducts structural steel construction services by applying the proposed decision-making model. The case study interviewed the structural steel fabricators and its suppliers and subcontractors to collect data. These data include

- primary business processes and activities performed in the steel construction services
- subcontractor and supplier selection criteria

- records about the performances of subcontractors and suppliers on existing and past projects.

The case study categorizes and analyzes these collected data and provides the choices by applying the proposed model and sensitivity analysis.

Decision-making System Implementation

The decision-making process involves a huge amount of data analysis. Therefore computerizing this process will relieve the management from data burden. This stage will figure out the structure of the system and implement this system.

Conclusions and Recommendations

The last stage of this research discusses the conclusions of the research, pointing out its limitation and makes recommendations for future research on the selection decision-making process in construction.

CHAPTER 3
LITERATURE REVIEW

Subcontractor Selection and Construction Supply Chain Management

Appropriate supplier and sub-contractor selection is a vital element on construction projects, since a high degree of specialization has evolved in the provision of various goods and services in the construction industry and a large proportion of construction activities may be subcontracted on a given project. However, improvements in the subcontractor selection processes have not received the critical attention needed. Price-based selection often squeezes out the more responsible subcontractors, driving down both prices and performance levels (Kumaraswamy et al; 2000). The importance of smarter project formulation is to begin with intelligent choices in the overall subcontractor procurement system design. It should be related to the project scenario rather than based on familiarity or convenience. The selection of project participants are critical procurement aspects that should be tailored to match project objectives and business goals. This also can in turn meet the strident demands for dramatic productivity increase in the construction industry (Kumaraswamy et al. 2000).

Probing deeper down the selection problem, it is a supply chain design and analysis problem. A deeper view of construction service procurement would draw attention to the supply chains of both goods and services (Kumaraswamy et al. 2000). Integrated supply chain management encompasses all activities associated with the flow and transformations of products from the raw materials stage through delivery to the final consumer. To achieve competitive advantages firms need to emphasize outsourcing in a way that adds value to the supply chain as a whole. A good supplier is a major component of this value creation, hence making the supplier selection decision critical (Bhutta and Huq 2002).

Supply chain management (SCM) has become a popular concept in the construction industry and research community since the 1990s. Effective integration and optimization of supply chains can have a tremendous, positive impact on project schedules, delivery time from concept development to turn-over, costs, customer satisfaction, and ultimately the bottom-line success of each project as well as the long-term success of every participant in the supply chain. Companies not engaging in SCM may find themselves falling rapidly behind in performance relative to their supply-chain competitors (Tommelein et al. 2003). A network of supply chains includes multiple layers of subcontractors and interlinked suppliers. Selection problems exist on each layer of supply chain. General contractors face the selection problems for choosing steel fabricator, concrete subcontractor, electricity subcontractor, etc. The subcontractors face the selection problems for having raw material suppliers, specialty treatment partners, etc. Multiple layers of subcontracting extend this chain even further and add variety to the construction supply chains. Selection methodologies of various project supply chain participants and decisions are critical to improving construction project performance. More proactive subcontractor procurement would enhance both project performance and supply chain performance.

Supply Chain Performance Measures

An important component in supply chain design and analysis is the establishment of appropriate performance measures. Performance measures are used to design a supply chain by determining the values of the decision variables that yield the most desirable levels of performance. A set of performance measures is used to determine the efficiency and effectiveness of a supply chain, or to compare competing alternative supply chains. Most available literature about performance measures is from the manufacturing industry and it identifies a number of performance measures in the evaluation of supply chain effectiveness and efficiency.

Carman and Conrad (2000) indicated that the supply chain should put the customer first. They proposed to use Key Performance Indicators (KPIs) to provide business organizations with the yardstick that indicates whether activities are meeting the needs of customers. They defined four dimensions of customer satisfaction, product quality, delivery service, product variety, and competitive pricing. Chopra and Meindl (2001) indicated that customer demand may vary along five attributes: product quality, response time, product variety, service level, product pricing and product innovation rate. Hugos (2003) provided performance measurements in another way. He defined four measurement categories to measure supply chain performance. They are: (1) Customer Service: measures the ability of the supply chain to meet the expectations of its customers; (2) Internal Efficiency: refers to the ability of a company or a supply chain to operate in such a way as to generate an appropriate level of profitability; (3) Demand Flexibility: measures the ability to respond to uncertainty in levels of product demand; (4) Product Development: encompasses a company and a supply chain’s ability to continue to evolve along with the markets it serves. Hugos (2003) defined two sets of customer service metrics depending on whether the company or supply chain is in a build to stock (BTS) or build to order (BTO) situation.

Beamon (1999) proposed a framework for supply chain performance measurements. The framework involved three main key elements that include the measurement of resources, output and flexibility. Each of the three types of measures has important characteristics and the measure of each of these affects the others. Resource measures include inventory levels, personnel requirements, equipment utilization, energy usage, and cost. Resources are generally measured in terms of the minimum requirements (quantity) or a composite efficiency measure. Efficiency measures the utilization of the resources in the system that is used to meet the system's objectives. One general goal of supply chain analysis is resource minimization. Some examples of supply chain resource performance measures are the following: (1) total cost; (2) distribution Cost; (3) manufacturing cost; (4) inventory; (5) return on investment (ROI). Output measures include customer responsiveness, quality, and the quantity of final product produced. Some output performance measures are represented numerically such as (1) number of on-time deliveries; (2) and time required to produce a particular item or set of items. However there are many output performance measures that are difficult to express numerically such as customer satisfaction. Flexibility measures a system's ability to accommodate volume and schedule fluctuations from suppliers, manufacturers, and customers. Flexibility is vital to the success of the supply chain since the supply chain exists in an uncertain environment. The interrelationship among these three types of measures is illustrated in Table 3-1.

Performance measurements analysis in the construction industry is usually related to productivity. A classical definition of productivity is a comparison of the output of a production process to its corresponding input, i.e., the output to input ratio. It is usually

Table 3-1. Performance Measure Types

Performance measure type	Goal	Purpose
Resources	High level of efficiency	Efficient resource management is critical to profitability
Output	High level of customer service	Without acceptable output, customers will turn to other supply chains
Flexibility	Ability to respond to a changing environment	In an uncertain environment, supply chains must be able to respond to change

(Source: Beamon, 1999)

measured to control project cost and schedule. Productivity is also defined as work-hours required per unit of work. It is the input divided by the output and is calculated for a finite time interval. The reason for this appears to be that typically the construction industry places paramount importance upon costs during both estimating and project execution (Thomas and Mathews 1986). The construction industry commonly tracks this change in progress in terms of work units completely attained during a given period of time and the associated costs in terms of man-hours or dollars (Thomas and Mathews 1986). The process of productivity measurement is to quantify the work-hours and completed quantities. Usually this process is associated with an activity or cost account. Traditionally productivity measurement involves the collection of information about various activities or cost accounts. These data can be examined to determine if productivity is improving or declining.

Cox et al. (2003) conducted a survey of management's perception of key performance indicators (KPIs) for construction. They defined the KPIs as compilations of data measures used to assess the performance of a construction operation. These evaluations typically compare the actual and estimated performance in terms of effectiveness, efficiency, and quality in terms of both workmanship and product. Their research indicates the need for seeking constant monitoring to improve the reporting methods. Those firms in determining construction performance should use the KPIs in conjunction with those indicators that monitor the internal objectives of their own company. Construction professionals need to better monitor and control their organization's performance at both the field level and office level (or both operation level and strategic level). Quantitative KPIs determined by the survey included:

- Total Cost
- Quality Control/Rework
- On-time
- Safety
- $/Unit
- Units/Man Hour

Differences in KPIs were found based on individual level of management. Project managers typically maintain a project level focus while executives tend to have a company-wide focus. However, supply chain performance evaluation is a more comprehensive analysis than productivity. Productivity is only one part of performance analysis. It is more focused on the internal control and management of an organization.

Through the performance literature review in both the manufacturing and the construction industries, the performance measures may be categorized as cost based measures which are quantitative and non-cost based measures which are either qualitative or quantitative.

Measures Based on Cost

Measures based on cost are quantitative performance measures that may be directly described numerically. These measures are objectives that are based directly on cost or profit.

- Cost minimization: the most widely used objective. Cost is typically minimized for an entire supply chain (total cost), or is minimized for an entire supply chain (total cost), or is minimized for particular business units or stages.

- Sales maximization: Maximize the amount of sales dollars or units sold.

- Profit maximization: Maximize revenues less costs.

- Inventory investment minimization: Minimize the amount of inventory costs (including product costs and holding costs).

- Return on investment maximization: Maximize the ratio of net profit to capital that was employed to product that profit.

Non-Cost Based Performance Measures

Non-cost based performance measures are those measures for which there is no single direct numerical measurement, although some aspects of them may be quantified.

- Customer satisfaction/customer service/service level/delivery service: The degree to which customers are satisfied with the product and service received, and may apply to internal customers or external customers. Customer satisfaction is comprised of three elements (Christopher 1994)

1. Pre-transaction satisfaction: satisfaction associated with service elements occurring prior to product purchase.

2. Transaction satisfaction: satisfaction associated with service elements directly involved in the physical distribution of products.

3. Post-transaction satisfaction: satisfaction associated with support provided for products while in use.

- Flexibility: The degree to which the supply chain can respond to random fluctuations in the demand pattern.

- Information and material flow integration (Nicoll 1994): the extent to which all functions within the supply chain communicates information and transport materials.

- Effective risk management (Johnson and Randolph 1995): All of the relationships within the supply chain contain inherent risk. Effective risk management describes the degree to which the effects of these risks are minimized.

- Supplier performance (Beamon 1998): With what consistency suppliers deliver raw materials to production facilities on time and in good condition. These measures may be presented as:

a. Fill rate maximization: Maximize the percentage of customer orders filled on time

b. Product lateness minimization: Minimize the amount of time between the promised product delivery date and the actual product delivery date.

c. Customer response time minimization: Minimize the amount of time required from the time an order is placed until the time the order is received by the customer. Usually refers to external customer only.

d. Lead time minimization: Minimize the amount of time required from the time a product has begun its manufacture until the time it is completely processed.

The performance measures presented above are important characteristics of a supply chain. However, their use in supply chain models is challenging, since the qualitative nature of such non-cost based performance measures makes them difficult to incorporate into quantitative models.

Analytic Hierarchy Process: A Method/Model for Non-cost-based Performance Measures

Analytic Hierarchy Process (AHP) is an excellent approach that can be used in a multifactor decision-making environment, and especially when subjective and intuitive consideration has to be incorporated. It provides a framework to cope with multiple criteria situations involving intuitive, rational, qualitative and quantitative aspects. It provides a structured approach for determining the scores and weights for each of the multiple criteria used, and it standardizes them so that they can be compared and decision is made.

AHP was developed by Saaty and published in his 1980 book, *The Analytic Hierarchy Process*. After its introduction in 1982, AHP has been widely used in many applications enabling decision-makers to represent the interaction of multiple factors in complex and unstructured situations. AHP is used as a framework to formulize the evaluation of trade-offs between the conflicting selection criteria associated with the various suppliers' offers (Nydick and Hill 1992; Render and Stair 2000). When a supplier selection decision has to be made, the buyer generally establishes a set of evaluation criteria, and the AHP process makes use of these criteria to help make the decision.

AHP is said to be a successful theory, because its assumptions are consistent with available experimental data, it makes testable predictions based on experiments, and it explains behavior. This is the main reason for selecting AHP as the decision support method or model for solving the supplier selection problem, which involves many intangible factors, but still requires a logical and rational control of decisions. It first structures the problem in the form of a hierarchy to capture the basic elements of a problem and then derives ratio scales to integrate the perceptions and purposes into a synthesis. Generally the hierarchy has three levels: the goal, the criteria, and the alternatives. For the supplier selection problem, the goal is the best supplier, the criteria could be quality, on-time delivery, prices, etc. and the alternatives are the suppliers or proposals of the suppliers (William et al., 2001). In the hierarchical structure, all the elements in a level are pair-wise compared with respect to the elements in the level above, and paired comparisons are used to elicit judgments. Then the synthesis of judgments is obtained as a result of hierarchic "re-composition" in order to find the best decision. Figure 3-1 shows the AHP hierarchic structure.

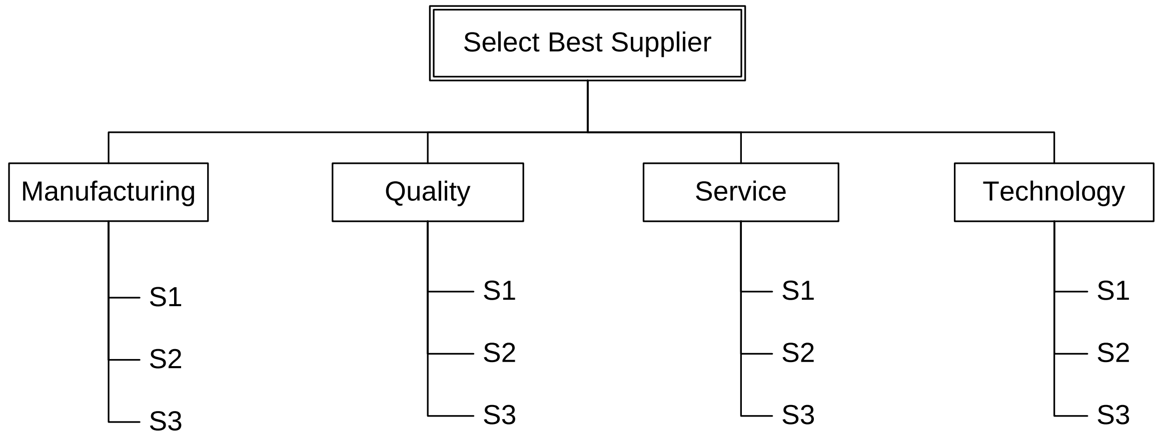


Figure 3-1. Hierarchy of Decision (Source: Bhutta and Huq 2002)

The AHP decision process requires the decision-maker to develop a hierarchical structure of the factors in the given problem and to provide judgments about the relative importance of each of these factors and ultimately to specify a preference for each decision alternative with respect to each factor. Managerial judgments are used to drive the AHP approach by assigning weights to different criteria and the alternative with the highest total weighted score is selected as the best. Although there are many scales that can be used for quantifying managerial judgments, the scale shown in Table 3-2 is commonly used for AHP analysis.

Table 3-2. Measurement Scale

Verbal judgment or preference	Numerical rating
Extremely preferred	9
Very strongly to extremely preferred	8
Very strongly preferred	7
Strongly to very strongly preferred	6
Strongly preferred	5
Moderately to strongly preferred	4
Moderately preferred	3
Equally to moderately preferred	2
Equally preferred	1

(Source: Bhutta and Huq, 2002)

The AHP offers a methodology to rank alternatives based on the decision maker's judgments concerning the importance of the criteria and the extent to which they are met by each alternative. This method allows the decision maker to structure complex problems in the form of a hierarchy. Managerial judgments are used to drive the AHP approach (Saaty 1980). These judgments are expressed in terms of pair-wise comparisons of items on a given level of the hierarchy with respect to their impact on the next higher level. Pair-wise comparisons express the relative importance of one item versus another in meeting a goal or a criterion. Each of pair-wise comparisons represents an estimate of the ratio of the weights of the two criteria being compared. Because AHP utilizes a ratio scale for human judgments, the alternative weights reflect the relative importance of the criteria in achieving the goal of the hierarchy. For this reason, AHP is ideally suited for the supplier selection problem. (Partovi et al. 1989).

Advantages and limitations of AHP: AHP is used for evaluating the sources of supply in a materials management situation. AHP can help managers in formulating decisions under the following scenarios (Bhutta and Huq 2002):

- Analyzing the impact of supply sources on multiple goals of an organization;
- Facilitating the interactive flow of inputs and evaluating the sources from a strategic perspective.

There are a variety of extensions to the AHP approach, which can increase its usefulness for managerial decision making. First, the AHP is a flexible modeling tool that accommodates a larger set of evaluation criteria. The criteria can be compared pair-wise first and then the individual criteria can be pair-wise compared within each category. In this way, a larger number of criteria can be included within the hierarchy without generating an extremely large pair-wise matrix. Similarly other extensions, including calculation of consistency indices and ratios have been suggested and used by various researchers; however, the core procedure remains identical across these extensions (Sun 2001; Render and Stair 2000).

On the other hand, several criticisms have also been made of this approach, some of them claim that like other judgmental techniques, this process is driven by judgments of the decision maker and there is no independent (analytic) way of verifying the results. All criteria are relative, so no absolute measures can be given to them and, when a new criterion is added, then the whole process has to be repeated (Bhutta and Huq 2002).

Methods for Developing Supply Chain Cost-Based Performance Measures

Supply chain costing provides a mechanism for developing cost-based performance measures for the activities comprising the key processes within the supply chain. The capabilities provided by supply chain costing include the ability to:

- determine the overall effectiveness of the supply chain,
- identify opportunities for further improvement or reengineering,
- measure performance of individual activities or processes,
- evaluate alternative supply chain structures or select supply chain partners,
- evaluate effects of technology improvements.

Supply chain costing also differs by including transaction, information, physical flow, and inventory carrying costs. Two of the popular methods employed by supply chain costing are Activity-based Costing (ABC) and Total Cost Ownership (TCO). They differ by costing activities across the entire supply chain. Total cost of ownership and activity-based costing provide useful information but do not satisfactorily address the entire supply chain.

Activity-based Costing (ABC)

Activity-based costing (ABC) emerged during the 1980s as a means to more accurately assign costs within an organization (Cooper 1989). ABC is a technique for assigning the direct and indirect costs of an organization to the activities consuming the organization's resources and then subsequently tracing the costs of performing these activities to the products, customers, or distribution channels consuming the activities (Cooper 1988).

Activity-based costing is a methodology that measures the cost and performance of activities, resources, and cost objects. Resources are assigned to cost objects based on their use. Activity-based costing recognizes the causal relationship of cost drivers to activities (Raffish and Turney 1991). An ABC model based on this definition has two main views. The first is the cost assignment view, which is the vertical part of the model shown in Figure 3-2. It reflects the need that organizations have to assign costs to activities and cost objects to analyze critical decisions. These decisions have to do with issues such as:

- Pricing,
- Product Mix,
- Sourcing,
- Product Design,
- Setting priorities for improvement efforts.

The second part of the ABC model is the process view, which is the horizontal part of the model shown in Figure 3-2. The process view reflects the need that organizations have for a new category of information, information about what causes work, and how well that work is done. Organizations use this type of information to help improve performance and to increase the value received by customers (Turney 1992).

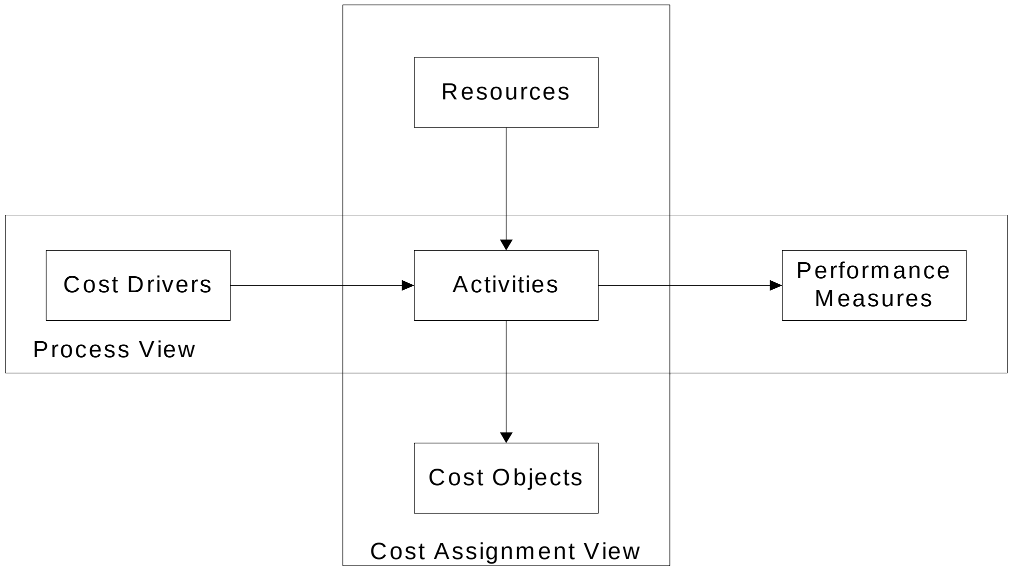


Figure 3-2. Cost Assignments and Process View for Activity-based Costing and Management (Source: Turney 1992)

Firms using ABC can obtain more accurate information of how specific products, customers, or supply chains affect costs and contribute to overall profitability (Pohlen and Bernard 1994; Fuller et al. 1993). ABC has gained considerable attention as a potential tool for evaluating supply chain performance. Firms can also use ABC to evaluate how the performance of other supply chain members drive their logistics costs and affect overall profitability (Tyndal 1990). Costs may vary based on factors such as cycle time, on-time delivery, promotional versus regular sales, type of customer, or order accuracy (Ellram 1994).

A framework for estimating ABC should have the following features:

- It should be hierarchical in relating to overheads to activities and then to cost objects.
- It should be analytic in nature allowing the analyst to integrate available quantitative measures of operations as well as non-quantitative comparative estimates of various cost drivers, activities and products.

- It should provide a tracking process for retracing the various steps and it should perform a comprehensive sensitivity analysis to investigate changing cost drivers or activities.
- The approach should be simple and flexible enough to be acceptable by many decision makers with possibly different points of view regarding selection of activities, cost drivers and their measures.

Activity-based Costing and Analytical Hierarchy Process: Because of the hierarchy features in both Activity-based costing (ABC) and Analytical Hierarchy Process (AHP), the two methods are integrated to be a decision making model. This kind of model is in the form of a hierarchy that includes two stages of ABC and AHP-based hierarchy. Partovi (1991) presented an estimating model for determining the overhead cost allocation to different products (Figure 3-3). The top of the hierarchy corresponds to the first stage of ABC. In this stage, total overhead costs are allocated to an organization's major activities using traditional overhead classifications such as purchasing, maintenance and rent as cost drivers. In the lower half of Figure 3-3, there are several hierarchies that correspond to major activities. The function of these hierarchies is to trace the activities' overhead to different cost objects (products) using appropriate cost drivers.

The objective of the model is shown at the highest level of the hierarchy in Figure 3-3 and the allocation of overhead costs to products is the objective. At the second level, various cost drivers for allocating overhead to activities are shown. Allocation of overhead with respect to cost drivers can be different for each activity. Activities are shown in the third level of the hierarchy. Identifying these activities is important because the cost of resources consumed is related to each product through these activities. Sometimes gathering the information about the consumption of each activity by each product can be prohibitively expensive. In others, by combining activities, the need to measure and track their individual performances is eliminated. At the fourth level of the hierarchy, a second set of cost drivers is presented linking activities to products. Each of the activities has a set of cost drivers, and those with the same cost drivers often have different overhead weights. The lowest level in the hierarchy of Figure 3-3 represents specific products, each of which makes use of the facilities of each activity. Once the activities, cost drivers, and products are defined, the evaluation process can proceed using the established AHP procedures. At that point the values of most cost drivers are known and managers need only determine their ratios at the second level. At the third level, allocating overhead costs can be accomplished by using combinations of quantitative as well as judgmental data to determine the significance of each activity in comparison to

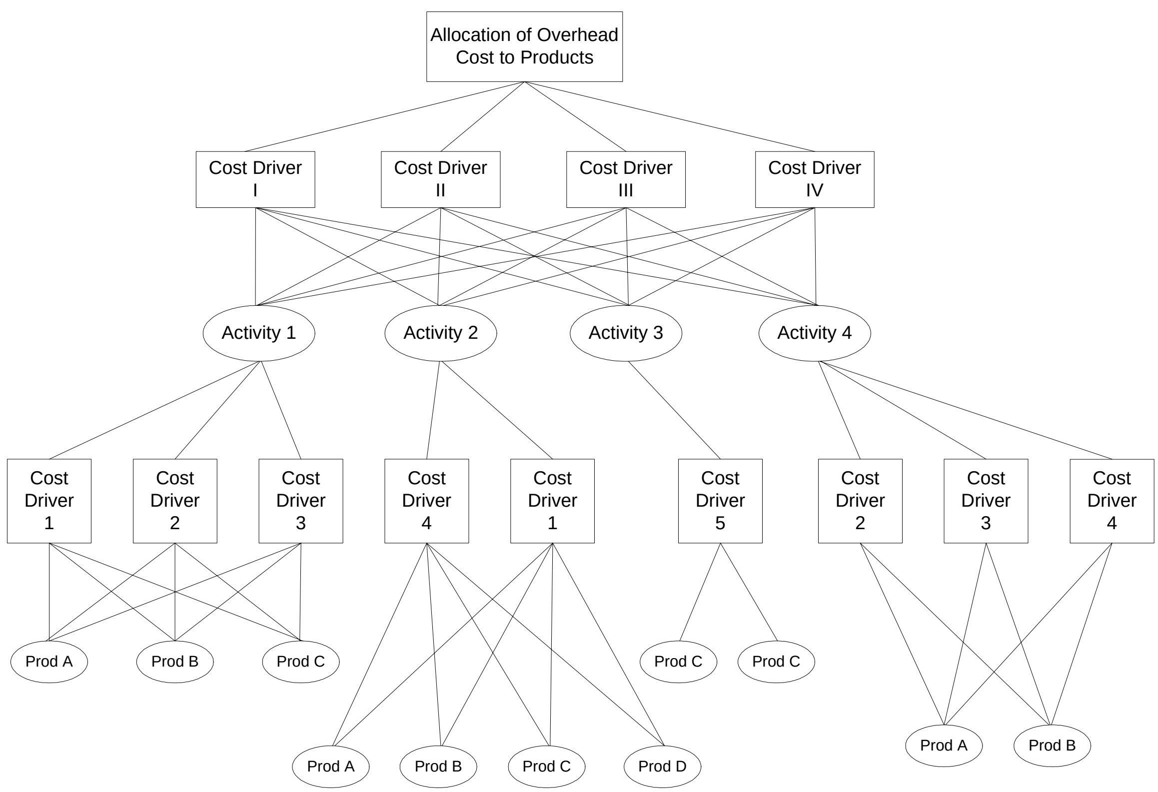


Figure 3-3. Hierarchical Structure of Overhead Allocation to Product (Source: Partovi 1991)

other activities with respect to a particular cost driver. Once the overhead costs have been allocated to activities, the next step is to find the importance of the second set of cost drivers with respect to each activity. Finally, a set of pair-wise comparisons evaluates different products with respect to the second set of cost drivers. Pair-wise comparisons at all levels of the hierarchy are required only if quantifiable measurements are not available.

The study proposes a model, which is based on ABC and incorporates Saaty's(1980) AHP, to estimate the overhead costs associated with each product. The proposed model uses overhead cost categories as well as subjective judgments by managers for determining specific overhead costs when "hard" data is not available. The proposed model contributes a rational and inclusive model for allocating overhead to different products, without requiring extensive cost information. The proposed model provides managers with a yardstick to help estimate if it makes any difference in overhead allocation if ABC is implemented. Decision makers may prefer this model for sensitivity analysis. By introducing various cost drivers, and observing their effect on overhead allocation, the decision makers will have a firmer base upon which to make decisions about selecting the appropriate cost drivers for implementing ABC in a particular setting.

Despite the advantages provided by ABC, the methodology does not provide a satisfactory solution to efficient supply chain management. Most research on ABC focuses on internal costs rather than the total supply chain. One study that looked at companies in a variety of manufacturing and service industries found that ABC management applications tended to focus within individual firms and looked at purchasing and procurement, operations, marketing and selling, distribution, and general and administrative expenses. These internal applications provide valuable information, however, they do not enable the supply chain participants to determine: where non-value-added activities may exist in the supply chain; what high cost activities or processes to target for continuous improvement or reengineering; or what the key factors driving supply chain costs are in order to strategically position logistics activities in the channel where the function can be best performed in terms of cost, time, or quality. The applications have not attempted to determine how the behaviors of individual firms have affected the total supply chain cost or the marketplace cost seen by the ultimate consumer.

Total Cost of Ownership (TCO)

Total cost of ownership (TCO) is a phrase used to describe “all costs associated with the acquisition, use, and maintenance” of a good or service. The TCO examines the cost associated with purchased goods and services throughout the entire supply chain. Thus, TCO considers costs all the way from idea inception, as in working with a supplier to develop a new or improved part, through warranty claims associated with that part once the final product is in use by the customer. TCO differs in two important ways from most models that attempt to look at the “cost” of doing business with a supplier. First, TCO considers a broader spectrum of acquisition costs than do most cost of ownership systems. Second, TCO attempts to look at life cycle costs, which consider costs associated with using a given item from a given supplier during the entire life of the item, including costs incurred once the item is in use (Ellram 1993).

Benefits of TCO

Ellram (1993) generalized several major TCO benefits as:

1. performance measurement,
2. decision making,
3. communication,
4. insight/understanding, and
5. the support of continuous improvement efforts.

The “performance measurement” category of benefits includes those that improve the quantitative measurement of supplier performance. It includes such issues as the following: TCO is a good way to evaluate suppliers; TCO provides a quantitative method for measuring the results of supplier performance improvement/quality improvement efforts; and TCO provides an excellent tool for benchmarking. In benchmarking, TCO data can be used to compare suppliers, or to track changes in a supplier’s cost performance over time.

TCO also supports improved decision-making. TCO forces the quantification of tradeoffs in terms of dollars. It also provides a good basis for supplier selection decisions, because it provides complete cost data on the important cost issues. Thus, TCO creates more informed decision-making, in a structured, systematic way.

TCO can also help improve both internal and external communications for the purchasing function. The system provides solid data to communicate to suppliers regarding their performance. It also represents an important way to get others within the firm involved in purchasing decisions – by providing data, or identifying relevant cost consideration.

The depth of the TCO approach also provides important insights and deeper understanding into the true nature of supplier performance. The information developed using TCO regarding a supplier’s total costs can be used to track the supplier’s costs over time, or to compare with other suppliers. Such detailed information provides excellent data for negotiations, and can help focus target pricing efforts. TCO also helps purchasing personnel develop an awareness of the significant non-price factors that affect their firm in the case of certain buys. This insight can help in negotiations, and in determining which non-price cost elements a supplier should provide, and which can be foregone or obtained more economically elsewhere. Finally, TCO provides a better understanding of purchase decisions by taking a long-term, big picture approach. It looks beyond price to explore how purchasing activity affects the firm's total costs both today and in the future.

All of these benefits above represent proactive means for purchasing to help continuously improve some aspect of the firm's or the supplier's operations. By identifying various critical cost elements and their values, TCO helps focus a supplier's efforts on improving the "right thing." TCO also uncovers cost savings opportunities by highlighting large cost elements. Internally, TCO allows firms to gain an understanding of how their requirements (delivery, inventory, unique specifications, and so on) may actually increase costs of ownership.

A review of the TCO literature indicates that costs such as quality and delivery are the most commonly included items in TCO models. A logical way to view the costs of ownership is based on the order in which the cost elements are incurred, that is as they relate to the transaction sequence: pre-transaction, transaction, and post-transaction.

Advantages and limitations of the TCO approach

TCO provides many benefits that are documented in the literature (Ellram 1993). Some of the primary benefits of adopting a TCO approach are that it provides a consistent supplier evaluation tool, improving the value of supplier performance comparisons among suppliers and over time. It helps clarify and define supplier performance expectations for both the buyer and the supplier. TCO also provides a focus and sets priorities regarding the areas in which supplier performance would be improved by eliminating any non-purchasing related cost in the future and creating major opportunities for cost savings. TCO improves the purchaser's understanding of supplier performance issues and cost structure and provides excellent data for negotiations. It also justifies higher initial prices based on better quality/lower total cost in the long run to managers. Specifically, TCO-supported supplier selection helps in providing a consistent framework for supplier performance recognition awards and measuring ongoing supplier performance. It also helps in comparing supplier performance against others and self over time and in building strategic alliance efforts.

Total Cost of Ownership (TCO) represents a more recent attempt to cost a specific portion of the supply chain. TCO is "a structured approach for determining the total costs associated with the acquisition and subsequent use of a given item or service from a given supplier" (Carr and Christopher 1992). The approach recognizes that the purchase price represents only a portion of the total cost of acquiring an item. Supplier performance also affects the costs of ordering, expediting, receiving, and inspecting. TCO attempts to identify the total acquisition price by including the costs of purchasing, holding, poor quality, and delivery failure. Assigning costs to activities affected by the buyer provides another tool in the supplier decision. Buyers can evaluate alternate suppliers based on the costs associated with the number of product returns, nonconformance, or late shipments. Companies incorporating these factors into their ownership analysis can better determine which suppliers offer the best overall value to them.

TCO provides the capability to assess how inter-firm relationships affect costs within the purchasing firm. It links supplier performance to specific activities performed throughout the purchasing firm and translates the activities into costs. When coupled with activity-based costing, TCO can provide an even more accurate depiction of the activities and resources consumed in dealing with specific vendors (Kurt Salmon Associates 1993). Companies employing TCO can use the information to negotiate with or select upstream channel members based on total acquisition costs and other performance criteria. Although TCO does provide more accurate information on how the performance of one firm in the supply chain affects the costs of another, it does not provide the total supply chain cost. The costs captured in a TCO analysis only include the costs of one member of the supply chain. TCO does not capture the upstream firm's costs. By not capturing these costs, TCO may miss opportunities for making inter-firm cost trade-offs. One of the firms may more efficiently perform some activities than the other such as transportation, packaging, warehousing, or inventory management. TCO also does not demonstrate how the buyer's behavior may affect the suppliers' costs. The lack of an integrated costing approach may preclude the supply chain from achieving a cost competitive position.

Cost-based supplier performance evaluation

The purchasing function directly affects the ability of a firm to compete through its impact on quality, cost, technology, and supplier responsiveness. Measurement and evaluation of suppliers' performance will be required to establish and accurately reflect the total cost of doing business with individual suppliers. Monczka (1988) proposed a cost-based supplier performance evaluation system that provides a justifiable and rational method for evaluating key supplier performance factors. The system is predicated on the recognition that material price is only a fraction of the cost of the purchased material. It identifies supplier non-performance costs and accurately reflects the actual cost of doing business with suppliers.

Cost-based performance evaluation of suppliers incorporates quality, delivery, and other related costs as measurable factors that should be included when evaluating the total purchase costs of buying from various suppliers. It recognizes the suppliers who can most positively impact on organization's profitability by providing the lowest total cost to the buying firm. By establishing cost performance indexes related to supplier non-performance costs, the hidden costs associated with inadequate quality, delivery, price, and other elements of performance can be identified. These costs can then be managed through the utilization of appropriate suppliers and concomitant efforts that focus on reducing non-productive supplier-related costs.

The basic logic of a cost-based system is provided through a supplier performance index (SPI) defined as follows:

$$SPI = \frac{Extended\ \ Purchase\ \ \mathrm{Pr}ice + Non - Performance\ \ Cost}{Extended\ \ Purchase\ \ \mathrm{Pr}ice}$$

The SPI recognizes costs attributed to non-performance by suppliers for delivery, material quality, and prices. These costs are identified and collected, and the total cost of the supplier's performance (unit price plus non-performance costs) is used to develop an index number for each supplier for each major item.

The index amortizes the non-performance costs across each purchased item to establish the total average cost of doing business with a supplier by item. It highlights the actual total material cost per supplier. The index may provide the quantitative justification required for awarding business to a supplier whose product quality and service is more reliable even though his price is higher, if his total cost is lower than that other suppliers.

Supplier performance such as delivery, quality, and service are recognized as another important factor in selecting suppliers. The importance of supplier service performance is measured by means of service factor ratings (SFR). The SFR includes performance factors that are difficult to quantify from a cost point of view, but ones that nevertheless are important to a supplier's success. Ratings for these performance factors are generated by "customers" of the supplier within the buying firm – personnel in internal operating units, such as purchasing, quality control, manufacturing, product engineering, and so on. For a given supplier, his ratings on all factors are summed for each "customer" and then averaged to obtain a total service rating. This rating is then divided by the total number of points possible to obtain the supplier's "service factor percentage". Although the rating reflects the individual's actual experience with the supplier, it is to a great extent a subjective assessment. Service factor ratings (SFRs) are not expressed in terms of costs. They are considered separately in the overall evaluation of a supplier as a measurement of their strengths and weakness.

The model can be used as a primary indicator to indicate when a supplier's relative performance is exhibiting variability. As the supplier's SPI increases, so does the buying organization's requirement for resources. Increases signify that corrective action should immediately be taken. Additionally, the SPI provides insight into the supplier's ability to maintain item performance consistency. The inability to maintain consistency can be directly related to the supplier's manufacturing process capability. This is key when establishing long-term relationships. Conversely, reductions of a supplier's SPI indicate the success of the supplier's efforts to provide consistently acceptable products.

A cost-based supplier evaluation program addresses buying needs by monitoring and evaluating suppliers on their actual performance, using dollars as the objective rating criteria. As a potential cost reduction tool, the cost-based supplier evaluation system focuses on the total cost of doing business with any supplier. This includes the examination of a variety of costs, from the manufacturing line to future quality/warranty problems. Given the high value of raw material dollars as a percentage of sales, this supplier management tool offers the opportunity for total cost reduction, positive competitive positioning, and long-term supply assurances.

Comparison of TCO, ABC and AHP

Activity-based costing and total cost of ownership are topics that have received considerable attention in recent years. Both represent attempts to understand the factors that actually drive costs and assign costs to the activities that generate them (as opposed to allocating costs on some basis that may not be directly relevant to those activities). An underlying assumption of both ABC and TCO is that proper cost assignment can improve decision-making. Both ABC and TCO concentrate on properly assigning costs to the activities that generate those costs. In ABC, however, the focus is to properly assign costs to the product, customer, or distribution channel that causes the cost to be incurred. TCO focus on costs associated with purchases. Further, TCO aims to group costs that are driven by doing business with a particular supplier for a particular item or service.

TCO in purchasing focuses on a better understanding of the true costs associated with the entire purchasing cycle. Unlike ABC, which focuses on the internal, administrative costs of a purchasing department, TCO focuses on all costs associated with a purchase. TCO considers all costs associated with the acquisition, use, maintenance, and follow-up of a purchased goods or service, not just the purchase price. TCO considers costs that occur before, during, and after a purchase.

Bhutta and Huq (2002) compare TCO and AHP methods in several ways. They consider that, first of all, TCO tends to focus more on the pricing issues and ignores qualitative issues, its strength is the ability to use the same model to evaluate suppliers across the board and identify the best supplier based on lowest transaction costs, and can be used effectively for supplier evaluation along with supplier selection. AHP provides a tool to help integrate and compare seemingly un-comparable issues and forces company management to make the required trade-offs to select the optimal supplier. AHP is more of a selection tool and is appropriate in decision-making situations, where both quantitative and qualitative factors have to be considered, whereas TCO is difficult to use in an environment where subjective assessments and judgments have to be used in comparing factors.

Second, TCO provides a consistent supplier evaluation tool, improving the value of supplier performance comparisons among suppliers and over time. It helps clarify and define supplier performance expectations for both the buyer and the supplier. Using a common model for both supplier selection and evaluation, TCO provides focus and a consistent message about what is important, creates less work, and the outcome of selection and evaluation can be used directly to pre-qualify suppliers, qualify suppliers, and even be part of the supplier certification process. Thus all the firms' supplier measurement tools will be linked and consistent. However the amount of complexity and the data requirements are major drawbacks of the approach.

AHP, on the other hand, is a flexible modeling tool that can accommodate a larger set of evaluation criteria and can address both qualitative and quantitative data. However, judgments that drive the process, along with being the strongest advantage, are also a limitation, because one person's judgment may differ from another's. Several consensus-building approaches have been adopted to overcome this concern.

In terms of the applications of TCO and AHP, TCO is better suited to those situations where cost is of high priority and detailed cost data are available to make comparisons. In the case of AHP, it is better suited to solve and decide between suppliers when several conflicting goals exist and, when although cost may be an important factor, it is not the overriding one (Bhutta and Huq 2002). Table 3-3 shows the features and relationships between AHP, ABC, and TCO.

Table 3-3. Features and Relationships of AHP, ABC, and TCO

	AHP **(Analytical Hierarchy Process)**	**ABC** **(Activity-based Costing)**	**TCO** **(Total Cost of Ownership)**
Features	• Selection method for decision-making situation where both quantitative and qualitative factors have to be considered;	• Assign costs (especially indirect costs) to the activities that generate them; • Cost objects are product, customers, etc. • Focus on the internal and administrative costs;	• Assign costs (especially indirect costs) to the activities that generate them; • Cost objects are product, customers, etc. • Focus on the internal and administrative costs;
Model Structure	Hierarchy	Hierarchy	
Relationship	ABC and AHP combined in the hierarchical structure, using AHP in the pair-wise comparison when quantitative measures are not available		
		Both of them assign costs to activities which generate them	

Summary

This chapter reviews the performance measurements in supply chain design. The performance measurements are categorized in two – one category is cost-based performance measurements and the other one is non-cost-based performance measurements. This chapter also reviews two popular supply chain models and concepts in supplier selection: activity-based costing (ABC) and total cost of ownership (TCO). Both of models and concepts has to cope with intangible and tangible factors. Analytical Hierarchy Process (AHP), which is a popular decision-making method, is usually utilized with ABC or TCO to quantify the intangible factors in the ABC or TCO models, and make decisions in supplier selection or product selection. However these models are proposed in manufacturing industry, no appropriate models are available for the construction industry to solve the supplier and subcontractor selection decision-making and form effective and efficient supply chains. The next chapter will present this research objectives, significance, contribution and limitation.

CHAPTER 4
RESEARCH OBJECTIVES, PROBLEMS, SIGNIFICANCE, AND LIMITATIONS

Supplier selection is generally a lengthy evaluation process. Suppliers are evaluated on several criteria such as pricing structure, delivery, product quality, and service (Bhutta and Huq 2002), encompassing many tangible and intangible factors in a hierarchical manner. The evaluation of intangible factors requires the assessment of expert judgment, and the hierarchical structure requires decomposition and synthesis of these factors (Prueitt, 2000). The objective of supplier selection is to find the optimal supplier, not necessarily the supplier offering the lowest price or the shortest delivery or the best technical service. The decision maker must consider multiple criteria in their attempts to distinguish between services offered by potential suppliers.

Supplier/subcontractor selection problem is a supply chain design and analysis problem. An important component in supply chain design and analysis is the establishment of appropriate performance measures. A lot of performance measurement studies have been conducted in manufacturing industries. In the Construction Industry, Performance measures are used to analyze project performance instead of the supply chain performance. Performance measures usually are related to the productivity measurements. Productivity analysis in the Construction Industry is actually operational level performance measurements concerning cost accounting, measuring daily man-hours and daily quantities. The performance measurement studies for the manufacturing industries and the construction industry can be categorized into cost-based performance measures and non-cost-based performance measures.

On the other hand, a network of supply chains includes multiple layers of subcontractors and interlinked suppliers. Selection problems exist on each layer of the supply chain. Selection methodologies for various project supply chain participants and decisions are critical to affect construction project performance. Supplier selection models such as analytical hierarchy process (AHP), activity-based costing (ABC) which is applied as a supply chain costing method to evaluate a supply chain efficiency, total cost ownership (TCO) which focuses on purchasing to evaluate both direct cost and hidden cost, and the combination of AHP, ABC, or TCO are proposed by researchers to evaluate the quantitative and qualitative performance criteria of a supply chain. Although these models solve some problems for the manufacturing industries, a model that addresses both quantitative and qualitative performance criteria for selecting a subcontractor/supplier in the construction industry does not currently exist. Therefore, the research objectives are to:

- Develop a supplier/subcontractor selection decision support model
- Synthesize the quantitative and qualitative performance criteria in evaluating a supply chain and selecting a supplier/subcontractor
- Assess the tangible and intangible performance criteria through tracing the activities between the inter-organizations.
- Identify value-added and non-value-added construction activities.
- Apply the proposed model to the steel construction service case study.

Based on the objectives of the research, the research questions are:

- For the construction industry, how do we synthesize both tangible and intangible performance measurements to evaluate a supply chain/supplier/subcontractor?
- How do we integrate the decision goal, performance measurements, and activities of inter-organizations to into one decision model?

- What data should be collected while quantifying intangible performance measurement?

This purpose of this research is to provide a handy methodology and decision support system for construction industry practitioners to make supplier/subcontractor selection decision or an efficient and effective supply chain design. The data produced by the system implementing the methodology can be used in a number of related managerial activities. Some of the more significant contributions are:

- Allows for the selection of the right construction supply chain for a project
- Provides non-cost-based performance data for use in various types of improvement programs
- Allows for the utilization of non-cost-based performance data as a point for negotiation with suppliers/subcontractors.
- Allows the forecasting of new supplier performance, based on relevant historical data.
- Improving construction project management through the use of a holistic management approach.

The research treats the performance criteria, activities, and suppliers and subcontractors in a hierarchical way. The performance criteria are treated as quantitative and qualitative measurements or cost-based and non-cost-based measurements. The qualitative measurements on the subcontractors and suppliers are evaluated on the activities undertaken by the potentials. The importance weight of activities, non-cost-based measurements on activities, and subcontractors/suppliers are quantified by using Analytic Hierarchy Process (AHP). The activities, qualitative measurements, subcontractors and suppliers are compared pair-wise on each level of the hierarchical structure. In this way, a larger number of criteria can be included within the hierarchy. Although obtaining the importance weights by using AHP is currently the best way to deal with the qualitative judgments, this model is driven by judgments of the decision maker and there is no analytic way of verifying the results. In addition, when a new criterion is added, the judgment process has to be repeated with generating a large pair-wise matrix. It also adds burden of recording data on the management personnel while tracking the performance of subcontractors and supplier.

No matter what the limitations of this research, it does disentangle the complexity of subcontractor selection in the construction industry using the activity-based performance and costing model. Furthermore, the proposed model is applied to structural steel services, addressing both quantitative and qualitative performance criteria and structuring and rationalizing the decision-making process. The next chapter, Chapter 5, describes how the research is conducted.

CHAPTER 5
DECISION SUPPORT MODEL FOR STRUCTURAL STEEL SUPPLY CHAIN

The proposed model is developed as a convenient tool for project managers or other construction industry practitioners during the pre-bid stage to make decisions on selecting the right supply chain alternatives based on tangible and intangible suppliers' performance. As described in Chapter 3, the suppliers'/subcontractors' performances are categorized into cost-based performances and non-cost-based performances, and the overall model will consider two impacts – cost impact and performance impact. The two impacts affect the decision-making in selecting supply chain alternatives. Two separate sub-models-- a cost model and activity-based performance model – are proposed to evaluate the cost impact and performance impact of the decision making process. The overall model operates at two levels: the decision-making level and the operational level. The decision-making level makes a supply decision based on the supply chain alternatives. All of the alternatives are evaluated based on the cost and performance scores. The cost model and the performance model are combined together at this level and are used to evaluate the alternatives. At the operational level, the cost model and the performance model are separated. In the performance model, the activities and performance criteria that reflect the activities in each work package are analyzed. The performance score of each subcontractor in a work package are evaluated. The overall model is depicted in Figure 5-1, and it consists of the following steps:

Step 1. Define the importance weights of cost and performance criteria based on the company's business strategy.

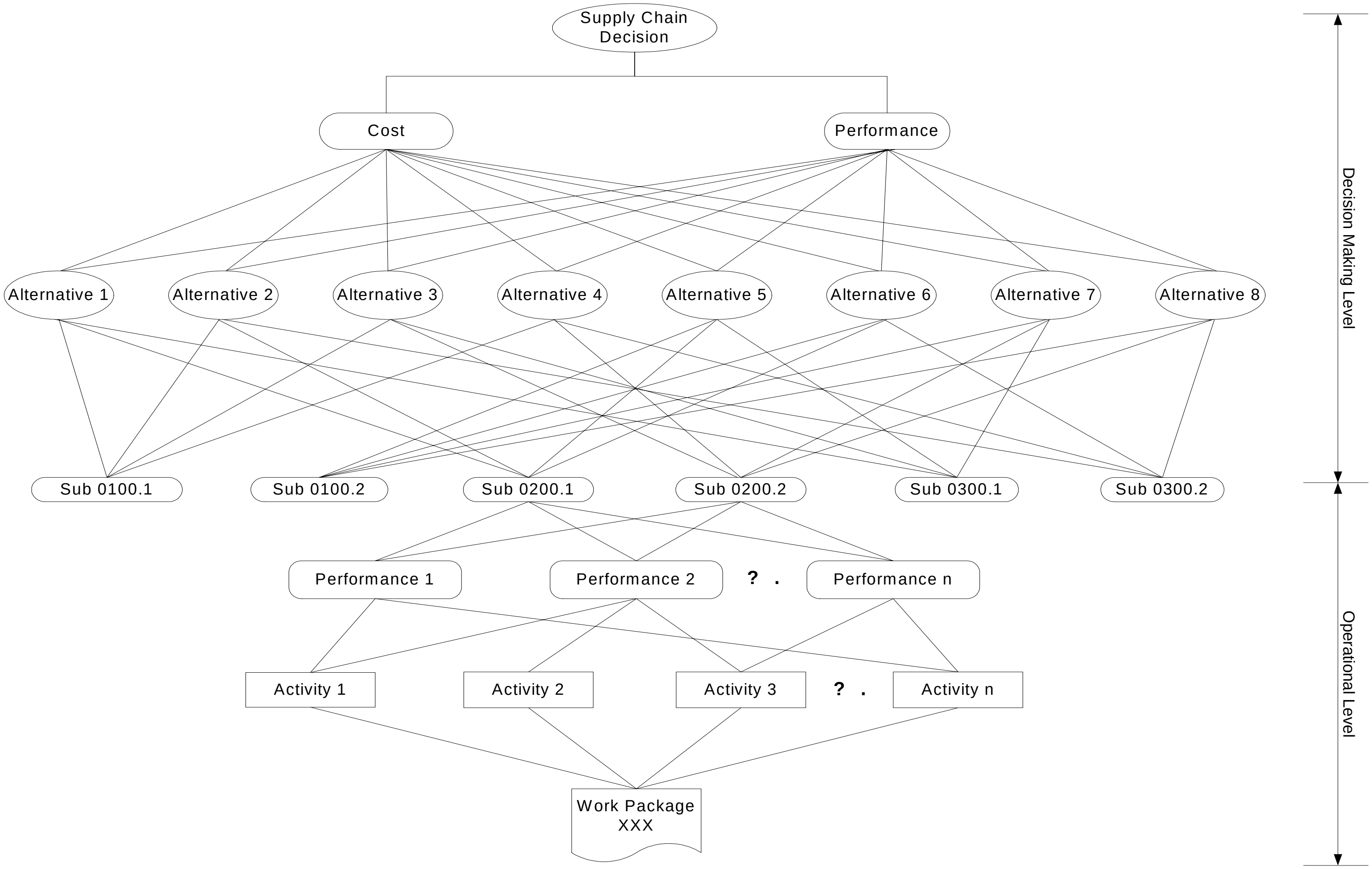


Figure 5-1. Overall Supply Chain Design Decision Making Model

Step 2. Using the cost model, identify the cost of each alternative and evaluate the cost score of each alternative by using analytical hierarchy process techniques.

Step 3. Using the performance model, identify essential activities and performance measures in each work package for a specific project, establish the relationship between activities and performance criteria (measures) by using analytical hierarchy process, and evaluate the subcontractors in each work package in terms of performance measures.

Step 4. Based on the information from Step 3, assess the performance score of each alternative.

Step 5. Based on the information from Step 1, Step 2, and Step 4, plots the cost and performance scores to select the best alternatives.

Next the cost and performance models will each be discussed individually.

Cost Model

The cost model is relatively simpler than the performance model. The main idea of the cost model is to figure out all the combination possibilities (supply chain alternatives) of the subcontractors within each work package and to sum are up the total cost for each combination. Finally the cost model evaluates the scores of each alternative by using analytical hierarchy process techniques. The procedure and steps included in the cost model illustrated below with an example.

Step 1. Identify work packages that will be subcontracted out or self-performed for a specific project. Figure 5-2 shows an example of a specific project that has three work packages; the notation of the first work package is 0100. The number of subcontractors that send quotes for this work package is two. They are identified as 0100.1 and 0100.2. Using the same notation method, the second and third work packages are identified as 0200 and 0300, and each of them has two different subcontractors submitting bids. Figure

5-2 also shows the combination possibilities of these subcontractors in three different work packages. A total of 8 alternatives are displayed ($2^3 = 8$).

Step 2. List all the quotes from potential subcontractors or the costs of self-performing the work. Based on these cost and quotes information, determine the total cost for each of the supply chain alternatives (See Figure 5-2).

Step 3. Use a matrix to store the cost difference between each pair of alternatives (See Table 5-1). This step is in preparation for the comparison of each pair of alternatives based on a specific measurement scale (See Table 5-2).

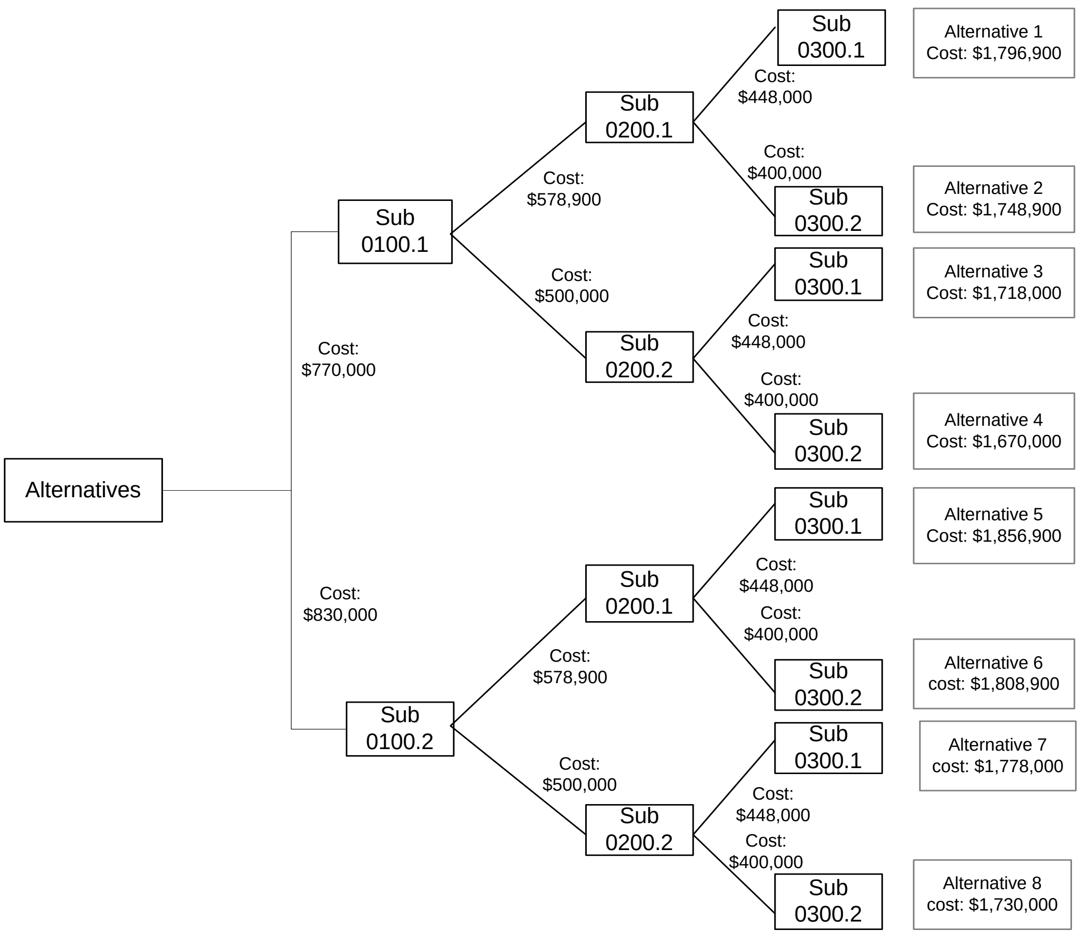


Figure 5-2. Total Cost for the Alternatives

Table 5-1. The Cost Differences Between Each Pair of Alternatives

		Alt 1	Alt 2	Alt 3	Alt 4	Alt 5	Alt 6	Alt 7	Alt 8
		$1,796,900	$1,748,900	$1,718,000	$1,670,000	$1,856,900	$1,808,900	$1,778,000	$1,730,000
Alt 1	$1,796,900	$0	($48,000)	($78,900)	($126,900)	$60,000	$12,000	($18,900)	($66,900)
Alt 2	$1,748,900		$0	($30,900)	($78,900)	$108,000	$60,000	$29,100	($18,900)
Alt 3	$1,718,000			$0	($48,000)	$138,900	$90,900	$60,000	$12,000
Alt 4	$1,670,000				$0	$186,900	$138,900	$108,000	$60,000
Alt 5	$1,856,900					$0	($48,000)	($78,900)	($126,900)
Alt 6	$1,808,900						$0	($30,900)	($78,900)
Alt 7	$1,778,000							$0	($48,000)
Alt 8	$1,730,000								$0

Table 5-2. Measurement Scale (Source: Nydick and Hill 1992)

Verbal Judgment or Preference	**Numerical Rating**
Extremely Preferred	9
Very Strongly Preferred	7
Strongly Preferred	5
Moderately Preferred	3
Equally Preferred	1
The intermediate values of 2,4,6, and 8 provide additional levels of discrimination	
Reciprocals: If activity *i* has a specific numerical rating with respect to activity *j*, then j has the reciprocal value when compared to *i*.	

Step 4. Set a cost measurement scale. This cost measurement scale is set based on the cost difference calculated in Step 3 and is measured using the scale of Nydick and Hill(1992). Choose the largest absolute cost difference and distribute it evenly into this 9-point cost measurement scale (See Table 5-3).

Step 5. Based on the information from Step 3 and Step 4, an original matrix with respect to cost comparison of each pair of alternatives is presented in Table 5-4.

Step 6. Adjust the original matrix with respect to cost comparison and finally obtain the cost weight for each alternative (See Table 5-5).

Table 5-3. 9-Point Cost Measurement Scale

Verbal Judgment	Numerical Rating	Cost Difference Range	
Extremely preferred	9	$224,999	$200,000
	8	$199,999	$175,000
Very strongly preferred	7	$174,999	$150,000
	6	$149,999	$125,000
Strongly preferred	5	$124,999	$100,000
	4	$99,999	$75,000
Moderately preferred	3	$74,999	$50,000
	2	$49,999	$25,000
Equally preferred	1	$24,999	($24,999)
	1/2	($25,000)	($49,999)
Moderately un-preferred	1/3	($50,000)	($74,999)
	1/4	($75,000)	($99,999)
Strongly un-preferred	1/5	($100,000)	($124,999)
	1/6	($125,000)	($149,999)
Very strongly un-preferred	1/7	($150,000)	($174,999)
	1/8	($175,000)	($199,999)
Extremely un-preferred	1/9	($200,000)	($224,999)

Table 5-4. Original Matrix With Respect To Cost Comparison

	Alt 1	Alt 2	Alt 3	Alt 4	Alt 5	Alt 6	Alt 7	Alt 8
Alt 1	1	1/2	1/4	1/6	3	1	1	1/3
Alt 2	2	1	1/2	1/4	5	3	2	1
Alt 3	4	2	1	1/2	6	4	3	1
Alt 4	6	4	2	1	8	6	5	3
Alt 5	1/3	1/5	1/6	1/8	1	1/2	1/4	1/6
Alt 6	1	1/3	1/4	1/6	2	1	1/2	1/4
Alt 7	1	½	1/3	1/5	4	2	1	1/2
Alt 8	3	1	1	1/3	6	4	2	1
Total	18.33	9.53	5.5	2.74	35	21.5	14.75	7.25

Table 5-5. Adjusted Matrix With Respect To Cost Comparison

	Alt 1	Alt 2	Alt 3	Alt 4	Alt 5	Alt 6	Alt 7	Alt 8	Weights (Row Avg.)
Alt 1	0.05	0.05	0.05	0.06	0.09	0.05	0.07	0.05	0.06
Alt 2	0.11	0.10	0.09	0.09	0.14	0.14	0.14	0.14	0.12
Alt 3	0.22	0.21	0.18	0.18	0.17	0.19	0.20	0.14	0.19
Alt 4	0.33	0.42	0.36	0.36	0.23	0.28	0.34	0.41	0.34
Alt 5	0.02	0.02	0.03	0.05	0.03	0.02	0.02	0.02	0.03
Alt 6	0.05	0.03	0.05	0.06	0.06	0.05	0.03	0.03	0.05
Alt 7	0.05	0.05	0.06	0.07	0.11	0.09	0.07	0.07	0.07
Alt 8	0.16	0.10	0.18	0.12	0.17	0.19	0.14	0.14	0.15
								Total	1.00

Activity-based Performance Measuring Model

The performance model is proposed to analyze and evaluate the subcontractors' performances on the activities of a work package rather than only focusing on the cost. The objective of the performance model is to measure the performance of subcontractors. Activity-based performance measures the each specific activity's performance. The total performance of activities of subcontractors can then be measured. There are several hierarchies in this model. The first level is the activity level. By analyzing the business process, a work package is decomposed into activities by management. These activities are needed to complete the requirements of specific work packages. The second level is the performance measure level. The performance measures or criteria are established based on the information that management needs to control and improve activities. The link from activities to performance is a natural result of implementing activity-based performance management. This link provides a tangible and measurable method for measuring the performance of each activity. The third level represents subcontractors. The other link from performance measures to subcontractors provides a tangible method to measure each subcontractor's performance on each activity performed in a specific work package. The performance model links activities and subcontractors through performance criteria or measures to finally evaluate the subcontractors' overall performance on these decomposed activities. All the priority weights in this performance model are derived from pair-wise comparisons using AHP techniques, including the development of priority weights linking activities to performance measures, and the development of priority weights linking performances to subcontractors. The following is a detailed process description of the activity-based performance model (Figure 5-3).

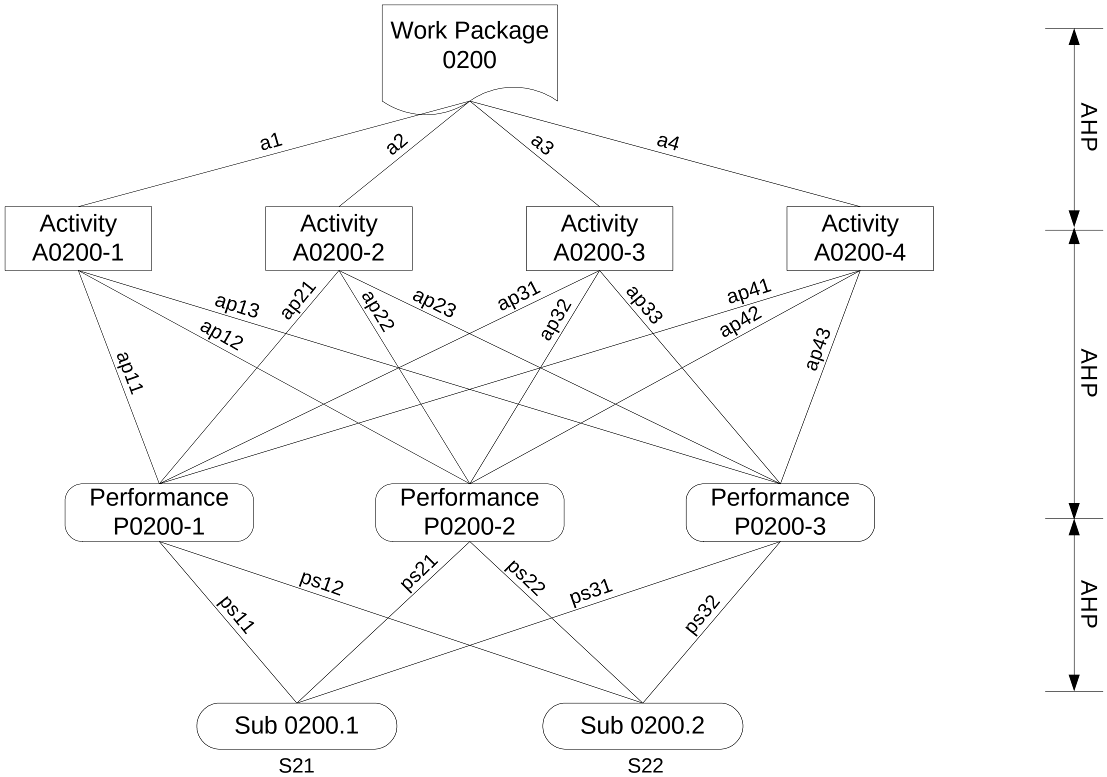


Figure 5-3. Activity-based Performance Model for A Specific Work Package

Step 1. Identify activities performed by suppliers/subcontractors, and establish the priority of each activity in a specific work package. The notation of the priority for each activity is $a_1, a_2, \ldots\ldots, a_n$. Figure 5-3 shows the business process in a work package 0200 is decomposed into four critical activities. They are identified as A0200-1, A0200-2, A0200-3, and A0200-4. The priority for each activity is identified respectively by management as $A = [a_1 \quad a_2 \quad \ldots\ldots \quad a_m]$.

Step 2. Identify performance measures for this specific work package. The performance measures or criteria are established based on the information the management needs to control and improve activities of suppliers/subcontractors. In the example shown in Figure 5-3, there are three performance measures identified to control the activities. The performance measures are represented as P0200-1, P0200-2, and P0200-3.

Step 3. Establish the links between each activity and performance measures. The links quantify the priority of performance measures to each activity based on the management needs. The measurement scale of Nydick and Hill (1992) or some other kind of measurement scale can be used to establish pair-wise performance comparison for each activity. Finally the weights for each link are determined by the AHP technique. Figure 5-3 shows the weights of each activity to each performance measure. For example, the weights of performance measures for the activity 0200-1 are shown as ap_{11}, ap_{12}, and ap_{13}. The total relationship matrix between activities and performances is represented as:

$$\begin{array}{c} \textit{Performances} \\ AP = \begin{bmatrix} ap_{11} & ap_{12} & ..... & ap_{1n} \\ ap_{21} & ap_{22} & ..... & ap_{2n} \\ ..... & .... & ..... & ..... \\ ap_{m1} & ap_{m2} & ..... & ap_{mn} \end{bmatrix} \quad \textit{Activities} \end{array}$$

Step 4. Evaluate the subcontractors' score on each performance criteria by using AHP. Figure 5-3 shows the scores on the links between performance measures and subcontractors. The subcontractors' performance scores are determined by using the measurement scale of Nydick and Hill (1992) and AHP techniques to avoid intuitive assessments. The whole subcontractors' performance matrix is represented as:

$$\begin{array}{c} \textit{Subcontractors} \\ PS = \begin{bmatrix} ps_{11} & ps_{12} & ..... & ps_{1t} \\ ps_{21} & ps_{22} & ..... & ps_{2t} \\ ..... & ..... & ..... & ..... \\ ps_{n1} & ps_{n2} & ..... & ps_{nt} \end{bmatrix} \quad \textit{Performances} \end{array}$$

Step 5. Based on the information from Steps 1, 3 and 4, multiply matrixes A • AP • PS to get the final overall performance weights of subcontractors and suppliers on the activities of a specific work package.

$$S = A \bullet AP \bullet PS$$

$$= \begin{bmatrix} a_1 & a_2 & ..... & a_m \end{bmatrix} \bullet \begin{bmatrix} ap_{11} & ap_{12} & ..... & ap_{1n} \\ ap_{21} & ap_{22} & ..... & ap_{2n} \\ ..... & ..... & ..... & ..... \\ ap_{m1} & ap_{m2} & ..... & ap_{mn} \end{bmatrix} \bullet \begin{bmatrix} ps_{11} & ps_{12} & ..... & ps_{1t} \\ ps_{21} & ps_{22} & ..... & ps_{2t} \\ ..... & ..... & ..... & .... \\ ps_{n1} & ps_{n2} & ..... & ps_{nt} \end{bmatrix}$$

$$= \begin{bmatrix} s_{11} & s_{12} & ..... & s_{1t} \end{bmatrix}$$

Step 6. Repeat Steps 1, 2, 3, and 4 for each specific work package of a project. The final overall performance weights of subcontractors and suppliers of all work packages will then be appropriately represented in this performance model. In the Figure 5-3 example there are three work packages, the overall performance weights of subcontractors and suppliers for each work package can be represented as:

$$S^1 = \begin{bmatrix} s_{11}^1 & s_{12}^1 \end{bmatrix}$$

$$S^2 = \begin{bmatrix} s_{11}^2 & s_{12}^2 \end{bmatrix}$$

$$S^3 = \begin{bmatrix} s_{11}^3 & s_{12}^3 \end{bmatrix}$$

Step 7. Calculate the performance weights of all alternatives based on the information from Step 6. For example the performance weight of alternative 1 are obtained from $Alt_{p,1} = s_{11}^1 \bullet s_{11}^2 \bullet s_{11}^3$

Integrating Cost and Performance Scores

The last step of the decision support model is to integrate the scores of each alternative from the cost and performance models, derived from the cost and the performance weights assigned by management and the scores for each alternative.

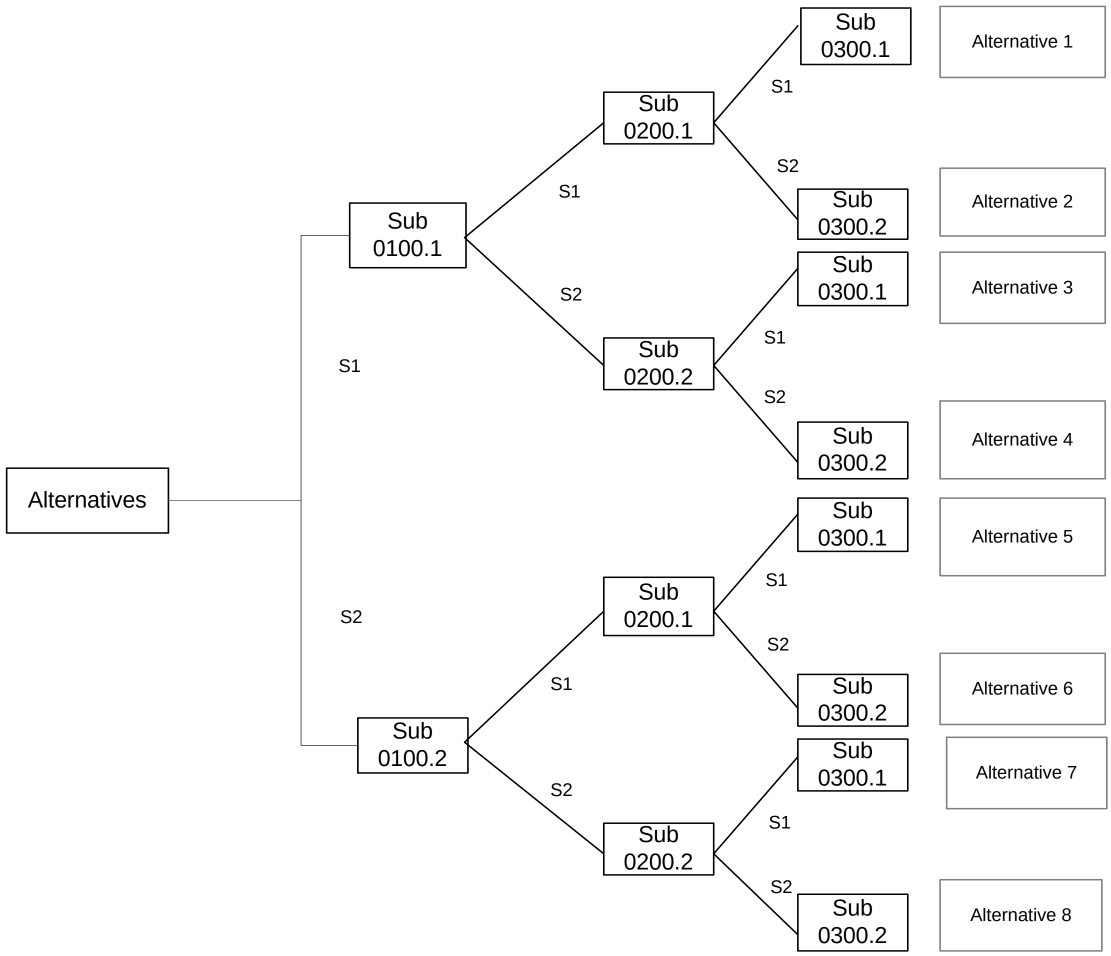


Figure 5-4. The Alternatives' Weights in Activity-based Performance Model

From the cost and the performance models, the final score of each alternative can be represented as:

$$Alt_k = w_c \bullet Alt_{c,k} + w_p \bullet Alt_{p,k}$$

where:

Alt_k : the kth Alternative, $k = 1,2,.....n$;

$Alt_{c,k}$: the cost score of kth Alternative, $k = 1,2,.....n$;

$Alt_{p,k}$: the performance score of kth Alternative, $k = 1,2,.....n$;

w_c : the cost weight;

w_p : the performance weight;

The best choice from these alternatives will be the highest score:

$Max\{Alt_1, Alt_2, ...Alt_n\}$

Summary

This chapter describes the activity-based performance and costing model in details. The model consists of two sub-models: a costing sub-model and an activity-based performance sub-model. The model also is divided into two levels: a decision-making level and operational level. The costing model is only on the decision-making level and handles the cost-based performance criterion. It gives out cost priority weights of different supply chain alternatives. The activity-based performance model is on the both decision-making level and an operational level and it handles the non-cost-based performance criteria. The activity-based performance sub-model handles the activities, performance criteria, and subcontractors/suppliers in the hierarchical structure. The importance weights of activities, performance criteria, and subcontractors/suppliers are quantified by AHP. The activities, the relationship between activities and performance criteria, and the relationship between performance criteria and subcontractors/suppliers are represented as corresponding matrices. The overall non-cost-based performance of subcontracts/suppliers is the product of these matrices. The non-cost-based performance model finally gives out the priority weight of performance of various supply chain alternatives. The right supply chain choice is determined by aggregating the cost priority weight and performance priority weight. The next chapter, Chapter 6, will introduce the application of the proposed model in the steel construction services area.

CHAPTER 6
CASE STUDY

Company Background

ABC Steel is a small fabricator and erector of structural steel located in north central Florida. Their average contract size is $100,000 and the largest contract it has undertaken is $500,000. Their work volume is around $5,000,000 per year. Figure 6-1 shows the organizational structure of the company. The company's primary projects are located in north Florida. The company has the following business characteristics:

- It is a small business in the structural steel market.
- It is organized in a job shop configuration operating in a make-to-order production process, where each project is customized and unique.

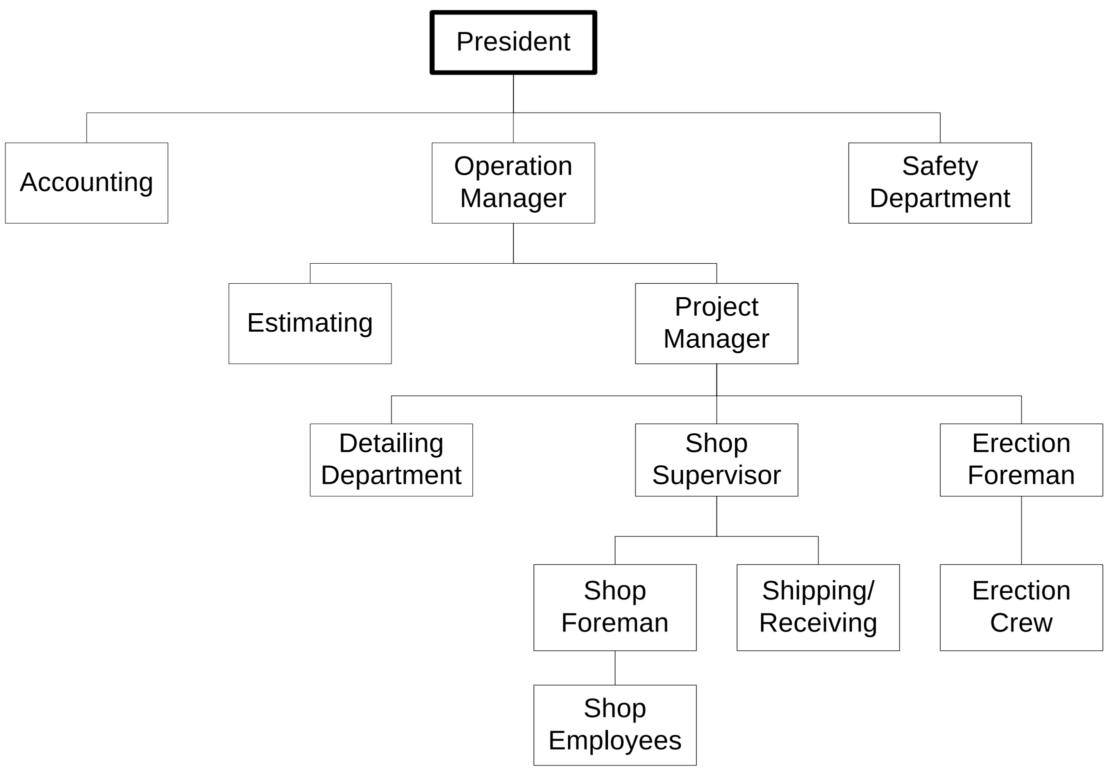


Figure 6-1. ABC Steel Fabrication Organizational Structure

The production process begins with an inquiry by a customer, generally a general contractor, for a quote on a particular job. The development of the quote typically involves multiple iterations with the general contractor for project information and requires working with subcontractors and material suppliers. If the quote is accepted by the general contractor, it is converted into a job. A job is classified as small if its quote is less than $20,000, medium if its quote ranges between $20,000 and $200,000, and large if its quote is greater than $200,000. Upon being awarded the contract, the job is scheduled according to its due date and the existing shop load. Structural steel must pass through various operations during the course of its fabrication, the typical fabrication process used in the shop is: material handling and cutting, template making, laying out, punching and drilling, fitting and reaming, fastening, finishing, machine shop operations, quality control inspecting, cleaning and painting, and shipping. Figure 6-2 shows the business processes within the company.

However, before the company sends a quote to the general contractor, it has to make selection decisions for its subcontractors and material suppliers. The right decision will help it work with its upstream suppliers and subcontractors to not only meet the general contractors' project requirements but also to gain competitive advantages in the fierce structural steel business market. The upstream business partners ABC Steel Fabrication it works with are described in the following sections.

ABC Steel Fabrication Upstream Supply Chain

Rolling Mill

The company has two main suppliers XYZ Bros. in Jacksonville, Florida and Best Steel in Macon, Georgia. On small projects a detailed cut-list is prepared by the company and then sent to the rolling mill. The rolling mill checks for availability and cuts the steel

Figure 6-2. ABC Steel Fabrication Flowchart

accordingly. On large jobs, the engineer's structural drawings are sent to the mill. With this approach, the duty of preparing the cut-list is shifted to the rolling mill. On average it takes 2-3 days to receive an order from the rolling mill. The shipping and receiving department is responsible for placing orders to the rolling mill.

Joist and Decking Subcontractors

In order to reduce its cost the company subcontracts all the joist and decking components. Joist subcontractors are responsible for the structural detailing of the products they supply. This joist detailing is normally coordinated with the main structural detailer to ensure a common base design and also to avoid errors. The two main subcontractors used by the company for joist and decking components are SC Inc. (South Carolina) and Steel Decking (Tampa, Florida). For most projects the lead-time for joist and decking is between 2 - 6 weeks.

Structural Detailers

The company has in-house detailers to do all structural detailing. Occasionally when their in-house detailers have too much work, it completely subcontracts the structural detailing work. On an average job it takes between 2-3 weeks to complete the job.

Connectors

The company has the capability to manufacture some connector plates and joist anchors but it is more economical to buy these connectors from local suppliers.

Galvanizing and Light Gauge Steel Work

There are instances where the drawing specifies the use of members that need to be galvanized after fabrication. The company does not have the expertise to do such job and hence subcontracts them to specialists like Steel Galvanizers in Tampa, FL. Materials to be galvanized are supplied by the company to Steel Galvanizers and it usually takes about one week to receive the order back from them.

Specialized Treatment and Fabrication

Occasionally there is the need to do specialized work such as the bending of plates and cutting and forming intricate shapes. In such instances the company uses the services of Steel Bending, Lake City, Florida. The required materials are supplied by the company and it normally takes about 5 days for an order to be processed. After the order from the mill is received and structural detailing is finalized the shop superintendent prepares a detailed cutting and processing instruction for the machine shop operators. The purpose of this set of instructions is to:

- Minimize wastage of material.
- Reduce idle fabrication time.
- Help control their schedule that is tied to that of the general contractor and erection firm.

After the components have been fabricated, they are cleaned and stacked ready to be delivered to the construction job site for erection. Figure 6-3 shows the related supply chain tiers and participants for ABC Steel.

The company usually awards its supplier and subcontractor work based on the low prices and familiarity. Although this selection method is convenient and offers low quotes to the general contractors, any failure in meeting delivery and quality requirements incurs additional costs for their company. ABC Steel Fabrication’s profits had declined during the previous several years. Although individual jobs appeared to be profitable, at an aggregate level the company was losing money and market share. This situation has led the company to develop an improved selection and decision support

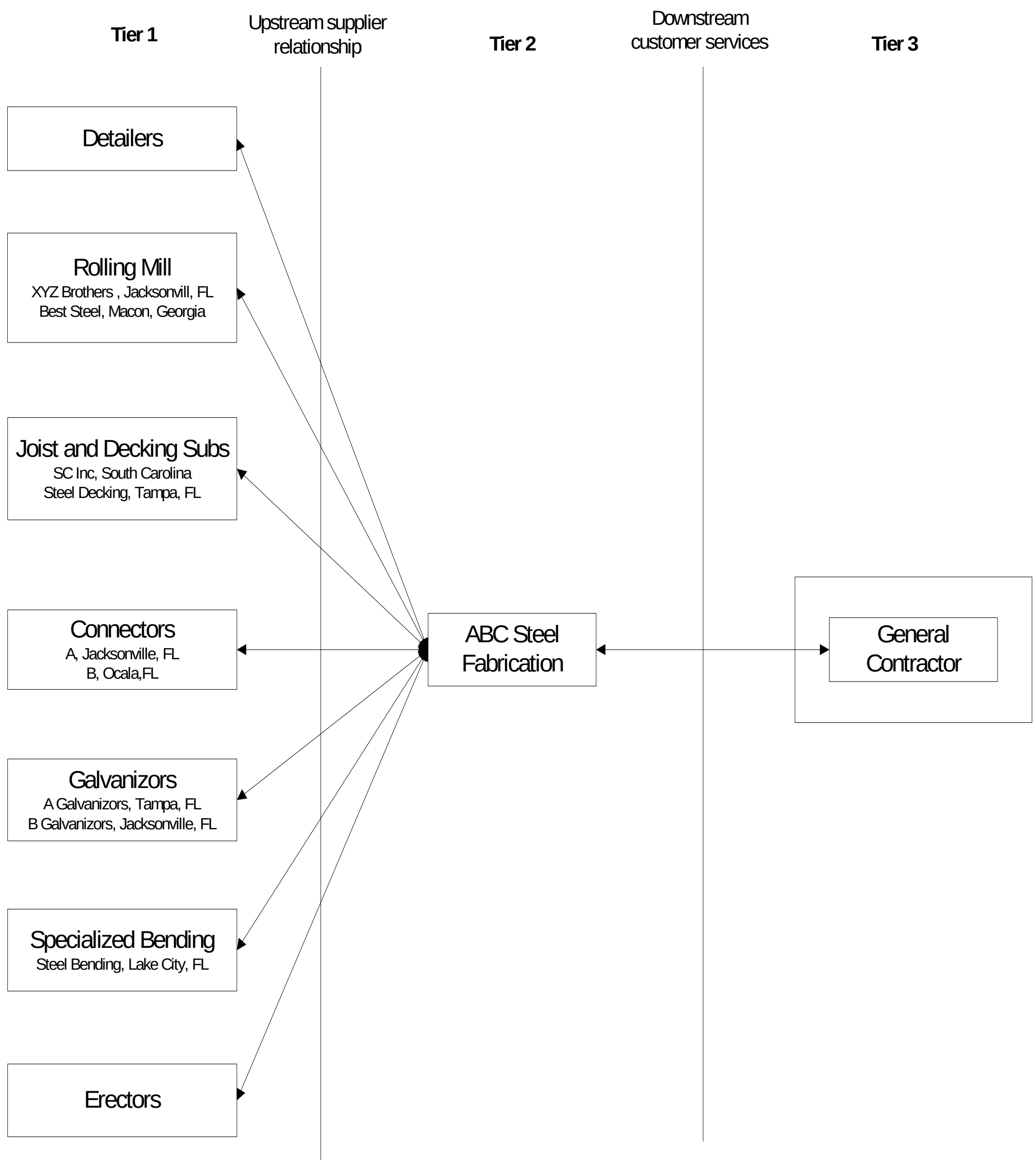


Figure 6-3. Supply Chain Tiers and Participants for ABC Steel

system. The managers of the company to list several factors that affect how they select their material suppliers and subcontractors:

- Material Delivery Schedule
- Communications
- Price
- Service
- Quality

These factors are associated with the activities that the suppliers and subcontractors perform in the procurement of materials and services to support the fabricator's internal operations. The goal of the subcontractor/supplier selection decision-making process is to provide:

- A tool for supplier or subcontractor selection for specific projects;
- An ability to identify the non-efficient activities and processes within the organization;
- Data for continuous improvement;

Defining Work Packages and Work Scope

ABC Steel Fabrication defines their work packages based on the parts of their subcontracted jobs. Their operating procedures include the following work packages:

- Rolling mill purchases
- Joist and decking subcontract
- Connectors
- Galvanizing and light gauge steel
- Structural detailer
- Specialized treatment and fabrication

This case study explains how to apply the proposed model using the first four work packages.

Documenting the Activities and Processes and Identify Performance Criteria

To better understand the structure and the operations of the company, the significant activities being carried out were identified. The interviewing technique was used to identify the activities within each work package and the performance criteria concerns. A bill of activities was generated by compilation of the responses gathered during the interviews. The activities generated are shown in Tables 6-1. The performance criteria used are shown in Table 6-2.

- Effective and efficient communication

- Quote response time
- Dispute minimization Higher quality
- On-time delivery
- Quality
- Cost certainty

Table 6-1. List of Activities for The Work Packages

Suppliers/Subcontractors	**Activities**
Rolling Mill	Send quotes, Process purchase order, shipping, Inspection, Billing
Joist and Decking	Send quotes, Detailing, fabrication, shipping, Inspection, Billing
Connector	Send quotes, Process purchase order, Inspection, Billing
Galvanizer and light gauge steel work	Send quotes, Process purchase order, Inspection, Billing

Table 6-2. Performance Criteria for The Work Packages

Suppliers/Subcontractors	**Performance Criteria**
Rolling Mill	Effective and efficient communication, Quote Response Time, Dispute Minimization, On-time delivery, Warranty Returns, Cost Certainty
Joist and Decking	Quote Response Time, Design Quality, Design Cooperation, Effective and efficient communication, Dispute Minimization, On-time delivery, Warranty Returns, Cost Certainty

Table 6-2 Continued

Suppliers/Subcontractors	***Performance Criteria***
Connector	Effective and efficient communication, Quote response time, Dispute minimization, On-time delivery, Warranty returns, Cost certainty
Galvanizer and light gauge steel work	Effective and efficient communication, Quote response time, Dispute minimization, On-time delivery, Warranty Returns, Cost certainty

The following sections introduce how the proposed method is applied in the ABC Steel Fabrication company for a specific project. The first part introduces the application of each of the cost and the performance sub-models, followed by the synthesis of both of them. The following section explains in detail the application of the performance sub-model to rolling mill and the joist and decking work packages. The performance sub-model is applied to other work packages such as the connector work package and the galvanizer and light gauge work packages similarly to the rolling mill work package. Therefore they are not explained in detail. Since there are many tables and figures used to record and calculate the performance of the subcontractors and suppliers, Table 6-3 is presented as a summary of the tables and figures used for explaining how the performance sub-model is applied. The tables listed in the same rows in Table 6-3 have similar functions although they are used for different work packages.

Applying the Model

Applying the Performance Model in the Rolling Mill Work Package

The company has two suppliers in the rolling mill work package. In order to make the right decisions, the company will not only select a supplier based on its quote but also on its performance. Figure 6-4 shows the performance model used in the rolling mill work package. In the hierarchical structure, the first level shows the primary business activities performed by the rolling mills: send quotes, process purchase order, and shipping. Because they have the same significance in the business transactions, the weight for each activity is identical (0.2). The second level is the performance criteria: effective and efficient communication, quoted customer response time, dispute minimization, on-time delivery, warranty returns, and cost certainty. The weights on the links between the activities (the first level) and performance criteria (the second level) represent the significance of each performance criteria on an activity. The weights are set by the company management personnel as shown in the Figure 6-4. The third level shows the suppliers. Some original data is collected and is used to evaluate the aggregate

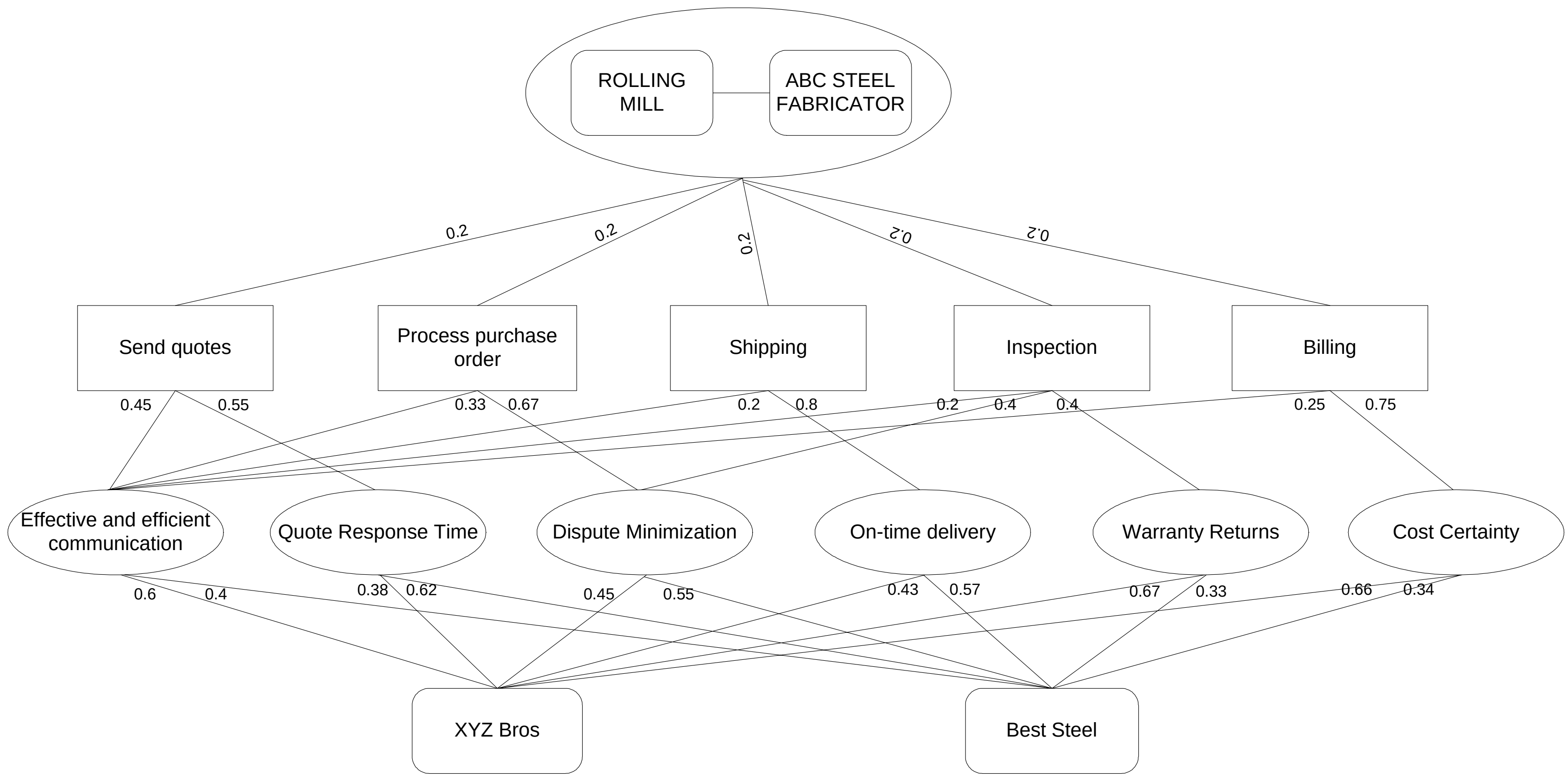


Figure 6-4. Performance Model Application in Rolling Mill Work Package

performance of the suppliers. The weights on the links between the second level and third level show the historical performance of the suppliers.

Table 6-3. List of Tables and Figures Used in Each Work Package

Work Package \ Performance Criteria	**Rolling Mill**	**Joist and Decking**	**Connector**	**Galvanizer and Light Gauge**
Performance Model Chart	Figure 6-4	Figure 6-5	Figure 6-6	Figure 6-7
Original Data Sheet	Table 6-4	Table 6-18	Table 6-36	Table 6-50
Weights for Effective and efficient communication	Table 6-5	Table 6-19	Table 6-37	Table 6-51
Weights for design cooperation		Table 6-20		
Design mistakes measurement scale		Table 6-21 Table 6-22		
Weights for design mistakes		Table 6-23		
Cost certainty Measurement Scale	Table 6-6	Table 6-24	Table 6-38	Table 6-52
Weights for Cost certainty	Table 6-7	Table 6-25	Table 6-39	Table 6-53
Quote response time measurement scale	Table 6-8	Table 6-26	Table 6-40	Table 6-54
Weights for quote response time	Table 6-9	Table 6-27	Table 6-41	Table 6-55
Dispute minimization measurement scale	Table 6-10 Table 6-11	Table 6-28 Table 6-29	Table 6-44 Table 6-45	Table 6-58 Table 6-59
Weights for dispute minimization	Table 6-12	Table 6-30	Table 6-46	Table 6-60
On-time delivery measurement scale	Table 6-13	Table 6-31	Table 6-42	Table 6-56
Weights for On-time delivery	Table 6-14	Table 6-32	Table 6-43	Table 6-57
Warranty return times measurement scale	Table 6-15 Table 6-16	Table 6-33 Table 6-34	Table 6-47 Table 6-48	Table 6-61 Table 6-62
Weights for Warranty return times	Table 6-17	Table 6-35	Table 6-49	Table 6-63

In order to quantify the intangible performance criteria and evaluate the tangible performance criteria, some original data are collected. Table 6-4 shows the original data collected for the rolling mill work package. Effective and efficient communication is judged by the company management using AHP. The final project price and initial quotes are used to assess the cost certainty performance criteria. The number of days taken by the suppliers to send quotes to ABC Steel Fabricator represents the Quote Response Time.

The Total Delivery Times and On-time Delivery Times on a project are collected to represent the delivery performance. The data is collected and compiled after a project is finished and a project is rewarded to only one supplier. Table 6-4 indicates that the project is awarded to XYZ Bros. Therefore, the data about the number of times of dispute, total delivery times, on-time delivery times, and warranty return times columns is zero. Next, how these original data are utilized to quantify the performance criteria is explained.

Effective and efficient communication performance: Table 6-5 represents the weights assigned by management to the suppliers based on Effective and Efficient Communication Performance by using AHP. For the case study project based on performance, the weights for XYZ Bros. and Best Steel are 0.833 and 0.167 respectively.

Table 6-4. Original Data Sheet for Project XXX (Rolling Mill Work Package)

Suppliers /Subcontractors	Effective and Efficient Communication	Final Project Price	Initial Quotes	Quote Response Time (Days)	Times of Dispute	Total Delivery Times	On-time Delivery Times	Warranty Return Times
XYZ Bros.	-	$42,000	$40,000	3	3	4	3	1
Best Steel	-	$0	$39,100	4	0	0	0	0

Table 6-5. Weights for Performance-Effective and Efficient Communication (Rolling Mill Work Package)

	Original Matrix		**Adjusted Matrix**		
	XYZ Bros.	Best Steel	XYZ Bros.	Best Steel	Weights (Row Avg.)
XYZ Bros.	1	5	0.833	0.833	0.833
Best Steel	1/5	1	0.167	0.167	0.167
Column Total	1.20	6	1.000	1.000	1.000

Cost Certainty: Table 6-6 shows the measurement scale for Cost Certainty performance criterion. The third column in Table 6-6 is the percentage of cost difference between the Final Project Price and Initial Quotes (see Table 6-4). By using Table 6-6 and AHP, the project manager assigns the original weights in Table 6-7. The final weights for cost certainty performance are obtained by calculating the original weights. For example, the initial quote for the sample project submitted by XYZ Bros. is $300,000, and that by Best Steel is $310,000. The sample project is awarded to XYZ Bros. After the project is completed, the final project price is $320,000. The cost difference is 0.067 (($320,000 - $300,000)/$300,000 = 0.067). In Table 6-7, insert 1/2 in the cell where XYZ Bros. is compared to Best Steel. That means XYZ Bros. is between “equally preferred” and “moderately un-preferred” compared to Best Steel in terms of cost certainty performance. When Best Steel is compared with XYZ Bros., insert 2, the reciprocal of the value 1/2 mention above, in the cell. The final adjusted weights on Cost Certainty for XYZ Bros. and Best Steel are 0.34 and 0.66 respectively.

Quote Response Time: Table 6-8 shows the Quote Response Time Measure Scale. The third column in Table 6-8 shows the Quote Response Time Comparison Percentage Range. Table 6-9 shows a comparison of results among the suppliers. For example, XYZ Bros. took three days to send their quote while Best Steel took four days to send their quotes (3/4 = 0.75). Using the AHP method, insert in Table 6-8 1/4 in the cell where Best Steel is compared with XYZ Bros. and insert 4 (the reciprocal of 1/4) in the cell where XYZ Bros. is compared with Best Steel. The final weights for these two suppliers on the Quote Response Time are 0.8 and 0.2 respectively as shown in Table 6-9.

Table 6-6. Cost Certainty Measurement Scale (Rolling Mill Work Package)

Verbal Judgment	**Numerical Rating**	**Cost Difference Percentage Range**	
Extremely preferred	9	-	-0.701
	8	-0.700	-0.601
Very strongly preferred	7	-0.600	-0.501
	6	-0.500	-0.401
Strongly preferred	5	-0.400	-0.301
	4	-0.300	-0.201
Moderately preferred	3	-0.200	-0.101
	2	-0.100	-0.001
Equally preferred	1	0.000	0.000
	1/2	0.001	0.100
Moderately un-preferred	1/3	0.101	0.200
	1/4	0.201	0.300
Strongly un-preferred	1/5	0.301	0.400
	1/6	0.401	0.500
Very strongly un-preferred	1/7	0.501	0.600
	1/8	0.601	0.700
Extremely un-preferred	1/9	0.701	-

Table 6-7. Weights for Performance-Cost Certainty (Rolling Mill Work Package)

	Original Matrix		**Adjusted Matrix**		
	XYZ Bros.	Best Steel	XYZ Bros.	Best Steel	Weights (Row Avg.)
XYZ Bros.	1	1 / 2	0.333	0.333	0.333
Best Steel	2	1	0.667	0.667	0.667
Column Total	3	1.5	1.000	1.000	1.000

Table 6-8. Quote Response Time Measure Scale (Rolling Mill Work Package)

Verbal Judgment	Numerical Rating	Customer Response Time Comparison Percentage Range (A/B)	
Equally preferred	1	1.000	1.000
	1/2	0.999	0.900
Moderately un-preferred	1/3	0.899	0.800
	1/4	0.799	0.700
Strongly un-preferred	1/5	0.699	0.600
	1/6	0.599	0.500
Very strongly un-preferred	1/7	0.499	0.400
	1/8	0.399	0.300
Extremely un-preferred	1/9	0.299	0.000

Dispute Minimization: For this performance criterion, there are two measurement scales based on the size of the projects. Table 6-10 shows the measurement scale for project sizes ranging from $5,000 to $100,000 and Table 6-11 for project sizes ranging from $100,000 and $250,000. The sample project size is within $100,000 to $250,000 and the times of dispute is three. So insert 1/4 in the cell of Table 6-12 where XYZ Bros. is compared with Best and insert 4 (reciprocal of 1/4) in the cell where Best is compared with XYZ Bros.. The final weights for these two suppliers on Dispute Minimization are 0.2 and 0.8 respectively.

Table 6-9. Weights for Performance- Quote Response Time (3/4=0.75) (Rolling Mill Work Package)

	Original Matrix		**Adjusted Matrix**		
	XYZ Bros.	Best Steel	XYZ Bros.	Best Steel	Weights (Row Avg.)
XYZ Bros.	1	4	0.800	0.800	0.800
Best Steel	1/4	1	0.200	0.200	0.200
Column Total	1.25	5	1.000	1.000	1.000

Table 6-10. Times of Dispute Measurement Scale ($5,000-$100,000) (Rolling Mill Work Package)

Verbal Judgment	**Numerical Rating**	**Times of Dispute**
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	1
	1/4	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

Table 6-11. Times of Dispute Measurement Scale ($100,000-250,000) (Rolling Mill Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	2
	1/4	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-12. Weights for Performance-Times of Dispute (Rolling Mill Work Package)

	Original Matrix		**Adjusted Matrix**		
	XYZ Bros.	Best Steel	XYZ Bros.	Best Steel	Weights (Row Avg.)
XYZ Bros.	1	1/4	0.20	0.20	0.20
Best Steel	4	1	0.80	0.80	0.80
Column Total	5	1.25	1.00	1.00	1.00

On-time Delivery: Table 6-13 shows the measurement scale for delivery performance. It shows the on-time delivery percentage range and its corresponding numerical rating. For example, XYZ Bros. in this sample project delivers materials to the fabricator a total of four times, but the delivery is on time only three times. The numerical rating of this performance is 4 (3/4=0.75). Insert 4 in the cell of Table 6-14 where XYZ Bros. is compared with Best Steel and insert 1/4 (the reciprocal of 4) in the cell where Best Steel is compared with XYZ Bros.

Warranty Returns: This performance criterion is related to the quality of delivered material or products. The measurement scale is divided into two based on the project size. Table 6-15 shows the measurement scale for project sizes ranging from $5,000 to $100,000 and Table 6-16 for project sizes ranging from $100,000 to $250,000. The sample project is within $100,000 to $250,000 and the warranty return times is one, so

Table 6-13. On-time Delivery Measurement Scale (Rolling Mill Work Package)

Verbal Judgment	Numerical Rating	On-time Delivery Percentage Range	
Extremely preferred	9	1.000	0.9751
	8	0.975	0.951
Very strongly preferred	7	0.900	0.851
	6	0.850	0.801
Strongly preferred	5	0.800	0.751
	4	0.750	0.701
Moderately preferred	3	0.700	0.651
	2	0.650	0.601
Equally preferred	1	0.600	0.600
	1/2	0.601	0.500
Moderately un-preferred	1/3	0.501	0.400
	1/4	0.401	0.350
Strongly un-preferred	1/5	0.351	0.300
	1/6	0.301	0.250
Very strongly un-preferred	1/7	0.251	0.200
	1/8	0.201	0.150
Extremely un-preferred	1/9	0.151	0.000

Table 6-14. Weights for Performance- On-time Delivery (Rolling Mill Work Package)

	Original Matrix		Adjusted Matrix		
	XYZ Bros.	Best Steel	XYZ Bros.	Best Steel	Weights (Row Avg.)
XYZ Bros.	1	4	0.800	0.800	0.800
Best Steel	1/4	1	0.200	0.200	0.200
Column Total	1.25	5	1.000	1.000	1.000

Table 6-15. Warranty Return Times Measurement Scale ($5,000-$100,000) (Rolling Mill Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	1
	1/4	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

the corresponding rating is 1/2. Insert 1/2 in the cell of Table 6-17 where XYZ Bros. is compared to Best Steel and insert 2 (the reciprocal of 1/2) in the cell where Best Steel is compared with XYZ Bros.

After collecting all the original data for performance on the completed projects and using the same method to evaluate the performance of each supplier in this work package, the average weights of each performance criterion for each supplier would be calculated through the following formula:

$$ps_{ij}^{n} = (\sum_{k=1}^{N} W_{ijk-n}) / N \qquad \text{(Equation 6-1)}$$

where:

n: represents nth work package.

ps_{ij}^{n} : the average weight of the ith supplier on the jth performance criterion in the nth work package.

W_{ijk-n} : the weight of the ith supplier on the jth performance criterion as perceived by the kth project for the nth work package.

N : the number of projects.

Table 6-16. Warranty Return Times Measurement Scale ($100,000-250,000) (Rolling Mill Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	2
	1/4	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-17. Weights for Performance-Warranty Return Times (Rolling Mill Work Package)

	Original Matrix		**Adjusted Matrix**		
	XYZ Bros.	Best Steel	XYZ Bros.	Best Steel	Weights (Row Avg.)
XYZ Bros.	1	1 / 2	0.333	0.333	0.333
Best Steel	2	1	0.667	0.667	0.667
Column Total	3	1.5	1.000	1.000	1.000

Several matrix operations are used to get the aggregate performance of each supplier in this work package. A^n represents the activity matrix. The superscript of A^n identifies the work package. As described above there are five primary activities in the rolling mill work package, and each activity has identical significance in the business transactions (Figure 4). Therefore the activity matrix for rolling mill is expressed as:

$$A^1 = \begin{bmatrix} a_1^1 & a_2^1 & a_3^1 & a_4^1 & a_5^1 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix} \qquad \text{(Equation 6-2)}$$

where:

A^1 : activity matrix for rolling mill work package. The rolling mill work package is identified as work package 1.

a_r^1: activities in rolling mill work package, r=1, 2…5

AP^n : represents the relationship between activities and performances, called activity-performance matrix. The superscript of AP^n refers to the work package, n. The management of ABC Steel Fabricator set six performance criteria to measure the activities in the rolling mill work package (Figure 6-4). The activity-performance matrix for the rolling mill work package is expressed as:

$$AP^1 = \begin{bmatrix} ap_{11}^1 & ap_{12}^1 & ap_{13}^1 & ap_{14}^1 & ap_{15}^1 & ap_{16}^1 \\ ap_{21}^1 & ap_{22}^1 & ap_{23}^1 & ap_{24}^1 & ap_{25}^1 & ap_{26}^1 \\ ap_{31}^1 & ap_{32}^1 & ap_{33}^1 & ap_{34}^1 & ap_{35}^1 & ap_{36}^1 \\ ap_{41}^1 & ap_{42}^1 & ap_{43}^1 & ap_{44}^1 & ap_{45}^1 & ap_{46}^1 \\ ap_{51}^1 & ap_{52}^1 & ap_{53}^1 & ap_{54}^1 & ap_{55}^1 & ap_{56}^1 \end{bmatrix} = \begin{bmatrix} 0.45 & 0.55 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.33 & 0.00 & 0.67 & 0.00 & 0.00 & 0.00 \\ 0.20 & 0.00 & 0.00 & 0.80 & 0.00 & 0.00 \\ 0.20 & 0.00 & 0.40 & 0.00 & 0.40 & 0.00 \\ 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.75 \end{bmatrix}$$

(Equation 6-3)

where:

AP^1: activity-performance matrix for rolling mill work package. The rolling mill work package is identified as work package 1.

ap_{rm}^1: weight of rth activity on the mth performance criteria in the rolling mill work package. Here, r = 1, 2…5, m = 1, 2, …6

PS^n: is the performance-sub matrix representing the suppliers' performance on each performance criterion, called. The superscript of PS^n refers to work package, n.

There are two suppliers measured on six performance criteria (Figure 6-4). The performance-sub matrix for the rolling mill work package is expressed as:

$$PS^1 = \begin{bmatrix} ps_{11}^1 & ps_{12}^1 \\ ps_{21}^1 & ps_{22}^1 \\ ps_{31}^1 & ps_{32}^1 \\ ps_{41}^1 & ps_{42}^1 \\ ps_{51}^1 & ps_{52}^1 \\ ps_{61}^1 & ps_{62}^1 \end{bmatrix} = \begin{bmatrix} 0.60 & 0.40 \\ 0.38 & 0.62 \\ 0.45 & 0.55 \\ 0.43 & 0.57 \\ 0.67 & 0.33 \\ 0.66 & 0.34 \end{bmatrix}$$

(Equation 6-4)

where:

PS^1: performance-sub matrix for rolling mill work package. The rolling mill work package is identified as work package 1.

ps_{mn}^1: weight of nth supplier on the mth performance criteria in the rolling mill work package. Here, m = 1, 2…6, m = 1, 2.

The values in the Equation (6-4) are the average values after the management evaluates suppliers' performance on the completed projects. They are calculated as shown in Equation (6-1). Finally, the aggregate performance of each supplier is represented by S^n, called aggregate performance matrix. The superscript of S^n refers to work package, n. It is the result obtained by multiplying matrices $A^n \bullet AP^n \bullet PS^n$. The aggregate performance of suppliers in the rolling mill work package is expressed as:

$$S^1 = A^1 \bullet AP^1 \bullet PS^1 = \begin{bmatrix} 0.5311 & 0.4689 \end{bmatrix} \qquad \text{(Equation 6-5)}$$

Applying the Performance Model to the Joist-Decking Work Package

Table 6-18 is the sample of original data collected for an ABC Steel Fabrication project. Six activities are identified in this work package:

- Send quotes
- Detailing
- Fabrication
- Shipping
- Inspection
- Billing

Eight performance criteria are identified:

- Effective and efficient communication
- Design cooperation
- Design quality
- Cost certainty
- Quote response time
- Dispute minimization
- On-time delivery
- Warranty returns

Two subcontractors SC Inc., and Steel Decking usually are invited to work with ABC Steel Fabricator. The hierarchical performance model for this work package is shown in Figure 6-5. Six activities are considered identical significance in this work package. So weight value 0.167 is assigned to each of the activity. Based on the judgment of management, different weights are assigned on the links between the activity and the performance criteria shown in Figure 6-5.

Effective and Efficient Communication/Design Cooperation: By using a similar approach as that for the rolling mill work package, management can evaluate the subcontractors on Effective and Efficient Communication and Design Cooperation (Table 6-19 and 6-20).

Design Quality: Based on the project size the measurement scale for this performance criterion is divided into two. One is for the project sizes ranging from $5,000 to $50,000(Table 6-21). The other is for project sizes ranging from $50,000 to $150,000(Table 6-22). The project whose size is within $5,000 to $50,000 range is awarded to Steel Decking Company, and the numbers of design mistakes is 3. Hence, insert 1/7 in the cell of Table 6-23 where Steel Decking, Inc. is compared with SC Inc. By using the AHP method, populate the other cells and the adjusted weights of SC Inc. and Steel Decking will respectively be 0.875 and 0.125.

Table 6-18. Original Data Sheet for Project XXX (Joist-Decking Work Package)

Suppliers/ Subcontractors	Effective and Efficient Communication	Design Cooperation	Numbers of Design Mistakes	Final Project Price	Initial Quotes	Quote Response Time (Days)	Times of Dispute	Total Delivery Times	On-time Delivery Times	Warranty Return Times
SC Inc.	-	`-	-	$0	$48,900	4	-	-	-	-
Steel Decking	-	-	3	$50,000	$48,000	5	2	4	4	1

Table 6-19. Weights for Performance-Effective and Efficient Communication (Joist-Decking Work Package)

	Original Matrix		**Adjusted Matrix**		
	SC Inc.	Steel Decking	SC Inc.	Steel Decking	Weights (Row Avg.)
SC Inc.	1	3	0.75	0.75	0.75
Steel Decking	1/3	1	0.25	0.25	0.25
Column Total	1.33	4	1.00	1.00	1.00

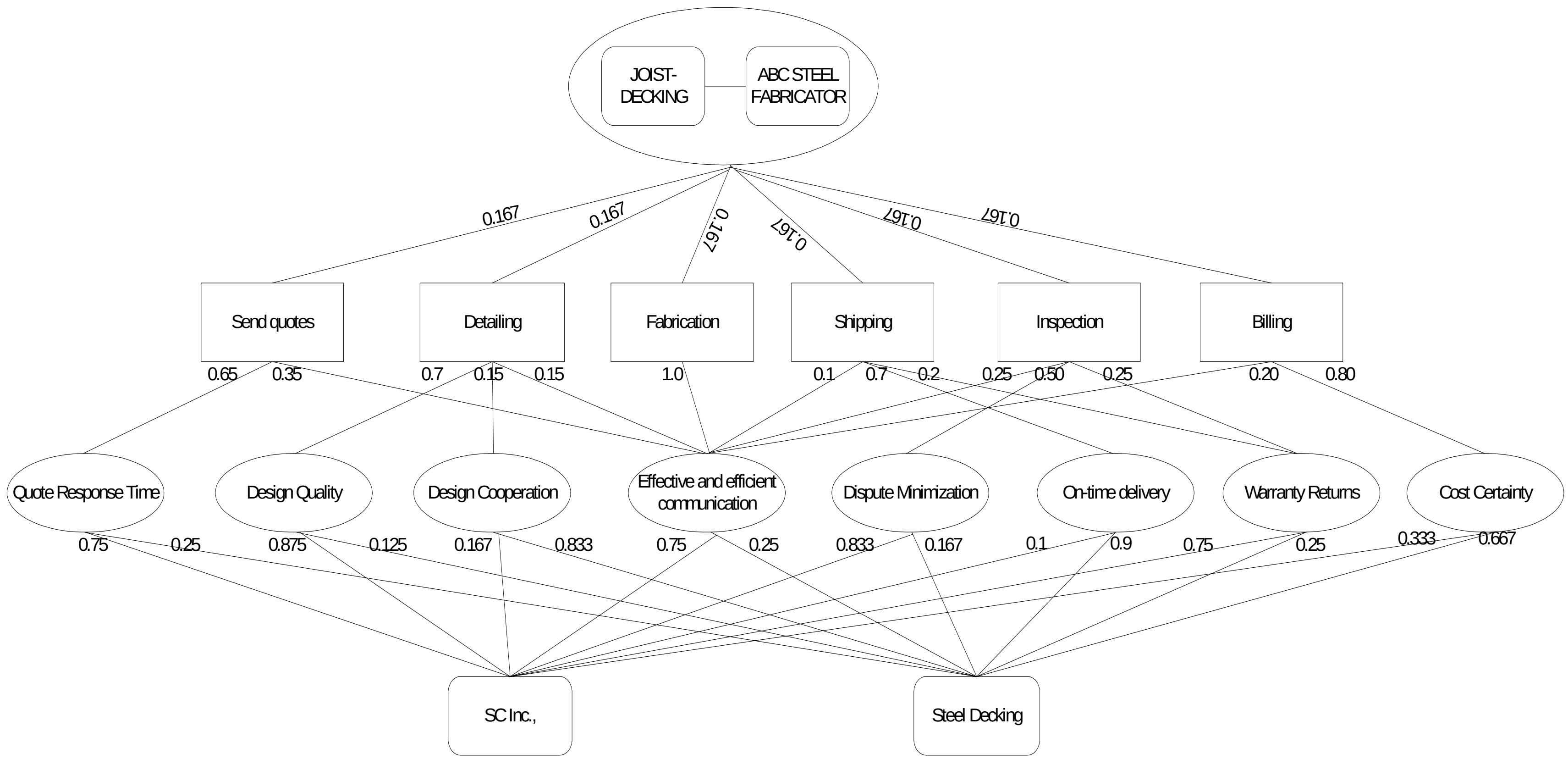


Figure 6-5. Performance Model Application in Joist-decking Work Package

Table 6-20. Weights for Design Cooperation (Joist-Decking Work Package)

	Original Matrix		Adjusted Matrix		
	SC Inc.	Steel Decking	SC Inc.	Steel Decking	Weights (Row Avg.)
SC Inc.	1	1/5	0.167	0.167	0.167
Steel Decking	5	1	0.833	0.833	0.833
Column Total	6	1.2	1.000	1.000	1.000

Table 6-21. Numbers of Design Mistakes Measurement Scale ($5,000-$50,000) (Joist-Decking Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	1
	1/4	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

Table 6-22. Numbers of Design Mistakes Measurement Scale ($50,000-150,000) (Joist-Decking Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	2
	1/4	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-23. Weights for Performance-Design Quality Performance (Joist-Decking Work Package)

	Original Matrix		Adjusted Matrix		
	SC Inc.	Steel Decking	SC Inc.	Steel Decking	Weights (Row Avg.)
SC Inc.	1	7	0.875	0.875	0.875
Steel Decking	1/7	1	0.125	0.125	0.125
Column Total	1.143	8	1.000	1.000	1.000

Table 6-24. Cost Certainty Measurement Scale (Joist-Decking Work Package)

Verbal Judgment	Numerical Rating	Cost Difference Percentage Range	
Extremely preferred	9	-	-0.701
	8	-0.700	-0.601
Very strongly preferred	7	-0.600	-0.501
	6	-0.500	-0.401
Strongly preferred	5	-0.400	-0.301
	4	-0.300	-0.201
Moderately preferred	3	-0.200	-0.101
	2	-0.100	-0.001
Equally preferred	1	0.000	0.000
	1/2	0.001	0.100
Moderately un-preferred	1/3	0.101	0.200
	1/4	0.201	0.300
Strongly un-preferred	1/5	0.301	0.400
	1/6	0.401	0.500
Very strongly un-preferred	1/7	0.501	0.600
	1/8	0.601	0.700
Extremely un-preferred	1/9	0.701	-

Table 6-25. Weights for Performance-Cost Certainty (Joist-Decking Work Package)

	Original Matrix		Adjusted Matrix		
	SC Inc.	Steel Decking	SC Inc.	Steel Decking	Weights (Row Avg.)
SC Inc.	1	1 / 2	0.333	0.333	0.333
Steel Decking	2	1	0.667	0.667	0.667
Column Total	3	1.5	1.000	1.000	1.000

Table 6-26. Quote Response Time Measure Scale (Joist-Decking Work Package)

Verbal Judgment	Numerical Rating	Quoted Customer Response Percentage Range	
Equally preferred	1	1.000	1.000
	1/2	0.999	0.900
Moderately un-preferred	1/3	0.899	0.800
	1/4	0.799	0.700
Strongly un-preferred	1/5	0.699	0.600
	1/6	0.599	0.500
Very strongly un-preferred	1/7	0.499	0.400
	1/8	0.399	0.300
Extremely un-preferred	1/9	0.299	0.000

Table 6-27. Weights for Performance- Quote Response Time (4/5=0.8) (Joist-Decking Work Package)

	Original Matrix		**Adjusted Matrix**		
	SC Inc.	Steel Decking	SC Inc.	Steel Decking	Weights (Row Avg.)
SC Inc.	1	3	0.750	0.750	0.750
Steel Decking	1/3	1	0.250	0.250	0.250
Column Total	1.333	4	1.000	1.000	1.000

Table 6-28. Times of Dispute Measurement Scale ($5,000-$50,000) (Joist-Decking Work Package)

Verbal Judgment	**Numerical Rating**	**Times of Dispute**
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	1
	1/4	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

Table 6-29. Times of Dispute Measurement Scale ($50,000-150,000) (Joist-Decking Work Package)

Verbal Judgment	**Numerical Rating**	**Times of Dispute**
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	2
	1/4	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-30. Weights for Performance-Times of Dispute (Joist-Decking Work Package)

	Original Matrix		**Adjusted Matrix**		
	SC Inc.	Steel Decking	SC Inc.	Steel Decking	Weights (Row Avg.)
SC Inc.	1	5	0.833	0.833	0.833
Steel Decking	1/5	1	0.167	0.167	0.167
Column Total	1.2	6	1.000	1.000	1.000

Table 6-31. On-time Delivery Measurement Scale (Joist-Decking Work Package)

Verbal Judgment	Numerical Rating	On-time Delivery Percentage Range	
Extremely preferred	9	1.000	0.9751
	8	0.975	0.951
Very strongly preferred	7	0.900	0.851
	6	0.850	0.801
Strongly preferred	5	0.800	0.751
	4	0.750	0.701
Moderately preferred	3	0.700	0.651
	2	0.650	0.601
Equally preferred	1	0.600	0.600
	1/2	0.601	0.500
Moderately un-preferred	1/3	0.501	0.400
	1/4	0.401	0.350
Strongly un-preferred	1/5	0.351	0.300
	1/6	0.301	0.250
Very strongly un-preferred	1/7	0.251	0.200
	1/8	0.201	0.150
Extremely un-preferred	1/9	0.151	0.000

Table 6-32. Weights for Performance- On-time Delivery (Joist-Decking Work Package)

	Original Matrix		Adjusted Matrix		
	SC Inc.	Steel Decking	SC Inc.	Steel Deckings	Weights (Row Avg.)
SC Inc.	1	1/9	0.10	0.10	0.10
Steel Decking	9	1	0.90	0.90	0.90
Column Total	10	1.111	1.00	1.00	1.00

Table 6-33. Warranty Return Times Measurement Scale ($5,000-$50,000) (Joist-Decking Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	1
	1/4	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

Table 6-34. Warranty Return Times Measurement Scale ($50,000-150,000) (Joist-Decking Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	2
	1/4	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-35. Weights for Performance-Warranty Return Times (Joist-Decking Work Package)

	Original Matrix		Adjusted Matrix		
	SC Inc.	Steel Decking	SC Inc.	Steel Decking	Weights (Row Avg.)
SC Inc.	1	3	0.750	0.750	0.750
Steel Decking	1/3	1	0.250	0.250	0.250
Column Total	1.333	4	1.000	1.000	1.000

Cost Certainty/Quote Response Time/Dispute Minimization/On-time Delivery/Warranty Returns: Subcontractors on these performance criteria are evaluated by the similar way described in the Rolling Mill work package. They are shown in Table 6-24 to Table 6-35. The average weights of each subcontractor on each performance criterion can be calculated by Equation (6-1). Using similar matrix operations the subcontractors aggregate performance can be represented as:

$$S^2 = A^2 \bullet AP^2 \bullet PS^2 = \begin{bmatrix} 0.6267 & 0.3753 \end{bmatrix} \qquad \text{(Equation 6-6)}$$

where:

$$A^2 = \begin{bmatrix} a_1^2 & a_2^2 & a_3^2 & a_4^2 & a_5^2 & a_6^2 \end{bmatrix} = \begin{bmatrix} 0.167 & 0.167 & 0.167 & 0.167 & 0.167 & 0.167 \end{bmatrix}$$

$$AP^2 = \begin{bmatrix} ap_{11}^2 & ap_{12}^2 & ap_{13}^2 & ap_{14}^2 & ap_{15}^2 & ap_{16}^2 & a_{17}^2 & ap_{18}^2 \\ ap_{21}^2 & ap_{22}^2 & ap_{23}^2 & ap_{24}^2 & ap_{25}^2 & ap_{26}^2 & ap_{27}^2 & ap_{28}^2 \\ ap_{31}^2 & ap_{32}^2 & ap_{33}^2 & ap_{34}^2 & ap_{35}^2 & ap_{36}^2 & ap_{37}^2 & ap_{38}^2 \\ ap_{41}^2 & ap_{42}^2 & ap_{43}^2 & ap_{44}^2 & ap_{45}^2 & ap_{46}^2 & ap_{47}^2 & ap_{48}^2 \\ ap_{51}^2 & ap_{52}^2 & ap_{53}^2 & ap_{54}^2 & ap_{55}^2 & ap_{56}^2 & ap_{57}^2 & ap_{58}^2 \\ ap_{61}^2 & ap_{62}^2 & ap_{63}^2 & ap_{64}^2 & ap_{65}^2 & ap_{66}^2 & ap_{67}^2 & ap_{68}^2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.65 & 0.00 & 0.00 & 0.35 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.70 & 0.15 & 0.15 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.10 & 0.00 & 0.70 & 0.20 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.25 & 0.50 & 0.00 & 0.25 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.20 & 0.00 & 0.00 & 0.00 & 0.80 \end{bmatrix}$$

$$PS^2 = \begin{bmatrix} ps_{11}^2 & ps_{12}^2 \\ ps_{21}^2 & ps_{22}^2 \\ ps_{31}^2 & ps_{32}^2 \\ ps_{41}^2 & ps_{42}^2 \\ ps_{51}^2 & ps_{52}^2 \\ ps_{61}^2 & ps_{62}^2 \\ ps_{71}^2 & ps_{72}^2 \\ ps_{81}^2 & ps_{82}^2 \end{bmatrix} = \begin{bmatrix} 0.750 & 0.250 \\ 0.875 & 0.125 \\ 0.167 & 0.833 \\ 0.750 & 0.250 \\ 0.833 & 0.167 \\ 0.100 & 0.900 \\ 0.750 & 0.250 \\ 0.333 & 0.667 \end{bmatrix}$$

Applying the Performance Sub-model to the Connector Work Package and the Galvanizer Work Package

The work packages left to be analyzed are the connector work package, the galvanizer work package, and the special treatment work package. The activities and performance criteria identified in these work packages are the same. The activities are:

- Send quotes
- Process purchase order
- Inspection
- Billing

The performance criteria are:

- Effective and efficient communication
- Quote response time

- Dispute minimization
- On-time delivery
- Warranty returns
- Cost certainty

Figures 6-6, 6-7, and 6-8 represent the hierarchical performance model for the connector work package, galvanizer work package, and special treatment work package. By using a method similar to that described for the rolling mill work package, the subcontractors' weights for the identified performance criteria in each of these work packages are calculated. Tables 6-36 to 6-49 represent data used for the connector work package, Tables 6-50 to 6-63 represent data for the galvanizer work package, and Tables 6-64 to 6-75 represent data for special treatment work package. Equations 6-7, 6-8, and 6-9 show the aggregate performance of subcontractors in these work packages.

$$S^3 = A^3 \bullet AP^3 \bullet PS^3 = [0.5335 \quad 0.4665] \qquad \text{(Equation 6-7)}$$

where:

$$A^3 = \begin{bmatrix} a_1^3 & a_2^3 & a_3^3 & a_4^3 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

$$AP^3 = \begin{bmatrix} ap_{11}^3 & ap_{12}^3 & ap_{13}^3 & ap_{14}^3 & ap_{15}^3 & ap_{16}^3 \\ ap_{21}^3 & ap_{22}^3 & ap_{23}^3 & ap_{24}^3 & ap_{25}^3 & ap_{26}^3 \\ ap_{31}^3 & ap_{32}^3 & ap_{33}^3 & ap_{34}^3 & ap_{35}^3 & ap_{36}^3 \\ ap_{41}^3 & ap_{42}^3 & ap_{43}^3 & ap_{44}^3 & ap_{45}^3 & ap_{46}^3 \end{bmatrix} = \begin{bmatrix} 0.45 & 0.55 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.33 & 0.00 & 0.00 & 0.67 & 0.00 & 0.00 \\ 0.20 & 0.00 & 0.40 & 0.00 & 0.40 & 0.00 \\ 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.75 \end{bmatrix}$$

$$PS^3 = \begin{bmatrix} ps_{11}^3 & ps_{12}^3 \\ ps_{21}^3 & ps_{22}^3 \\ ps_{31}^3 & ps_{32}^3 \\ ps_{41}^3 & ps_{42}^3 \\ ps_{51}^3 & ps_{52}^3 \\ ps_{61}^3 & ps_{62}^3 \end{bmatrix} = \begin{bmatrix} 0.60 & 0.40 \\ 0.55 & 0.45 \\ 0.30 & 0.70 \\ 0.77 & 0.23 \\ 0.45 & 0.55 \\ 0.37 & 0.63 \end{bmatrix}$$

$$S^4 = A^4 \bullet AP^4 \bullet PS^4 = \begin{bmatrix} 0.5346 & 0.4654 \end{bmatrix} \qquad \text{(Equation 6-8)}$$

where:

$$A^4 = \begin{bmatrix} a_1^4 & a_2^4 & a_3^4 & a_4^4 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

$$AP^4 = \begin{bmatrix} ap_{11}^4 & ap_{12}^4 & ap_{13}^4 & ap_{14}^4 & ap_{15}^4 & ap_{16}^4 \\ ap_{21}^4 & ap_{22}^4 & ap_{23}^4 & ap_{24}^4 & ap_{25}^4 & ap_{26}^4 \\ ap_{31}^4 & ap_{32}^4 & ap_{33}^4 & ap_{34}^4 & ap_{35}^4 & ap_{36}^4 \\ ap_{41}^4 & ap_{42}^4 & ap_{43}^4 & ap_{44}^4 & ap_{45}^4 & ap_{46}^4 \end{bmatrix} = \begin{bmatrix} 0.45 & 0.55 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.33 & 0.00 & 0.00 & 0.67 & 0.00 & 0.00 \\ 0.20 & 0.00 & 0.40 & 0.00 & 0.40 & 0.00 \\ 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.75 \end{bmatrix}$$

$$PS^4 = \begin{bmatrix} ps_{11}^4 & ps_{12}^4 \\ ps_{21}^4 & ps_{22}^4 \\ ps_{31}^4 & ps_{32}^4 \\ ps_{41}^4 & ps_{42}^4 \\ ps_{51}^4 & ps_{52}^4 \\ ps_{61}^4 & ps_{62}^4 \end{bmatrix} = \begin{bmatrix} 0.55 & 0.45 \\ 0.50 & 0.50 \\ 0.70 & 0.30 \\ 0.40 & 0.60 \\ 0.66 & 0.34 \\ 0.50 & 0.50 \end{bmatrix}$$

Figure 6-8 shows all the possible team combinations of material suppliers and subcontractors, which are identified as alternatives. There are four levels from the left to the right of the tree structure shiwn in Figure 6-8. Each level represents one work package. The aggregate performance weights of each material supplier and subcontractor are shown on the corresponding links in Figure 6-8. The final performance weights for each of the alternatives come out by multiplying the related aggregate performance weights for each alternative. They are shown on the left-most side of the Figure 6-8.

Table 6-36. Original Data Sheet for Project XXX (Connector Work Package)

Suppliers/Sub contractors	**Effective and Efficient Communication**	**Quote Response Time (Days)**	**Final Project Price**	**Initial Quotes**	**Promised Process Time**	**Actual Process Time**	**Times of Dispute**	**Warranty Return Times**
A Connector	-	2	$9,500	$9,500	3	4	1	-
B Connector	-	3	$0	$9,800	3	3	-	-

Table 6-37. Weights for Performance-Effective and Efficient Communication (Connector Work Package)

	Original Matrix		Adjusted Matrix		
	A Connector	B Connector	A Connector	B Connector	Weights (Row Avg.)
A Connector	1	6	0.857	0.857	0.857
B Connector	1/6	1	0.143	0.143	0.143
Column Total	1.167	7	1.000	1.000	1.000

Table 6-38. Cost Certainty Measurement Scale (Connector Work Package)

Verbal Judgment	Numerical Rating	Cost Difference Percentage Range	
Extremely preferred	9	-	-0.701
	8	-0.700	-0.601
Very strongly preferred	7	-0.600	-0.501
	6	-0.500	-0.401
Strongly preferred	5	-0.400	-0.301
	4	-0.300	-0.201
Moderately preferred	3	-0.200	-0.101
	2	-0.100	-0.001
Equally preferred	1	0.000	0.000
	1/2	0.001	0.100
Moderately un-preferred	1/3	0.101	0.200
	1/4	0.201	0.300
Strongly un-preferred	1/5	0.301	0.400
	1/6	0.401	0.500
Very strongly un-preferred	1/7	0.501	0.600
	1/8	0.601	0.700
Extremely un-preferred	1/9	0.701	-

Table 6-39. Weights for Performance-Cost Certainty (Connector Work Package)

	Original Matrix		Adjusted Matrix		
	A Connector	B Connector	A Connector	B Connector	Weights (Row Avg.)
A Connector	1	1	0.50	0.50	0.50
B Connector	1	1	0.50	0.50	0.50
Column Total	2	2	1.00	1.00	1.00

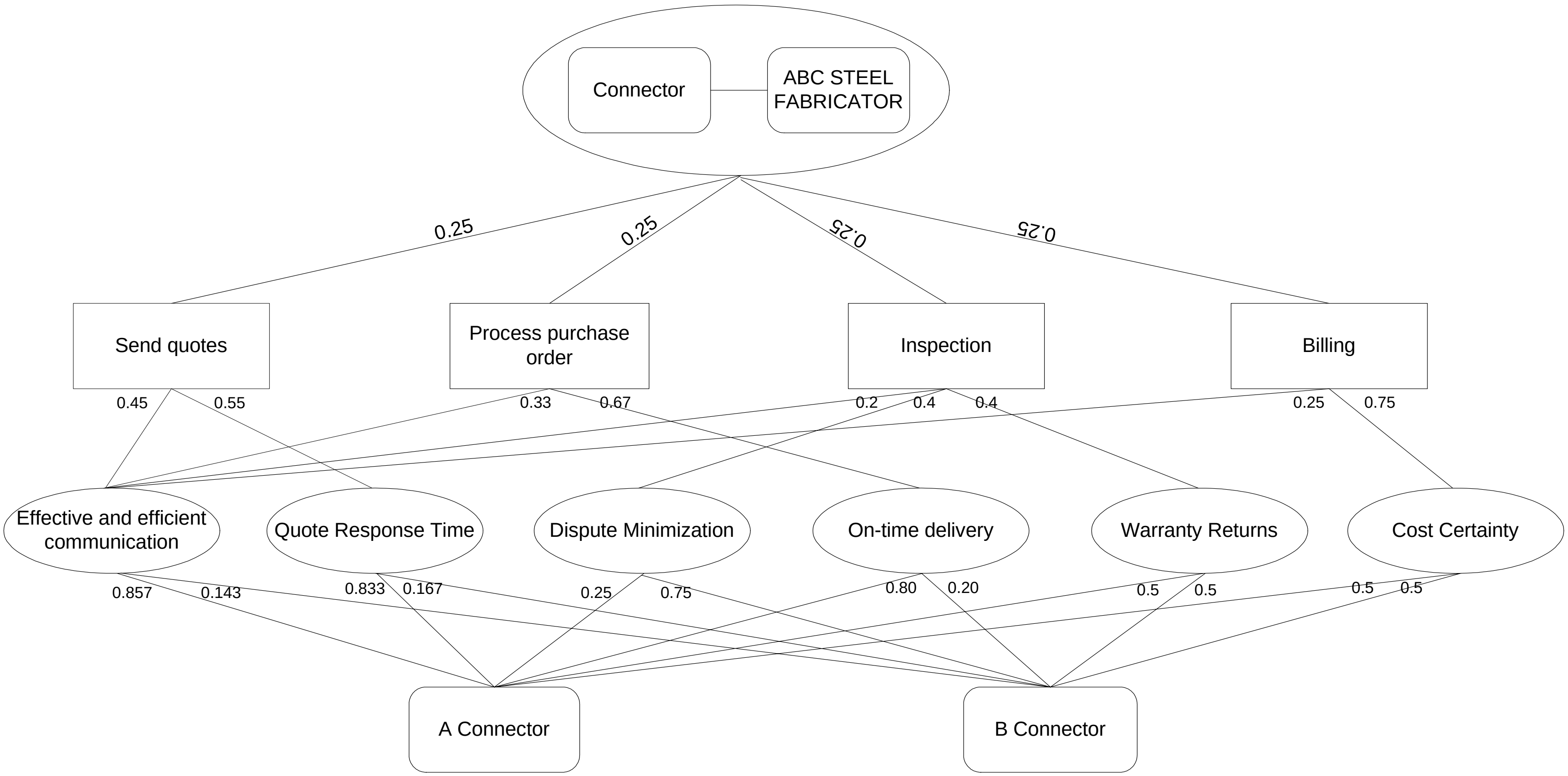


Figure 6-6. Performance Model Application in Connector Work Package

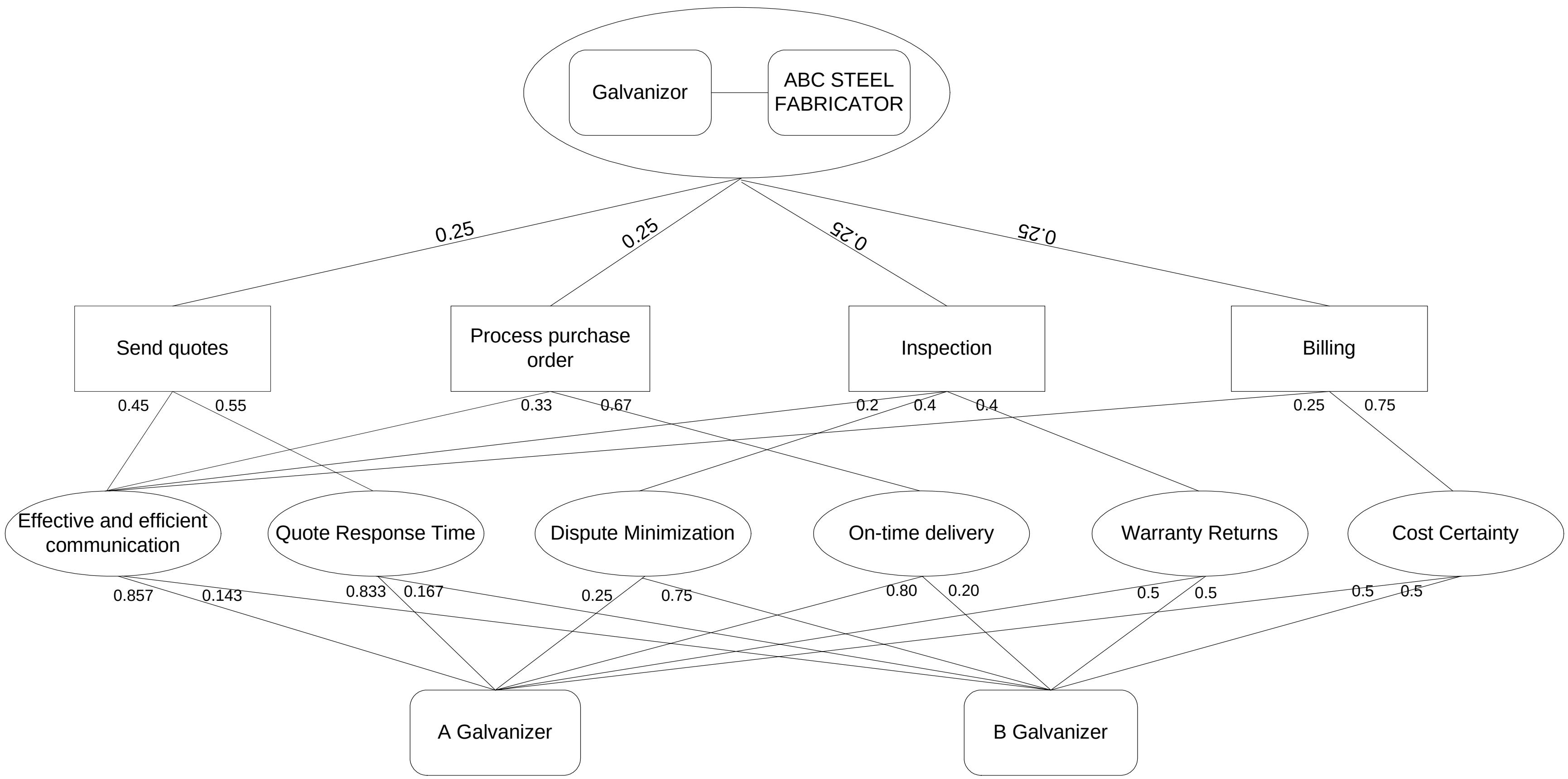


Figure 6-7. Performance Model Application in Galvanizer Work Package

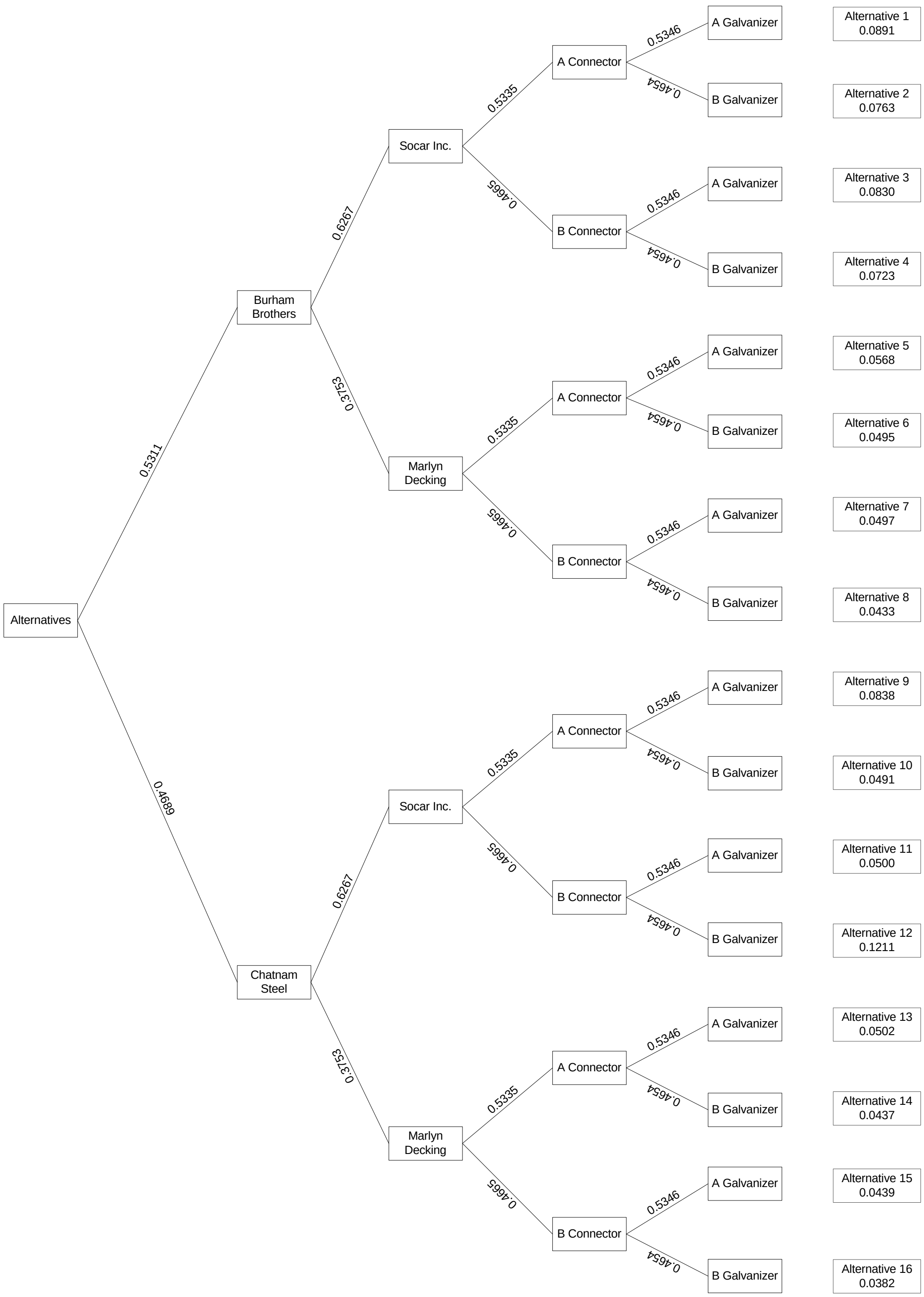


Figure 6-8. Supply Chain Alternatives and Performance Weights of the Material Suppliers and Subcontractors

Table 6-40. Quoted Customer Response Time Measure Scale (Connector Work Package)

Verbal Judgment	Numerical Rating	Quoted Customer Response Percentage Range	
Equally preferred	1	1.000	1.000
	1/2	0.999	0.900
Moderately un-preferred	1/3	0.899	0.800
	1/4	0.799	0.700
Strongly un-preferred	1/5	0.699	0.600
	1/6	0.599	0.500
Very strongly un-preferred	1/7	0.499	0.400
	1/8	0.399	0.300
Extremely un-preferred	1/9	0.299	0.000

Table 6-41. Weights for Performance- Quoted Customer Response Time (2/3=0.8) (Connector Work Package)

	Original Matrix		Adjusted Matrix		
	A Connector	B Connector	A Connector	B Connector	Weights (Row Avg.)
A Connector	1	5	0.833	0.833	0.833
B Connector	1/5	1	0.167	0.167	0.167
Column Total	1.2	6	1.000	1.000	1.000

Table 6-42. On-time Delivery Measurement Scale (Connector Work Package)

Verbal Judgment	Numerical Rating	On-time Delivery Percentage Range	
Extremely preferred	9	1.000	0.9751
	8	0.975	0.951
Very strongly preferred	7	0.900	0.851
	6	0.850	0.801
Strongly preferred	5	0.800	0.751
	4	0.750	0.701
Moderately preferred	3	0.700	0.651
	2	0.650	0.601
Equally preferred	1	0.600	0.600
	1/2	0.601	0.500
Moderately un-preferred	1/3	0.501	0.400
	1/4	0.401	0.350
Strongly un-preferred	1/5	0.351	0.300
	1/6	0.301	0.250
Very strongly un-preferred	1/7	0.251	0.200
	1/8	0.201	0.150
Extremely un-preferred	1/9	0.151	0.000

Table 6-43. Weights for Performance- On-time Delivery (promised/actual = 3/4 = 0.75, since nominator must be less than denominator) (Connector Work Package)

	Original Matrix		Adjusted Matrix		
	A Connector	B Connector	A Connector	B Connector	Weights (Row Avg.)
A Connector	1	4	0.80	0.80	0.80
B Connector	1 / 4	1	0.20	0.20	0.20
Column Total	1.25	5	1.00	1.00	1.00

Table 6-44. Times of Dispute Measurement Scale ($5,000-$50,000) (Connector Work Package)

Verbal Judgment	**Numerical Rating**	**Times of Dispute**
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	1
	1/4	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

Table 6-45. Times of Dispute Measurement Scale ($50,000-150,000) (Connector Work Package)

Verbal Judgment	**Numerical Rating**	**Times of Dispute**
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	2
	1/4	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-46. Weights for Performance-Times of Dispute (Connector Work Package)

	Original Matrix		Adjusted Matrix		
	A Connector	B Connector	A Connector	B Connector	Weights (Row Avg.)
A Connector	1	1/3	0.250	0.250	0.250
B Connector	3	1	0.750	0.750	0.750
Column Total	4	1.333	1.000	1.000	1.000

Table 6-47. Warranty Return Times Measurement Scale ($5,000-$50,000) (Connector Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	1
	1/4	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

Table 6-48. Warranty Return Times Measurement Scale ($50,000-150,000) (Connector Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	1/2	1
Moderately un-preferred	1/3	2
	1/4	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-49. Weights for Performance-Warranty Return Times (Connector Work Package)

	Original Matrix		Adjusted Matrix		
	A Connector	B Connector	A Connector	B Connector	Weights (Row Avg.)
A Connector	1	1	0.50	0.50	0.50
B Connector	1	1	0.50	0.50	0.50
Column Total	2	2	1.00	1.00	1.00

Table 6-50. Original Data Sheet for Project XXX (Galvanizer Work Package)

Suppliers/ Subcontractors	**Effective and Efficient Communication**	**Quoted Customer Response Time (Days)**	**Final Project Price**	**Initial Quotes**	**Promised Process Time**	**Actual Process Time**	**Times of Dispute**	**Warranty Return Times**
A Galvanizer	-	2	$12,500	$12,500	3	4	1	-
B Galvanizer	-	3	$0	$13,000	3	3	-	-

Table 6-51. Weights for Performance-Effective and Efficient Communication (Galvanizer Work Package)

	Original Matrix		Adjusted Matrix		
	A Galvanizer	B Galvanizer	A Galvanizer	B Galvanizer	Weights (Row Avg.)
A Galvanizer	1	6	0.857	0.857	0.857
B Galvanizer	1/6	1	0.143	0.143	0.143
Column Total	1.167	7	1.000	1.000	1.000

Table 6-52. Cost Certainty Measurement Scale (Galvanizer Work Package)

Verbal Judgment	**Numerical Rating**	**Cost Difference Percentage Range**	
Extremely preferred	9	-	-0.701
	8	-0.700	-0.601
Very strongly preferred	7	-0.600	-0.501
	6	-0.500	-0.401
Strongly preferred	5	-0.400	-0.301
	4	-0.300	-0.201
Moderately preferred	3	-0.200	-0.101
	2	-0.100	-0.001
Equally preferred	1	0.000	0.000
	1/2	0.001	0.100
Moderately un-preferred	1/3	0.101	0.200
	1/4	0.201	0.300
Strongly un-preferred	1/5	0.301	0.400
	1/6	0.401	0.500
Very strongly un-preferred	1/7	0.501	0.600
	1/8	0.601	0.700
Extremely un-preferred	1/9	0.701	-

Table 6-53. Weights for Performance-Cost Certainty (Galvanizer Work Package)

	Original Matrix		Adjusted Matrix		
	A Galvanizer	B Galvanizer	A Galvanizer	B Galvanizer	Weights (Row Avg.)
A Galvanizer	1	1	0.50	0.50	0.50
B Galvanizer	1	1	0.50	0.50	0.50
Column Total	2	2	1.00	1.00	1.00

Table 6-54. Quote Response Time Measure Scale (Galvanizer Work Package)

Verbal Judgment	Numerical Rating	Quoted Customer Response Percentage Range	
Equally preferred	1	1.000	1.000
	1/2	0.999	0.900
Moderately un-preferred	1/3	0.899	0.800
	1/4	0.799	0.700
Strongly un-preferred	1/5	0.699	0.600
	1/6	0.599	0.500
Very strongly un-preferred	1/7	0.499	0.400
	1/8	0.399	0.300
Extremely un-preferred	1/9	0.299	0.000

Table 6-55. Weights for Performance- Quote Response Time (2/3=0.8) (Galvanizer Work Package)

	Original Matrix		Adjusted Matrix		
	A Galvanizer	B Galvanizer	A Galvanizer	B Galvanizer	Weights (Row Avg.)
A Galvanizer	1	5	0.833	0.833	0.833
B Galvanizer	1/5	1	0.167	0.167	0.167
Column Total	1.2	6	1.000	1.000	1.000

Table 6-56. On-time Delivery Measurement Scale (Galvanizer Work Package)

Verbal Judgment	Numerical Rating	On-time Delivery Percentage Range	
Extremely preferred	9	1.000	0.9751
	8	0.975	0.951
Very strongly preferred	7	0.900	0.851
	6	0.850	0.801
Strongly preferred	5	0.800	0.751
	4	0.750	0.701
Moderately preferred	3	0.700	0.651
	2	0.650	0.601
Equally preferred	1	0.600	0.600
	1/2	0.601	0.500
Moderately un-preferred	1/3	0.501	0.400
	1/4	0.401	0.350
Strongly un-preferred	1/5	0.351	0.300
	1/6	0.301	0.250
Very strongly un-preferred	1/7	0.251	0.200
	1/8	0.201	0.150
Extremely un-preferred	1/9	0.151	0.000

Table 6-57. Weights for Performance- On-time Delivery (promised/actual = 3/4 = 0.75, since nominator must be less than denominator) (Galvanizer Work Package)

	Original Matrix		**Adjusted Matrix**		
	A Galvanizer	B Galvanizer	A Galvanizer	B Galvanizer	Weights (Row Avg.)
A Galvanizer	1	4	0.80	0.80	0.80
B Galvanizer	1 / 4	1	0.20	0.20	0.20
Column Total	1.25	5	1.00	1.00	1.00

Table 6-58. Times of Dispute Minimization Measurement Scale ($5,000-$50,000) (Galvanizer Work Package)

Verbal Judgment	**Numerical Rating**	**Times of Dispute**
Equally preferred	1	0
	½	1
Moderately un-preferred	1/3	1
	¼	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

Table 6-59. Times of Dispute Minimization Measurement Scale ($50,000-150,000) (Galvanizer Work Package)

Verbal Judgment	**Numerical Rating**	**Times of Dispute**
Equally preferred	1	0
	½	1
Moderately un-preferred	1/3	2
	¼	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-60. Weights for Performance-Times of Dispute (Galvanizer Work Package)

	Original Matrix		**Adjusted Matrix**		
	A Galvanizer	B Galvanizer	A Galvanizer	B Galvanizer	Weights (Row Avg.)
A Galvanizer	1	1/3	0.250	0.250	0.250
B Galvanizer	3	1	0.750	0.750	0.750
Column Total	4	1.333	1.000	1.000	1.000

Table 6-61. Warranty Return Times Measurement Scale ($5,000-$50,000) (Galvanizer Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	½	1
Moderately un-preferred	1/3	1
	¼	2
Strongly un-preferred	1/5	2
	1/6	3
Very strongly un-preferred	1/7	3
	1/8	4
Extremely un-preferred	1/9	4

Table 6-62. Warranty Return Times Measurement Scale ($50,000-150,000) (Galvanizer Work Package)

Verbal Judgment	Numerical Rating	Times of Dispute
Equally preferred	1	0
	½	1
Moderately un-preferred	1/3	2
	¼	3
Strongly un-preferred	1/5	4
	1/6	5
Very strongly un-preferred	1/7	6
	1/8	7
Extremely un-preferred	1/9	8

Table 6-63. Weights for Performance-Warranty Return Times (Galvanizer Work Package)

	Original Matrix		Adjusted Matrix		
	A Galvanizer	B Galvanizer	A Galvanizer	B Galvanizer	Weights (Row Avg.)
A Galvanizer	1	1	0.50	0.50	0.50
B Galvanizer	1	1	0.50	0.50	0.50
Column Total	2	2	1.00	1.00	1.00

Integrating the Activity-based Performance Model and the Cost Model

Figure 6-9 shows all the possible combinations of material suppliers and subcontractors and the quotes offered. By applying the cost model described in Chapter 5, the final weights for each cost alternative are shown in Figure 6-9. Table 6-64 displays the cost weights and performance weights of the 16 supply chain alternatives. The most right column of Table 6-64 shows the synthesized weights of the alternatives while management set cost weight of 0.4 and performance weight of 0.6 for the project. Figure 6-10 and Figure 6-11 are the visualizations of results from Table 6-64. Figure 6-11 shows that alternative 12 is the best choice for this case study while both cost weight and performance weight are 0.5. However Figure 6-12 shows alternative 12 is the right choice while the cost weights are respectively 0.20 or 0.40 even though the cost rate of alternative 11 is the best one. Alternative 11 is an appropriate choice when the cost weights are 0.60 or 0.80.

Table 6-64 Weights for Alternatives

Alternatives	**Cost Weights (0.4)**	**Performance Weights (0.6)**	**Final Weights**
(1)	(2)	(3)	(4) = 0.5*(2) + 0.5 * (3)
1	0.0469	0.0891	0.0680
2	0.0274	0.0763	0.0763
3	0.0737	0.0830	0.0830
4	0.0430	0.0723	0.0723
5	0.0158	0.0568	0.0568
6	0.0108	0.0495	0.0495
7	0.0235	0.0497	0.0497
8	0.0145	0.0433	0.0433
9	0.1430	0.0838	0.0838
10	0.0861	0.0491	0.0491
11	0.1972	0.0500	0.0500
12	0.1272	0.1211	0.1211
13	0.0469	0.0502	0.0502
14	0.0274	0.0437	0.0437
15	0.0737	0.0439	0.0439
16	0.0430	0.0382	0.0382
Total	1.0000	1.0000	1.0000

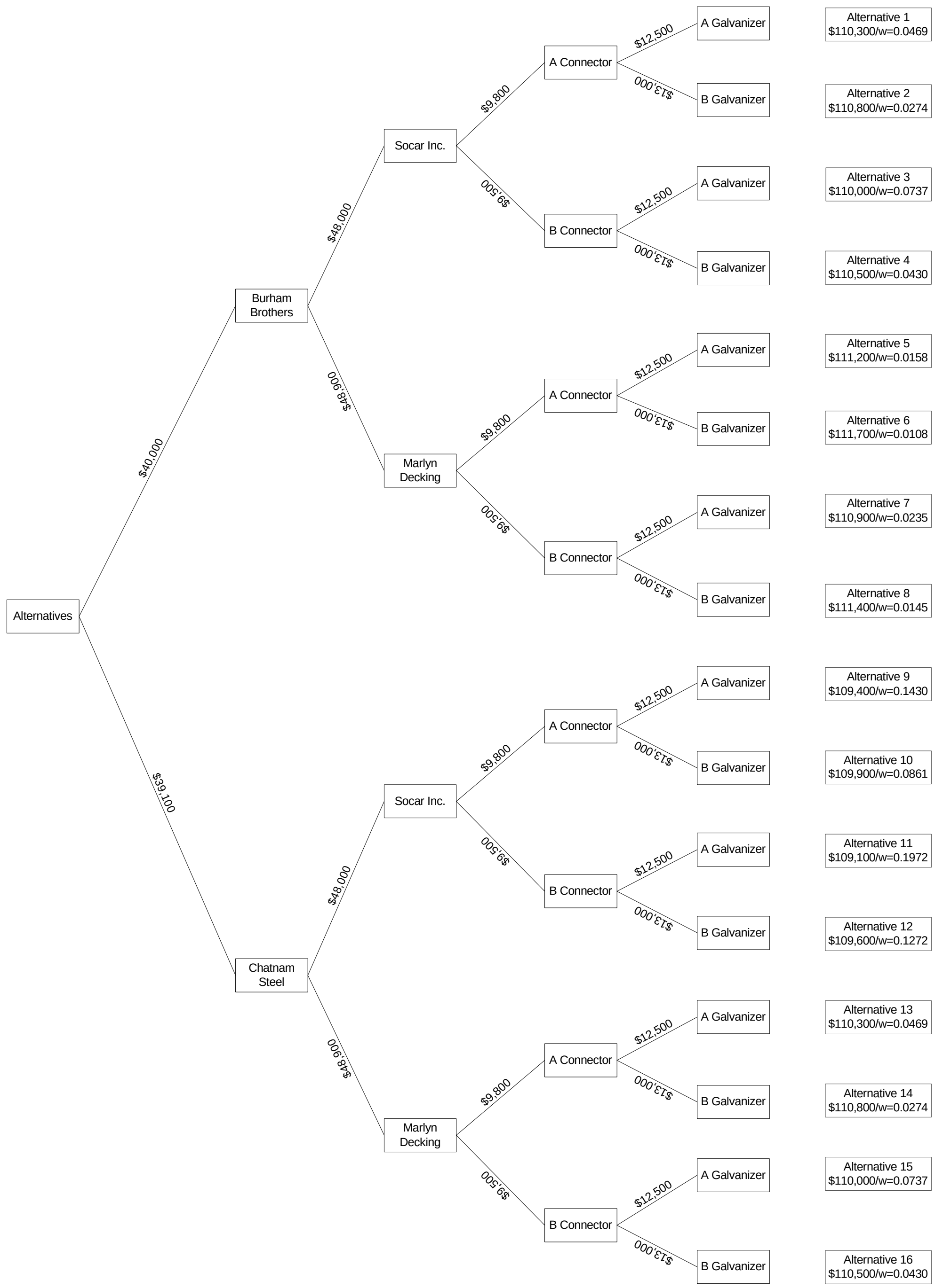


Figure 6-9. Total Cost of Supply Chain Alternatives and Quotes of the Material Suppliers and Subcontractors for a Project

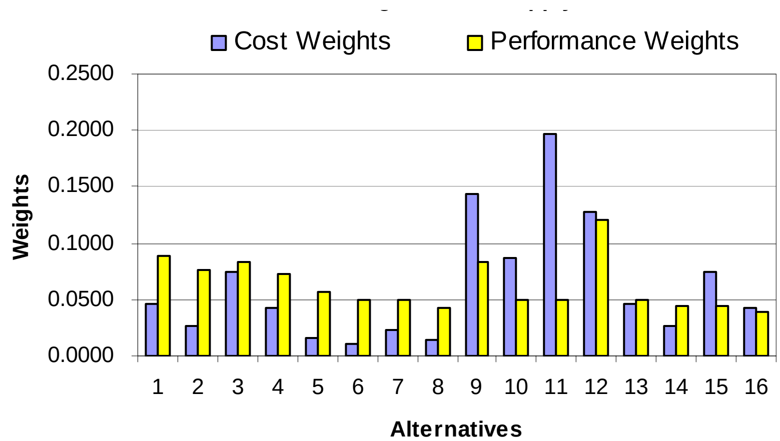


Figure 6-10 Cost and Performance Weights for the Supply Chain Alternatives

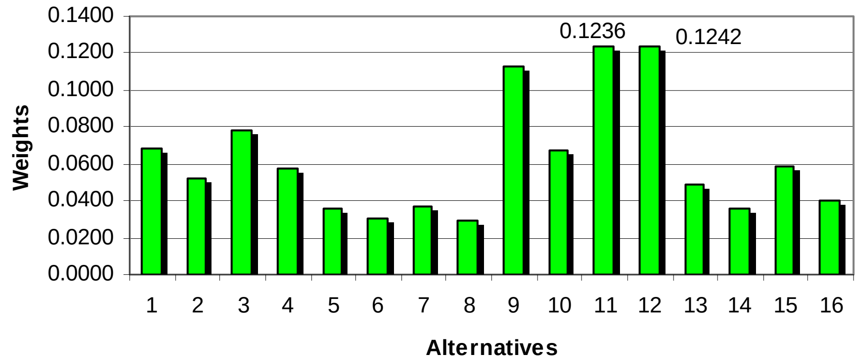


Figure 6-11 Synthesized Weights for the Supply Chain Alternatives

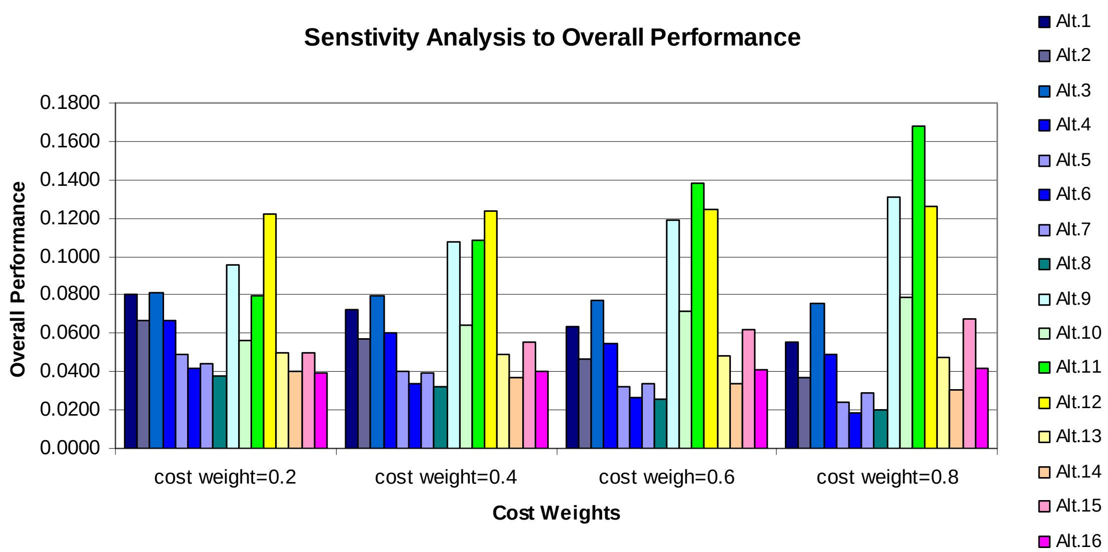


Figure 6-12 Sensitivity Analysis to Supply Chain Alternatives

Summary

This chapter presents a case study based on interviews with a structural steel fabricator. This case study classifies the work packages for the ABC Steel Fabricator based on is business model. Then it explains how to run the proposed model in this company. It first illustrates the performance model running in different work packages and then integrates the performance model with the cost model to display the synthesized weights for each potential supply chain alternatives. This gives management a rational basis to make the rational decisions for a specific construction project. In order to use the proposed model, each company has to define their own work packages and apply the model based on their own business background, specific project requirements, and project sizes. However, because the decision-making process in this case study involves many calculations and data entries, computerizing the decision-making process becomes necessary. The next chapter, Chapter 7, will introduce the structure of the computerized decision-making system and its underlying databases.

CHAPTER 7
IMPLEMENTATION OF MODEL

Introduction of the System

A computer environment to support the decision-making process in selecting the project partners and supply chain configuration is presented in this chapter. The goal of the system is to evaluate the performance of suppliers and subcontractors and analyze the supply chain alternatives, based on evaluation information and upcoming project information. The system has a highly interactive interface that provides increased support for the modeling process. It also offers flexibility to define the work packages, performance criteria, activities, and relationships between them. Theoretically it does not have a limit on the number of supply chain alternatives that can be analyzed. It is implemented in a Windows XP environment, using the development environment of Microsoft's Visual Basic programming language. The following sections will introduce the system and the underlying databases.

Figure 7-1 shows the architecture of the implemented system. Basically the system is composed of three components:

- Business definition – it allows the users to define their own work packages, activities, performance criteria, and relationships between them based on their business goal, improvement area, and specific project. This extends the usage of the system to any kind of contractor, general contractor or subcontractors. Figures 7-2 to 7-5 show snap shots of the business definition component.

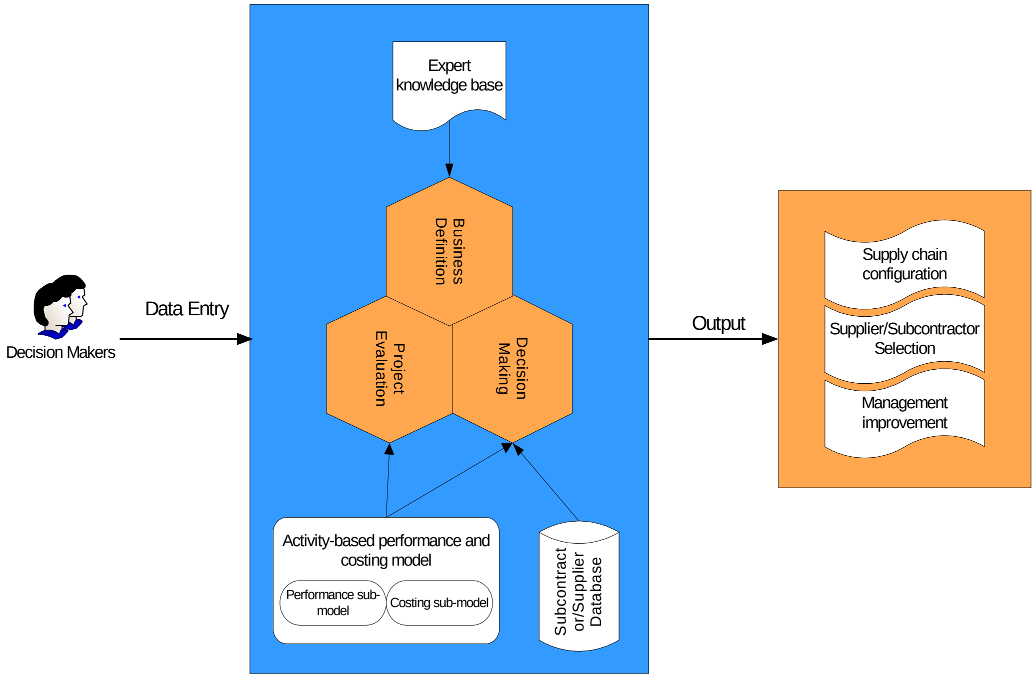


Figure 7-1. System Architecture

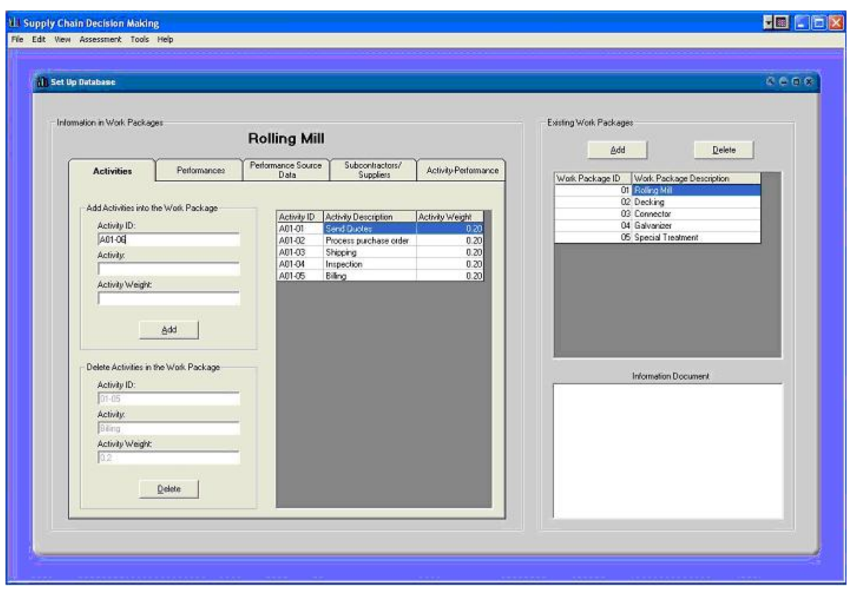


Figure 7-2. Snapshot of Business Definition Component – Define Work Packages and Activities

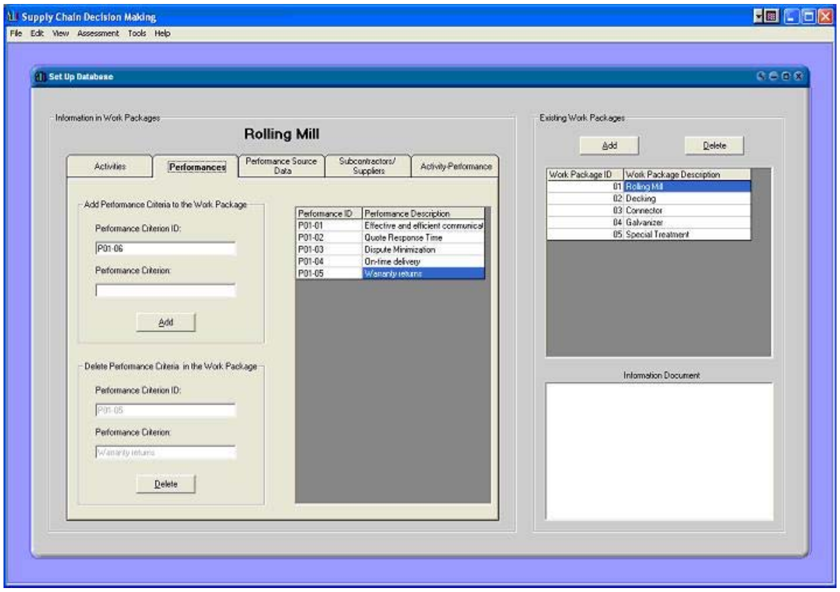


Figure 7-3. Snapshot of Definition of Performance Criteria

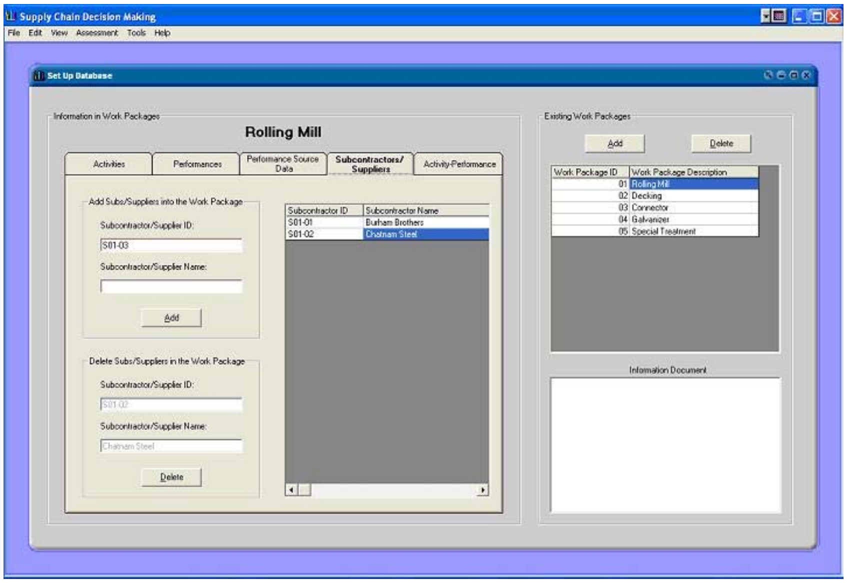


Figure 7-4. Snapshot of Subcontractors or Suppliers in the Work Package Rolling Mill

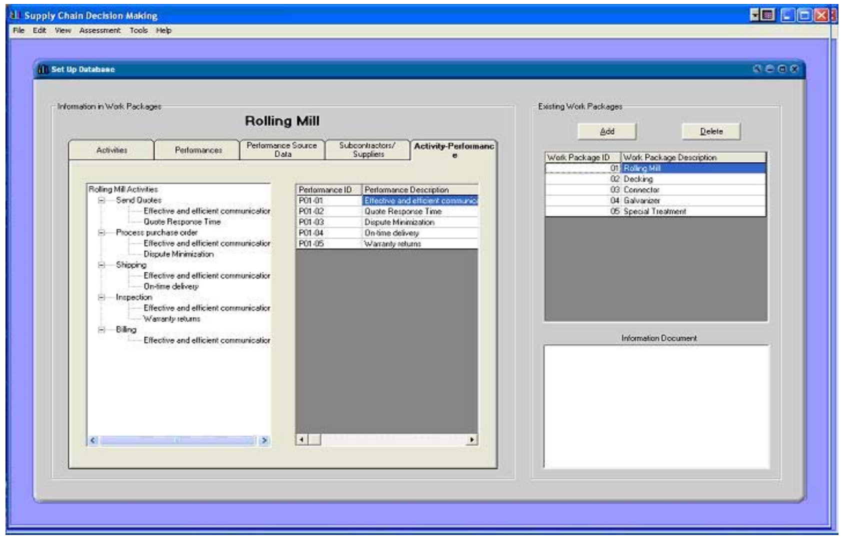


Figure 7-5.Snapshot of Business Definition Component – Relationships between Activities and Performance Criteria

- Project evaluation – it requires the user to enter data which is utilized to evaluate the subcontractors' performance on the completed projects. The data entered will become historical data or a knowledgebase which will be processed by algorithms such as AHP (analytical hierarchy process) to support the decision making process. Figure 7-6 shows the flow of the evaluation process in the system.

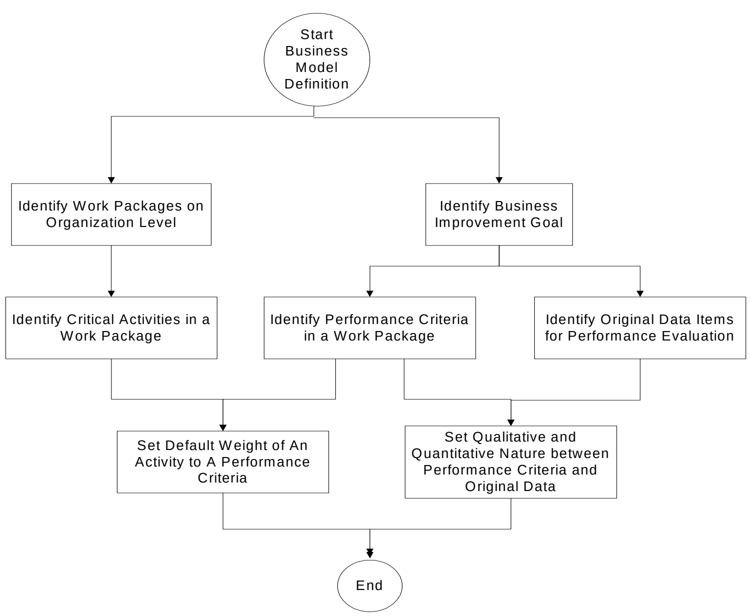


Figure 7-6. Flow of Business Definition

- Decision-making on supply chain and subcontractors/suppliers selection -- it provides an interface for the system users to enter information on an upcoming project, such as work packages, subcontractors/suppliers' quotes, activities and performance criteria. This component will analyze potential supply chain alternatives, performance and cost information for suppliers/subcontractors to provide the system users with tabulated and graphic results. It also provides the functionality for sensitivity analysis to give the user flexibility of decision-making in configuring the supply chain configuration and selecting the subcontractors and suppliers. Figure 7-7 shows the flow of the decision making process in this system. Figures 7-8 to 7-15 show the snapshots for a specific project decision-making process.

- Figure 7-8 shows a new project. The list on the top right hand side shows the work packages that a company may have. The list on the bottom right hand side shows the suppliers or subcontractors in each work package. The left side tree view shows the work packages involved in this new project and the subcontractors who submit quotes for this new project. Figure 7-9 shows all potential supply chain alternatives for the upcoming project. This new project has eight alternatives.
- Figure 7-10 shows the alternatives and gives detailed information for each alternative such as partners, total cost, cost weights, performance weights, etc.
- Figure 7-11shows two graphs. The upper one shows the total cost of each alternative and the lower one gives the cost weights of these alternatives. It shows that alternative 6 has the lowest cost with the highest cost weight.

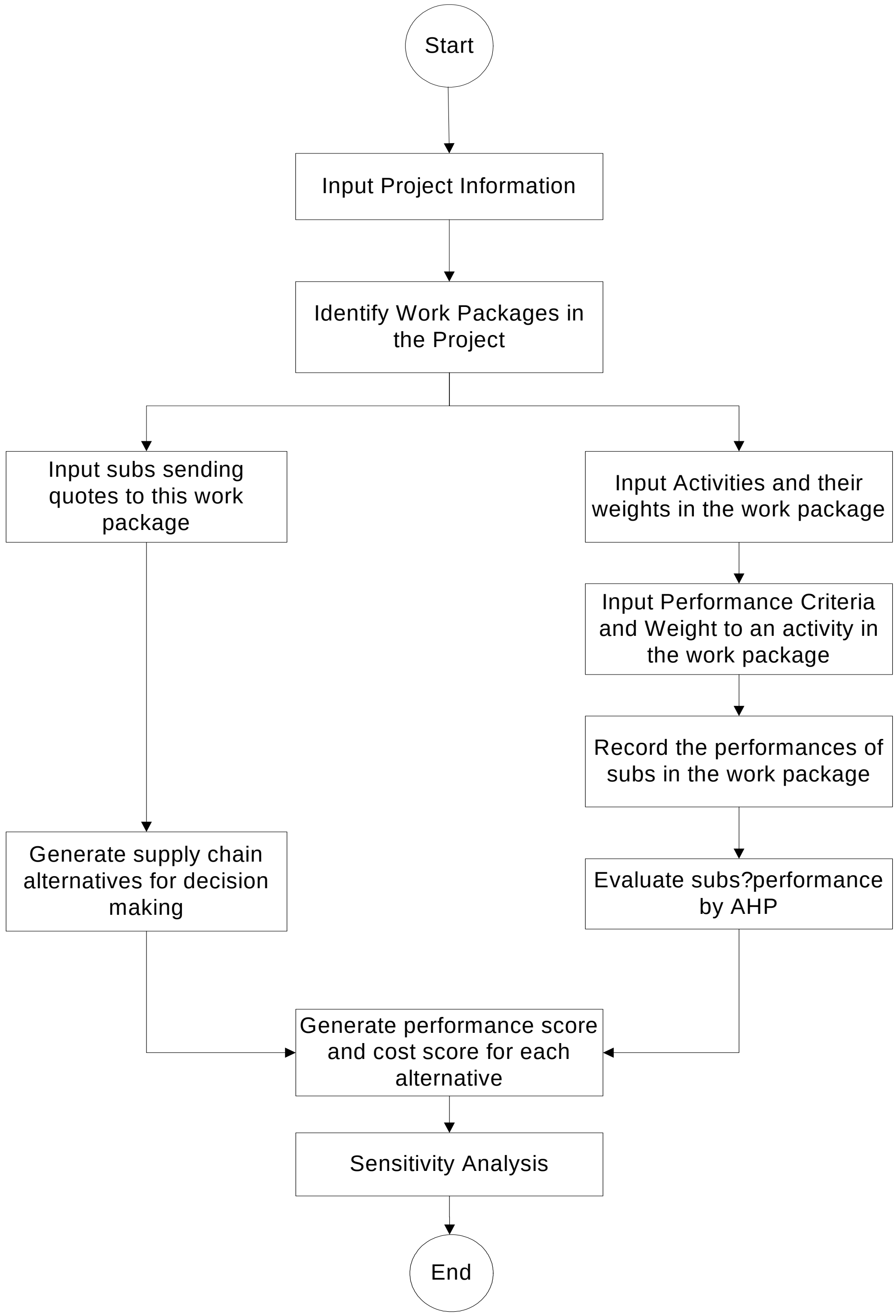


Figure 7-7. Flow of Subcontractor/Supplier Evaluation in Completed Projects and Decision Making in Upcoming Projects

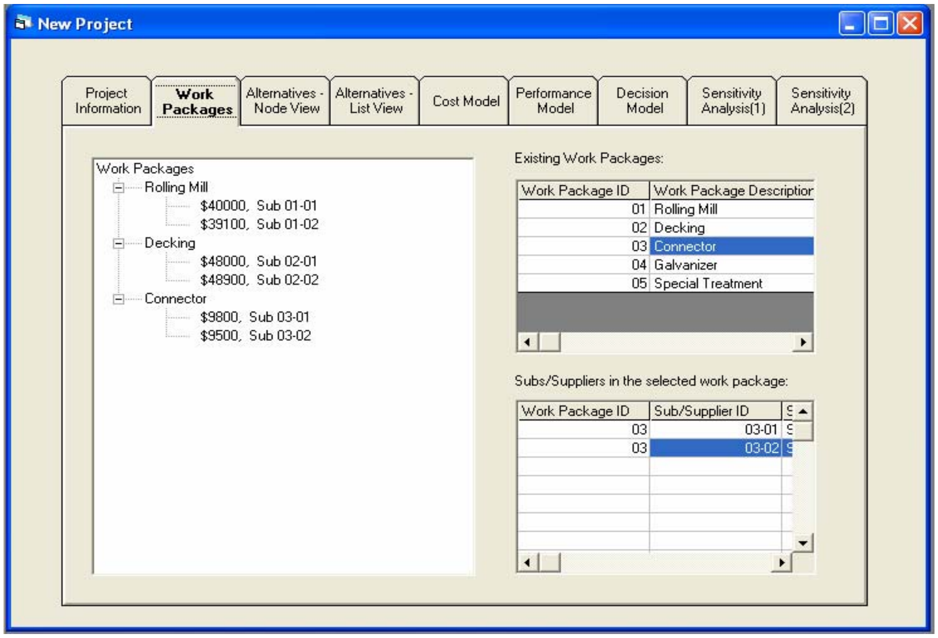


Figure 7-8. Data Entry for an Upcoming Project, Such As Work Packages, Subcontractors and Quotes Snap Shot

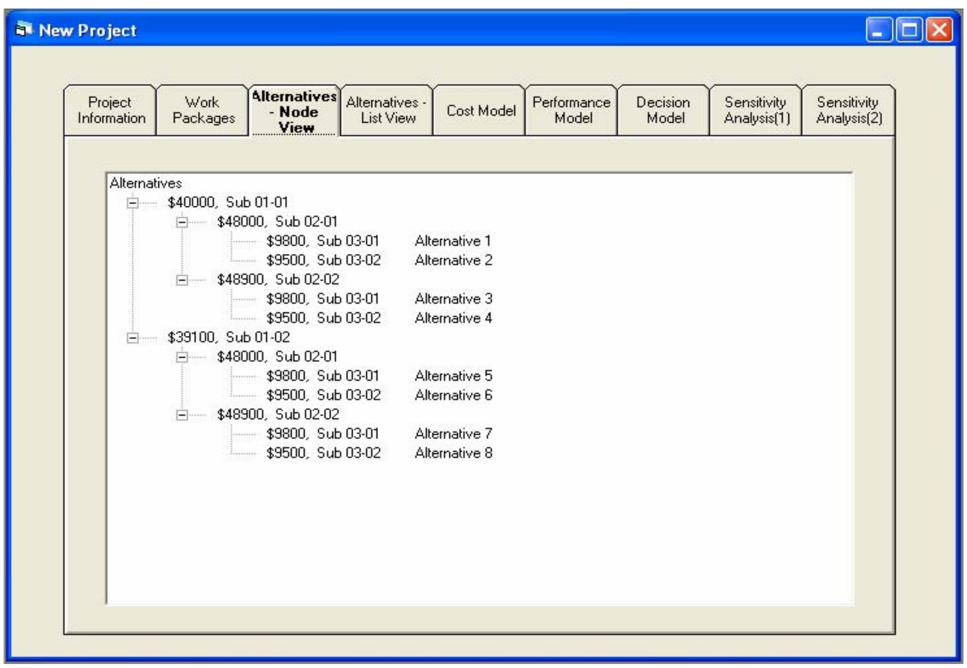
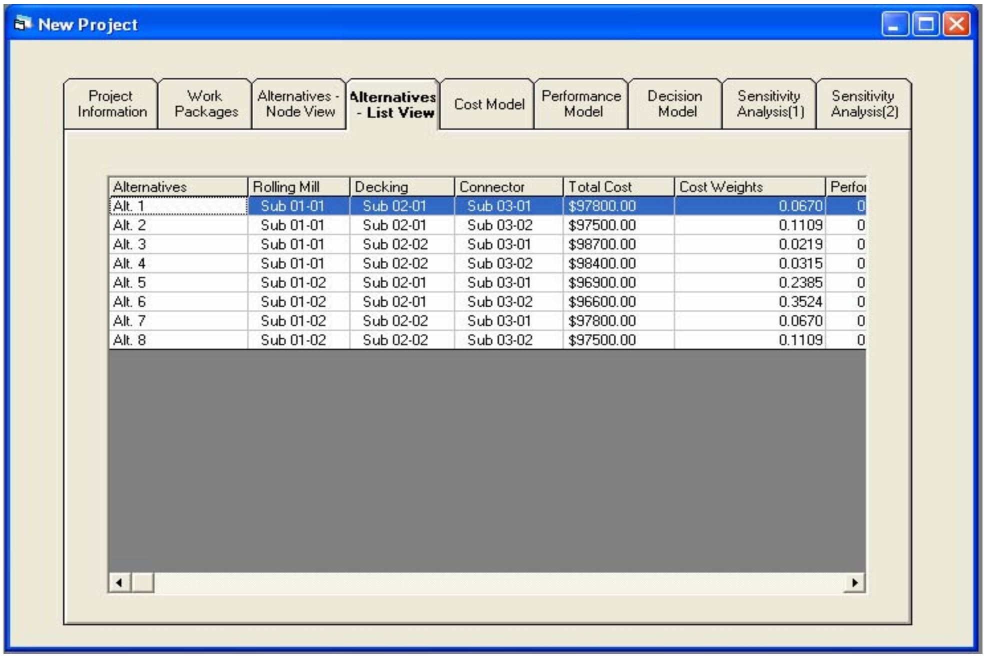


Figure 7-9. Supply Chain Configurations or Alternatives for the Upcoming Project Snap Shot (Tree View)



Alternatives	Rolling Mill	Decking	Connector	Total Cost	Cost Weights	Perfor
Alt. 1	Sub 01-01	Sub 02-01	Sub 03-01	$97800.00	0.0670	0
Alt. 2	Sub 01-01	Sub 02-01	Sub 03-02	$97500.00	0.1109	0
Alt. 3	Sub 01-01	Sub 02-02	Sub 03-01	$98700.00	0.0219	0
Alt. 4	Sub 01-01	Sub 02-02	Sub 03-02	$98400.00	0.0315	0
Alt. 5	Sub 01-02	Sub 02-01	Sub 03-01	$96900.00	0.2385	0
Alt. 6	Sub 01-02	Sub 02-01	Sub 03-02	$96600.00	0.3524	0
Alt. 7	Sub 01-02	Sub 02-02	Sub 03-01	$97800.00	0.0670	0
Alt. 8	Sub 01-02	Sub 02-02	Sub 03-02	$97500.00	0.1109	0

Figure 7-10. Supply Chain Configurations or Alternatives for the Upcoming Project Snap Shot (Table View, Which Provides More Detailed Information for the Users)

Figure 7-11. Cost Model Shows the Analysis Results (Weights) of Cost for Each Supply Chain Alternative. Red Bars Show the True Cost of Each Supply Chain Alternative and Green Bars Show the Cost Weight for Each Supply Chain Configuration.

- Figure 7-12 shows the performance model tab and gives the performance weights for all supply chain alternatives. Alternatives 3, 5, 6, and 7 have low performance weights and alternative 1 has the highest performance weight.

Figure 7-13 shows how the decision model tab provides the tool to set the cost weight and performance weight for a specific project. The blue bars show performance weight and the green bars show the cost weight. The stack of a pair of blue and green bar shows a synthesized weight for a specific supply chain alternative. This snap shot shows the alternative 6 is the best choice when cost weight is 0.60 and performance weight is 0.40.

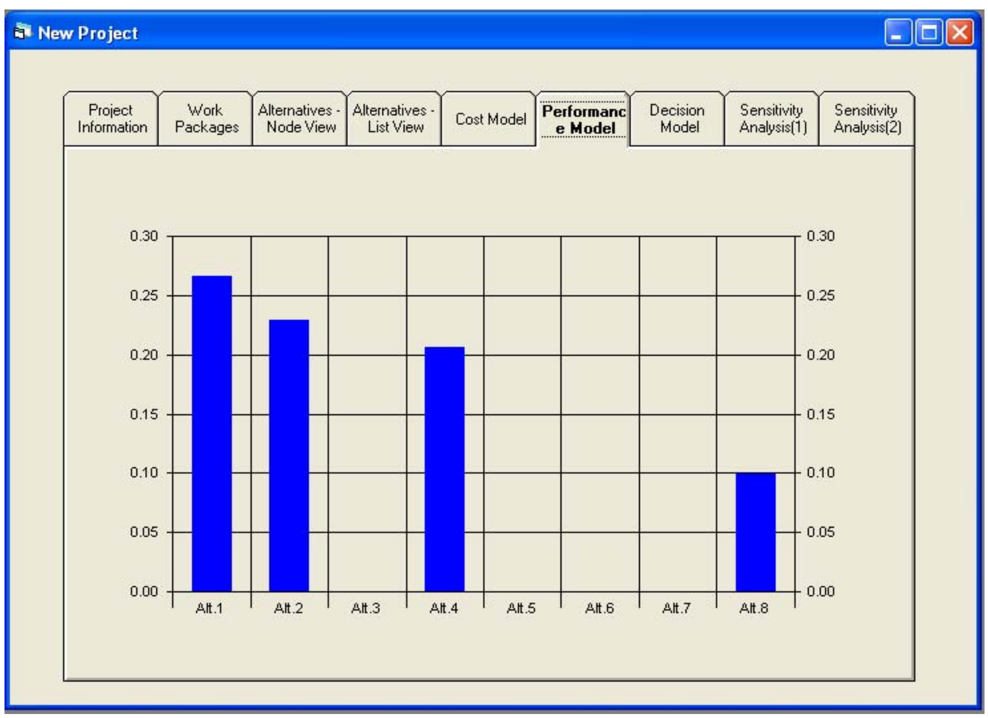


Figure 7-12. Snap Shot of Decision-making Component – Performance Model Shows the Analysis Results (Weights) of Performance for Each Supply Chain Alternative.

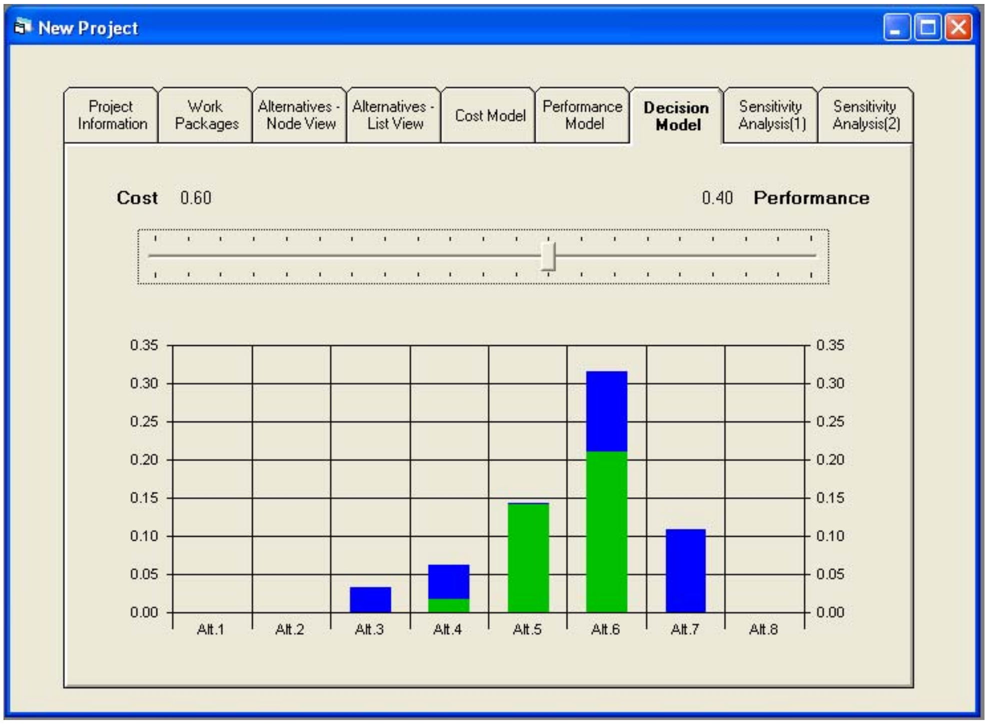


Figure 7-13. Snap Shot of Decision-making Component – Synthesis of the Performance and Cost Model Result.

- Figures 7-14 and 7-15 show the overall performance for each alternative and are synthesized from specified cost and performance priority weights. Although Figure 7-15 has similar function as Figure 7-14, it is more readable than Figure 7-14. It shows that alternative 2 is the reasonable choice when cost weight is 0.20 or 0.40 and the corresponding performance priority weights equal 0.8 or 0.6. Similarly alternative 6 is the appropriate selection when the cost priority weight is set to 0.6 or 0.8.

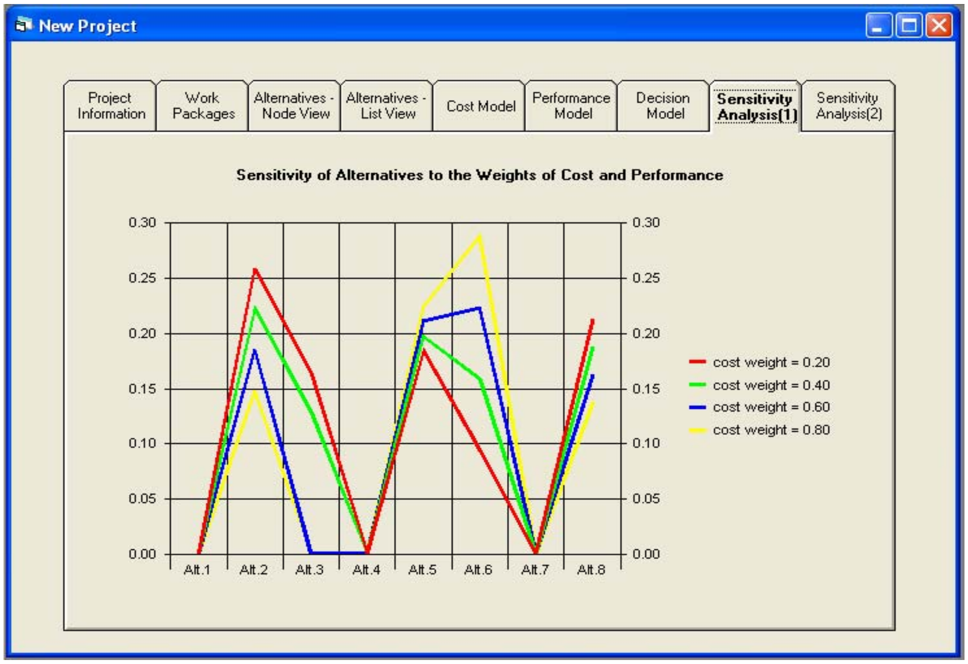


Figure 7-14. Snap Shot of Decision-making Component – Sensitivity of Synthesized Cost and Performance Weight for Each Supply Chain Alternative

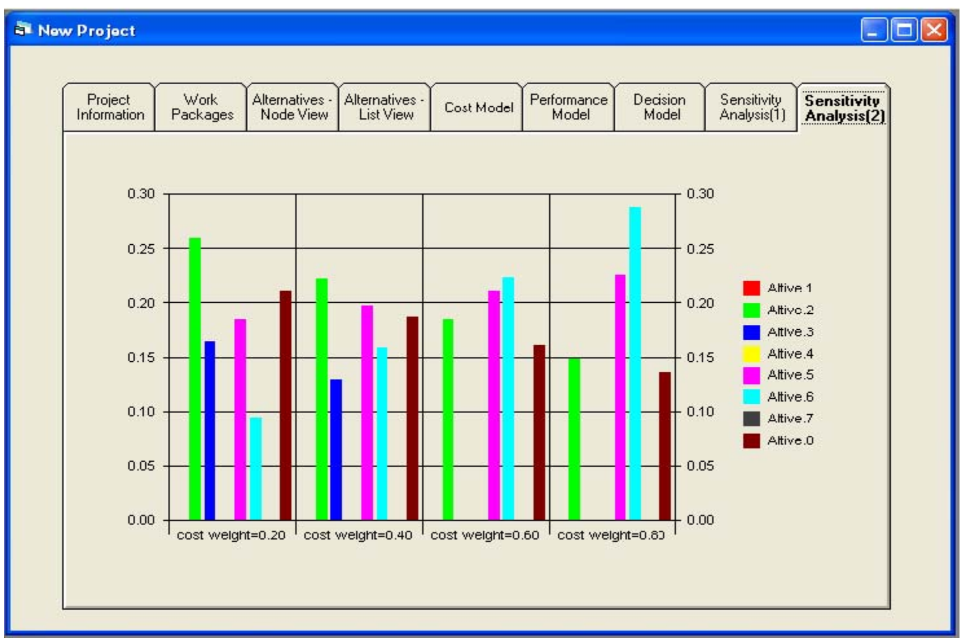


Figure 7-15. Snap Shot of Decision-making Component – Sensitivity of Synthesized Cost and Performance Weight for Each Supply Chain Alternative.

System Conceptual Data Schema

The underlying data schemas offer the most appropriate match to data usage. The development of the schema for the software development is carried out by documenting the data fields used during information entry and exporting them from the software as shown in Figure 7-16. The basic entity-relationship modeling notation (Chen 1976) uses three main constructs: data entities, relationships, and their associated attributes. Figure 7-16 shows the E-R diagram for the Activity-based performance and costing system. The following describes the data entities and associative entities, and their associated attributes from the E-R diagram:

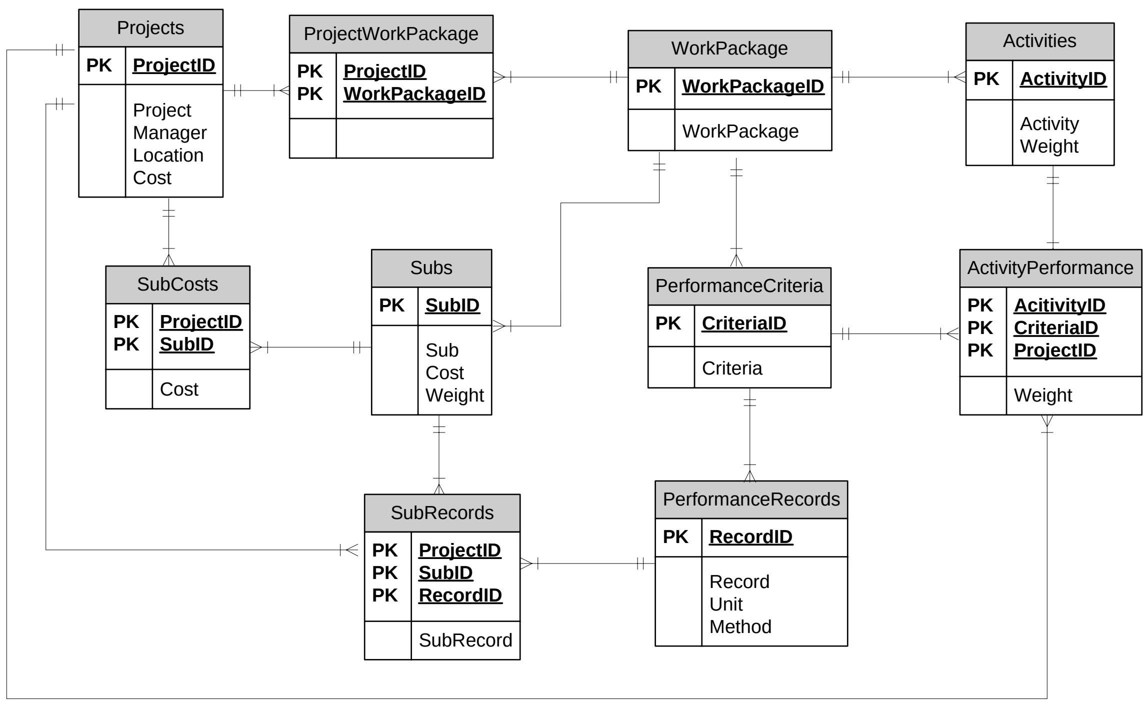


Figure 7-16 Conceptual Data Model of E-R Diagram for Activity-based Performance and Costing system

6. Entities

- PROJECTS;
  - ProjectID (primary key): String;
  - Project : String;
  - Manager : String;
  - Location : String;
  - Start Date: Date;
  - End Date: Date;

- WORKPACKAGE;
  - WorkPackageID (primary key): String
  - WorkPackage: String

- ACTIVITIES;
  - ActivityID (primary key): String;
  - Activity: String;
  - Weight: Number;

- SUBS;
  - SubID (primary key): String;
  - Sub: String;
  - Phone: String;
  - Fax: String;
  - Address: String;

- PERFORMANCECRITERIA;
  - CriteriaID (primary key): String;
  - Criteria: String;

- PERFORMANCERECORDS;
  - RecordID (primary key): String;
  - Record: String;
  - Unit: String;

7. Associative Entities

- PROJECTWORKPACKAGE;
  - ProjectID: String;
  - WorkPackageID: String;

- ACTIVITYPERFORMANCE;
  - ActivityID (primary key): String;
  - CriteriaID (primary key): String;

- Weight: Number;

- SUBCOSTS;
  - ProjectID (primary key): String;
  - SubID (primary key): String;
  - Cost: Currency;

- SUBRECORDS;
  - ProjectID (primary key): String;
  - SubID (primary key): String;
  - RecordID (primary key): String;
  - Data: Number;

Figure 7-16 also shows the relationships between the data entities. These relationships are described as follows:

- A PROJECT includes one or more WORKPACKAGE, A WORKPACKAGE may be in one or more PROJECT. Therefore an associative entity PROJECTWORKPACKAGE exists between PROJECT and WORKPACKAGE. Two foreign keys ProjectID from PROJECT and WorkPackageID from WORKPACKAGE combined as primary key for this entity.

- A WORKPACKAGE includes one or more SUBS. An object of SUBS only belongs to one and only one WORKPACKAGE.

- A WORKPACKAGE includes one or more PERFORMANCECRITERIA. An object of PERFORMANCECRITERIA belongs to one and only one WORKPACKAGE.

- A WORKPACKAGE includes one or more ACTIVITIES while an object of ACTIVITIES belongs to one and only one WORKPACKAGE.

- An ACTIVITIES may have one or more PERFORMANCECRITERIA while PERFORMANCECRITERIA may have one or more ACTIVITIES. Therefore, an associative entity ACTIVITYPERFORMANCE exists between ACTIVITIES and PERFORMANCECRITERIA. Two foreign keys, ActivityID from ACTIVITIES and CriteriaID from PERFORMANCECRITERIA combined as primary key for this entity. An item of entity belongs to one and only one ACTIVITIES and one only one PERFORMANCECRITERIA.

- PERFORMANCERECORDS stores original data to record the performance of subs/suppliers. One or several items from PERFORMANCERECORDS are used to calculate scores of an object of SUBS on one and only one object of PERFORMANCECRITERIA. Therefore an object of

PERFORMANCERECORDS belongs to one and only one PERFORMANCECRITERIA. An object of PERFORMANCECRITERIA may include one or more PERFORMANCERECORDS.

- An associative entity SUBRECORDS exists between SUBS and PERFORMANCERECORDS to store the original data of an object of SUBS.
- An associative entity SUBCOSTS exists between SUBS and PROJECTS to store the cost information of each sub/supplier for a specific project

Transforming Entities in E-R diagram to relations depends on both the degree of the cardinalities of the relationships between the entities. The following shows the final relations for this Activity-based performance and costing decision-making system based on the analysis of relationship and entities in E-R diagram:

- PROJECTS (ProjectID, Project, Manager, Location, Cost)
- WORKPACKAGE (WorkPackaeID, WorkPackage)
- PROJECTWORKPACKAGE (ProjectID, WorkPackageID)
- ACTIVITIES (AcitivityID, Activity, Weight, WorkPackageID)
- PERFORMANCECRITERIA (CriteriaID, Criteria, WorkPackageID)
- ACTIVITYPERFORMANCE (ActivityID, CriteriaID, ProjectID,Weight)
- SUBS (SubID, Sub, WorkPackageID)
- SUBCOSTS (SubID, ProjectID, Cost)
- SUBRECORDS (SubID, RecordID, ProjectID, SubRecord)
- PERFORMANCERECORDS (RecordID, Record, Unit, CriteriaID)

The subcontractor and supplier selection system (supply chain configuration decision-making system) consists mainly of three components: (1) Business definition; (2) Project evaluation; (3) Decision-making on supply chain and subcontractors/ suppliers. The underlying database provides an expert knowledgebase for the system and the proposed model implemented in this system provides the method for evaluating and analyzing the performance of subcontractors and suppliers, and the potential supply chain alternatives by using this expert knowledgebase.

Summary

This chapter describes the implementation of the proposed activity-based performance and costing decision-making system. It introduces the structure of the system and the underlying database schema. It also shows the snapshots of decision-making process of the case study. The system is interactive, flexible, and easy to use. Although the snap shots show the decision-making process in the structural steel construction service, the system also can be extended to the other construction trades or general contractors. Next, Chapter 8 will discuss the conclusions and limitations of this research and will list recommendations future areas of research.

CHAPTER 8
CONCLUSIONS AND RECOMMENDATIONS

Conclusions

A high degree of specialization has evolved in the provision of various goods and services in the construction industry and a large proportion of construction activities may be subcontracted out on any given project. The subcontractor and material supplier selection process in the construction industry is usually based on pricing and pre-existing business relationships. This may overlook more responsible subcontractors or suppliers. Improvements in the subcontractor and supplier selection processes have not received the critical attention for improvement. However, the subcontractor/supplier selection process encompasses different functions such as purchasing, quality, production, etc. within the company. It is a multi-criteria problem, encompassing many tangible and intangible factors. Frequently, these evaluation criteria involve trade-offs. This study focuses on the supplier and subcontractor selection decision support using structural steel procurement procedures as a case study. It shows that an Activity-based performance and costing model which includes two sub-models, a cost model and a performance model can be successfully used in the decision making process. The performance model is for recording and evaluating the performance of subcontractors/suppliers on the completed projects. The historical data forms a sound foundation to support decision-making. The Analytical Hierarchy Process (AHP) method is used to analyze both tangible and intangible factors. The model provides the potential supply chain alternatives and their weights respectively for decision-making. It treats the supplier selection criteria, potential alternatives, and activities involved in procurement procedures in a hierarchical manner. The hierarchical structure decomposes and synthesizes these factors to find the right suppliers and subcontractors for a project. The computerized decision-making system provides an interactive and convenient environment for industry practitioners to utilize the proposed model to make the right decision on subcontractor/supplier selection, configuring an efficient supply chain for a project.

Limitations

Although this study uses a structural steel fabricator as an example case study, the model is flexible enough to be further used by general contractors and any other subcontractors for selecting suppliers and sub-subcontractors for other types of building components. Performance measurement is a critical step in the evaluation of subcontractors/suppliers and the design of supply chains. It will vary depending on the construction trades, the type of projects, and the business goal of the construction company. The more trades are involved, the more challenging it becomes to measure effectively. Second, many activities involve several business partners. Identifying critical activities for each trade/work package for performance measuring affects directly the evaluation and performance analysis process. Third, the original data collection for performance analysis requires long term efforts and commitments by management to collect historical data to be used in the model for selecting the right supply chain.

Recommendations for Future Research

This study proposed a cost and activity-based performance measure model and applied this model to the supply chain decision making process for a structural steel fabricator. However, to enhance the model, future research is required in the following areas:

- Extending and testing the model to include other construction industry trades and their supply chain selection needs.

  Successful completion of construction projects involves many work packages and many trade specialty areas. Each trade has its special techniques, supply chain formation, work performance criteria, and activities involved. The future effort should not only apply the model to the special construction industry trades, but also compare the completed project results with the decision making system analysis for a specific project. This will help in testing the decision-making analysis system, and improving the analysis ability of the system by giving feedback to the system.

- Developing the selection decision-making process on each layer of the construction project supply chain

  Selection problems are related to the procurement of construction products and services. Selections are made at each level of the construction supply chain. In the typical construction project supply chain, the project owner has to make decisions on choosing appropriate general contractors, the general contractors must intelligently select the most appropriate subcontractors, and the subcontractors have to choose the right suppliers and sub-subcontractors, and so on. Intelligent selection at each level of the construction service supply chain would assure the successful completion of the project. However, the selection criteria, decision variables, and work allocation structures, etc. vary at each layer of decision making process. In addition, contractors, subcontractors, and sub-subcontractors and suppliers also face the need to make selections for upcoming projects, in a process called reverse selection decision-making. Further research efforts are needed for each layer of construction service, supply chain design and analysis and reverse selection decision-making.

- Adapting the model to incorporate fuzzy set theory to compensate for the lack of information

  Fuzzy set theory provides knowledge about existing uncertainty and imprecision and allows management personnel to incorporate this knowledge into their decision-making process. A Fuzzy set contains elements that have varying degrees of membership in the set. The membership function $\mu(x)$ consists of real numbers in the interval [0, 1] that represent the degree of membership to which an object belongs within the fuzzy set. The higher the degree of membership (closer to 1), the stronger the object belongs to that set. This theory is beneficial when the decision-maker:

  1. lacks performance information about the potential construction service supplier;
  2. does not have accurate and/or adequate historical data available by the potential business partners

  The decision makers can benefit from the quantification of data uncertainty and can represent potential risks in the analytical models through using fuzzy data.

- Integrating neural network technology to improve the current decision-making system

  The capability of neural networks to develop knowledge from historical project data provides the following possibilities in improving the selection decision-making analysis:

  1. Training the decision-making system using a neural network methods,will involve utilizing a lot of historical project data about construction service suppliers. This may help solve the continuous data entry burden after the system is well trained.

2. Comparing the cost and activity-based performance measure model with the neural network methods. This may provide more analysis tools for decision-makers.

- Adding the ability to determine non-valued activities in the model.

  In this study, the activity-based performance model is used to measure the business partners' performance on the activity level. Non-valued activities or activities with unfavorable performance are identified and suggestions for improvement would be generated for use in future decision support systems to help to improve project participants' performances.

- Developing a more adaptive database management system to meet industrial needs

  The study uses Visual Basic to implement the model and Microsoft Access to manage the database. However many construction companies, even large companies, use Microsoft Excel or other convenient tools to record and maintain project data. Most of them are reluctant to transform their existing data management system to a new one. Three potential approaches could solve this problem:

1. Transform the existing data management system to the Microsoft Access database management system, and make decisions based on the transformed database.
2. Extract data from the current data management system, and make decisions based on the current data management system.
3. Implement a decision-making system totally differently from this study. For example, for the companies who are using Microsoft Excel, the decision-making system could be implemented by Visual Basic for Applications (VBA). Allowing

the decision-making system to adapt to a variety of database management systems is important in making the system easier to use and in encourage its adoption.

- Making the selection decision-making system a more comprehensive construction service procurement system

  The selection decision-making process and the supply chain performance measurement are only parts of the construction service outsourcing and the project procurement system. Delivering a comprehensive solution would respond to the sourcing deployment needs from the construction industry and significantly reduce costs and streamline the sourcing process in the procurement of construction services. The dashed lines in Figure 8-1 show the future areas of the Construction Service Outsourcing /Procurement Decision Making System that need to be studied.

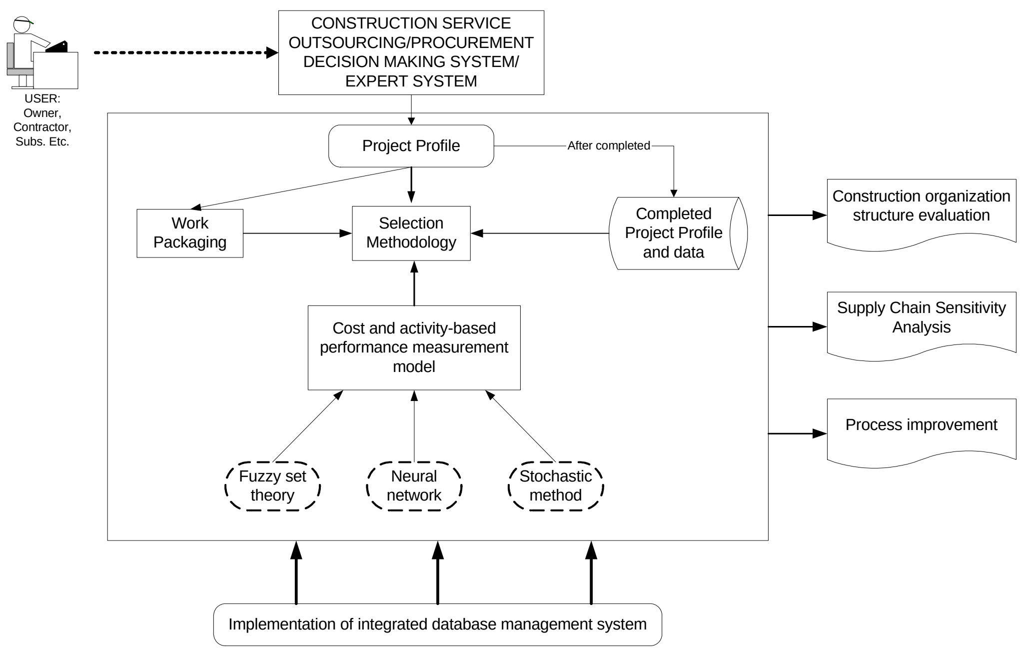


Figure 8-1 Current and Future Construction Service Outsourcing/Procurement Decision Making System

APPENDIX A
COST AND ACTIVITY-BASED PERFORMANCE MEASURMENT DECISION MAKING SYSTEM VB CODE — BUSINESS DEFINITION COMPONENT

```
Option Explicit
Dim dbABPC As Database
Dim datWorkPackages As Recordset
Dim datSubs As Recordset
Dim datPerformances As Recordset
Dim datActPerf As Recordset
Dim datActivities As Recordset
Dim datPerfData As Recordset

Dim Node3Num As Integer

Private Sub cmdAddAct_Click()
    Dim i As Integer
    Dim msg As String

    For i = 1 To grdActivities.Rows - 1
        If grdActivities.TextMatrix(i, 0) = txtActivityIDAdd.Text
Or _
           grdActivities.TextMatrix(i, 1) = txtActivityAdd.Text
Then
            msg = MsgBox("The activity has already existed. Please
reenter information.", vbInformation, "Add New Subcontractors")
            Exit Sub
        End If
    Next i


    If datActivities.RecordCount <> 0 Then
        datActivities.MoveFirst

        For i = 0 To datActivities.RecordCount - 1
            If datActivities("ActivityID") =
Trim$(txtActivityIDAdd.Text) Then
                msg = MsgBox("The activity ID has already existed.
Please reenter information.", vbInformation, "Add New Work Package")
                Exit Sub
            End If
            datActivities.MoveNext
        Next i
    End If

    datActivities.AddNew
    datActivities("WorkPackageID") =
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 0)
    datActivities("ActivityID") = txtActivityIDAdd.Text
```

```
        datActivities("ActivityDescription") = txtActivityAdd.Text
        datActivities("Weights") = txtActWeightAdd.Text
        datActivities.Update

        grdActivities.AddItem (txtActivityIDAdd.Text & vbTab &
txtActivityAdd.Text & vbTab & txtActWeightAdd.Text)
        txtActivityIDAdd.Text = ""
        txtActivityAdd.Text = ""
        txtActWeightAdd.Text = ""
    End Sub

    Private Sub cmdAddActs_Click()
        Dim i As Integer
        Dim msg As String

        For i = 1 To grdSubs.Rows - 1
            If grdSubs.TextMatrix(i, 0) = txtSubIDAdd.Text Or _
               grdSubs.TextMatrix(i, 1) = txtSubNameAdd.Text Then
                msg = MsgBox("The subcontractor has already existed.
Please reenter information.", vbInformation, "Add New Subcontractors")
                Exit Sub
            End If
        Next i

        If datSubs.RecordCount <> 0 Then
            datSubs.MoveFirst
            For i = 0 To datSubs.RecordCount - 1
                If datSubs("SubID") = Trim$(txtSubIDAdd.Text) Then
                    msg = MsgBox("The subcontractor/supplier ID has
already existed. Please reenter information.", vbInformation, "Add New
Work Package")
                    Exit Sub
                End If
                datSubs.MoveNext
            Next i
        End If

        datSubs.AddNew
        datSubs("WorkPackageID") =
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 0)
        datSubs("SubID") = txtSubIDAdd.Text
        datSubs("SubName") = txtSubNameAdd.Text
        datSubs.Update

        grdSubs.AddItem (txtSubIDAdd.Text & vbTab & txtSubNameAdd.Text)
        txtSubIDAdd.Text = ""
        txtSubNameAdd.Text = ""
    End Sub

    Private Sub cmdAddPerf_Click()
        Dim i As Integer
        Dim msg As String

        For i = 1 To grdPerformances.Rows - 1
            If grdPerformances.TextMatrix(i, 0) = txtPerfIDAdd.Text Or
_
               grdPerformances.TextMatrix(i, 1) = txtPerfAdd.Text Then
```

```
                msg = MsgBox("The performance criterion has already
existed. Please reenter information.", vbInformation, "Add New
Subcontractors")
                Exit Sub
            End If
        Next i


        If datPerformances.RecordCount <> 0 Then
            datPerformances.MoveFirst

            For i = 0 To datPerformances.RecordCount - 1
                If datPerformances("PerformanceID") =
Trim$(txtPerfIDAdd.Text) Then
                    msg = MsgBox("The performance criterion ID has
already existed. Please reenter information.", vbInformation, "Add New
Work Package")
                    Exit Sub
                End If
                datPerformances.MoveNext
            Next i
        End If

        datPerformances.AddNew
        datPerformances("WorkPackageID") =
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 0)
        datPerformances("PerformanceID") = txtPerfIDAdd.Text
        datPerformances("Description") = txtPerfAdd.Text
        datPerformances.Update

        grdPerformances.AddItem (txtPerfIDAdd.Text & vbTab &
txtPerfAdd.Text)
        txtPerfIDAdd.Text = ""
        txtPerfAdd.Text = ""

        SetupGridActPerf
        FillGridActPerf
    End Sub

    Private Sub cmdAddWP_Click()
        frmNewWP.Show vbModal
    End Sub

    Private Sub cmdDataAdd_Click()
        Dim i As Integer
        Dim msg As String

        For i = 1 To grdPerfData.Rows - 1
            If grdPerfData.TextMatrix(i, 0) = txtDataIDAdd.Text Or _
               grdPerfData.TextMatrix(i, 1) = txtDataAdd.Text Then
                msg = MsgBox("The Data Item has already existed. Please
reenter information.", vbInformation, "Add New Subcontractors")
                Exit Sub
            End If
        Next i

        If datPerfData.RecordCount <> 0 Then
```

```
            datPerfData.MoveFirst
            For i = 0 To datPerfData.RecordCount - 1
                If datPerfData("DataID") = Trim$(txtDataIDAdd.Text)
Then
                    msg = MsgBox("The Data Item ID has already existed.
Please reenter information.", vbInformation, "Add New Work Package")
                    Exit Sub
                End If
                datPerfData.MoveNext
            Next i
        End If

        datPerfData.AddNew
        datPerfData("WorkPackageID") =
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 0)
        datPerfData("WorkPackage") =
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 1)
        datPerfData("DataID") = txtDataIDAdd.Text
        datPerfData("Data") = txtDataAdd.Text
        datPerfData("Unit") = txtUnitAdd.Text
        datPerfData.Update

        grdPerfData.AddItem (txtDataIDAdd.Text & vbTab &
txtDataAdd.Text & vbTab & txtUnitAdd.Text)
        txtDataIDAdd.Text = ""
        txtDataAdd.Text = ""
        txtUnitAdd.Text = ""
    End Sub

    Private Sub cmdDataDel_Click()
        Dim Idd As String
        Dim Message As String
        Dim Response As Integer

        Idd = grdPerfData.TextMatrix(grdPerfData.RowSel, 0)
        Message = "Are you sure to delete this Data Item?"

        Response = MsgBox(Message, vbQuestion + vbYesNo)
        If Response = vbYes And grdPerfData.RowSel = 1 Then
            grdPerfData.Clear

            With datPerfData
                .Requery
                .FindFirst "DataID = " & "'" & Idd & "'"
                .Delete
            End With
            SetupGridPerfData
            FillGridPerfData
        ElseIf Response = vbYes Then
            grdPerfData.RemoveItem (grdPerfData.RowSel)
            With datPerfData
                .Requery
                .FindFirst "DataID = " & "'" & Idd & "'"
                .Delete
            End With
        End If
    End Sub
```

```
Private Sub cmdDelAct_Click()
    Dim Idd As String
    Dim Message As String
    Dim Response As Integer

    Idd = grdActivities.TextMatrix(grdActivities.RowSel, 0)
    Message = "Are you sure to delete this activity?"

    Response = MsgBox(Message, vbQuestion + vbYesNo)
    If Response = vbYes Then
        grdActivities.RemoveItem (grdActivities.RowSel)
        With datActivities
            .Requery
            .FindFirst "ActivityID = " & "'" & Idd & "'"
            .Delete
        End With
    End If
End Sub

Private Sub cmdDeleteActs_Click()
    Dim Idd As String
    Dim Message As String
    Dim Response As Integer

    Idd = grdSubs.TextMatrix(grdSubs.RowSel, 0)
    Message = "Are you sure to delete this sub/supplier?"

    Response = MsgBox(Message, vbQuestion + vbYesNo)
    If Response = vbYes And grdSubs.RowSel = 1 Then
        grdSubs.Clear

        With datSubs
            .Requery
            .FindFirst "SubID = " & "'" & Idd & "'"
            .Delete
        End With
        SetupGridSub
        FillGridSub
    ElseIf Response = vbYes Then
        grdSubs.RemoveItem (grdSubs.RowSel)
        With datSubs
            .Requery
            .FindFirst "SubID = " & "'" & Idd & "'"
            .Delete
        End With
    End If
End Sub

Private Sub cmdDeleteWP_Click()
   Dim Idd As String
   Dim Message As String
   Dim Response As Integer

   Message = "Are you sure to delete this work package?"
   Idd = grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 0)
```

```
        Response = MsgBox(Message, vbQuestion + vbYesNo)
        If Response = vbYes Then
             With datWorkPackages
                 .Requery
                 .FindFirst "WorkPackageID = " & "'" & Idd & "'"
                 If .NoMatch = False Then
                  .Delete
                 End If
             End With
             grdWorkPackages.RemoveItem (grdWorkPackages.RowSel)
             lblWP.Caption = ""
         Else
             Exit Sub
         End If
     End Sub

     Private Sub cmdDelPerf_Click()
         Dim Idd As String
         Dim Message As String
         Dim Response As Integer

         Idd = grdPerformances.TextMatrix(grdPerformances.RowSel, 0)
         Message = "Are you sure to delete this performance criterion?"

         Response = MsgBox(Message, vbQuestion + vbYesNo)
         If Response = vbYes Then
             grdPerformances.RemoveItem (grdPerformances.RowSel)
             With datPerformances
                 .Requery
                 .FindFirst "PerformanceID = " & "'" & Idd & "'"
                 .Delete
             End With
         End If
     End Sub

     Private Sub Form_Load()
         Dim SQLQuery1 As String
         Dim SQLQuery2 As String
         Dim SQLQuery3 As String
         Dim SQLQuery4 As String
         Dim SQLQuery5 As String

         Me.Height = 9825
         Me.Width = 16005

         Set dbABPC = OpenDatabase(App.Path & "\Database\ABPC.mdb")
         Set datWorkPackages = dbABPC.OpenRecordset("WorkPackages",
dbOpenDynaset)
         Set datSubs = dbABPC.OpenRecordset("Subcontractors",
dbOpenDynaset)
         Set datPerformances = dbABPC.OpenRecordset("Performances",
dbOpenDynaset)
         Set datActPerf = dbABPC.OpenRecordset("ActivityPerformance",
dbOpenDynaset)
         Set datActivities = dbABPC.OpenRecordset("Activities",
dbOpenDynaset)
```

```
        Set datPerfData = dbABPC.OpenRecordset("DataItem",
dbOpenDynaset)


        SetupGridWP
        FillGridWP
        SetupGridSub
        SetupGridPerf
        SetupGridActPerf
        SetupGridAct
        SetupGridPerfData
    End Sub

    Private Sub SetupGridWP()

        Dim WideString1, WideString2 As String
        Dim C As Integer, GridWidth As Integer

        WideString1 = String$(15, "X")
        WideString2 = String$(25, "X")

        With grdWorkPackages
            .SelectionMode = flexSelectionByRow
            .Cols = 2
            .FixedCols = 0
            .ColWidth(0) = TextWidth(WideString1)
            .ColWidth(1) = TextWidth(WideString2)
        End With

        grdWorkPackages.TextMatrix(0, 0) = "Work Package ID"
        grdWorkPackages.TextMatrix(0, 1) = "Work Package Description"

    End Sub
    Private Sub SetupGridSub()
        Dim WideString1, WideString2 As String
        Dim C As Integer, GridWidth As Integer

        WideString1 = String$(15, "X")
        WideString2 = String$(25, "X")

        With grdSubs
            .SelectionMode = flexSelectionByRow
            .Cols = 2
            .FixedCols = 0
            .ColWidth(0) = TextWidth(WideString1)
            .ColWidth(1) = TextWidth(WideString2)
        End With

        grdSubs.TextMatrix(0, 0) = "Subcontractor ID"
        grdSubs.TextMatrix(0, 1) = "Subcontractor Name"

    End Sub
    Private Sub SetupGridPerf()
        Dim WideString1, WideString2 As String

        WideString1 = String$(13, "X")
        WideString2 = String$(24, "X")
```

```
    With grdPerformances
        .SelectionMode = flexSelectionByRow
        .Cols = 2
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)
    End With

    grdPerformances.TextMatrix(0, 0) = "Performance ID"
    grdPerformances.TextMatrix(0, 1) = "Performance Description"

End Sub
Private Sub SetupGridActPerf()
    Dim WideString1, WideString2 As String
    Dim C As Integer, GridWidth As Integer

    WideString1 = String$(13, "X")
    WideString2 = String$(24, "X")

    With grdActPerf
        .SelectionMode = flexSelectionByRow
        .Cols = 2
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)
    End With

    grdActPerf.TextMatrix(0, 0) = "Performance ID"
    grdActPerf.TextMatrix(0, 1) = "Performance Description"

End Sub
Private Sub SetupGridAct()
    Dim WideString1, WideString2, WideString3 As String
    Dim C As Integer, GridWidth As Integer

    WideString1 = String$(9, "X")
    WideString2 = String$(18, "X")
    WideString3 = String$(13, "X")

    With grdActivities
        .SelectionMode = flexSelectionByRow
        .Cols = 3
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)
        .ColWidth(2) = TextWidth(WideString3)
    End With

    grdActivities.TextMatrix(0, 0) = "Activity ID"
    grdActivities.TextMatrix(0, 1) = "Activity Description"
    grdActivities.TextMatrix(0, 2) = "Activity Weight"
End Sub


Private Sub SetupGridPerfData()
    Dim WideString1, WideString2, WideString3 As String
```

```
    Dim C As Integer, GridWidth As Integer

    WideString1 = String$(9, "X")
    WideString2 = String$(18, "X")
    WideString3 = String$(13, "X")

    With grdPerfData
        .SelectionMode = flexSelectionByRow
        .Cols = 3
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)
        .ColWidth(2) = TextWidth(WideString3)
    End With

    grdPerfData.TextMatrix(0, 0) = "Data Item ID"
    grdPerfData.TextMatrix(0, 1) = "Data Item"
    grdPerfData.TextMatrix(0, 2) = "Unit"
End Sub


Private Sub FillGridWP()
    Dim NewRow As String
    Do Until datWorkPackages.EOF = True
        NewRow = datWorkPackages("WorkPackageID") & vbTab &
datWorkPackages("WorkPackageDescription")
        grdWorkPackages.AddItem NewRow
        datWorkPackages.MoveNext
    Loop

End Sub

Private Sub FillGridPerfData()
Dim i As Integer

i = 1
If datPerfData.RecordCount <> 0 Then
datPerfData.MoveFirst

    Do Until datPerfData.EOF = True
        grdPerfData.TextMatrix(i, 0) = datPerfData("DataID")
        grdPerfData.TextMatrix(i, 1) = datPerfData("Data")
        grdPerfData.TextMatrix(i, 2) = datPerfData("Unit")
        i = i + 1
        datPerfData.MoveNext
    Loop

End If

End Sub

Private Sub FillGridSub()
Dim i As Integer

i = 1
If datSubs.RecordCount <> 0 Then
datSubs.MoveFirst
```

```
        Do Until datSubs.EOF = True
            grdSubs.TextMatrix(i, 0) = datSubs("SubID")
            grdSubs.TextMatrix(i, 1) = datSubs("SubName")
            i = i + 1
            datSubs.MoveNext
        Loop

    End If

    End Sub
    Private Sub FillGridPerf()
        Dim i As Integer

        i = 1
        If datPerformances.RecordCount <> 0 Then
        datPerformances.MoveFirst
        End If

        Do Until datPerformances.EOF = True
            grdPerformances.TextMatrix(i, 0) =
datPerformances("PerformanceID")
            grdPerformances.TextMatrix(i, 1) =
datPerformances("Description")
            i = i + 1
            datPerformances.MoveNext
        Loop
    End Sub
    Private Sub FillGridActPerf()
        Dim i As Integer

        i = 1
        If datPerformances.RecordCount <> 0 Then
            datPerformances.MoveFirst
        End If

        Do Until datPerformances.EOF = True
            grdActPerf.TextMatrix(i, 0) =
datPerformances("PerformanceID")
            grdActPerf.TextMatrix(i, 1) =
datPerformances("Description")
            i = i + 1
            datPerformances.MoveNext
        Loop

    End Sub
    Private Sub FillGridAct()
        Dim i As Integer

        i = 1
        If datActivities.RecordCount <> 0 Then
        datActivities.MoveFirst
        End If

        Do Until datActivities.EOF = True
            grdActivities.TextMatrix(i, 0) =
datActivities("ActivityID")
```

```
            grdActivities.TextMatrix(i, 1) =
datActivities("ActivityDescription")
            grdActivities.TextMatrix(i, 2) = datActivities("Weights")
            i = i + 1
            datActivities.MoveNext
        Loop
    End Sub
    Private Sub FillTVWActPerf()
        'Populate activity-performance tree view
        Dim Node1, Node2, Node3 As Node
        Dim j, k As Integer


        Set Node1 = tvwActPerf.Nodes.Add(, , "R",
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 1) + " " +
"Activities")


        If datActivities.RecordCount <> 0 Then
            datActivities.MoveFirst

            For j = 0 To datActivities.RecordCount - 1
                Set Node2 = tvwActPerf.Nodes.Add("R", tvwChild, "A" +
Str$(j + 1), datActivities("ActivityDescription"))

                If datActPerf.RecordCount <> 0 Then
                    datActPerf.MoveFirst
                    Do Until datActPerf.EOF = True
                        If datActPerf("WorkPackageID") =
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 0) And _
                            datActPerf("ActivityID") =
datActivities("ActivityID") Then
                            Set Node3 = tvwActPerf.Nodes.Add("A" +
Str$(j + 1), tvwChild, "P" + Str$(Node3Num), datActPerf("Performance"))
                            Node3.Key = datActPerf("WAPID")
                            Node3Num = Node3Num + 1
                        End If
                        datActPerf.MoveNext
                    Loop
                End If
                If Node2.Index > 0 Then
                    Node2.Expanded = True
                End If
                datActivities.MoveNext
            Next j

        End If

        If Node1.Index > 0 Then
            Node1.Expanded = True
        End If
    End Sub


    Private Sub grdActivities_Click()
        txtActivityIDDel.Text =
grdActivities.TextMatrix(grdActivities.RowSel, 0)
```

```
        txtActivityDel.Text =
grdActivities.TextMatrix(grdActivities.RowSel, 1)
        txtActWeightDel.Text =
grdActivities.TextMatrix(grdActivities.RowSel, 2)
    End Sub

    Private Sub grdPerfData_Click()
        txtDataIDDel.Text = grdPerfData.TextMatrix(grdPerfData.RowSel,
0)
        txtDataDel.Text = grdPerfData.TextMatrix(grdPerfData.RowSel, 1)
        txtUnitDel.Text = grdPerfData.TextMatrix(grdPerfData.RowSel, 2)
    End Sub

    Private Sub grdPerformances_Click()
        txtPerfIDDel.Text =
grdPerformances.TextMatrix(grdPerformances.RowSel, 0)
        txtPerfDel.Text =
grdPerformances.TextMatrix(grdPerformances.RowSel, 1)
    End Sub

    Private Sub grdSubs_Click()
        txtSubIDDel.Text = grdSubs.TextMatrix(grdSubs.RowSel, 0)
        txtSubNameDel.Text = grdSubs.TextMatrix(grdSubs.RowSel, 1)
    End Sub

    Private Sub grdWorkPackages_Click()
        lblWP = grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 1)

        Dim SQLQuery As String
        Dim msg As String
        Dim i As Integer
        Dim count As Integer

        i = grdWorkPackages.RowSel

        'Populate subcontractors' flexible grids
        SQLQuery = "Select * From Subcontractors Where WorkPackageID =
'"
        SQLQuery = SQLQuery + grdWorkPackages.TextMatrix(i, 0) + "'"

        Set datSubs = dbABPC.OpenRecordset(SQLQuery, dbOpenDynaset)

        count = 0
        Do Until datSubs.EOF = True
            count = count + 1
            datSubs.MoveNext
        Loop
        grdSubs.Rows = count + 1

        FillGridSub

        'Populate data items' flexible grids
        SQLQuery = "Select * From DataItem Where WorkPackageID = '"
        SQLQuery = SQLQuery + grdWorkPackages.TextMatrix(i, 0) + "'"


        Set datPerfData = dbABPC.OpenRecordset(SQLQuery, dbOpenDynaset)
```

```
        count = 0
        Do Until datPerfData.EOF = True
            count = count + 1
            datPerfData.MoveNext
        Loop
        grdPerfData.Rows = count + 1

        FillGridPerfData

        'Populate performances' flexible grids
        SQLQuery = "Select * From Performances Where WorkPackageID = '"
        SQLQuery = SQLQuery + grdWorkPackages.TextMatrix(i, 0) + "'"

        Set datPerformances = dbABPC.OpenRecordset(SQLQuery,
dbOpenDynaset)

        SQLQuery = "Select * From ActivityPerformance Where
WorkPackageID='"
        SQLQuery = SQLQuery + grdWorkPackages.TextMatrix(i, 0) + "'"
        Set datActPerf = dbABPC.OpenRecordset(SQLQuery, dbOpenDynaset)

        count = 0
        Do Until datPerformances.EOF = True
            count = count + 1
            datPerformances.MoveNext
        Loop
        grdPerformances.Rows = count + 1
        grdActPerf.Rows = count + 1

        FillGridPerf
        FillGridActPerf

        'Populate activities' flexible grids
        SQLQuery = "Select * From Activities Where WorkPackageID = '"
        SQLQuery = SQLQuery + grdWorkPackages.TextMatrix(i, 0) + "'"

        Set datActivities = dbABPC.OpenRecordset(SQLQuery,
dbOpenDynaset)

        count = 0
        Do Until datActivities.EOF = True
            count = count + 1
            datActivities.MoveNext
        Loop
        grdActivities.Rows = count + 1

        tvwActPerf.Nodes.Clear
        FillGridAct
        FillTVWActPerf



    End Sub

    Private Sub mnuActivities_Click()
        frmActivities.Show
```

```
    End Sub

    Private Sub mnuActivityPerformance_Click()
        frmActPerf.Show
    End Sub

    Private Sub mnuCostModel_Click()
        CreatePriceNodes
        CreateAlternativePrice
        CalPriceWeight
        frmCostModel.Show
    End Sub

    Private Sub mnuExit_Click()
        End
    End Sub

    Private Sub mnuPerformances_Click()
        frmPerformances.Show
    End Sub

    Private Sub mnuProjectInformation_Click()
        frmNewProject.Show
    End Sub

    Private Sub mnuProjectWorkPackages_Click()
        frmNewProjectWorkPacks.Show
    End Sub

    Private Sub mnuSub_Click()
        frmSubs.Show
    End Sub

    Private Sub mnuSubWorkPack_Click()
        frmNewProjectSubs.Show
    End Sub

    Private Sub mnuWorkPackages_Click()
        frmWorkPackages.Show
    End Sub

    Private Sub tvwActPerf_NodeClick(ByVal Node As MSComctlLib.Node)
        Dim Node3 As Node
        Dim msg As String
        Dim i As Integer
        Dim WAPID As String
        Dim NodeKey As String

        'identify NodeKey and WAPID
        For i = 1 To grdActivities.Rows - 1
            If Node.Text = grdActivities.TextMatrix(i, 1) Then
                    NodeKey = grdActivities.TextMatrix(i, 0)
            End If
        Next i
        WAPID = grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 0) +
"-" + NodeKey + "-" + grdActPerf.TextMatrix(grdActPerf.RowSel, 0)
```

```
        If Node.Key = "R" Then
            Exit Sub
        ElseIf Node.Parent.Key = "R" Then
            msg = MsgBox("Are you sure to add this performance
criterion to this activity?", vbYesNo + vbQuestion, "Question")
            If msg = vbYes Then
                If datActPerf.RecordCount <> 0 Then
                    datActPerf.MoveFirst
                End If

                Do Until datActPerf.EOF = True
                    If WAPID = datActPerf("WAPID") Then
                        msg = MsgBox("The relationship has already
existed. Try again.", vbOKOnly, "Information")
                        Exit Sub
                    End If
                    datActPerf.MoveNext
                Loop
                Set Node3 = tvwActPerf.Nodes.Add(Node, tvwChild, ,
grdActPerf.TextMatrix(grdActPerf.RowSel, 1))
                Node3.Key = WAPID
                datActPerf.AddNew
                datActPerf("WAPID") = WAPID
                datActPerf("WorkPackageID") =
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 0)
                datActPerf("WorkPackage") =
grdWorkPackages.TextMatrix(grdWorkPackages.RowSel, 1)
                datActPerf("ActivityID") = NodeKey
                datActPerf("Activity") = Node.Text
                datActPerf("PerformanceID") =
grdActPerf.TextMatrix(grdActPerf.RowSel, 0)
                datActPerf("Performance") =
grdActPerf.TextMatrix(grdActPerf.RowSel, 1)

                datActivities.MoveFirst
                Do Until datActivities.EOF = True
                    If datActivities("ActivityID") = NodeKey Then
                        datActPerf("Weights") =
datActivities("Weights")
                    End If
                    datActivities.MoveNext
                Loop
                datActPerf.Update
            End If

        ElseIf Node.Parent.Parent.Key = "R" Then
            msg = MsgBox("Are you sure to delete this performance
criterion for this activity?", vbYesNo + vbQuestion, "Question")

            If msg = vbYes Then
                tvwActPerf.Nodes.Remove (Node.Index)
                With datActPerf
                    .Requery
                    .FindFirst "WAPID = " & "'" & Node.Key & "'"
                    .Delete
                End With
             End If
```

```
    Else: Exit Sub
    End If

End Sub
```

APPENDIX B

COST AND ACTIVITY-BASED PERFORMANCE MEASURMENT DECISION MAKING SYSTEM VB CODE — PROJECT DECISION MAKING COMPONENT

```
Option Explicit
Dim dbABPC As Database
Dim datProjectWP As Recordset
Dim datProjectSubs As Recordset
Dim datProjectWPSubs As Recordset

Dim WPNum As Integer
Dim SubNum As Integer
Dim WPNodeArray(0 To 100, 0 To 2) As String
'SubNodeArray(i,0)=Sub,
SubNodeArray(i,1)=NodeKey,SubNodeArray(i,2)=WorkPackage
Dim SubNodeArray(0 To 100, 0 To 2) As String
Dim TotalWeights() As Double
Dim a1, a2 As Integer

Private Sub DrawSensiChart()
    Dim i, j As Integer
    Dim X() As Variant
    Dim Y() As Variant
    Dim label(0 To 4) As String
    Dim Alternatives As Integer
    Dim w As Single


    With ChartSensi
        .chartType = VtChChartType2dLine
        .ShowLegend = True
        .ColumnCount = 11
    End With

    Alternatives = 1
    For i = 1 To WorkPacks
        Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
    Next i

    ReDim X(0 To Alternatives, 0 To 2)
    ReDim Y(1 To Alternatives, 0 To 4)

    X(0, 0) = "Alternatives"
    For i = 1 To Alternatives
        X(i, 0) = "Alt." & i
        Y(i, 0) = "Alt." & i
    Next i
    X(0, 1) = "Cost Weight"
    X(0, 2) = "Performance Weight"
```

```
    For j = 1 To 4
        w = j * 2 / 10
        For i = 1 To Alternatives
            X(i, 1) = w * PriceComp(i - 1, Alternatives)
            X(i, 2) = (1 - w) * PerfWeights(i - 1)
            Y(i, j) = (X(i, 1) + X(i, 2)) * 100
        Next i

        label(j) = "cost weight = " + Format$(w, "0.00")

    Next j

    ChartSensi.ChartData = Y

    For j = 1 To 4
        ChartSensi.Column = j
        ChartSensi.ColumnLabel = label(j)
    Next j
End Sub
Private Sub DrawSensiChart2()
    Dim i, j As Integer
    Dim X() As Variant
    Dim Y() As Variant
    Dim label(0 To 4) As String
    Dim Alternatives As Integer
    Dim w As Single


    With chartSensi2
        .chartType = VtChSeriesType2dBar
        .ShowLegend = True
        .ColumnCount = 11
    End With

    Alternatives = 1
    For i = 1 To WorkPacks
        Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
    Next i

    ReDim X(0 To Alternatives, 0 To 2)
    ReDim Y(0 To 4, 0 To Alternatives)

    X(0, 0) = "Alternatives"
    For i = 1 To Alternatives
        X(i, 0) = "Alt." & i
        Y(0, i) = "Altive." & i
    Next i
    X(0, 1) = "Cost Weight"
    X(0, 2) = "Performance Weight"

    For j = 1 To 4
        w = j * 2 / 10

        'ReDim PerfWeights(0 To Alternatives) As Double
        Y(j, 0) = "cost weight=" + Format$(w, "0.00")
        For i = 1 To Alternatives
            X(i, 1) = w * PriceComp(i - 1, Alternatives)
```

```
            X(i, 2) = (1 - w) * PerfWeights(i - 1)
            Y(j, i) = (X(i, 1) + X(i, 2)) * 100
        Next i

        label(j) = "cost weight = " + Format$(w, "0.00")

    Next j


    chartSensi2.ChartData = Y

    For j = 1 To 4
        ChartSensi.Column = j
        ChartSensi.ColumnLabel = label(j)
    Next j

End Sub

Private Sub DrawCostChart()
    Dim i, j As Integer
    Dim X() As Variant
    Dim Alternatives As Integer

    With chartCost
        .chartType = VtChChartType2dBar
        .Plot.Wall.Pen.Style = VtPenStyleNull
    End With


    Alternatives = 1
    For i = 1 To WorkPacks
        Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
    Next i

    ReDim X(0 To Alternatives, 0 To 1)

    X(0, 0) = "Alternatives"
    For i = 1 To Alternatives
        X(i, 0) = "Alt." & i
    Next i

    X(0, 1) = "Cost"
    For i = 1 To Alternatives
        X(i, 1) = AltTotalPrice(i - 1)
    Next i

    chartCost.ChartData = X
End Sub


Private Sub DrawPerfChart()
    Dim i, j As Integer
    Dim X() As Variant
    Dim Alternatives As Integer

    With chartPerf
        .chartType = VtChChartType2dBar
```

```
        .Plot.Wall.Pen.Style = VtPenStyleNull
    End With


    Alternatives = 1
    For i = 1 To WorkPacks
        Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
    Next i

    ReDim X(0 To Alternatives, 0 To 1)

    X(0, 0) = "Alternatives"
    For i = 1 To Alternatives
        X(i, 0) = "Alt." & i
    Next i

    X(0, 1) = "Performance"
    For i = 1 To Alternatives
        X(i, 1) = PerfWeights(i - 1) * 100
    Next i

    chartPerf.ChartData = X
End Sub

Private Sub DrawCostWeightChart()
    Dim i, j As Integer
    Dim X() As Variant
    Dim Alternatives As Integer

    With chartCostWeight
        .chartType = VtChChartType2dBar
        .Plot.Wall.Pen.Style = VtPenStyleNull
    End With


    Alternatives = 1
    For i = 1 To WorkPacks
        Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
    Next i

    ReDim X(0 To Alternatives, 0 To 1)

    X(0, 0) = "Alternatives"
    For i = 1 To Alternatives
        X(i, 0) = "Alt." & i
    Next i

    X(0, 1) = "Cost Weight"
    For i = 1 To Alternatives
        X(i, 1) = PriceComp(i - 1, Alternatives) * 100
    Next i

    chartCostWeight.ChartData = X
End Sub


Private Sub DrawCostPerfWeightChart()
```

```
    Dim i, j As Integer
    Dim X() As Variant
    Dim Alternatives As Integer

    With chartCostWeight
        .chartType = VtChChartType2dBar
        .Plot.Wall.Pen.Style = VtPenStyleNull
    End With


    With chartTotalWeight
        .chartType = VtChChartType2dBar
        .Stacking = True
    End With

    Alternatives = 1
    For i = 1 To WorkPacks
        Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
    Next i

    ReDim X(0 To Alternatives, 0 To 2)

    X(0, 0) = "Alternatives"
    For i = 1 To Alternatives
        X(i, 0) = "Alt." & i
    Next i

    'ReDim PerfWeights(0 To Alternatives) As Double
    X(0, 1) = "Cost Weight"
    X(0, 2) = "Performance Weight"
    For i = 1 To Alternatives
        X(i, 1) = CostWt * PriceComp(i - 1, Alternatives) * 100
        X(i, 2) = PerfWt * PerfWeights(i - 1) * 100
    Next i

    chartTotalWeight.ChartData = X
End Sub


Private Sub SetTVWAlt()
Dim ANode As Node    'Add a node to the tree
Dim nd1 As Node      'store "R" node in tvwProjectWP
Dim nd2 As Node      'store "WP" node in tvwProjectWP
Dim nd3 As Node      'store "Sub" node in tvwProjectWP
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim kk As Integer
Dim m As Integer
Dim NNode As Integer
Dim nd3Text As String

tvwAlt.Nodes.Clear
Set ANode = tvwAlt.Nodes.Add(, , "L0-1", "Alternatives")
ANode.Expanded = True
Set nd1 = tvwProjectWP.Nodes("R")
Set nd2 = nd1.Child
```

```
NNode = 1
kk = 1
'this level is used to control the level of tree
For i = 1 To nd1.Children
    If nd2.Children <> 0 Then

    'this level is used to populate the nodes on a specific level
    NNode = NNode * nd2.Children
    Set nd3 = nd2.Child
    m = 1
    For k = 1 To kk

        For j = 1 To nd2.Children
            If i = nd1.Children Then
                nd3Text = nd3.Text + "         " + "Alternative " +
Trim$(Str$(m))
                Set ANode = tvwAlt.Nodes.Add("L" + Trim$(i - 1) +
"-" + Trim$(k), tvwChild, "L" + Trim$(i) + "-" + Trim$(m), nd3Text)
                ANode.Expanded = True
                Set nd3 = nd3.Next
                m = m + 1
            Else
                Set ANode = tvwAlt.Nodes.Add("L" + Trim$(i - 1) +
"-" + Trim$(k), tvwChild, "L" + Trim$(i) + "-" + Trim$(m), nd3.Text)
                ANode.Expanded = True
                Set nd3 = nd3.Next
                m = m + 1
            End If
        Next j
        Set nd3 = nd2.Child
    Next k

    kk = NNode
    Set nd2 = nd2.Next

    End If
Next i

End Sub

Private Sub SetupProjectInfGrid()
    Dim WideString1, WideString2 As String
    Dim C As Integer, GridWidth As Integer

    WideString1 = String$(40, "X")
    WideString2 = String$(44, "X")

    With grdProjectInf
        .SelectionMode = flexSelectionByRow
        .Cols = 2
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)
    End With


End Sub
```

```
Private Sub SetupProjectGridWP()

    Dim WideString1, WideString2 As String
    Dim C As Integer, GridWidth As Integer

    WideString1 = String$(15, "X")
    WideString2 = String$(25, "X")

    With grdProjectWP
        .SelectionMode = flexSelectionByRow
        .Cols = 2
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)
    End With

    grdProjectWP.TextMatrix(0, 0) = "Work Package ID"
    grdProjectWP.TextMatrix(0, 1) = "Work Package Description"

End Sub
Private Sub SetupProjectGridSubs()
    Dim WideString1, WideString2, WideString3 As String
    Dim C As Integer, GridWidth As Integer

    WideString1 = String$(15, "X")
    WideString2 = String$(15, "X")
    WideString3 = String$(12, "X")

    With grdProjectSubs
        .SelectionMode = flexSelectionByRow
        .Cols = 3
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)
        .ColWidth(2) = TextWidth(WideString3)
    End With

    grdProjectSubs.TextMatrix(0, 0) = "Work Package ID"
    grdProjectSubs.TextMatrix(0, 1) = "Sub/Supplier ID"
    grdProjectSubs.TextMatrix(0, 2) = "Sub/Supplier"
End Sub
Private Sub FillProjectGridWP()
    Dim NewRow As String

    Do Until datProjectWP.EOF = True
        NewRow = datProjectWP("WorkPackageID") & vbTab &
datProjectWP("WorkPackageDescription")
        grdProjectWP.AddItem NewRow
        datProjectWP.MoveNext
    Loop

End Sub

Private Sub FillProjectInfGrid()

    grdProjectInf.TextMatrix(0, 0) = "Project ID"
    grdProjectInf.TextMatrix(0, 1) = ProjectID
```

```
        grdProjectInf.TextMatrix(1, 0) = "Project"
        grdProjectInf.TextMatrix(1, 1) = ProjectName
        grdProjectInf.TextMatrix(2, 0) = "Project Manager"
        grdProjectInf.TextMatrix(2, 1) = Manager
        grdProjectInf.TextMatrix(3, 0) = "Project Location"
        grdProjectInf.TextMatrix(3, 1) = Location
        grdProjectInf.TextMatrix(4, 0) = "Square Footage"


    End Sub


    Private Sub FillProjectGridSubs()
        Dim NewRow As String
        Dim i As Integer

        grdProjectSubs.Clear
        SetupProjectGridSubs

        i = 1
        If datProjectSubs.RecordCount <> 0 Then
            'grdProjectSubs.Rows = datProjectSubs.Recordset.RecordCount
+ 2
            datProjectSubs.MoveFirst
            Do Until datProjectSubs.EOF = True
                grdProjectSubs.TextMatrix(i, 0) =
datProjectSubs("WorkPackageID")
                grdProjectSubs.TextMatrix(i, 1) =
datProjectSubs("SubID")
                grdProjectSubs.TextMatrix(i, 2) =
datProjectSubs("SubName")
                i = i + 1
                datProjectSubs.MoveNext
            Loop
        ElseIf datProjectSubs.RecordCount = 0 Then
            grdProjectSubs.Rows = 1
        End If
    End Sub

    Private Sub SetupGrid(ByVal grdObject As Object)
        Dim WideString1, WideString2, WideString3, WideString4,
WideString5 As String
        Dim C As Integer, GridWidth As Integer

        WideString1 = String$(15, "X")
        WideString2 = String$(12, "X")
        WideString3 = String$(14, "X")
        WideString4 = String$(21, "X")
        WideString5 = String$(15, "X")

        With grdObject
            .SelectionMode = flexSelectionByRow
            .Cols = WorkPacks + 5
            .FixedCols = 0
            .ColWidth(0) = TextWidth(WideString1)
            .ColWidth(WorkPacks + 1) = TextWidth(WideString2)
            .ColWidth(WorkPacks + 2) = TextWidth(WideString3)
```

```
            .ColWidth(WorkPacks + 3) = TextWidth(WideString4)
            .ColWidth(WorkPacks + 4) = TextWidth(WideString5)
        End With

        For C = 1 To WorkPacks
            grdObject.ColWidth(C) = TextWidth(ProjectWorkPacks(C - 1,
1)) + 5
            grdObject.TextMatrix(0, C) = ProjectWorkPacks(C - 1, 1)
        Next C

        'grdObject.Width = GridWidth
        grdObject.TextMatrix(0, 0) = "Alternatives"
        grdObject.TextMatrix(0, WorkPacks + 1) = "Total Cost"
        grdObject.TextMatrix(0, WorkPacks + 2) = "Cost Weights"
        grdObject.TextMatrix(0, WorkPacks + 3) = "Performance Weights"
        grdObject.TextMatrix(0, WorkPacks + 4) = "Total Weights"
    End Sub


    Private Sub FillGridCostModel()
        Dim NewRow As String
        Dim i, j, k As Integer
        Dim Alternatives As Integer
        Dim Subs As Integer
        Dim FormatWeight As String


        grdCostModel.Rows = Alternatives + 1

        Alternatives = 1
        For i = 0 To WorkPacks - 1
            Alternatives = Alternatives * SubsInWorkPack(i, 1)
        Next i

        For i = 0 To WorkPacks
            Subs = Subs + SubsInWorkPack(i, 1)
        Next i

        'Sum up the cost weights and performance weights to each
alternative
        ReDim TotalWeights(0 To Alternatives) As Double
        For i = 0 To Alternatives
            TotalWeights(i) = PriceComp(i, Alternatives) +
PerfWeights(i)
        Next i

        For i = 1 To Alternatives
            NewRow = "Alt." + Str$(i) + vbTab
            For j = 1 To WorkPacks
                For k = 0 To Subs - 1
                    If Alternative(i - 1, j) = WorkPackSubPrice(k, 1)
Then
                        NewRow = NewRow & WorkPackSubPrice(k, 2) &
vbTab
                    End If
                Next k
            Next j
```

```
            FormatWeight = Format$(PriceComp(i - 1, Alternatives),
"0.0000")
            NewRow = NewRow & Format$(AltTotalPrice(i - 1), "$.00") &
vbTab & FormatWeight
            FormatWeight = Format$(PerfWeights(i - 1), "0.0000")
            NewRow = NewRow & vbTab & FormatWeight
            FormatWeight = Format$(CostWt * PriceComp(i - 1,
Alternatives) + PerfWt * PerfWeights(i - 1), "0.0000")
            NewRow = NewRow & vbTab & FormatWeight
            grdCostModel.AddItem NewRow
            NewRow = ""
        Next i

    End Sub

    Private Sub datProjectWP_Validate(Action As Integer, Save As
Integer)

    End Sub

    Private Sub Form_Load()
        Dim SQLQuery1 As String
        Dim SQLQuery2 As String
        Dim Node1 As Node
        Dim i, j As Integer

        WPNum = 0
        SubNum = 0
        a1 = 0
        a2 = 0
        setPerfWeight = True

        For i = 0 To 100
            For j = 0 To 2
                WPNodeArray(i, j) = ""
                SubNodeArray(i, j) = ""
            Next j
        Next i

        tvwProjectWP.Refresh

        Me.Height = 8115
        Me.Width = 11025

        'set cost weight and performance weight
        CostWt = 0.5
        PerfWt = 0.5

        SetupProjectInfGrid
        FillProjectInfGrid

        Set dbABPC = OpenDatabase(App.Path & "\Database\ABPC.mdb")
        Set datProjectWP = dbABPC.OpenRecordset("WorkPackages",
dbOpenDynaset)
        Set datProjectWPSubs = dbABPC.OpenRecordset("ProjectSubs",
dbOpenDynaset)
```

```
        SetupProjectGridWP
        FillProjectGridWP
        SetupProjectGridSubs

        Set Node1 = tvwProjectWP.Nodes.Add(, , "R", "Work Packages")
        Node1.Expanded = True

    End Sub

    Private Sub grdProjectSubs_Click()
        Dim msg As String
        Dim str1 As String
        Dim Node3 As Node
        Dim i As Integer
        Dim NodeKey As String

        For i = 0 To tvwProjectWP.Nodes.count
            If SubNodeArray(i, 0) =
grdProjectSubs.TextMatrix(grdProjectSubs.RowSel, 2) Then
                Exit Sub
            End If
        Next i


        For i = 0 To tvwProjectWP.Nodes.count
            If WPNodeArray(i, 0) =
grdProjectWP.TextMatrix(grdProjectWP.RowSel, 1) Then
                NodeKey = WPNodeArray(i, 1)
            End If
        Next i

        str1 = "Please input cost for this " +
grdProjectWP.TextMatrix(grdProjectWP.RowSel, 1)
        msg = InputBox(str1, "Information", 0)

        If SubNum > 0 Then
            For i = 0 To SubNum - 1
                If tvwProjectWP.Nodes("S" + Str$(i)).Text =
grdProjectSubs.TextMatrix(grdProjectSubs.RowSel, 1) Then
                   Exit Sub
                End If
            Next i
        End If

        Set Node3 = tvwProjectWP.Nodes.Add(NodeKey, tvwChild, "S" +
Str$(SubNum), "  $" + Trim$(msg) + ",  " +
grdProjectSubs.TextMatrix(grdProjectSubs.RowSel, 2))
        Node3.Tag = grdProjectSubs.TextMatrix(grdProjectSubs.RowSel, 1)

        SubNodeArray(SubNum, 0) =
grdProjectSubs.TextMatrix(grdProjectSubs.RowSel, 2)
        SubNodeArray(SubNum, 1) = "S" + Str$(SubNum)
        SubNodeArray(SubNum, 2) =
grdProjectWP.TextMatrix(grdProjectWP.RowSel, 1)
        SubNum = SubNum + 1

        datProjectWPSubs.AddNew
```

```
        datProjectWPSubs("PWSPID") = ProjectID + "-" +
grdProjectWP.TextMatrix(grdProjectWP.RowSel, 0) + "-" +
grdProjectSubs.TextMatrix(grdProjectSubs.RowSel, 1)
        datProjectWPSubs("ProjectID") = ProjectID
        datProjectWPSubs("ProjectName") = ProjectName
        datProjectWPSubs("WorkPackageID") =
grdProjectWP.TextMatrix(grdProjectWP.RowSel, 0)
        datProjectWPSubs("WorkPackage") =
grdProjectWP.TextMatrix(grdProjectWP.RowSel, 1)
        datProjectWPSubs("SubID") =
grdProjectSubs.TextMatrix(grdProjectSubs.RowSel, 1)
        datProjectWPSubs("SubName") =
grdProjectSubs.TextMatrix(grdProjectSubs.RowSel, 2)
        datProjectWPSubs("Price") = Trim$(msg)
        datProjectWPSubs.Update

    End Sub

    Private Sub grdProjectWP_Click()
        Dim Node2 As Node
        Dim i As Integer
        Dim msg As String
        Dim SQLQuery1 As String
        Dim Crit As String

        Crit = grdProjectWP.TextMatrix(grdProjectWP.RowSel, 0)
        SQLQuery1 = "Select * From Subcontractors Where WorkPackageID="
+ "'" + Crit + "'"
        Set datProjectSubs = dbABPC.OpenRecordset(SQLQuery1,
dbOpenDynaset)
        SetupProjectGridSubs
        FillProjectGridSubs

        If WPNum > 0 Then
            For i = 0 To WPNum - 1
                If tvwProjectWP.Nodes("WP" + Str$(i)).Text =
grdProjectWP.TextMatrix(grdProjectWP.RowSel, 1) Then
                    Exit Sub
                End If
            Next i
        End If

        Set Node2 = tvwProjectWP.Nodes.Add("R", tvwChild, "WP" +
Str$(WPNum), grdProjectWP.TextMatrix(grdProjectWP.RowSel, 1))
        Node2.Tag = grdProjectWP.TextMatrix(grdProjectWP.RowSel, 0)
        WPNodeArray(WPNum, 0) =
grdProjectWP.TextMatrix(grdProjectWP.RowSel, 1)
        WPNodeArray(WPNum, 1) = "WP" + Str$(WPNum)
        Node2.Expanded = True
        WPNum = WPNum + 1

    End Sub

    Private Sub SSTab1_Click(PreviousTab As Integer)

        Dim i As Integer
        Dim j As Integer
```

```
        Dim k As Integer
        Dim Pos1, Pos2 As Integer
        Dim SubName As String
        Dim SubPrice As String
        Dim nd1, nd2 As Node


        WorkPacks = 0

             Set nd1 = tvwProjectWP.Nodes("R").Child
             For i = 1 To tvwProjectWP.Nodes("R").Children
                    ProjectWorkPacks(i - 1, 0) = nd1.Tag
                    ProjectWorkPacks(i - 1, 1) = nd1.Text
                    WorkPacks = WorkPacks + 1

                    SubsInWorkPack(i - 1, 0) = nd1.Tag
                    SubsInWorkPack(i - 1, 1) = nd1.Children

                    Set nd2 = nd1.Child
                    For k = 1 To nd1.Children
                        Pos1 = InStr(nd2.Text, "$")
                        Pos2 = InStr(nd2.Text, ",")
                        SubName = Mid$(nd2.Text, Pos2 + 1)
                        SubPrice = Mid$(nd2.Text, Pos1 + 1, Pos2 - 1)

                        WorkPackSubPrice(j, 0) = nd2.Parent.Tag
'WorkPackageID
                        WorkPackSubPrice(j, 1) = nd2.Tag
'SubID
                        WorkPackSubPrice(j, 2) = SubName
'SubName
                        WorkPackSubPrice(j, 3) = SubPrice
'Price
                        j = j + 1

                        Set nd2 = nd2.Next
                    Next k
                    Set nd1 = nd1.Next
             Next i

        CreatePriceNodes
        CreateAlternativePrice
        CalPriceWeight

        a1 = CalAltNum()
        If a1 <> a2 Then
            setPerfWeight = True
        End If

        If SSTab1.Tab = 2 Or SSTab1.Tab = 3 Or SSTab1.Tab = 4 Or
SSTab1.Tab = 5 Or SSTab1.Tab = 6 Then
            a2 = CalAltNum()
            If setPerfWeight = True Then
               PopulatePerfWeights
               setPerfWeight = False
            End If
```

```
        SetTVWAlt

        SetupGrid grdCostModel
        FillGridCostModel

        DrawCostChart
        DrawCostWeightChart
        DrawPerfChart
        DrawCostPerfWeightChart

    ElseIf SSTab1.Tab = 7 Then
        DrawSensiChart

    ElseIf SSTab1.Tab = 8 Then
        DrawSensiChart2
    End If
End Sub

Private Sub TreeView1_BeforeLabelEdit(Cancel As Integer)

End Sub

Private Sub sldCostPerf_Scroll()
    sldCostPerf.Text = Format(sldCostPerf.Value / 100, "0.00")

    CostWt = sldCostPerf.Value / 100
    PerfWt = 1 - CostWt
    lblCost = Format(CostWt, "0.00")
    lblPerf = Format(PerfWt, "0.00")

    DrawCostPerfWeightChart
End Sub
```

APPENDIX C

COST AND ACTIVITY-BASED PERFORMANCE MEASURMENT DECISION MAKING SYSTEM VB CODE — PROJECT EVALUATION COMPONENT

```
     Option Explicit

     Dim dbABPC As Database
     Dim datProjectSubs As Recordset
     Dim datProjectData As Recordset
     Dim datProjectAct As Recordset
     Dim datProjectPerf As Recordset
     Dim datProjectActPerf As Recordset
     Dim datAddProjectActPerf As Recordset
     Dim datObject As Recordset
     Dim datAct As Recordset
     Dim datAct1 As Recordset
     Dim datPerf1 As Recordset

     Dim WP(0 To 100, 0 To 2) As String   'WP(i,0)=WorkPackageID,
WP(i,1)=WorkPackage
     Dim WPNum As Integer                 'Number of Work Package

     Private Sub SaveProjectActPerf()
         Dim i As Integer
         Dim j As Integer
         Dim k As Integer
         Dim Node1 As Node
         Dim Node2 As Node
         Dim Node3 As Node
         Dim Pos1 As Integer
         Dim Act As String
         Dim ActWeight As String
         Dim Pos2 As Integer
         Dim Perf As String
         Dim PerfWeight As String

         If datAddProjectActPerf.RecordCount <> 0 Then
             datAddProjectActPerf.MoveFirst
         End If

         Set Node1 = tvwProjectActPerf.Nodes("R").Child
         For i = 1 To WPNum
             Set Node2 = Node1.Child
             For j = 1 To Node1.Children

                 Set Node3 = Node2.Child
                 For k = 1 To Node2.Children

                     If datAddProjectActPerf.RecordCount <> 0 Then
                         datAddProjectActPerf.MoveFirst
```

```
                    End If
                    Do While datAddProjectActPerf.EOF = False
                        If datAddProjectActPerf("PWAPID") =
SelProjectID + "-" + WP(i - 1, 0) + "-" + Node2.Tag + "-" + Node3.Tag
Then
                            GoTo SetNode
                        End If
                        datAddProjectActPerf.MoveNext
                    Loop

                    datAddProjectActPerf.AddNew
                    datAddProjectActPerf("PWAPID") = SelProjectID + "-"
+ WP(i - 1, 0) + "-" + Node2.Tag + "-" + Node3.Tag
                    datAddProjectActPerf("ProjectID") = SelProjectID
                    datAddProjectActPerf("Project") = SelProject
                    datAddProjectActPerf("WorkPackageID") = WP(i - 1,
0)
                    datAddProjectActPerf("WorkPackage") = WP(i - 1, 1)

                    Pos1 = InStr(Node2.Text, ",")
                    ActWeight = Mid$(Node2.Text, 1, Pos1 - 1)
                    Act = Mid$(Node2.Text, Pos1 + 3)
                    datAddProjectActPerf("ActivityID") = Node2.Tag
                    datAddProjectActPerf("Activity") = Act
                    datAddProjectActPerf("ActivityWeight") = ActWeight

                    Pos2 = InStr(Node3.Text, ",")
                    PerfWeight = Mid$(Node3.Text, 1, Pos2 - 1)
                    Perf = Mid$(Node3.Text, Pos2 + 3)
                    datAddProjectActPerf("PerformanceID") = Node3.Tag
                    datAddProjectActPerf("Performance") = Perf
                    datAddProjectActPerf("APWeight") = PerfWeight
                    datAddProjectActPerf.Update

     SetNode:       Set Node3 = Node3.Next
                Next k
                Set Node2 = Node2.Next
            Next j
            Set Node1 = Node1.Next
        Next i

     End Sub



     Private Sub SetupTVWProjectData()
        Dim Node1 As Node
        Dim Node2 As Node
        Dim Node3 As Node
        Dim Node4 As Node
        Dim i As Integer
        Dim j As Integer
        Dim k As Integer
        Dim db1, db2 As Database
        Dim rs1, rs2 As Recordset
        Dim SQL1, SQL2 As String
```

```
        tvwProjectData.Nodes.Clear
        Set Node1 = tvwProjectData.Nodes.Add(, , "R", "Data Items in
Work Packages")
        Node1.Expanded = True

        For i = 1 To WPNum
            Set Node2 = tvwProjectData.Nodes.Add("R", tvwChild, "WP" +
WP(i - 1, 0), WP(i - 1, 1))
            Node2.Tag = WP(i - 1, 0)
            Node2.Expanded = True

            SQL1 = "Select Distinct SubID, Sub from ProjectData Where
ProjectID='" & SelProjectID & "' and WorkPackageID='" & WP(i - 1, 0) &
"'"
            Set db1 = OpenDatabase(App.Path & "\Database\ABPC.mdb")
            Set rs1 = db1.OpenRecordset(SQLQuery1, dbOpenDynaset)

            For j = 1 To rs1.RecordCount
                Set Node3 = tvwProjectData.Nodes.Add("WP" + WP(i - 1,
0), tvwChild, "S" + rs1("SubID"), rs1("Sub"))
                Node3.Tag = rs1("SubID")
                Node3.Expanded = True

                SQL2 = "Select Distinct DataItemID, DataItem, Units
from ProjectData Where ProjectID='" & SelProjectID & "' and
WorkPackageID='" & WP(i - 1, 0) & "' and SubID='" & rs1("SubID") & "'"
                Set db2 = OpenDatabase(App.Path & "\Database\ABPC.mdb")
                Set rs2 = db1.OpenRecordset(SQL2)
                For k = 1 To rs2.RecordCount
                    Set Node4 = tvwProjectData.Nodes.Add("S" +
rs1("SubID"), tvwChild, , rs2("DataItem"))
                    Node4.Tag = rs2("DataItemID")
                    rs2.MoveNext
                Next k

                rs1.MoveNext
            Next j

        Next i

    End Sub

    Private Sub SetupTVWProjectSubs()
        Dim ANode As Node
        Dim i, j As Integer
        Dim m As Integer
        Dim Subs(0 To 100, 0 To 2) As String
        Dim TP As String

        tvwProjectSubs.Refresh
        Set ANode = tvwProjectSubs.Nodes.Add(, , "R", "Subs/suppliers
in Work Packages")
        ANode.Expanded = True
        For i = 0 To WPNum - 1
            Set ANode = tvwProjectSubs.Nodes.Add("R", tvwChild, "WP" +
WP(i, 0), WP(i, 1))
```

```
            ANode.Expanded = True

            m = 1
            TP = grdProjectSubs.TextMatrix(1, 2)
            Subs(0, 0) = grdProjectSubs.TextMatrix(1, 2)
            Subs(0, 1) = grdProjectSubs.TextMatrix(1, 3)
            For j = 1 To grdProjectSubs.Rows - 1
                If grdProjectSubs.TextMatrix(j, 0) = WP(i, 0) And
grdProjectSubs.TextMatrix(j, 2) <> TP Then
                    Subs(m, 0) = grdProjectSubs.TextMatrix(j, 2)
                    Subs(m, 1) = grdProjectSubs.TextMatrix(j, 3)
                    Set ANode = tvwProjectSubs.Nodes.Add("WP" + WP(i,
0), tvwChild, "Sub" + Subs(m, 0), Subs(m, 1))
                    TP = grdProjectSubs.TextMatrix(j, 1)
                    m = m + 1
                End If
            Next j
        Next i
    End Sub

    Private Sub SetupTVWProjectAct()

        Dim Node1 As Node
        Dim Node2 As Node
        Dim Node3 As Node
        Dim i As Integer
        Dim j As Integer
        Dim db1 As Database
        Dim rs1 As Recordset
        Dim SQL1 As String


        tvwProjectAct.Nodes.Clear
        Set Node1 = tvwProjectAct.Nodes.Add(, , "R", "Activities in
Work Packages")
        Node1.Expanded = True

        For i = 0 To WPNum - 1
            Set Node2 = tvwProjectAct.Nodes.Add("R", tvwChild, "WP" +
WP(i, 0), WP(i, 1))
            Node2.Expanded = True

            SQL1 = "Select Distinct ActivityID, Activity,
ActivityWeight from ProjectActPerf where ProjectID='" & SelProjectID &
"' and WorkPackageID='" & WP(i, 0) & "'"
            Set db1 = OpenDatabase(App.Path & "\Database\ABPC.mdb")
            Set rs1 = db1.OpenRecordset(SQL1)

            For j = 0 To rs1.RecordCount - 1

                Set Node3 = tvwProjectAct.Nodes.Add("WP" + WP(i, 0),
tvwChild, "A" + rs1("ActivityID"), rs1("ActivityWeight") + ",  " +
rs1("Activity"))
                Node3.Tag = rs1("ActivityID")
                rs1.MoveNext
            Next j
        Next i
```

```
    End Sub

    Private Sub SetupTVWProjectActPerf()

        Dim i As Integer
        Dim j As Integer
        Dim k As Integer
        Dim Node1 As Node
        Dim Node2 As Node
        Dim Node3 As Node
        Dim Node4 As Node
        Dim db1 As Database
        Dim rs1 As Recordset
        Dim SQL1 As String
        Dim db2 As Database
        Dim rs2 As Recordset
        Dim SQL2 As String

        If datProjectActPerf.Recordset.RecordCount <> 0 Then
            datProjectActPerf.Recordset.MoveFirst
        End If

        Set Node1 = tvwProjectActPerf.Nodes.Add(, , "R", "Work
Packages")
        Node1.Expanded = True
        For i = 1 To WPNum

            Set Node2 = tvwProjectActPerf.Nodes.Add("R", tvwChild, "WP"
+ WP(i - 1, 0), WP(i - 1, 1))
            Node2.Expanded = True

            SQL1 = "Select Distinct ActivityID,Activity,ActivityWeight
from ProjectActPerf where ProjectID='" & SelProjectID & "' and
WorkPackageID='" & WP(i - 1, 0) & "' "
            Set db1 = OpenDatabase(App.Path & "\Database\ABPC.mdb")
            Set rs1 = db1.OpenRecordset(SQL1)

            For j = 0 To rs1.RecordCount - 1

                Set Node3 = tvwProjectActPerf.Nodes.Add("WP" + WP(i -
1, 0), tvwChild, "A" + rs1("ActivityID"), rs1("ActivityWeight") + ",  "
+ rs1("Activity"))
                Node3.Tag = rs1("ActivityID")
                Node3.Expanded = True

                SQL2 = "Select Distinct PerformanceID, Performance,
APWeight from ProjectActPerf where ProjectID='" & SelProjectID & "' and
WorkPackageID='" & WP(i - 1, 0) & "' and ActivityID='" &
rs1("ActivityID") & "'"
                Set db2 = OpenDatabase(App.Path & "\Database\ABPC.mdb")
                Set rs2 = db2.OpenRecordset(SQL2)

                For k = 0 To rs2.RecordCount - 1
                    Set Node4 = tvwProjectActPerf.Nodes.Add("A" +
rs1("ActivityID"), tvwChild, , rs2("APWeight") + ",  " +
rs2("Performance"))
                    Node4.Tag = rs2("PerformanceID")
```

```
                    rs2.MoveNext
                Next k

                rs1.MoveNext
            Next j

        Next i

    End Sub

    Private Sub SetupTVWProjectPerf()

        Dim Node1 As Node
        Dim Node2 As Node
        Dim Node3 As Node
        Dim i As Integer
        Dim j As Integer
        Dim db1 As Database
        Dim rs1 As Recordset
        Dim SQL1 As String
        Dim msg As String

        tvwProjectPerf.Nodes.Clear
        Set Node1 = tvwProjectPerf.Nodes.Add(, , "R", "Performance
Criteria in Work Packages")
        Node1.Expanded = True

        For i = 0 To WPNum - 1
            Set Node2 = tvwProjectPerf.Nodes.Add("R", tvwChild, "WP" +
WP(i, 0), WP(i, 1))
            Node2.Expanded = True

            SQL1 = "Select Distinct PerformanceID, Performance from
ProjectActPerf where ProjectID='" & SelProjectID & "' and
WorkPackageID='" & WP(i, 0) & "'"
            Set db1 = OpenDatabase(App.Path & "\Database\ABPC.mdb")
            Set rs1 = db1.OpenRecordset(SQL1)

            For j = 0 To rs1.RecordCount - 1
                Set Node3 = tvwProjectPerf.Nodes.Add("WP" + WP(i, 0),
tvwChild, "P" + rs1("PerformanceID"), rs1("Performance"))
                Node3.Tag = rs1("PerformanceID")
                rs1.MoveNext
            Next j
        Next i

    End Sub

    Private Sub IdentifyWP()
        Dim i As Integer
        Dim m As Integer
        Dim TP As String

        datProjectSubs.MoveFirst
        m = 1
        TP = datProjectSubs("WorkPackageID")
        WP(0, 0) = datProjectSubs("WorkPackageID")
```

```
        WP(0, 1) = datProjectSubs("WorkPackage")
        For i = 0 To datProjectSubs.RecordCount - 1
            If datProjectSubs("WorkPackageID") <> TP Then
                WP(m, 0) = datProjectSubs("WorkPackageID")
                WP(m, 1) = datProjectSubs("WorkPackage")
                TP = datProjectSubs("WorkPackageID")
                m = m + 1
            End If
            datProjectSubs.MoveNext
        Next i

        WPNum = m
    End Sub
    Private Sub SetupGridProjectSubs()

        Dim WideString1, WideString2, WideString3, WideString4,
WideString5 As String

        WideString1 = String$(10, "X")
        WideString2 = String$(25, "X")
        WideString3 = String$(10, "X")
        WideString4 = String$(25, "X")
        WideString5 = String$(15, "X")

        With grdProjectSubs
            .SelectionMode = flexSelectionByRow
            .Cols = 5
            .FixedCols = 0
            .ColWidth(0) = TextWidth(WideString1)
            .ColWidth(1) = TextWidth(WideString2)
            .ColWidth(2) = TextWidth(WideString3)
            .ColWidth(3) = TextWidth(WideString4)
            .ColWidth(4) = TextWidth(WideString5)
        End With

        grdProjectSubs.TextMatrix(0, 0) = "Work Package ID"
        grdProjectSubs.TextMatrix(0, 1) = "Work Package"
        grdProjectSubs.TextMatrix(0, 2) = "Sub/Supplier ID"
        grdProjectSubs.TextMatrix(0, 3) = "Sub/Supplier"
        grdProjectSubs.TextMatrix(0, 4) = "Price"
    End Sub

    Private Sub SetupGridProjectData()

        Dim WideString1, WideString2, WideString3 As String

        WideString1 = String$(10, "X")
        WideString2 = String$(25, "X")
        WideString3 = String$(10, "X")

        With grdData
            .SelectionMode = flexSelectionByRow
            .Cols = 3
            .FixedCols = 0
            .ColWidth(0) = TextWidth(WideString1)
            .ColWidth(1) = TextWidth(WideString2)
            .ColWidth(2) = TextWidth(WideString3)
```

```
    End With

    grdData.TextMatrix(0, 0) = "Data Item ID"
    grdData.TextMatrix(0, 1) = "Data Item"
    grdData.TextMatrix(0, 2) = "Unit"
End Sub

Private Sub SetupGridProjectAct(grdObject As Object)

    Dim WideString1, WideString2 As String

    WideString1 = String$(10, "X")
    WideString2 = String$(25, "X")


    With grdObject
        .SelectionMode = flexSelectionByRow
        .Cols = 2
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)

    End With

    grdObject.TextMatrix(0, 0) = "Activity ID"
    grdObject.TextMatrix(0, 1) = "Activity"

End Sub

Private Sub SetupGridProjectPerf(grdObject As Object)

    Dim WideString1, WideString2 As String

    WideString1 = String$(10, "X")
    WideString2 = String$(25, "X")

    With grdObject
        .SelectionMode = flexSelectionByRow
        .Cols = 2
        .FixedCols = 0
        .ColWidth(0) = TextWidth(WideString1)
        .ColWidth(1) = TextWidth(WideString2)

    End With

    grdObject.TextMatrix(0, 0) = "PerformanceID"
    grdObject.TextMatrix(0, 1) = "Description"

End Sub

Private Sub FillGridProjectData()
    Dim count As Integer
    Dim i As Integer

    grdData.Clear
    SetupGridProjectData
    count = 0
```

```
        If datDataItem.RecordCount <> 0 Then
            datDataItem.MoveFirst
            Do Until datDataItem.EOF = True
                count = count + 1
                datDataItem.MoveNext
            Loop

            grdData.Rows = count + 1
            datDataItem.MoveFirst
            For i = 1 To count
                grdData.TextMatrix(i, 0) =
datDataItem.Recordset("DataID")
                grdData.TextMatrix(i, 1) =
datDataItem.Recordset("Data")
                grdData.TextMatrix(i, 2) =
datDataItem.Recordset("Unit")
                datDataItem.MoveNext
            Next i
        End If

    End Sub

    Private Sub FillGridProjectAct(datObject As Object, grdObject As
Object)

        Dim i As Integer
        Dim Rows As Integer

        i = 1
        Rows = 0
        If datObject.RecordCount <> 0 Then
            Do Until datObject.EOF = True
                Rows = Rows + 1
                datObject.MoveNext
            Loop
            grdObject.Rows = Rows + 1
            datObject.MoveFirst
            Do Until datObject.EOF = True
                grdObject.TextMatrix(i, 0) =
datObject.Recordset("ActivityID")
                grdObject.TextMatrix(i, 1) =
datObject.Recordset("ActivityDescription")

                i = i + 1
                datObject.MoveNext
            Loop
        ElseIf datObject.RecordCount = 0 Then
            grdObject.Rows = 1
        End If

    End Sub

    Private Sub FillGridProjectPerf(datObject As Object, grdObject As
Object)

        Dim i As Integer
        Dim Rows As Integer
```

```
        i = 1
        Rows = 0
        If datObject.RecordCount <> 0 Then
            Do Until datObject.EOF = True
                Rows = Rows + 1
                datObject.MoveNext
            Loop
            grdObject.Rows = Rows + 1
            datObject.MoveFirst
            Do Until datObject.EOF = True
                grdObject.TextMatrix(i, 0) = datObject("PerformanceID")
                grdObject.TextMatrix(i, 1) = datObject("Description")

                i = i + 1
                datObject.MoveNext
            Loop
        ElseIf datObject.RecordCount = 0 Then
            grdObject.Rows = 1
        End If

    End Sub


    Private Sub FillGridProjectSubs()
        Dim NewRow As String
        Do Until datProjectSubs.EOF = True
            NewRow = datProjectSubs("WorkPackageID") & vbTab &
datProjectSubs("WorkPackage") & vbTab & datProjectSubs("SubID") & vbTab
& datProjectSubs("SubName") & vbTab & Format(datProjectSubs("Price"),
"$.00")
            grdProjectSubs.AddItem NewRow
            datProjectSubs.MoveNext
        Loop

    End Sub

    Private Sub datProjectSub_Validate(Action As Integer, Save As
Integer)

    End Sub

    Private Sub datProjectData1_Validate(Action As Integer, Save As
Integer)

    End Sub

    Private Sub Form_Load()
        Dim SQLQuery1 As String
        Dim SQLQuery2 As String
        Dim SQLQuery3 As String
        Dim SQLQuery4 As String
        Dim SQLQuery5 As String

        WPNum = 0
        Me.Height = 9630
        Me.Width = 11715
```

```
        lblProject.Caption = SelProject

        SQLQuery1 = "Select * From ProjectSubs Where ProjectID='" &
SelProjectID & "'"
        Set dbABPC = OpenDatabase(App.Path & "\Database\ABPC.mdb")
        Set datProjectSubs = dbABPC.OpenRecordset(SQLQuery1,
dbOpenDynaset)

        SetupGridProjectSubs
        FillGridProjectSubs
        IdentifyWP
        SetupTVWProjectSubs

        'Prepare data item tab by using same sub "SetupTVWProjectSubs
        SQLQuery2 = "Select * From ProjectData Where ProjectID='" &
SelProjectID & "'"
        Set datProjectData = dbABPC.OpenRecordset(SQLQuery2,
dbOpenDynaset)

        SetupGridProjectData
        SetupTVWProjectSubs

        'Prepare activity tab
        SQLQuery2 = "Select * From ProjectAct where ProjectID='" &
SelProjectID & "'"
        Set datProjectAct = dbABPC.OpenRecordset(SQLQuery2,
dbOpenDynaset)

        SetupTVWProjectAct
        SetupGridProjectAct grdAct

        'Prepare performance tab
        SQLQuery3 = "Select * From ProjectPerf where ProjectID='" &
SelProjectID & "'"
        Set datProjectPerf = dbABPC.OpenRecordset(SQLQuery3,
dbOpenDynaset)

        SetupTVWProjectPerf
        SetupGridProjectPerf grdPerf

        'Prepare activity-performance tab
        SQLQuery4 = "Select * From ProjectActPerf where ProjectID='" &
SelProjectID & "'"
        Set datProjectActPerf = dbABPC.OpenRecordset(SQLQuery4,
dbOpenDynaset)

        SQLQuery5 = "Select * From ProjectActPerf"
        Set datAddProjectActPerf = dbABPC.OpenRecordset(SQLQuery5,
dbOpenDynaset)

        SetupTVWProjectActPerf
        SetupGridProjectAct grdProjectAct
        SetupGridProjectPerf grdProjectPerf

    End Sub

    Private Sub grdAct_Click()
```

```
        Dim msg As String
        Dim str1 As String
        Dim Node3 As Node
        Dim i As Integer
        Dim ANode As Node
        Dim NodeKey As String


        str1 = "Please input weight for this " +
grdAct.TextMatrix(grdAct.RowSel, 1)
        msg = InputBox(str1, "Information", 0)

        'Check if there is the repeated node
        Set ANode = tvwProjectAct.Nodes("WP" +
lblActWPID.Caption).Child
        For i = 1 To tvwProjectAct.Nodes("WP" +
lblActWPID.Caption).Children

            If ANode.Key = "A" + grdAct.TextMatrix(grdAct.RowSel, 0)
Then
                msg = MsgBox(grdAct.TextMatrix(grdAct.RowSel, 1) + "
has already been in this work package. try again", vbOKOnly,
"Information")
                Exit Sub
            End If

            Set ANode = ANode.Next

        Next i

        Set Node3 = tvwProjectAct.Nodes.Add("WP" + lblActWPID.Caption,
tvwChild, "A" + grdAct.TextMatrix(grdAct.RowSel, 0), Format(Trim$(msg),
"0.00") + ",  " + grdAct.TextMatrix(grdAct.RowSel, 1))

        datProjectAct.AddNew
        datProjectAct("PWAID") = SelProjectID + "-" + lblActWPID + "-"
+ grdAct.TextMatrix(grdAct.RowSel, 0)
        datProjectAct("ProjectID") = SelProjectID
        datProjectAct("Project") = SelProject
        datProjectAct("WorkPackageID") = lblActWPID.Caption
        datProjectAct("WorkPackage") = lblActWP.Caption
        datProjectAct("ActivityID") = grdAct.TextMatrix(grdAct.RowSel,
0)
        datProjectAct("Activity") = grdAct.TextMatrix(grdAct.RowSel, 1)
        datProjectAct("Weight") = Trim$(msg)
        datProjectAct.Update
    End Sub

    Private Sub grdData_Click()
        Dim msg As String
        Dim str1 As String
        Dim Node3 As Node
        Dim i As Integer
        Dim ANode As Node
        Dim NodeKey As String

        'Check if there is the repeated node
```

```
        Set ANode = tvwProjectData.Nodes("WP" +
lblDataWPID.Caption).Child
        For i = 1 To tvwProjectData.Nodes("WP" +
lblDataWPID.Caption).Children

            If ANode.Key = "D" + grdData.TextMatrix(grdData.RowSel, 0)
Then
                msg = MsgBox(grdData.TextMatrix(grdData.RowSel, 1) + "
has already been in this work package. try again", vbOKOnly,
"Information")
                Exit Sub
            End If

            Set ANode = ANode.Next

        Next i

        Set Node3 = tvwProjectData.Nodes.Add("WP" +
lblDataWPID.Caption, tvwChild, "D" + grdData.TextMatrix(grdData.RowSel,
0), grdData.TextMatrix(grdData.RowSel, 1))
        Node3.Tag = grdData.TextMatrix(grdData.RowSel, 0)

        datProjectPerf.AddNew
        datProjectPerf("PWPID") = SelProjectID + "-" + lblPerfWPID + "-
" + grdPerf.TextMatrix(grdPerf.RowSel, 0)
        datProjectPerf("ProjectID") = SelProjectID
        datProjectPerf("Project") = SelProject
        datProjectPerf("WorkPackageID") = lblPerfWPID.Caption
        datProjectPerf("WorkPackage") = lblPerfWP.Caption
        datProjectPerf("PerformanceID") =
grdPerf.TextMatrix(grdPerf.RowSel, 0)
        datProjectPerf("Performance") =
grdPerf.TextMatrix(grdPerf.RowSel, 1)
        datProjectPerf.Update
    End Sub

    Private Sub grdPerf_Click()
        Dim msg As String
        Dim str1 As String
        Dim Node3 As Node
        Dim i As Integer
        Dim ANode As Node
        Dim NodeKey As String

        'Check if there is the repeated node
        Set ANode = tvwProjectPerf.Nodes("WP" +
lblPerfWPID.Caption).Child
        For i = 1 To tvwProjectPerf.Nodes("WP" +
lblPerfWPID.Caption).Children

            If ANode.Key = "A" + grdPerf.TextMatrix(grdPerf.RowSel, 0)
Then
                msg = MsgBox(grdPerf.TextMatrix(grdAct.RowSel, 1) + "
has already been in this work package. try again", vbOKOnly,
"Information")
                Exit Sub
            End If
```

```
            Set ANode = ANode.Next

        Next i

        Set Node3 = tvwProjectPerf.Nodes.Add("WP" +
lblPerfWPID.Caption, tvwChild, "A" + grdPerf.TextMatrix(grdPerf.RowSel,
0), grdPerf.TextMatrix(grdPerf.RowSel, 1))

        datProjectPerf.AddNew
        datProjectPerf("PWPID") = SelProjectID + "-" + lblPerfWPID + "-
" + grdPerf.TextMatrix(grdPerf.RowSel, 0)
        datProjectPerf("ProjectID") = SelProjectID
        datProjectPerf("Project") = SelProject
        datProjectPerf("WorkPackageID") = lblPerfWPID.Caption
        datProjectPerf("WorkPackage") = lblPerfWP.Caption
        datProjectPerf("PerformanceID") =
grdPerf.TextMatrix(grdPerf.RowSel, 0)
        datProjectPerf("Performance") =
grdPerf.TextMatrix(grdPerf.RowSel, 1)
        datProjectPerf.Update
    End Sub

    Private Sub grdProjectAct_Click()
        Dim msg As String
        Dim str1 As String
        Dim Node3 As Node
        Dim i As Integer
        Dim ANode As Node
        Dim NodeKey As String

        'Check if there is the repeated node
        Set ANode = tvwProjectActPerf.Nodes("WP" +
lblWPID1.Caption).Child
        For i = 1 To tvwProjectActPerf.Nodes("WP" +
lblWPID1.Caption).Children

            If ANode.Key = "A" +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 0) Then
                msg =
MsgBox(grdProjectAct.TextMatrix(grdProjectAct.RowSel, 1) + " has
already been in this work package. try again", vbOKOnly, "Information")
                Exit Sub
            End If

            Set ANode = ANode.Next

        Next i

        str1 = "Please input weight for this " +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 1)
        msg = InputBox(str1, "Information", 0)

        Set Node3 = tvwProjectActPerf.Nodes.Add("WP" +
lblWPID1.Caption, tvwChild, "A" +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 0), Format(Trim$(msg),
"0.00") + ",  " + grdProjectAct.TextMatrix(grdProjectAct.RowSel, 1))
```

```
        Node3.Tag = grdProjectAct.TextMatrix(grdProjectAct.RowSel, 0)
        Node3.Expanded = True
    End Sub

    Private Sub grdProjectData_Click()

    End Sub

    Private Sub grdProjectPerf_Click()
        Dim msg As String
        Dim str1 As String
        Dim Node3 As Node
        Dim i As Integer
        Dim ANode As Node
        Dim NodeKey As String
        Dim NKey1 As String
        Dim Nkey2 As String

        'Check if the activity node existed
        NKey1 = "WP" + lblWPID1.Caption
        If tvwProjectActPerf.Nodes(NKey1).Children = 0 Then
            Exit Sub
        End If

        'Check if there is the repeated node
        Set ANode = tvwProjectActPerf.Nodes("A" +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 0)).Child
        For i = 1 To tvwProjectActPerf.Nodes("A" +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 0)).Children

            Nkey2 = "A" +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 0) + "P" +
grdProjectPerf.TextMatrix(grdProjectPerf.RowSel, 0)
            If ANode.Key = Nkey2 Then
                msg =
MsgBox(grdProjectPerf.TextMatrix(grdProjectPerf.RowSel, 1) + " has
already been under activity " +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 1) + " , try again",
vbOKOnly, "Information")
                Exit Sub
            End If

            Set ANode = ANode.Next

        Next i

        str1 = "Please input weight between " +
grdProjectPerf.TextMatrix(grdProjectPerf.RowSel, 1) + " and " +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 1)
        msg = InputBox(str1, "Information", 0)

        NodeKey = "A" + grdProjectAct.TextMatrix(grdProjectAct.RowSel,
0) + "P" + grdProjectPerf.TextMatrix(grdProjectPerf.RowSel, 0)
        Set Node3 = tvwProjectActPerf.Nodes.Add("A" +
grdProjectAct.TextMatrix(grdProjectAct.RowSel, 0), tvwChild, NodeKey,
Format(msg, "0.00") + ",  " +
grdProjectPerf.TextMatrix(grdProjectPerf.RowSel, 1))
```

```
        Node3.Tag = grdProjectPerf.TextMatrix(grdProjectPerf.RowSel, 0)
        Node3.Expanded = True
    End Sub

    Private Sub SSTab1_Click(PreviousTab As Integer)
        If SSTab1.Tab <> 2 Then
            SaveProjectActPerf
            SetupTVWProjectAct
            SetupTVWProjectPerf
        End If
    End Sub

    Private Sub tvwProjectAct_NodeClick(ByVal Node As MSComctlLib.Node)
        Dim SQLQuery1 As String

        If Mid$(Node.Key, 1, 2) = "WP" Then
            lblActWPID.Caption = Mid$(Node.Key, 3)
            lblActWP.Caption = Node.Text
            SQLQuery1 = "Select * From Activities Where
WorkPackageID='" & Mid$(Node.Key, 3) & "'"
            datAct.DatabaseName = App.Path & "\Database\ABPC.mdb"
            Set datAct = dbABPC.OpenRecordset(SQLQuery1, dbOpenDynaset)

            FillGridProjectAct datAct, grdAct
        End If
    End Sub

    Private Sub tvwProjectActPerf_NodeClick(ByVal Node As
MSComctlLib.Node)
        Dim SQLQuery1 As String
        Dim SQLQuery2 As String

        If Mid$(Node.Key, 1, 2) = "WP" Then
            lblWPID1.Caption = Mid$(Node.Key, 3)
            lblWPID2.Caption = Mid$(Node.Key, 3)
            lblWP1.Caption = Node.Text
            lblWP2.Caption = Node.Text

            SQLQuery1 = "Select * From Activities Where
WorkPackageID='" & Mid$(Node.Key, 3) & "'"
            Set datAct1 = dbABPC.OpenRecordset(SQLQuery1,
dbOpenDynaset)

            SQLQuery2 = "Select * From Performances Where
WorkPackageID='" & Mid$(Node.Key, 3) & "'"
            Set datPerf1 = dbABPC.OpenRecordset(SQLQuery2,
dbOpenDynaset)

            FillGridProjectAct datAct1, grdProjectAct
            FillGridProjectPerf datPerf1, grdProjectPerf

        End If
    End Sub

    Private Sub tvwProjectData_NodeClick(ByVal Node As
MSComctlLib.Node)
        Dim SQLQuery1 As String
```

```
        If Mid$(Node.Key, 1, 2) = "WP" Then
            lblDataWPID.Caption = Mid$(Node.Key, 3)
            lblDataWP.Caption = Node.Text
            SQLQuery1 = "Select * From DataItem Where WorkPackageID='"
& Mid$(Node.Key, 3) & "'"
            Set dbABPC = OpenDatabase(App.Path & "\Database\ABPC.mdb")
            Set datDataItem = dbABPC.OpenRecordset(SQLQuery1,
dbOpenDynaset)

            FillGridProjectData
        End If
    End Sub

    Private Sub tvwProjectPerf_NodeClick(ByVal Node As
MSComctlLib.Node)
        Dim SQLQuery1 As String

        If Mid$(Node.Key, 1, 2) = "WP" Then
            lblPerfWPID.Caption = Mid$(Node.Key, 3)
            lblPerfWP.Caption = Node.Text
            SQLQuery1 = "Select * From Performances Where
WorkPackageID='" & Mid$(Node.Key, 3) & "'"
            Set dbABPC = OpenDatabase(App.Path & "\Database\ABPC.mdb")
            Set datPerf = dbABPC.OpenRecordset(SQLQuery1,
dbOpenDynaset)

            FillGridProjectPerf datPerf, grdPerf
        End If
    End Sub
```

APPENDIX D
COST AND ACTIVITY-BASED PERFORMANCE MEASURMENT DECISION MAKING SYSTEM VB CODE — FUNCTION MODULE

```
Option Explicit
Public ProjectID, ProjectName, Location, Manager As String
'Store new projectID, name, manager, and location
Public SelProjectID, SelProject As String
'Store selected project ID and name
Public Const MaxWorkPack = 20
Public ProjectWorkPacks(0 To MaxWorkPack, 0 To 1) As String
'Store WorkPackageID and WorkPackage
Public WorkPacks As Integer 'Store number of WorkPackages
Public WorkPackSubPrice(0 To 200, 0 To 3) As String
'Store workPackID,subId, subs and their price in each work package
Public SubsInWorkPack(0 To MaxWorkPack, 0 To 1)
'Store WorkPackID and sub numbers in that WorkPack
Public NodePrice(0 To 10000, 0 To 1) As String
'Store SubId and sub prices in each node
Public Nodes As Integer
'Store number of nodes
Public WorkPackLevelStartNode(0 To MaxWorkPack, 0 To 1)
'Store WorkPackID and Level Start Node Number in the treee
Public Alternative(0 To 2000, 0 To MaxWorkPack) As String
'Store Alternative Node Path
Public AlternativePrice(0 To 2000, 0 To MaxWorkPack) As Currency
'Store Alternative Node Price
Public PriceDif(0 To 2000, 0 To 2000) As Currency
'Store price differences in an array
Public PriceComp(0 To 2000, 0 To 2000) As Double
'Store Comparsion Results in this array
Public AltTotalPrice(0 To 2000) As Currency
'Store total price for each alternative
Public CostWt, PerfWt As Double
'Store cost weight and performance weight
Public PerfWeights() As Double
'store randonized performance weights
Public setPerfWeight As Boolean

Public Function CalAltNum()
    Dim a As Integer
    Dim i As Integer

    a = 1
    For i = 1 To WorkPacks
        a = a * SubsInWorkPack(i - 1, 1)
    Next i
End Function
Public Sub PopulatePerfWeights()
```

```
    Dim TotlRnd As Double
    Dim i As Integer
    Dim Alternatives As Integer

    Alternatives = 1
    For i = 1 To WorkPacks
        Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
    Next i

    ReDim PerfWeights(0 To Alternatives) As Double

    TotlRnd = 0
    For i = 0 To Alternatives - 1
        PerfWeights(i) = Rnd
        TotlRnd = TotlRnd + PerfWeights(i)
    Next i
    For i = 0 To Alternatives - 1
        PerfWeights(i) = PerfWeights(i) / TotlRnd
    Next i
End Sub

Public Sub CalPriceWeight()
    Dim Alternatives As Integer
    'Store total number of alternatives

    Dim i, j, k As Integer
    Dim Min, Max As Currency
    Dim sum As Double
    'Store the sum of comparisons
    Dim ULPrice, LLPrice As Currency
    'Store Upper Limit Price Scale and Lower Limit Price Scale
    Dim UnitRate As Currency
    'Store Unite Scale in scale table
    Dim AHPScales(-9 To 9, 0 To 1) As Double
    'Store the scale and cost difference in this array

    'Sum up total number of alternatives
    Alternatives = 1
    For i = 1 To WorkPacks
        Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
    Next i

    'Store total price for each alternative
    For i = 0 To Alternatives - 1
        AltTotalPrice(i) = 0
    Next i

    For i = 0 To Alternatives - 1
        For j = 1 To WorkPacks
            AltTotalPrice(i) = AltTotalPrice(i) +
                AlternativePrice(i, j)
        Next j
    Next i

    'Store price difference in PriceDif() and get Max and Min
    Max = 0
```

```
Min = 0
For i = 0 To Alternatives - 1
    For j = i To Alternatives - 1
        PriceDif(i, j) = AltTotalPrice(j) - AltTotalPrice(i)
        If PriceDif(i, j) > Max Then
            Max = PriceDif(i, j)
        ElseIf PriceDif(i, j) < Min Then
            Min = PriceDif(i, j)
        End If
    Next j
Next i

'Find the highest positive cost difference and the lowest
 negative cost difference
If Abs(Max) > Abs(Min) Then
    Max = Abs(Max)
Else
    Max = Abs(Min)
End If

ULPrice = Max + 1
LLPrice = -Max - 1
UnitRate = ULPrice / 9

'Populate Scales Array
AHPScales(0, 0) = 1
AHPScales(0, 1) = 0
AHPScales(-1, 0) = 1
AHPScales(-1, 1) = -UnitRate
AHPScales(1, 0) = 1
AHPScales(1, 1) = UnitRate
For i = 2 To 9
    AHPScales(i, 0) = i
    AHPScales(i, 1) = i * UnitRate
Next i
For i = -9 To -2
    AHPScales(i, 0) = -1 / i
    AHPScales(i, 1) = i * UnitRate
Next i

'Populate comparison table
For i = 0 To Alternatives - 1
    For j = i To Alternatives - 1
        For k = -9 To 8
            If PriceDif(i, j) = 0 Then
                PriceComp(i, j) = 1
            End If

            If k <= 0 And PriceDif(i, j) <= 0 Then
                If PriceDif(i, j) > AHPScales(k, 1) And
                  PriceDif(i, j) <= AHPScales(k + 1, 1) Then
                    PriceComp(i, j) = AHPScales(k, 0)
                    PriceComp(j, i) = 1 / AHPScales(k, 0)
                End If
            ElseIf k >= 0 And PriceDif(i, j) >= 0 Then
                If PriceDif(i, j) >= AHPScales(k, 1) And
                  PriceDif(i, j) < AHPScales(k + 1, 1) Then
```

```
                        PriceComp(i, j) = AHPScales(k + 1, 0)
                        PriceComp(j, i) = 1 / AHPScales(k + 1, 0)
                    End If
                End If

            Next k
        Next j
    Next i

    'Normalize the comparison table
    For i = 0 To Alternatives - 1
        sum = 0
        For j = 0 To Alternatives - 1
            sum = sum + PriceComp(j, i)
        Next j
        PriceComp(Alternatives, i) = sum
        For j = 0 To Alternatives - 1
            PriceComp(j, i) = PriceComp(j, i) /
               PriceComp(Alternatives, i)
        Next j
    Next i

    'Caculate the weight for each alternative
    For i = 0 To Alternatives - 1
        sum = 0
        For j = 0 To Alternatives - 1
            sum = sum + PriceComp(i, j)
        Next j
        PriceComp(i, Alternatives) = sum / Alternatives
    Next i

    'test
    sum = 0
    For i = 0 To Alternatives - 1
        sum = sum + PriceComp(i, Alternatives)
    Next i
End Sub

Public Sub CreateAlternativePrice()
    Dim i As Integer
    Dim j, m As Integer
    Dim k As Integer
    Dim T As Double
    Dim mi, mr As Integer
    'Store resultent of node order calculation
    Dim MiddleNode As Integer
    Dim StartNode As Integer
    'Store first alternative node
    Dim EndNode As Integer
    'Store last alternative node
    Dim SSN, SN, EN As Integer
    'Store first node and end node on any level of tree
    Dim NodeOnLastLevel As Integer
    'Store Nodes on the last level(Higher level or left level)
    Dim Alternatives As Integer
    'Store total number of alternatives
    Dim NodeOrder As Integer
```

```
'Calculated the order of the node on the higher level

'Calculate first alternative node
StartNode = 0
NodeOnLastLevel = 1
For i = 1 To WorkPacks - 1
    StartNode = StartNode + NodeOnLastLevel *
         SubsInWorkPack(i - 1, 1)
    NodeOnLastLevel = NodeOnLastLevel *
         SubsInWorkPack(i - 1, 1)
Next i
StartNode = StartNode + 1

'Calculate last alternative node
EndNode = 0
NodeOnLastLevel = 1
For i = 1 To WorkPacks
    EndNode = EndNode + NodeOnLastLevel *
         SubsInWorkPack(i - 1, 1)
    NodeOnLastLevel = NodeOnLastLevel *
         SubsInWorkPack(i - 1, 1)
Next i

'Calculate the total alternative numbers
Alternatives = 1
For i = 1 To WorkPacks
    Alternatives = Alternatives * SubsInWorkPack(i - 1, 1)
Next i

'Store alternative path and price on alternative level
For i = StartNode To EndNode
    Alternative(i - StartNode, WorkPacks) = NodePrice(i, 0)
    AlternativePrice(i - StartNode, WorkPacks) =
        Val(NodePrice(i, 1))
Next i

'Store alternative path and price on other levels
For i = StartNode To EndNode
    MiddleNode = i
    'For mi = 1 To 3
    For m = 1 To WorkPacks - 1
        j = WorkPacks - m
        'Calculate the start node on higher level of the tree
        SN = 0
        NodeOnLastLevel = 1
        For k = 1 To j - 1
            SN = SN + NodeOnLastLevel *
                SubsInWorkPack(k - 1, 1)
            NodeOnLastLevel = NodeOnLastLevel *
                SubsInWorkPack(k - 1, 1)
        Next k
        SN = SN + 1

        'Calculate the start node on lower level of the tree
        SSN = 0
        NodeOnLastLevel = 1
        For k = 1 To j
```

```
                SSN = SSN + NodeOnLastLevel * SubsInWorkPack(k - 1, 1)
                    NodeOnLastLevel = NodeOnLastLevel *
                         SubsInWorkPack(k - 1, 1)
                Next k
                SSN = SSN + 1

                'Calculate the order of node on any level of the tree
                T = Val(SubsInWorkPack(j, 1))
                mi = (MiddleNode - SSN + 1) \ T
                mr = (MiddleNode - SSN + 1) Mod T
                If mr <> 0 Then
                    NodeOrder = mi + 1
                Else
                    NodeOrder = mi
                End If

                Alternative(i - StartNode, j) = NodePrice(SN +
NodeOrder - 1, 0)
                AlternativePrice(i - StartNode, j) = Val(NodePrice(SN +
NodeOrder - 1, 1))
                MiddleNode = NodeOrder + SN - 1
            Next m
        Next i

    End Sub

    Public Sub CreatePriceNodes()
        Dim i As Integer
        'Controller of tree levels
        Dim j As Integer
        'Controller of nodes on a tree level
        Dim k As Integer
        'Controller of WorkPackSubPrice Array
        Dim NodeOnLevel As Integer
        'Store the number of nodes on this level
        Dim NumOfWorkPackSub As Integer
        'Store Number of items in WorkPackSubPrice array
        Dim LevelStartNode As Integer
        'Store the start node number on each level
        Dim LevelEndNode As Integer
        'Store the end node number on each level

        NumOfWorkPackSub = 0
        For i = 0 To WorkPacks
            NumOfWorkPackSub = NumOfWorkPackSub + SubsInWorkPack(i, 1)
        Next i

        NodePrice(0, 0) = 0
        NodePrice(0, 1) = 0
        Nodes = 1
        NodeOnLevel = 1
        LevelStartNode = 1

        'Start Tree level through workpackage controller
        For i = 0 To WorkPacks - 1
            LevelStartNode = Nodes
            NodeOnLevel = NodeOnLevel * SubsInWorkPack(i, 1)
```

```
        LevelEndNode = LevelStartNode + NodeOnLevel - 1
        WorkPackLevelStartNode(i, 0) = SubsInWorkPack(i, 0)
        WorkPackLevelStartNode(i, 1) = LevelStartNode

        'Store Node information such as subId and sub price
        j = LevelStartNode
        Do While j <= LevelEndNode
            For k = 0 To NumOfWorkPackSub - 1
                If WorkPackSubPrice(k, 0) = SubsInWorkPack(i, 0)
                     Then
        'Check if WorkPackId in WorkPackSubPrice = WorkPackId in
         SubsInWorkPack
                    NodePrice(Nodes, 0) = WorkPackSubPrice(k, 1)
        'Store Sub ID
                    NodePrice(Nodes, 1) = WorkPackSubPrice(k, 3)
        'Store Sub price
                    Nodes = Nodes + 1
                    j = j + 1
                End If
            Next k
        Loop

    Next i

End Sub
```

LIST OF REFERENCES

Alfeld, L. E. (1988). Construction Productivity, On-site Measurement and Management, McGraw-Hill, New York.

American Institute of Steel Construction (1983), Detailing for Steel Construction, Chicago, III, Illinois.

Arbel, A. and Seidmann, A. (1984), Selecting a Microcomputer for Process Control and Data Acquisition, IIE Transactions, Vol. 16, No. 1 (March 1984), pp. 73-80

Back, W. E. Maxwell, D. A. and Isidore, L. J. (2000), Activity-Based Costing as a Tool for Process Improvement Evaluations, Journal of Management in Engineering, Vol. 16, No. 2, pp. 48

Beamon, B. M. (1996), Performance Measures in Supply Chain Management, Proceedings of 1996 Conference on Agile and Intelligent Manufacturing Systems, Rensselaer Polytechnic Institute, Troy, New York

Beamon, B. M. (1998), Supply Chain Design and Analysis: Models and Methods, International Journal of Production Economics, Vol. 55 (3), pp. 281-294

Beamon, B. M. (1999), Measuring Supply Chain Performance, International Journal of Operation & Production Management, Vol. 19 (3), pp. 275-292

Bharara, A., and Lee, C. Y. (1996), Implementation of an Activity-Based Costing System in a Small Manufacturing Company, International Journal of Production Research, Vol. 34, No. 4, pp. 1109-1130

Bhutta, K. S. and Huq, F. (2002), Supplier Selection Problem: A Comparison Of The Total Cost of Ownership And Analytic Hierarchy Process Approaches, Supply Chain Management: An International Journal, Vol. 7(3), pp.126-135

Carman, R. and Conrad, S. (2000), Key Performance Indicators: Putting the Customer First, Supply Chain Management Review, Nov/Dec 2000, pp. 90-95

Carr, L. P. and Christopher D. I. (1992), Measuring the Cost of Ownership, Journal of Cost Management, Fall 1992, Vol. 6, No 3, pp. 42

Chen, P. P-S. (1976), The Entity-Relationship Model – Toward a Unified View of Data. ACM Transactions on Database System, Vol.1 (March): pp. 9-36

Chopra, S., and Meindl, P.(2001) Supply Chain Management: Strategy, Planning and Operation. Prentice-Hall, Inc., Upper Saddle River, New Jersey

Christopher, M. (1994), Logistics and Supply Chain Management, Richard D. Irwin, Inc., Financial Times, New York.

Cooper, S.E., and Chen, A.C. (1985). Designing Steel Structures: Methods and Cases, Pentice-Hall, London.

Cooper, R. (1987), The Two-Stage Procedure in Cost Accounting — Part One, Journal of Cost Management, Vol. 1, No. 2, pp. 43-51

Cooper, R. (1987), Two-Stage Procedure in Cost Accounting — Part Two, Journal of Cost Management, Vol. 1, No. 3, pp. 39-45

Cooper, R. (1988), The Rise of Activity-Based Costing—Part One: What Is an Activity-Based Cost System? Journal of Cost Management, Summer 1988, pp. 45-54.

Cooper, R. (1988), The Rise of Activity-Based Costing — Part Two: When Do I Need an Activity-Based Cost System, Journal of Cost Management for Manufacturing Industry, Vol. 2, No. 3, pp. 41-48.

Cooper, R. (1989), The Rise of Activity-Based Costing—Part Three: How Many Cost Drivers Do You Need, and How Do You Select Them? Journal of Cost Management, Winter 1989, pp. 34-46.

Cox, R. F., Issa, R.R. A., and Ahrens, D. (2003), Management's Perception of Key Performance Indicators for Construction, Journal of Construction Engineering and Management, March/April 2003.

Eggen, A. P., and Sandaker, B. N. (1995). Steel, Structure and Architecture, Watson-Guptill Publications, New York.

Ellram, L. M. (1993), Total Cost of Ownership: Elements and Implementation, International Journal of Purchasing of Material Management, Vol.29 (2), PP. 3-11

Ellram, L. M. (1994), Total Cost Modeling in Purchasing, Center for Advanced Purchasing Studies, Tempe, Arizona.

Ellram, L. M. and Siferd, S. P. (1993), Purchasing: The Cornerstone Of The Total Cost Of Ownership Concept. Journal of Business Logistics, Vol. 14, No. 1, 1993

Engel. I. (1988). Structural Steel in Architecture and Building Technology, Prentice Hall, Upper Saddle River, New Jersey.

Fellows, R., and Liu, A.(1997). Research Methods for Construction, Blackwell Science, Malden, Massachuset.

Fuller, J. B., O' Conor, J. and Rawlinson, R. (1993), Tailored Logistics: The Next Advantage, Harvard Business Review, May-June 1993, pp. 87-97.

Halpin, D. W., Escalona, A. L. and Szmurlo, P. M. (1987), Work Packaging For Project Control, Construction Industry Institute, Aug. 1987

Harker, P.T. and Vargas, L.G. (1987), The Theory of Ratio Scale Estimation: Satty's Analytic Hierarchy Process. Management Science, 33(11): pp. 1383-1403

Horngren, C.Y., Foster, G., and Datar, S.M. (1999). Cost Accounting, tenth edition, Prentice Hall, Upper Saddle River, New Jersey

Hugos, M. (2003), Essentials of Supply Chain Management, John Wiley & Sons, Inc. Hoboken, New Jersey

Johnson, J. B. and Randolph, S. (1995), Brief: Making Alliances Work – Using A Computer-based Management System To Integrate The Supply Chain, Journal of Petroleum Technology 47(6) (1995) pp. 512-513

Kaplan, R. S. and Cooper, R. (1997), Cost and Effect: Using Integrated Cost Systems to Drive Profitability and Performance, Harvard Business School Press, Boston, Massachusetts.

Kumaraswamy, M., Palaneeswaran, E., and Humphreys, P. (2000), Selection Matters - in Construction Supply Chain Optimization, International Journal of Physical Distribution & Logistics Management, Vol. 30, No. 7/8, pp. 661-680.

Kurt Salmon Associates Inc. (1993), Efficient Consumer Response: Enhancing Customer Value in the Grocery Industry, Kurt Salmon Associates Inc., New York.

LaLonde, B. J. and Zinszer, P. H. (1976), Customer Service: Meaning & Measurement (Chicago III.; National Council of Physical Distribution Management, 1976), pp. 281

Liberatore, M. J., (1987) An Extension of The Analytic Hierarchy Process for Industrial R& D Project Selection and Resource Allocation. IEEE Transactions on Engineering Management. Vol. EM-34, No. 1 (Feb. 1987), pp. 12-18;

Liberatore, M. J.,(1988) A Decision Support System Linking Research and Development Project Selection with Business Strategy, Project Management Journal, Vol. 19, No. 5 (Nov. 1988), pp. 14-21

Liberatore, M. J., and Nydick, R. L.,(1990) An Analytic Hierarchy Approach for Evaluating Product Formulation, Computer Aided Formulation: A Manual for Implementation, Alan H. Bohl, ed. (VCH Publishing Company, 1990) pp. 179-194

Marcus, S.H. (1981) Basics of Structural Steel Design, Second Edition, Reston Publishing Company Inc., Reston, Virginia.

Monczka, R. M. and Trecha, S. J. (1988), Cost-Based Supplier Performance Evaluation, Journal of Purchasing of Material Management, Vol. 24, No.1, pp. 2-7

Needy, K. L. and Bidanda, B. (2000), A Model to Develop, Assess, and Validate and Activity-Based Costing System for Small Manufacturers, Engineering Management Journal, Vol. 12, No. 1, pp. 31

Needy, K. L., Nachtmann, H., Roztocki, N., Warner, R. C., and Bidanda, B. (2003), Implementing Activity-Based Costing Systems in Small Manufacturing Firms: A Field Study, Engineering Management Journal, Vol. 15, No. 1, pp. 3

Nicoll, A. D. (1994), Integrating Logistics Strategies, Annual International Conference Proceedings, American Production and Inventory Control Society, 1994, pp. 590-594.

Nobbs, H. (1993), Future Role Of Construction Specialists. The Business Round Table, London.

Nydick, R.L. and Hill, R. P.(1992), Using the Analytic Hierarchy Process to Structure the Supplier Selection Procedure, International journal of Purchasing and Material Management, Vol. 28,No. 2, pp.31-36

Partovi, F. Y. (1991), An Analytic Hierarchy Approach to Activity-Based Costing, International Journal of Production Economics, Vol. 22 (2), pp.151-161

Partovi, F. Y., Burton, J. and Banarjee, A. (1989), Application of Analytical Hierarchy Process in Operations Management, International Journal of Operations and Production Management, Vol.10, No. 3 (1989), pp. 5-19

Pohlen, T. L. and Bernard J. L. (1994), Implementing Activity-Based Costing (ABC) in Logistics, Journal of Business Logistics, Vol. 15, No 2, 1994, pp. 1-24.

Prueitt, G.C. (2000), Case Study: US Army Utility Helicopter Fleet Modernization Analysis, The Engineering Economist, Vol. 45 No. 3, PP. 271-289

Raffish, N. and Turney, P. B. B. (1991), Glossary of Activity-based Management, Journal of Cost Management, Fall 1991, pp. 53-63

Render, B. and Stair, R. M. (2000), Quantitative Analysis Management, 7th ed., Prentice-Hall, Englewood Cliffs, New Jersey.

Saaty, T. J. (1980), The Analytic Hierarchy Process, McGraw-Hill, New York.

Shulitz, H.C., Sobek, W., and Habermann, K. J. (2000). Steel Construction Manual, Publishers for Architecture, Basel, Switzerland.

Slack, N. (1983), Flexibility As A Manufacturing Objective, International Journal of Operations & Production Management, Vol. 3, No. 3, pp. 4-13

Sun S. (2001), Base Closure: An Application Of The Analytic Hierarchy Process, INFOR, Vol. 39 No.1, pp. 17-31

Thomas, R. H., and Mathews, C. T. (1986). An Analysis of the Methods for Measuring Construction Productivity, Construction Industry Institute, Pennsylvania State Univ., University Park, Pennsylvania.

Thomas, H. R. and Kramer, D. F. (1988), The Manual of Construction Productivity Measurement and Performance Evaluation, Construction Industry Institute, May 1988

Tommelein, I. D., Walsh, K. D., and Hershauer, J. C. (2003), Improving Capital Projects Supply Chain Performance, PT-172 Research Report, March 2003

Turney, P. B. B. (1992), What an ABC Model Looks Like, Journal of Cost Management, Vol. 5, No. 4, pp. 54-60

Tyndal, G. R. (1990), Logistics Costs and Service Levels: Evaluating the Trade-Offs, Chapter 4-6, Emerging Practices in Cost Management, edited by Barry J. Brinker, Boston, MA: Warren, Gorham, & Lamont, 1990, pp. 211-217.

William, C. W., Eng, U. C. and Schoner, B. (2001), Magnitude Adjustment for AHP Benefit/Cost Ratios, European Journal of Operational Research, Vol. 133, No. 2, pp. 587-602

BIOGRAPHICAL SKETCH

Aiyin Jiang was born in Shanghai, China. She earned a B.S. in civil engineering and M.S. in structural engineering from the Southwest Jiaotong University, China. She received a master's degree in decision and information science from the University of Florida in 2002. In 2005 she completed her Ph.D. in design, construction and planning from the University of Florida. In the future, she is looking forward to having a successful academic career in the construction management field.